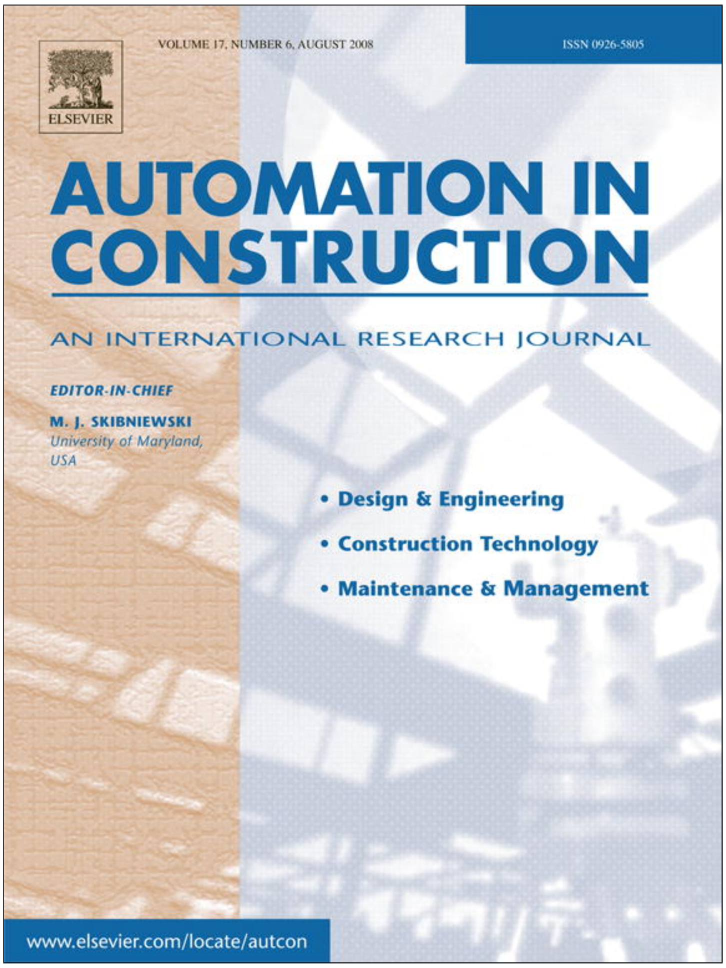
VOLUME 17, NUMBER 6, AUGUST 2008
ISSN 0926-5805
ELSEVIER
AUTOMATION IN CONSTRUCTION
AN INTERNATIONAL RESEARCH JOURNAL
EDITOR-IN-CHIEF
M. J. SKIBNIEWSKI
University of Maryland, USA
• Design & Engineering
• Construction Technology
• Maintenance & Management
www.elsevier.com/locate/autcon

Automation in Construction 17 (2008) 729–736

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Wireless sensor networks as part of a web-based building environmental monitoring system

Won-Suk Jang [a,*], William M. Healy [b], Mirosław J. Skibniewski [a]

[a] Department of Civil and Environmental Engineering, University of Maryland, College Park, MD, USA
[b] Building and Fire Research Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA

ARTICLE INFO

Article history:
Accepted 8 February 2008

Keywords:
Wireless sensor networks
Building monitoring
Sensor database
Motes

ABSTRACT

The presented research shows how advanced wireless sensor technology can be used by engineers to monitor conditions in and around buildings. The objective is split into three different tasks. First, wireless sensor hardware is programmed to process signals from sensors and transmit the data in a suitable format. This task was accomplished through an open-source operating system and a programming language designed specifically for wireless sensor hardware. The second task involved the processing of signals sent by the wireless sensor nodes. In this application, a Java program was written that deciphered messages transmitted from a wireless receiver over a computer's serial port and then placed the data in a database. The structure of that database is discussed to help identify the key pieces of information that are needed to make use of the data. The third piece of the proposed monitoring system is an interface to review the data. A Web-based system was developed that allows a user to mine the database using parameters such as the type of data, location of sensor, and the time of data acquisition. It is anticipated that this research will demonstrate the potential of using wireless sensor networks for monitoring buildings.

1. Introduction

Emerging wireless sensor technology promises to enable enhanced monitoring of conditions in and around buildings [1]. Wireless data communication between the sensor and a viewing or storage location opens up a range of possibilities because of the ease and the low cost by which the sensors can be deployed. There is no need to run unsightly signal wires through various parts of a constructed building, and significant time savings can be obtained in setting up the sensors. With these sensing possibilities, one can think of the scenarios in which such information could be used. For example, forensic analyses within buildings to determine the cause of problems would greatly benefit from the ability to deploy wireless sensors on a short term basis. Wireless sensors could be placed on critical pieces of equipment in buildings to help detect and diagnose faults. Buildings lacking a whole building automation system could utilize more sensing points to more efficiently control its lighting, heating, ventilating, and air-conditioning (HVAC) equipment, and plug loads [2]. In hazardous situations such as fires, deployment of wireless sensors could provide more information about the conditions within and around a building for first responders.

Researchers, investigators, or maintenance personnel can use a variety of methods to monitor conditions in a building by measuring such quantities as temperature, relative humidity, light, or energy consumption. Wired sensors could certainly be installed at each sensing location, but such a step requires significant effort and an additional set of wires throughout a building. Additionally, estimates of the cost to run signal wire in 2002 ranged from $2.20 per meter for new construction to $7.19 per meter for existing construction [3]. Another example of the costs involved in wired systems can be found in a recent structural monitoring system, where up to 75% of total testing time and 25% of system cost involved the installation of signal wire [4]. Anecdotal evidence suggests the costs to be much higher for more critical facilities, such as nuclear power plants. One way to avoid the need for signal wires is through the use of small data loggers with integrated sensors at each location. These devices do not have signal wires running to and from them, but they require personnel to periodically download their data. This step adds to the cost and difficulty of using these devices. Transmitting the data wirelessly provides a significant benefit to those investigating buildings by allowing them to deploy the sensors and monitor the data from a remote location [5]. Wireless systems, however, have their own set of disadvantages, such as higher equipment cost, the potential for radiofrequency interference to damage the data stream, and the need to provide power to these "wireless" devices. All things considered, however, the potential applications for wireless sensing abound.

A wireless sensor network consists of various pieces of hardware and software. At the heart of the system is the wireless sensor device. This piece of equipment consists of the physical sensors, a microprocessor to analyze the raw data signal and generate the data message, a radiofrequency transmitter to deliver the data, and a power source. The package will often be referred to as a node or "mote." A key

* Corresponding author.
E-mail address: wsjang@umd.edu (W.-S. Jang).

0926-5805/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2008.02.001

aspect of a wireless sensor network is that the microprocessor on each mote can be programmed to ensure that all sensors in a given region work as a coherent system. While sensor nodes in a wireless sensor network are capable of exchanging information with other nodes, most applications will involve the delivery of data and information from each sensor node to a central data collection point. That point will typically be a computer that archives the data, and software is needed to ensure that the data delivered from the wireless receiver is interpreted, displayed, and stored in a usable manner. The third component of any successful wireless sensor network system involves retrieval of the data produced from the sensor network. Software applications must somehow be able to query the data generated by the sensor network in a logical manner. This work will discuss issues related to data acquisition, storage, and retrieval from wireless sensor networks.

With current wireless technology, a great challenge arises because of the level of expertise needed to fully make use of the sensors. The most sophisticated hardware has emerged from university laboratories and often requires advanced knowledge of embedded programming to achieve the level of performance desired. Such knowledge is not common among the civil and mechanical engineers who are often tasked to use the devices on construction sites or in buildings. An added complication lies in the fact that much of the work involves proprietary programming methods, which makes it difficult to develop a standardized method of setting up a sensor network or to customize the sensor network for a particular application.

The purpose of the present work is to explore methods to make wireless sensors more easily used by engineers involved with buildings and construction. Initial discussion will focus on the programming of wireless sensor hardware using open-source software. The second aspect of the work will discuss database specifications for storing the data. The manuscript will conclude with a description of a method to easily view the data via a Web browser. It is hoped that the present work will describe a system that can effectively be used in a range of applications for monitoring purposes and will present a clear path that engineers can take to use existing wireless sensor technology in their particular application.

2. Background

A number of articles in the literature have discussed the use of wireless sensing for buildings and monitoring. Kintner-Meyer and Brambley [3] and Kintner-Meyer et al. [6] installed 30 temperature sensors throughout an office building and additional sensors on a rooftop HVAC unit to monitor its performance. For an office building such as the one discussed, wiring costs (labor plus material) make up approximately 45% of the installed cost for a new building and nearly 75% of the installed cost for a retrofit application. Healy [7] examined the use of wireless sensing devices to monitor conditions within a residence, finding that the sensor networks were easy to set up but required more programming than desired for easy deployment. Wills [8] discusses the move towards wireless technologies for control applications in buildings, foreshadowing the move by the BACnet committee of the American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE) to work with the ZigBee wireless standard committee [9] to implement wireless communications in building controls. Raimo [10] discusses the emergence of mesh networking for connecting controllers in a building. Ruiz [11] discussed a range of ways to implement various wireless applications in buildings, from voice communications to low data rate monitoring. In structural health monitoring, Lynch [12] reviews the emerging interest in using wireless sensors for monitoring structures. The article indicates that visual inspection of steel structures can cost between $200 and $1000 per welded connection, while deployment of wired sensors to monitor structural health in a building can cost up to $5000 per sensor for a system with 12 to 15 sensors. The use of wireless sensors would greatly cut this cost through savings in the labor required to install the system. From the range of articles, it is apparent that the building industry is very curious about the use of wireless in buildings, but questions still remain about the ease of use, the dependability, and the cost of wireless monitoring.

Recent advances in wireless technology have made deployment of these systems easier; these advances are documented by a number of authors (Pottie and Kaiser [13], Akyildiz et al. [14], Estrin et al. [15], Hill et al. [16], Szewczyk et al. [17], Azia et al. [18]). Of particular note is the use of mesh networking. Mesh networking describes a topology of a sensor network in which each sensor node communicates with its neighbors and can relay messages from that neighbor through the network. A contrasting topology would be a star network, in which each sensor node communicates directly to a base station that collects the data. The benefit of the mesh networking scheme lies in the fact that each sensor node need only communicate with a neighboring node as opposed to directly to a base station. The most advanced mesh networking schemes allow the sensors to arrange themselves in an ad-hoc manner, so that the routing path of a message can change should an obstruction prevent communication between two nodes or should another radio enter the network. This type of system adds robustness to the sensor network and permits it to be easily expanded. The downside of such a system is the increased complexity in the software (and corresponding increased cost of the system) and the fact that each sensor node serves as a repeater and therefore consumes more energy.

When considering sensing of the building environment, a large range of constituents could potentially be of interest to engineers. Of these, the most significant are likely to include temperature, relative humidity, light, carbon dioxide, carbon monoxide, power, smoke, occupancy, and flow rate. Numerous companies produce products that will archive data from such sensors and present the data via a graphical user interface. Such systems, however, are typically developed using proprietary protocols, and customization by the user is difficult. Alternatively, systems may follow common standard communication protocols such as BACnet [19], but such a system may not be appropriate for all applications around a building. Standard methods of exchanging and storing data and metadata from sensors for use by software applications would encourage the development of interoperable solutions to improve monitoring in buildings.

3. Methodology

Data representation challenges arise when dealing with wireless sensors. First, the data must be packaged by the wireless sensing node and sent in an understandable manner over the airwaves. That signal must be interpreted at a computer that stores or displays the data. To assist in making the data accessible by a large range of applications, efforts should be made to determine the best structures for storing the data in databases. To encourage interoperability between the different components of a sensing and monitoring system, standard data formats for the message being sent by the sensor, the database storing the sensor data, and the data stream from the database to the end application are needed. As part of this effort, the formats for transmission of data from the sensors and for storage of that data will be discussed.

To explore these questions, freely available, open-source tools for software development were sought. The software platform that was selected for programming the sensor nodes was the TinyOS operating system along with the nesC programming language (Culler [20], Gay et al. [21]).[1] These platforms were developed at the University of California at Berkeley specifically for sensor nodes [20], and an active user community has developed various software tools. While many programming languages could be used for developing software to run

[1] Certain commercial software are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software identified are necessarily the best available for the purposes.

on a desktop to process incoming signals, Java was selected here based on its popularity and the availability of compilers and libraries. For retrieving and viewing data, it was decided to use a Web-based system instead of developing a platform-specific application.

4. Implementation

4.1. Overview

In implementing a wireless sensor network to monitor the building environment, commercially-available hardware was used. To turn these motes into a usable sensor network, additional software was needed to ensure that the data generated by the sensors could be easily accessed by a range of applications. The following sections will describe the implementation of the wireless sensor network system in detail. Fig. 1 describes a brief methodology of the implementation as 1) data acquisition, 2) data collection, and 3) data retrieval. The application envisioned for this demonstration is one in which an engineer seeks to improve the monitoring of conditions within a building. Such a study could occur to evaluate the performance of the building's systems or to help diagnose a problem such as poor indoor air quality. The engineer would like to place sensors throughout the building and monitor the conditions for several weeks to help determine the performance of the building and its systems. While kits may be available to carry out such a study, the engineer will likely want to fine-tune the system to install particular sensors on the nodes, to modify the data acquisition rate, or to process the data in different ways. The implementation described here aims to provide that flexibility.

4.2. Data acquisition

4.2.1. Hardware

As mentioned previously, the sensor nodes are the heart of a wireless sensor network. For this implementation, nodes from a single vendor were used. Fig. 2 shows an image of such a node. The node consists of the main board which contains a microprocessor and a radio, a sensor board, and a battery holder for 2 AA batteries. For this implementation, the sensor board contained temperature, light, acceleration, and magnetic sensors and plugged into the main board via a 51-pin expansion connector. As can be seen, the two boards are designed to fit over one another. It is conceivable that any suite of sensors can be placed on a sensor board and simply plugged into the main board to provide data for any particular application.

On the main board lie the microprocessor and the radio hardware. The microprocessor (ATMEGA128L) carries out the conversion of the analog signals from the sensor board to digital form, ensures that data are taken at the correct interval, processes the raw data, packages the data into a message, sends the message to the radio hardware, and orchestrates the routing of messages through the network. The intelligence provided by the microcontroller provides the power of the wireless sensor network and is achieved in this implementation by a simple chip containing 128 kB of programmable flash memory, 4 kB of static Random Access Memory (RAM), and 4 kB of EEPROM (Electrically Erasable Programmable Read-Only Memory). Considering these constraints, care must be taken in writing software that controls these motes.

The final two components making up the sensor node are the radio hardware and the power source. The radio transmits messages on the 2.4 GHz band and complies with the IEEE 802.15.4 specification [22]. This specification is intended for low data rate applications such as those described here and aims to achieve the necessary data transmission at distance of 25–30 m indoor and up to 100 m outdoor while using little power [23]. The standard has been adopted by the ZigBee Alliance, a consortium that aims to speed the adoption of wireless sensors. The radio provides extensive hardware support for packet handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment, link quality indication, packet timing information, and mesh network. The nodes are powered by 2 AA batteries with lifetime up to 1 year, and power consumption rate is 30 μW at sleep mode, 33 mW at active mode, and 45 mW at radio transmission [24]. A recent lifetime analysis indicated that average mote lifetime was 61 days at an average power consumption of 13.1 mV/day and with a maximum data rate of 4.64 packets/sec [25]. It is clear that the need to replace batteries certainly makes these sensors less appealing. Other technologies such as vibration energy harvesting or photovoltaic charging are being examined closely as an alternative means of providing power to the nodes.

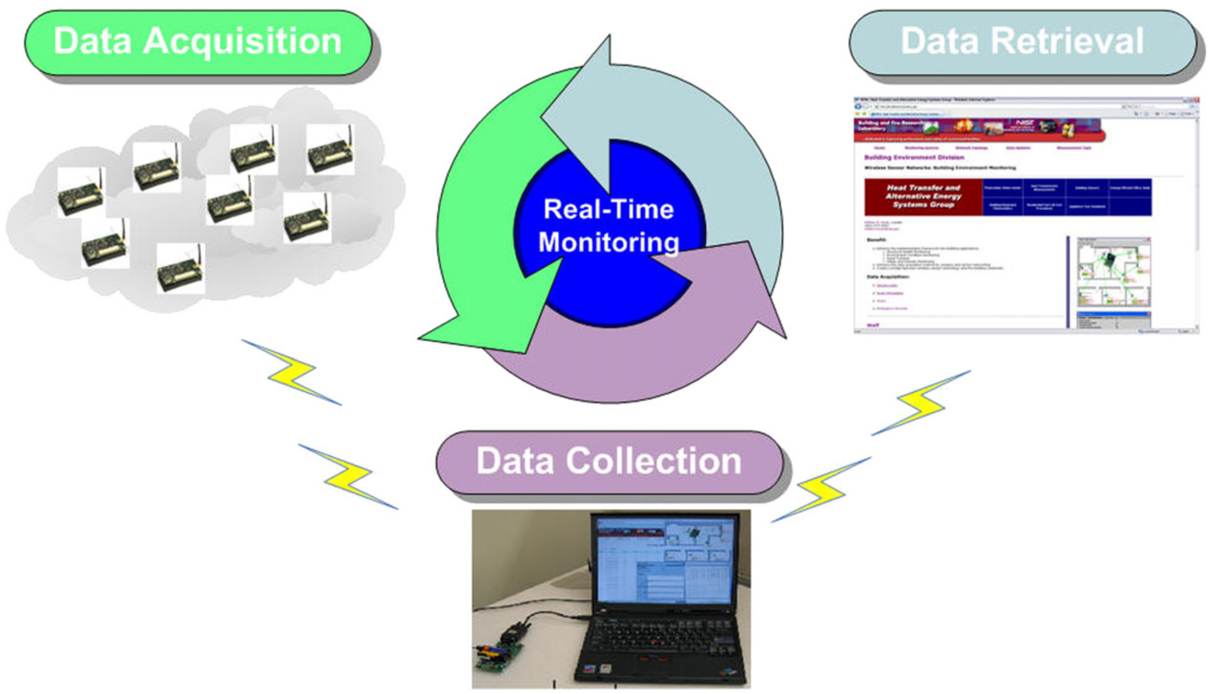

Fig. 1. Application diagram for a building monitoring system.

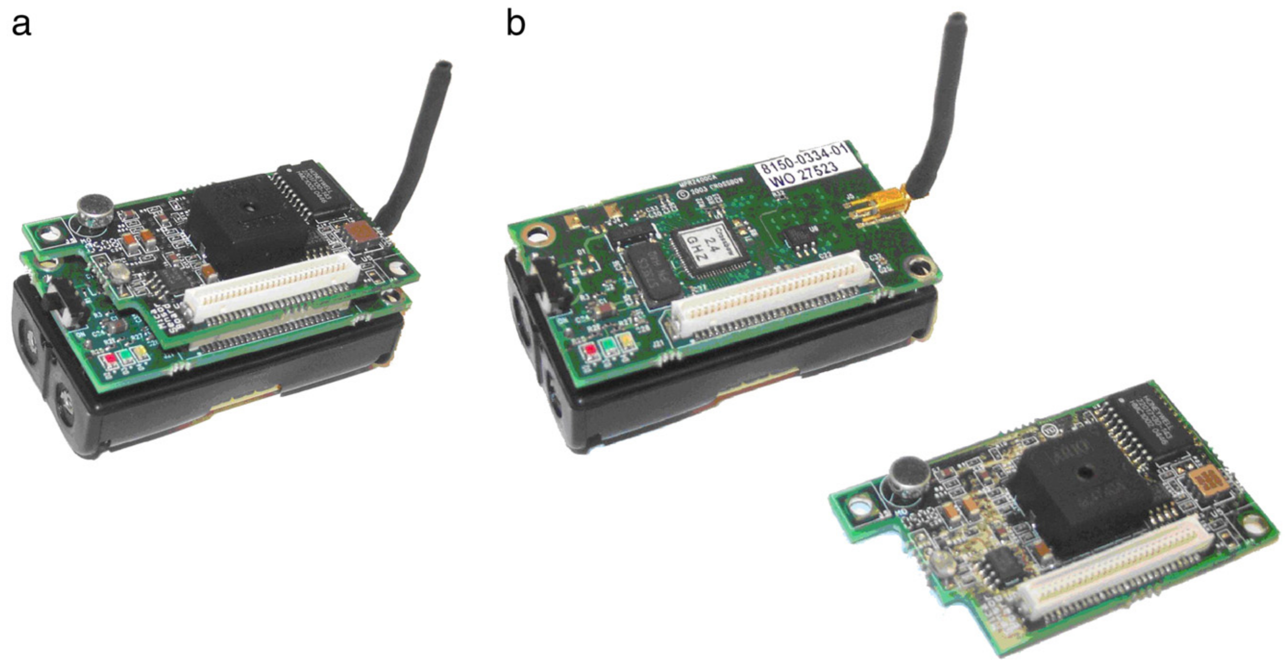

Fig. 2. Wireless sensor node. (a) sensor board and radio node connected, (b) sensor board and radio node separated.

4.2.2. Software

To make these motes function as desired, software must be developed and written to the on-board microprocessor. Applications are programmed on a desktop using the nesC programming language in a conventional editor; a compiler creates the machine code that is then transferred to each node through a temporarily attached wire. The wire is disconnected, and the program automatically runs when the sensor node is turned on.

The TinyOS operating system and the nesC language are specially designed for resource-constrained microprocessors. They aim to achieve efficient processing of events such as sensor data acquisition, radio transmission, and data processing. Another key aspect of the platform is the fact that it can be configured to ensure that the node uses very little energy when no action is needed by the sensor or radio.

Publicly-available software was used as the basis for the activity undertaken here. The software set up the sensors in a wireless mesh network, providing the algorithms needed to operate the radio and route the messages. The availability of such software is a boon to building engineers in that it provides them the wireless infrastructure on which they can build their specific sensing applications. The key software features that required attention were application-specific. The important pieces that the user would want to control are the time interval between acquisition of data from the sensor board, specification of analog to digital conversion for any particular suite of sensors attached to the node, and packaging of the data into a message that is sent over the airwaves. For the hardware used here, the time interval of 8 s for data acquisition is set in a header file, allowing simple modification of that value. For the sensor board, software was available to achieve the signal processing necessary to convert the sensor's

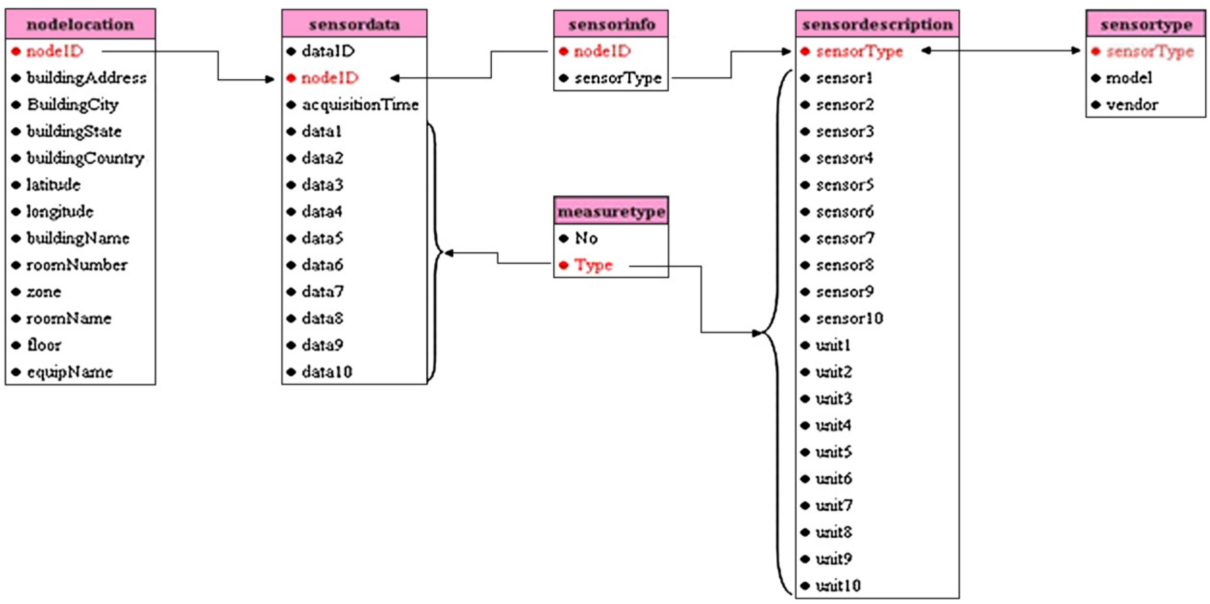

Fig. 3. Structure of database to store data from and information about the sensors in a sensor network.

signals to digital values. If other sensors are desired, however, this task would fall to the user or code would be needed from the developer of the sensor board. Knowledge of the input to each sensor (e.g. voltage levels) as well as how the output relates to the physical phenomenon being measured is needed. Alternatively, the sensor signals could be maintained in digital format and converted to engineering units by the end application. This approach was taken here; the message sent by the sensor node contained the data for each channel as a two-byte voltage reading. The message sent by the radio node starts with radio-specific features, such as an identification of the group to which the node belongs and a particular identification number for the node. This message can be generated automatically from code already developed, but the user may need to modify a group number to ensure that all sensors are part of the same network. After this preamble, the message then contains the data payload. For this application, it is assumed that the maximum number of sensors on any node will be 10. Each sensor's readings are represented in the message by 2 bytes of data. Clearly, the application written on the node must match with the application written to decipher the code. In this instance, the publicly-available code that was used as the basis for the mote programming has an accompanying suite of code that can be used to process the message.

4.3. Data collection

4.3.1. Hardware

Data collection takes place at a desktop computer. A radio node that is identical to the sensing nodes is connected to a special board that has an RS-232 connection. The node is programmed to act as a gateway node, receiving the data sent by all nodes in the network and sending the message across the RS-232 connection to the computer. In the programming interface, bytes of raw data are converted from/to asynchronous start–stop bit streams by a Universal Asynchronous Receiver/Transmitter (UART). More recent hardware uses a USB connection for this data transfer. The gateway is considered to be the bridge between the wireless and the wired networks.

4.3.2. Software

At the computer, software was written in Java to check for messages transmitted over the serial port. Each message is parsed to determine the sensor node from which it came and to separate the 10 possible data points (which can be extended up to 126 data points in a group network). The data points are converted to engineering units in this software application. A record is maintained as to which sensor node contains which sensors, and the appropriate transformations are

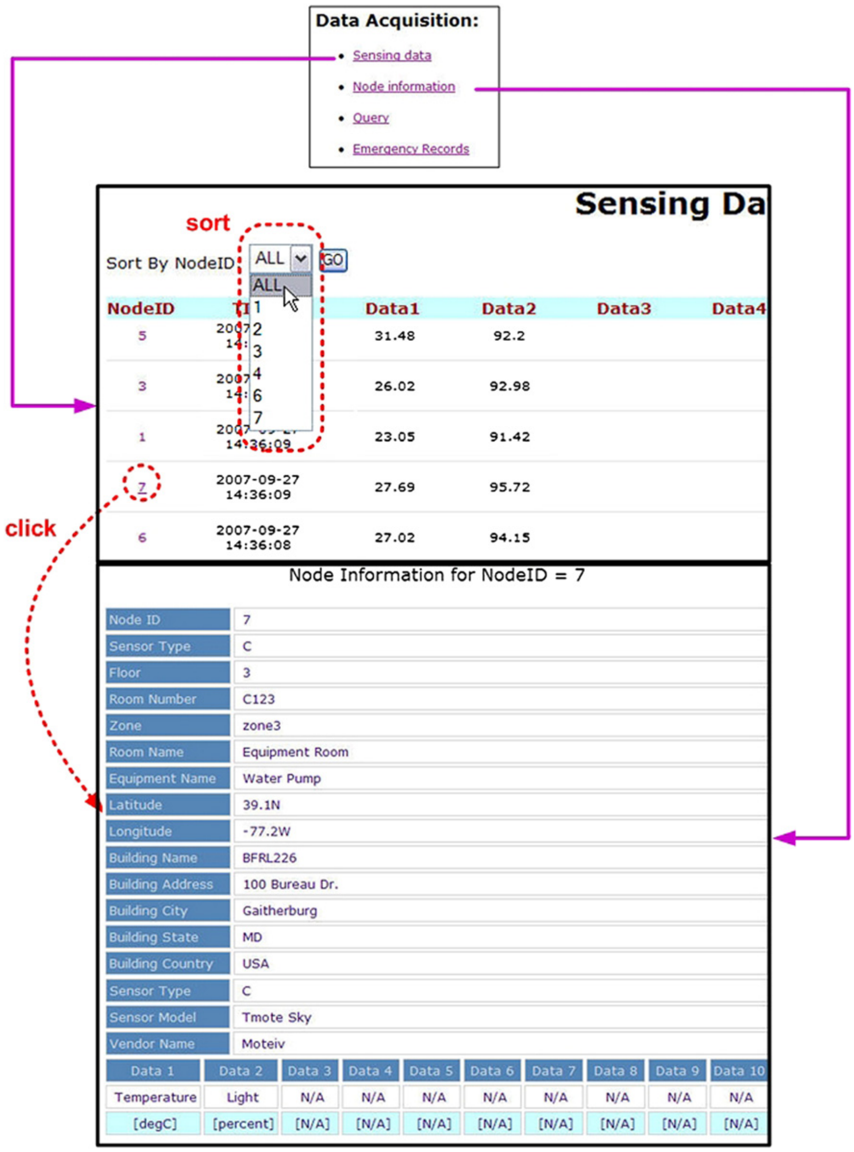

Fig. 4. Example of web-based display of sensing data and node information.

then made from raw digital values to physical values for each channel that is active on the sensor node. This application allows for real-time monitoring and plotting of the data. An additional component of this program allows one to see the wireless connections between nodes in the sensor network. A feature that was added as part of this work is the ability to archive the data into a standard database format. Discussion of the structure of the database is valuable in identifying the pieces of information that are needed to best make use of data from sensor networks.

The database was created using MySQL [26], a free database server, and was situated at the same computer to which the gateway was connected. Java code is available to ease the communication between Java programs and MySQL databases and was used in this work to place the data into the database. The structure of the database was devised to help provide all of the necessary metadata to describe the data generated by the sensor network and to store the data for any suite of sensors placed on a node. Fig. 3 shows a diagram of the structure of the database. Arrows show how each table links to other tables.

The database comprises 6 tables. Data in engineering units are stored in the SensorData table every time a transmission is received from a sensor node. Each entry contains a counter to provide a unique identification of the data value (dataID), the number of the node from which the data came (nodeID), the time at which the data were received at the gateway (acquisitionTime), and the 10 data values that were reported by that transmission. It should be noted that there is little latency expected in the sensor network, so the time when the data are received at the gateway and placed in the database deviates a negligible amount from when the data were actually taken at the sensor node. If a sensor node has less than 10 sensors, those slots will be left with null values. Definition of the quantity being measured is referenced by the SensorInfo, SensorDescription, and SensorType tables. The SensorInfo table indicates which sensor board is present on each node. The SensorType table stores information regarding the manufacturer and model of the sensor board. The SensorDescription table indicates which actual sensors are on a particular board. For each of the 10 possible sensors, a description of the quantity being measured (e.g. "temperature") is provided along with the units for each data value. Once again, if a sensor board has less than 10 sensors, null values are present in both the sensor and unit fields. This approach was taken in constructing the database under the assumption that users would typically use the same sensor board on many nodes. Therefore, specifying the types of sensor on each node is made easier by referencing the table of sensor boards. The final table,

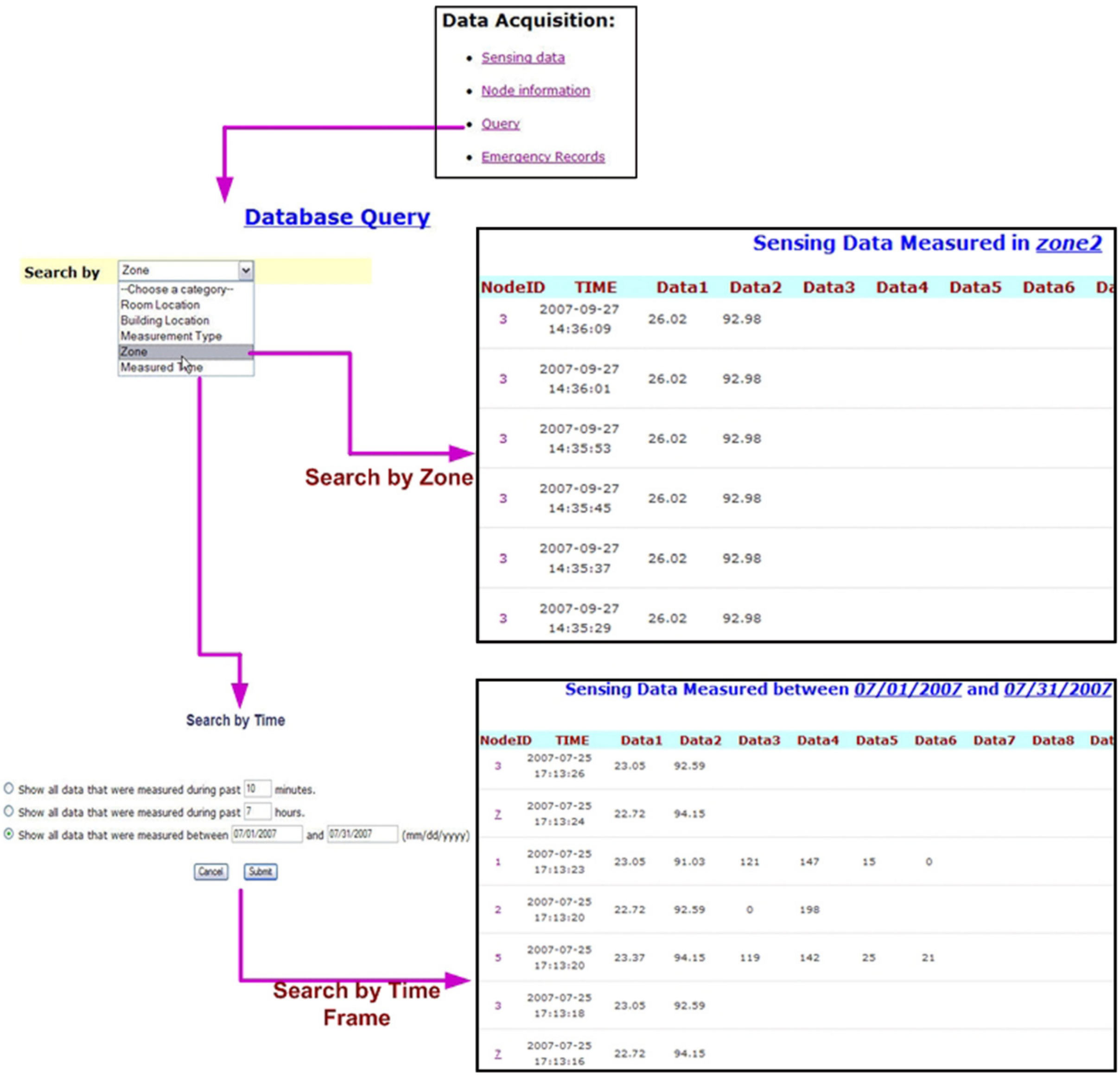

Fig. 5. Example indicating the function of the query menu (search by zone and time).

NodeLocation, contains location information for each sensor node. Describing the location within in a building is no trivial task considering the range of descriptions that one could have for a building. To describe the location of the building in which the sensor resides, fields are provided for the address, city, state, country, latitude, longitude, and building name. These fields are all optional so that the most popular method of identifying the building in a certain locale can be utilized. Within the building, node locations can be described by the room number, room name, floor number, zone number, or the equipment that is being monitored by the sensor (e.g. “Air Handler Unit #1”). These values are all stored as strings, so the user has flexibility in specifying according to his or her needs. This flexibility can cause problems with automating data exchange, but the first step in ensuring data interoperability lies in developing standard database structures for storing sufficient information about sensors as well as the data from those sensors. Future efforts into standardized vocabulary for these terms may further help interoperability.

4.4. Data retrieval

The question now becomes: how does one get to the data that are stored in the database? For this demonstration, we aimed to develop a system by which the data were accessible from anywhere, not just at the computer attached to the gateway. Software was, therefore, developed to access the sensor database over the Internet. The scripting software to access the database and display results was written in PHP [27]. It lets the user modify the database to add more sensor nodes or sensor boards and allows him to query the database to find necessary data.

The key parts of this web system include access to the data, information about each node in the network, and an image of the topology of the network. The site allows the user to both read data that have been recorded and to modify the sensor network by adding descriptions of nodes or sensor boards. These tasks are carried out in a familiar web environment.

Fig. 4 shows web pages that appear when one accesses the Sensing data or Node information items. Sensing data in reverse chronological order are presented when “Sensing data” is selected. A complete description of the location of and sensors on a node are presented when “Node information” is selected.

The query feature allows the user to select data from the database according to acquisition time, location, or measurement type. Fig. 5 shows the pages that enable such queries. The PHP script carries out the MySQL commands and directly communicates with the database that is being populated with data from the sensor networks. The server running the website in this case sits on the same computer as the database, and data are available as soon as the sensor network transmits the measurements to the computer.

The last menu entry, “Emergency records,” is provided to find instances in the database where adverse events, such as fires, leakage or power outage, could be flagged by predefined criteria. For example, the system does not issue any emergency notices under normal temperature conditions in the building. However, when the temperature value is over 50 °C in a particular location, then the system will

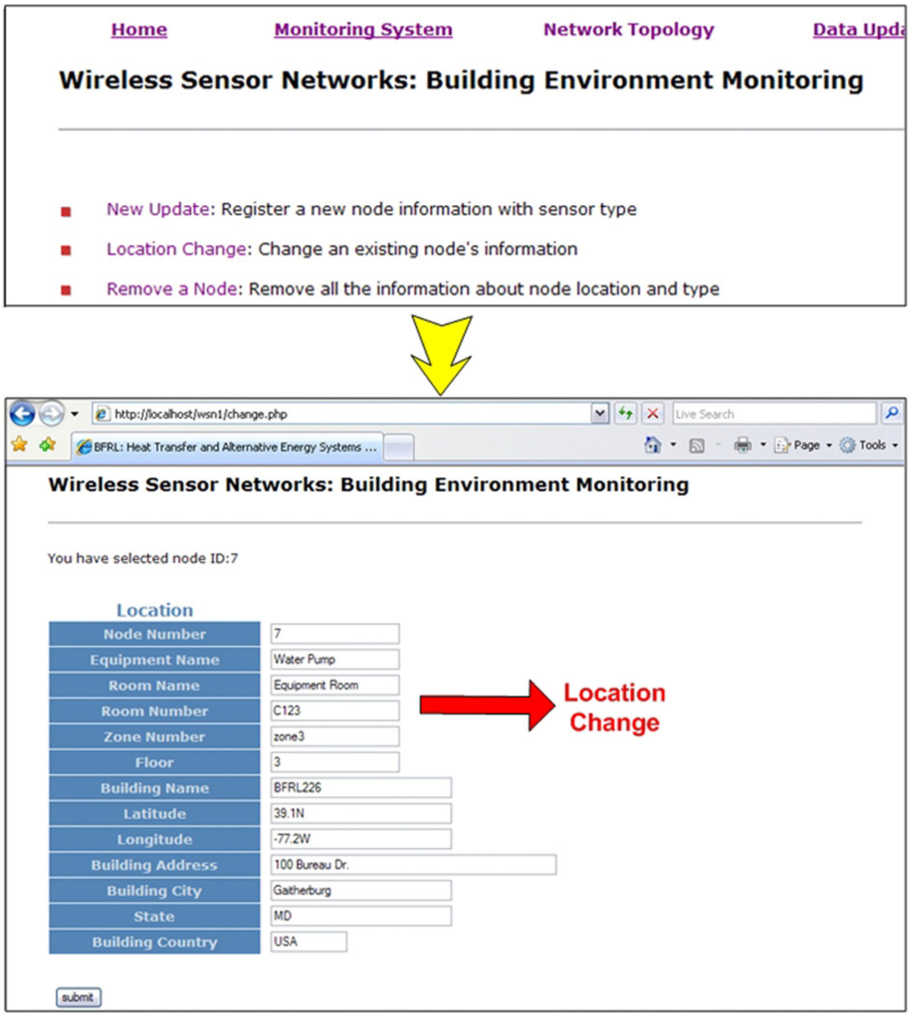

Fig. 6. Screenshot of node update.

flag that data value as a potential emergency situation. The threshold values can be modified as needed.

Another function implemented in the website is the ability to update the information on the sensor network, as shown in Fig. 6. This function is useful when nodes are moved to a different location, when a specific node number is no longer needed, or when a new node is introduced into the network. Only minor efforts were made in this work to examine security issues that arise when making the data available over the Internet; this issue will obviously be important in any actual implementation of a web-based monitoring system, so the user should consider ways to restrict access to the data. In this case, password protection was implemented for access to this website.

This setup proved to work quite well for our application. Data were easily accessed via the website, and sensor nodes could be added to the network with little trouble. The value of the database was demonstrated when a Web Service located on a remote computer mined the database to issue alerts as part of a separate project. The database structure allowed for easy determination of the location of an excessive temperature in the building or of a light level that was too low for adequate visibility. By breaking down the monitoring system into data acquisition, data storage, and data retrieval components, a flexible system was created that could be modified by users to meet their needs while allowing for advances in wireless sensing and networking technologies to augment monitoring capabilities without disrupting the operation of the entire system.

The benefits of a system such as the one described in this work lie in the ease with which engineers can obtain information from a building with finer granularity. As mentioned previously, the cost to run signal wires limits the number of sensors that can be installed in a building. That cost can be eliminated through wireless sensing, but the increased cost from the wireless hardware negates some of these savings. Forecasts call for wireless hardware prices to reach levels of around $4 per mote by 2010 [28], but off-the-shelf costs for the hardware used in this study (purchased in 2007) were approximately $100 per mote for radios that provide wireless capabilities to the sensors. While no data were found to quantify time savings, it is apparent that the use of wireless sensors will result in much more rapid deployment than would the use of wired sensors. The system used here would require some software development for the motes, but much of that code would be re-used since many motes will possess the same suite of sensors. By using freely available software tools, the only costs that will be incurred by the user will involve the cost of the wireless hardware and the cost of software development. With time, it is anticipated that the cost of the hardware will decrease rapidly, thereby permitting large numbers of sensors to be placed in and around buildings to monitor energy consumption, equipment performance, durability, and safety.

5. Summary

The discussion of this implementation is meant to demonstrate the steps needed to make use of advanced wireless sensor network technology. The end user will generally need to develop software for three purposes. First, the sensor hardware must be programmed to enable processing of the sensor's analog signals, to set the rate of measurement, to assign nodes to particular groups, and to package the data into a usable message. Fortunately for the end user, much of the software needed to control the radios and the routing of messages through the network has already been written, so he or she can focus on issues related to the end application.

An open-source platform for programming these nodes is the TinyOS operating system along with the nesC computer language. The second place where a software application is needed is in moving the data from the sensor network to a storage area. In this demonstration, a standard database server is used, and a Java application converts digital data coming over an RS-232 connection into engineering units and inserts the data into a database. Careful thought should be placed into structuring the database to ensure that all necessary information about the sensors is included. One such database design is provided here as a suggested method of structuring the data for building applications. To access the data, this work relies on a web-based viewer that allows a user to view the data via a conventional web browser. This user interface requires no advanced knowledge of programming, so the sensor network data will be available to users with limited skills in programming. To build this Web interface, conventional scripting tools can be used to access the database and make modifications to it as necessary. By describing the details of one such implementation to monitor the building environment, it is hoped that engineers will face less mystery when contemplating the use of wireless sensor networks to monitor conditions in and around buildings.

References

[1] Ali M. Malkawi, Ravi S. Srinivasan, A new paradigm for human-building interaction: the use of CDF and augmented reality, Automation in Construction, Elsevier 14 (1) (2005) 71–84.
[2] Z. Riaz, D.J. Edwards, A. Thorpe, SightSafety: a hybrid information and communication technology system for reducing vehicle/pedestrian collisions, Automation in Construction, Elsevier 15 (6) (2006) 719–728.
[3] M. Kintner-Meyer, M. Brambley, Pros & cons of wireless, ASHRAE Journal 44 (11) (2002) 54–61.
[4] Yang Wang, Jerome P. Lynch, Kincho H. Law, Validation of an integrated network system for real-time wireless monitoring of civil structures, Proceedings of the 5th International Workshop on Structural Health Monitoring, Stanford, CA, September 12–14, 2005, pp. 275–282.
[5] Jesus M. de la Ganza, Ivan Howitt, Wireless communication and computing at the construction jobsite, Automation in Construction, Elsevier 7 (4) (1998) 327–347.
[6] M. Kintner-Meyer, M.R. Brambley, T.A. Carlon, N.N. Bauman, Wireless sensors: technology and cost-savings for commercial buildings, Teaming for Efficiency: Proceedings of the 2002 ACEEE Summer Study on Energy Efficiency in Buildings, vol. 7, 2002, pp. 7.121–7.134.
[7] W.M. Healy, Lessons learned in wireless monitoring, ASHRAE Journal 47 (10) (2005) 54–58.
[8] J. Wills, Will HVAC control go wireless? ASHRAE Journal 46 (7) (2004) 46–52.
[9] J. Adams, Meet the ZigBee standard, Sensors 20 (6) (2003) 14–19.
[10] Jeff Raimo, Wireless mesh controller networks, ASHRAE Journal 48 (10) (2006) 34–38.
[11] John Ruiz, Going wireless, ASHRAE Journal 49 (6) (2007) 33–43.
[12] J.P. Lynch, An overview of wireless structural health monitoring for civil structures, Philosophical Transactions of the Royal Society of London. A 365 (2007) 345–372.
[13] G. Pottie, W. Kaiser, Wireless integrated network sensors, Communications of the ACM 43 (5) (2000) 51–58.
[14] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[15] D. Estrin, D. Culler, K. Pister, G. Sukhatme, Connecting the physical world with pervasive networks, IEEE Pervasive Computing 1 (1) (2002) 59–69.
[16] J. Hill, M. Horton, R. Kling, L. Krishnamurthy, The platforms enabling wireless sensor networks, Communications of the ACM 47 (6) (2004) 41–46.
[17] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, D. Estrin, Habitat monitoring with sensor networks, Communications of the ACM 47 (6) (2004) 34–40.
[18] Z. Aziz, C.J. Anumba, D. Ruikar, P. Carrillo, D. Bouchlaghem, Intelligent wireless web-services for construction — a review of the enabling technologies, Automation in Construction, Elsevier 15 (2) (2006) 113–123.
[19] American Society of Heating, Refrigerating, and Air-Conditioning Engineers Standard 135: BACnet - A Data Communication Protocol for Building Automation and Control Networks 2004.
[20] David E. Culler, TinyOS: operating system design for wireless sensor networks, Sensors 23 (5) (2006) 14–20.
[21] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, The nesC language: a holistic approach to network embedded systems, ACM SIGPLAN Notices 38 (5) (2003) 1–11.
[22] IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), The Institute of Electrical and Electronic Engineers Inc., New York 2003.
[23] Micaz 2.4 GHz Datasheet, Crossbow Technology Inc., Available online at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf, accessed in January 2008.
[24] Joseph Polastre, Robert Szewczyk, Cory Sharp, David Culler, The mote revolution: low power wireless sensor network devices, Symposium on High Performance Chips, August 22–24, 2004, Stanford, CA, 2004.
[25] Thomas Schmid, Henri Dubois-Ferriere, Martin Vetterli, SensorScope: experiences with a wireless building monitoring sensor network, Workshop on Real-World Wireless Sensor Networks, Stockholm, Swenden, June 20–21, 2005.
[26] "MySQL AB: The world's most popular open source database" www.mysql.com.
[27] "PHP: Hypertext Preprocessor" www.php.net.
[28] M. Kintner-Meyer, R. Conant, Opportunities of Wireless Sensors and Controls for Building Operation, ACEEE Summer Study on Energy Efficiency in Buildings, 2004.