

# Chapter 69

## Wireless Sensor Network Platforms

**Reinhard Bischoff, Jonas Meyer and Glauco Feltrin**

*Structural Engineering Research Laboratory, Empa, Swiss Federal Laboratories for Materials Testing and Research, Dübendorf, Switzerland*

---

1 Introduction	1
2 Hardware Architectures	3
3 Software Platforms	8
Related Articles	9
References	9

---

### 1 INTRODUCTION

Basically, every structural health monitoring (SHM) system is made up of various sensors measuring specific physical parameters, a data acquisition unit, and a storage device to save the acquired data. Traditional SHM systems show a starlike topology where each deployed sensor is connected via long cable runs to a central computer acting as data acquisition and storage device. The installation of such systems tends to be time consuming and therefore expensive (Figure 1). Especially in the field of civil engineering where the structures are typically large, the sensors can be located long way away from the data acquisition unit, resulting in high installation costs. These costs have proved to be a major issue, preventing a

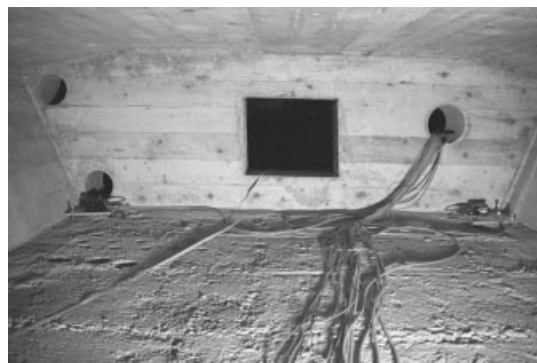
broad application of monitoring techniques to large-scale infrastructure. Furthermore, long cable runs are prone to pick up noise, reducing the effective accuracy of the acquired data, and hence require expensive high-quality cables. Moreover, these cables are susceptible to mechanical damage involving considerable maintenance efforts. Cabled systems also tend to offer a limited flexibility in terms of rearrangement of sensors and scalability. The adoption of wireless sensor network (WSN) techniques to SHM applications promises to overcome these drawbacks.

#### 1.1 Wireless sensor network

A WSN is essentially a computer network consisting of many small, intercommunicating computers equipped with one or several sensors. Each small computer represents a node of the network. These nodes are called sensor nodes or motes. The communication within the network is established using radio frequency transmission techniques. The sensor nodes typically form a multihop mesh network by establishing communication links to neighbor nodes. Multihop networks offer different advantages when monitoring data has to be transmitted over long distances. Mainly, the network robustness to sensor node failure and the high power efficiency [1] make multihop networks attractive for monitoring applications. Figure 2 illustrates a schematic multihop



(a)



(b)

**Figure 1.** Traditional, wired SHM installation.



**Figure 2.** Wireless sensor network deployed on a road bridge. The spots illustrate the sensor nodes, the straight lines the communication links.

network deployed on a road bridge. The network consists of several dozens of sensor nodes. Theoretically, the number of sensor nodes is unlimited. All sensor nodes are equipped with specific sensors tailored to their measurement tasks. On the one hand, these nodes act as data sources, and on the other hand, they act as relaying stations, receiving and forwarding

data from adjacent nodes. One or more particular sensor nodes act as base station and represent the data sink in the network. It aggregates all the data generated within the network. In addition, the base station establishes a communication link to a data logging unit or a remote site (e.g., control center), using standard wired or wireless communication technologies like universal mobile telecommunications system (UMTS) or wireless local area network (WLAN).

The initial research into WSNs was mainly driven by military applications like battlefield reconnaissance and surveillance, nuclear, biological, and chemical attack detection, etc. These projects focused on *ad hoc*, multihop WSNs that consisted of thousands of immobile nodes randomly distributed over a large geographical area (e.g., Smart Dust). The nodes were tiny (hardly noticeable), severely resource constrained, and homogeneous (identical hardware and software). Subsequently, the emergence of civilian applications of WSNs in different fields (environmental monitoring, home automation, health applications, production, inventory, delivery control,

etc.) produced a significant diversification of requirements with respect to deployment, mobility, size, cost, network topology, lifetime, etc., and therefore a flourishing of academic and commercial WSN platforms. To cope with these requirements, the platforms increased in size, computational resources, and hardware, as well as in software complexity.

The first commercial platforms appeared in the late 1990s. The most important platform was Crossbow's Rene mote, which emerged from the weC mote developed at the University of California, Berkeley, and which evolved later to the popular Mica platform. These platforms were the precursors of the recent Mica2 and MicaZ platforms (Table 1). A major reason for the popularity of Crossbow's early mote platforms was their open source policy with both hard- and software design open to the public. This policy built the base for the widespread diffusion of TinyOS as operating system for WSNs. Today, various commercial platforms with different characteristics in terms of computing resources, sensor interfaces, software architecture, etc., are available, which allow to cope with a wide spectrum of civilian applications.

## 2 HARDWARE ARCHITECTURES

The sensor nodes are the fundamental components of a WSN. To enable WSN-based SHM applications, the sensor nodes have to provide the following basic functionality (Figure 3):

- signal conditioning and data acquisition for different sensors;
- temporary storage of the acquired data;
- processing of the data;
- analysis of the processed data for diagnosis and, potentially, alert generation;
- self monitoring (e.g., supply voltage);
- scheduling and execution of the measurement tasks;
- management of the sensor node configuration (e.g., changing the sampling rate and reprogramming of data processing algorithms);
- reception, transmission, and forwarding of data packets;
- coordination and management of communication and networking.

### 2.1 General architecture

To provide the functionality described above, a sensor node is composed of one or more sensors, a signal conditioning unit, an analog-to-digital conversion module (ADC), a processing unit with memory, a radio transceiver, and a power supply (Figure 4).

If the sensor nodes are actually deployed in the field, especially in harsh environments like construction sites, they have to be protected against chemical and mechanical impacts. Therefore, an adequate packaging of the hardware is required (*see Microelectromechanical Systems (MEMS)*).

### 2.2 Hardware platform categories

Sensor node hardware platforms can be divided into three categories [2]. Each category shows a different hardware setup matched to diverse monitoring applications.

#### • Adapted general-purpose computers

These platforms are low-power personal computers (PCs), embedded PCs, and personal digital assistants (PDAs). These platforms mainly run on Windows CE, Linux, or other operating systems developed for mobile devices. These platforms are predominantly equipped with standard wireless communication devices like Wireless LAN (IEEE 802.11) and/or Bluetooth (IEEE 802.15.1). Because of the high processing ability and the high bandwidth communication, these platforms offer the opportunity to use higher level programming languages, which makes it easier to develop and implement software components. But in turn, they consume a considerable amount of energy and this can be prohibitive in some application scenarios. Additionally, they support networking protocols like Internet Protocol (IP). This simplifies the integration into a monitoring system.

#### • Embedded sensor modules

These platforms are assembled from commercial off-the-shelf (COTS) Chips. Using COTS offers several benefits. These components are widely used, making them cheap because of big production quantities, and are well supported by the manufacturers and communities. The microcontroller unit (MCU) of these platforms is mostly programmed in C. This

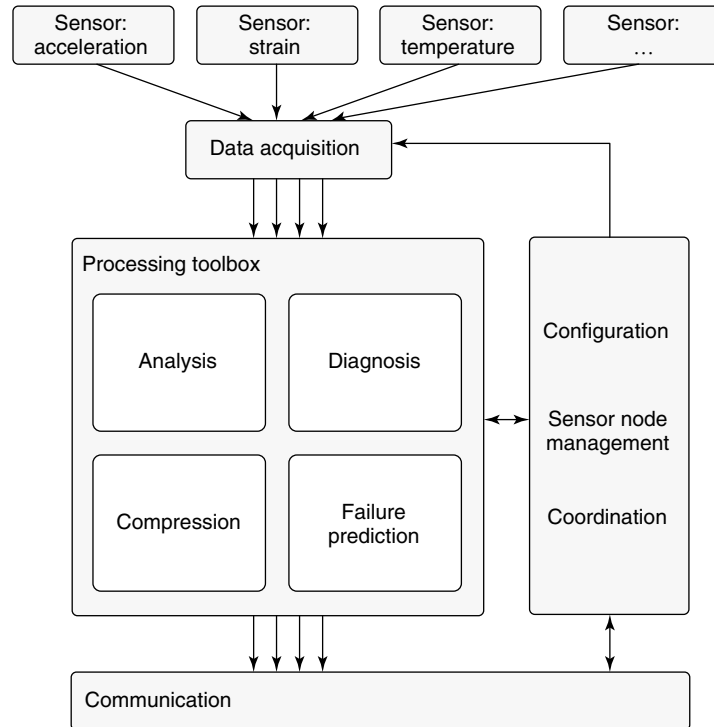
Table 1. Selection of wireless sensor network platforms

Name	Tmote	Mica2	MicaZ	Imote2	JN5121	Sun SPOT	Agile (V-Link)	BTnode rev3
MCU	Chip manufacturer Texas Instrument	Atmel	Atmel	Intel	OpenCores	ARM		Atmel
	MSP430F1611	ATMega 128L	ATMega 128L	PXA271 XScale	OpenRISC1000	ARM920T		ATmega 128L
Frequency (MHz)	8	7.383	7.383	13-416	16	180		7.383
Type (bit)	16	8	8	32	32	32		8
ROM, RAM (kB)	48, 10	128, 4	128, 4	32 MB, 32 MB	64, 96	4M, 512		64 + 180, 128
Interfaces	I <sup>2</sup> C, UART, SPI	I <sup>2</sup> C, UART, SPI	I <sup>2</sup> C, UART, SPI	UART, SPI, I <sup>2</sup> C, AC97, I2S, Camera	SPI, UART			ISP, UART, SPI, I <sup>2</sup> C
A/D, D/A	8, 2	8, 0	8, 0					
A/D channels	8	8	8		4	6	8	
Maximum sampling rate (kHz)		1	1				2	
Resolution (bit)	12	10	10					
D/A channels	2				12		12	
Maximum sampling rate (kHz)					2			
Resolution	12							11

Radio	Chip manufacturer	Chipcon	Chipcon	Chipcon	Chipcon	Chipcon	Zeevo, Chipcon ZV4002, CC1000 433 or 868/916, 2400
	Chip model	CC2420	CC2420	CC2420	CC2420	CC2420	
	Frequencies (kHz)	2400	2400	2400	2400	2400	
	Raw data rate (kbps)	250	250	250	250	250	
	Standard (IEEE)	802.15.4	802.15.4	802.15.4	802.15.4	802.15.4	Bluetooth, 802.15.1
	Range outdoor (m) <sup>(a)</sup>	125	100	30	30	70	
External memory	Chip manufacturer	ST	Atmel	Atmel	Atmel	Atmel	
	Chip model	M25P80	AT45DB41B	AT45DB41B	AT45DB41B	AT45DB41B	
	Size (KB)	1024	512	512	512	512	2048
Power	Supply voltage min, max (V)	2.1, 3.6	2.7, 3.3	2.7, 3.3	3.2, 5	2.7, 3.6	3.7
	Current consumption (normal, radio off) (mA) <sup>(b)</sup>	21.8, 1.8	39, 12	29.4, 12	44–66, 31	50, 5	90, 25
Dimensions	(cm × cm × cm)	6.6 × 3.2 × 0.7	5.8 × 3.2 × 0.7	5.8 × 3.2 × 0.7	4.8, 3.6, 0.9	3.0, 1.8, 1.0	3.5, 2.5
Manufacturer		Moteiv	Crossbow	Crossbow	Crossbow	Jennic	Sun Microsystems
							ETH Zürich

<sup>(a)</sup> Using integrated antenna.

<sup>(b)</sup> Values declared by manufacturer or typical datasheet values (power consumption computed by individual summation of system core, flash memory, and radio component values).



**Figure 3.** Basic functionality of a sensor node.

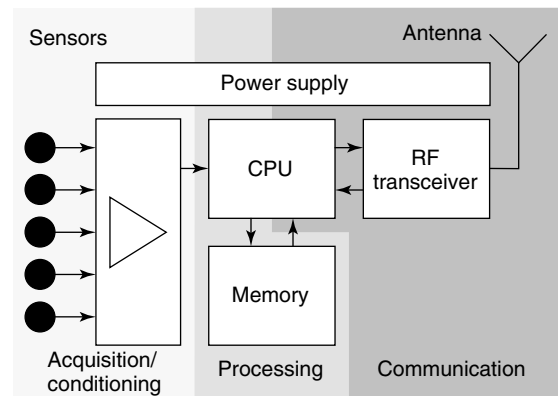
enables the development of a tight code that fits the limited memory size. Application developers have full access to hardware, but at the same time need to take care of all the resources. Examples from this category are Tmote from Moteiv, Mica2, MicaZ (Mica family), and Imote2 from Crossbow.

- **System on chip (SoC)**

These platforms integrate micro electromechanical systems (MEMS) sensors, microcontroller, and wireless transceiver technologies on one chip, an application-specific integrated circuit (ASIC). Because of this integration, platforms of this category are extreme low-power devices and have a small footprint/size. The smart dust node [3] is such an example.

### 2.3 Energy-related aspects

The advantages of WSNs over wired sensing systems only have an effect if an unattended operation of the



**Figure 4.** Sensor node (mote): hardware structure of a sensor node.

motes for a reasonably long period of time can be achieved. In terms of energy resources, this calls for a self-contained power source and has shown to be the most restrictive requirement in WSN applications. One approach to provide sufficient energy to operate the device over the desired period is to estimate the

total amount of energy that will be consumed and to equip the mote with adequately dimensioned energy storage upon deployment. Although this approach is a viable solution for some lowest-duty cycle applications, the required energy storage tends to become too big for long-term SHM systems. For monitoring applications that target overall system lifetime of several years, the dissipated energy has to be regenerated from mote-extern sources. This process is referred to as energy harvesting or scavenging. These sources absorb energy from the mote's environment and convert it to electrical energy.

Numerous different types of energy storage and harvesting concepts exist, but only the most important ones for SHM systems are presented here. An overview of other, partly inconvenient concepts is given in [4].

### 2.3.1 Communication versus computation

In terms of power consumption, wireless data transmission is much more expensive than data processing. To extend system lifetime, it is preferable to preprocess the raw sensor readings to reduce the data items needed to be transmitted to the base station. Many recent WSN-based SHM systems transmit the raw data streams to the base station and analyze them in the traditional centralized way. Without introducing huge batteries, this is not a viable solution if a system lifetime of several months to years is targeted. For long-term monitoring applications, distributed analysis algorithms have to be introduced, which allow for decentralized data reduction or even condition assessment.

### 2.3.2 Energy storage devices

#### ● Batteries

The most popular energy storage is batteries. Many battery types with different characteristics have been developed. Every battery has its own advantages and drawbacks and the suitable battery technology has to be selected according to the application requirements. Rechargeable batteries are utilized if energy is harvested from the mote's environment.

#### ● Ultracapacitors

The features of supercapacitors lie between those of capacitors and rechargeable batteries. Supercapacitors

exhibit virtually unlimited charge–discharge cycles like capacitors, but offer a much higher capacity. These components are adequate as energy storage if it is emptied and replenished in short intervals.

#### ● Fuel cells

This is a more recent but promising technology. Fuel cells oxidize hydrogen or hydrocarbon fuels and convert the heat into electrical energy. Currently, the commercially available fuel cells are too big in terms of size and converted energy to be applied to motes. However, much effort is being put into the development of small fuel cells for laptops and mobile phones. These devices will suit WSN applications well.

### 2.3.3 Energy harvesting and scavenging devices

Because of their nature, environmental energy scavenging devices do not provide a constant energy flow. Therefore, these devices are predominantly operated in conjunction with a storage device like a supercapacitor or a rechargeable battery. It stores excess energy and provides it later, when not enough can be harvested from the environment.

#### ● Solar cells

The most popular energy scavenging sources are solar cells. A reasonably small panel delivers enough energy to power a sensor node. Solar cells are predominately operated in conjunction with a supercapacitor or a rechargeable battery. This energy storage is needed to provide energy, when the panel does not. Obviously, solar cells are only an option for outdoor applications.

#### ● Wind mills

More unusual energy scavenging devices are small-scale wind mills or turbines. Like solar cells, this concept is only suitable for outdoor applications.

#### ● Vibration

An energy harvesting method that is considered for civil engineering applications is to convert vibration energy. Civil engineering structures contain a lot of vibration energy, but it is extremely hard to extract it. The energy levels that current prototypes provide are far too low for monitoring applications. But it could evolve to an interesting source in the future.

## 2.4 Overview of recent architectures

Various hardware platforms for WSNs are available today and new ones emerge regularly. This diversity offers the possibility to choose a platform that best fits the needs of a specific application. An overview of recently used platforms is given in Table 1. This table only shows a selection. Further platforms are presented in [5] and regularly updated lists in the Internet can be found at [6–9].

## 2.5 Tmote Sky from Moteiv Corporation

Tmote [10] from Moteiv Corporation is presented as an example of a popular WSN platform (Figure 5). Many comparable platforms with similar hardware setups exist today. All these platforms are based on the Texas Instruments microcontroller family MSP430 and the Chipcon radio CC2420.

The main components of Tmote sensor node platform are the TI MSP430F1611 microcontroller, the FTDI FT232BM USB interface, which allows for programming the microcontroller over USB, and the Chipcon CC2420 low-power radio chip for the wireless communication.

The ultra-low-power microcontroller features 10 kB of RAM and 48 kB of program memory (flash). This 16-bit processor features several power-down modes with extremely low sleep-current consumption that permits the sensor node to run for a long period of time from a limited energy resource. The MSP430 has an internal, digitally controlled oscillator (DCO) that may operate up to 8 MHz. The microcontroller may be turned on from sleep mode in 6  $\mu$ s, which allows for short reaction time upon the occurrence of an event. When the DCO is off, the MSP430 is clocked from an external 32 768-Hz watch crystal.

The MSP430 has eight external 12-bit ADC ports of which six are accessible on a pin header on the Tmote. The ADC input ranges from 0 to

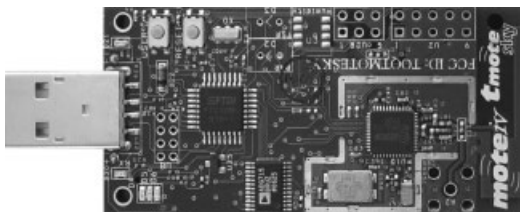
3.0 V. The maximum total sampling rate for all ports is 200 kHz at 12-bit resolution. The internal ADC ports may be used to monitor the internal processor temperature and the supply voltage. A variety of peripherals are available, including serial peripheral interface (SPI) and universal asynchronous receiver/transmitter (UART), enabling the communication to digital output sensors, digital I/O ports, a watchdog timer, and timers with capture and compare functionality. The I<sup>2</sup>C port, which is also integrated into the microcontroller, is mainly used to communicate to additional sensors and signal conditioning boards. The MSP430 also includes a 2-port 12-bit digital-to-analog converter (DAC) module, a supply-voltage supervisor, and a 3-port direct memory access (DMA) controller. Detailed features of the MSP430F1611 are presented in the Texas Instruments MSP430x1xx Family User’s Guide [11].

The Tmote platform is equipped with the Chipcon CC2420 radio, enabling IEEE802.15.4 standard compliant wireless communication. It offers reliable wireless communication and power management capabilities to ensure low-power consumption. The CC2420 is controlled by the TI MSP430 microcontroller through the SPI port and a series of digital I/O. The radio may be shut off by the microcontroller for reducing the power consumption. The CC2420 provides a digital receive signal strength indicator (RSSI) that may be read at any time. The programmable transmitter output power enables to optimize the power consumption. The theoretically achievable maximum data throughput rate of the system is 250 kbps, without framing and packet headers.

## 3 SOFTWARE PLATFORMS

Unlike general-purpose operating systems for standard PCs such as Windows or Linux, the WSN software platforms are highly tailored to the limited node hardware. These WSN software frameworks are not full-blown operating systems, since they lack a powerful scheduler, memory management, and file system support. However, these frameworks are widely referred to as *WSN operating systems*. Therefore, this term is retained in the following section.

TinyOS [12], one of the most widespread operating systems, is presented in more detail in the following section. Other operating systems developed for WSNs are Contiki [13], Mantis [14], and SOS [15].



**Figure 5.** Picture of a Tmote Sky form Moteiv (top view).



### 3.1 TinyOS

TinyOS is written in nesC [16], an extension to the C language, which supports event-driven component-based programming. The basic concept of component-based programming is to decompose the program into functionally self-contained components. These components interact by exchanging messages through interfaces. The components are event-driven. Events can originate from the environment (a certain sensor reading exceeds a threshold) or from other components, triggering a specific action. The main advantage of this component-based approach is the reusability of components.

The nesC language extension introduces several additional keywords to describe a TinyOS component and its interfaces. nesC and TinyOS are both Open Source projects supported by a fast growing community.

TinyOS has been ported to over a dozen WSN platforms (Table 1) and is also the native operating system of the presented Tmote platform. It provides a concurrency model and mechanisms for structuring, naming, and linking software components into a robust network embedded system. Today, TinyOS is a sort of *de facto* standard in WSN programming and widely used in the WSN community. As a result, a huge amount of software components for various sensors, network protocols, algorithms, and other WSN related topics is freely available on the Internet.

## RELATED ARTICLES

### Sensor Network Paradigms

#### Nondestructive Evaluation of Cooperative Structures (NDECS)

#### On the Way to Autonomy: the Wireless-interrogated and Self-powered “Smart Patch” System

#### Energy Harvesting using Thermoelectric Materials

## REFERENCES

- [1] Karl H, Willig A. *Protocols and Architecture for Wireless Sensor Networks*, ISBN 0-470-09510-5. John Wiley & Sons: Chichester, 2005, pp. 60–62.
- [2] Zhao F, Guibas L. *Wireless Sensor Networks: An Information Processing Approach*, ISBN 1-5860-914-8. Morgan Kaufmann: San Francisco, CA, 2004, pp. 240–245.
- [3] Kahn JM, Katz RH, Pister K. Mobile networking for smart dust. *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*. Seattle, WA, 17–19 August 1999.
- [4] Roundy S, Steingart D, Frechette L, Wright P, Rabaey J. Power sources for wireless sensor networks. In *Proceedings of the 1st European Workshop on Wireless Sensor Networks (EWSN), Lecture Notes in Computer Science*, Karl H, Willig A, Wolisz A (eds). Springer: Berlin/Heidelberg, 2004; Vol. 2920, pp. 1–17.
- [5] Lynch JP, Loh K. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*. Sage Publications, 2005, pp. 91–128.
- [6] SNM—The Sensor Network Museum.™ <http://www.btnode.ethz.ch/Projects/SensorNetworkMuseum>.
- [7] Bokareva T. *Mini Hardware Survey*, [http://www.cse.unsw.edu.au/~sensar/hardware/hardware\\_survey.html](http://www.cse.unsw.edu.au/~sensar/hardware/hardware_survey.html).
- [8] Body Sensor Networks. <http://ubimon.doc.ic.ac.uk/bsn/m206.html>.
- [9] Wireless Sensor Network (WSN) Wiki. [http://wsn.oversigma.com/wiki/index.php?title=WSN\\_Platforms](http://wsn.oversigma.com/wiki/index.php?title=WSN_Platforms).
- [10] Polastre J, Szewczyk R, Culler D. *Telos: Enabling Ultra-Low Power Research*. Information Processing in Sensor Networks/SPOTS: Berkeley, 2005.
- [11] MSP430x1xx Family User’s Guide. <http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>, 2006.
- [12] Levis P, *et al.* TinyOS: an operating system for wireless sensor networks. *Ambient Intelligence*. Springer-Verlag: New York, 2005.
- [13] Dunkels A, Gronvall B, Voigt T. Contiki—a lightweight and flexible operating system for tiny networked sensors. *Proceedings of the 29th Annual IEEE international Conference on Local Computer Networks (Lcn’04)*. LCN IEEE Computer Society: Washington, DC, 2004; pp. 455–462.
- [14] Abrach H, Bhatti S, Carlson J, Dai H, Rose J, Sheth A, Shucker B, Deng J, Han R. MANTIS: system support for multimodal networks of in-situ sensors. *Proceedings 2nd ACM International Conference on Wireless Sensor Networks and Applications*. ACM Press: New York, 2003; pp. 50–59.

- [15] Han C, Kumar R, Shea R, Kohler E, Srivastava M. SOS: a dynamic operating system for sensor nodes. *Proceedings of the 3rd international Conference on Mobile Systems, Applications, and Services (Seattle, Washington, June 06–08, 2005)*. *MobiSys'05*. ACM Press: New York, 2005; pp. 163–176.
- [16] Gay D, Levis P, von Behren R, Welsh M, Brewer E, Culler D. The nesC language: a holistic approach to networked embedded systems. *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation*. San Diego, CA, 2003; pp. 1–11.