



Wireless sensor networks for personal health monitoring: Issues and an implementation

Aleksandar Milenković *, Chris Otto, Emil Jovanov

Electrical and Computer Engineering Department, The University of Alabama in Huntsville, 301 Sparkman Drive, Huntsville, AL 35899, USA

Abstract

Recent technological advances in sensors, low-power integrated circuits, and wireless communications have enabled the design of low-cost, miniature, lightweight, and intelligent physiological sensor nodes. These nodes, capable of sensing, processing, and communicating one or more vital signs, can be seamlessly integrated into wireless personal or body networks (WPANs or WBANs) for health monitoring. These networks promise to revolutionize health care by allowing inexpensive, non-invasive, continuous, ambulatory health monitoring with almost real-time updates of medical records via the Internet. Though a number of ongoing research efforts are focusing on various technical, economic, and social issues, many technical hurdles still need to be resolved in order to have flexible, reliable, secure, and power-efficient WBANs suitable for medical applications. This paper discusses implementation issues and describes the authors' prototype sensor network for health monitoring that utilizes off-the-shelf 802.15.4 compliant network nodes and custom-built motion and heart activity sensors. The paper presents system architecture and hardware and software organization, as well as the authors' solutions for time synchronization, power management, and on-chip signal processing.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Wireless sensor networks; Health monitoring; Hardware; Software; Signal processing; Time synchronization; Energy efficiency

1. Introduction

Current health care systems – structured and optimized for reacting to crisis and managing illness – are facing new challenges: a rapidly growing population of elderly and rising health care spending. According to the U.S. Bureau of the Census, the number of adults age 65–84 is expected to double from 35 million to nearly 70 million by 2025 when the youngest Baby Boomers retire. This trend is global, so the worldwide population over age 65 is expected to more than double from 357 million in 1990 to 761 million in 2025. Also, overall health care expenditures in the United States reached \$1.8 trillion in 2004 with almost 45 million Americans uninsured. In addition, a recent study found that almost one third of U.S. adults, most of whom held full-time jobs, were serving as informal caregivers – mostly

to an elderly parent. It is projected that health care expenditures will reach almost 20% of the Gross Domestic Product (GDP) in less than 10 years, threatening the wellbeing of the entire economy [1]. All these statistics suggest that health care needs a major shift toward more scalable and more affordable solutions. Restructuring health care systems toward proactive managing of wellness rather than illness, and focusing on prevention and early detection of disease emerge as the answers to these problems. Wearable systems for continuous health monitoring are a key technology in helping the transition to more proactive and affordable healthcare.

Wearable health monitoring systems allow an individual to closely monitor changes in her or his vital signs and provide feedback to help maintain an optimal health status. If integrated into a telemedical system, these systems can even alert medical personnel when life-threatening changes occur. In addition, patients can benefit from continuous long-term monitoring as a part of a diagnostic procedure, can achieve optimal maintenance of a chronic condition,

* Corresponding author.

E-mail addresses: milenka@ece.uah.edu (A. Milenković), chrisaotto@yahoo.com (C. Otto), jovanov@ece.uah.edu (E. Jovanov).

or can be supervised during recovery from an acute event or surgical procedure. Long-term health monitoring can capture the diurnal and circadian variations in physiological signals. These variations, for example, are a very good recovery indicator in cardiac patients after myocardial infarction [2]. In addition, long-term monitoring can confirm adherence to treatment guidelines (e.g., regular cardiovascular exercise) or help monitor effects of drug therapy. Other patients can also benefit from these systems; for example, the monitors can be used during physical rehabilitation after hip or knee surgeries, stroke rehabilitation, or brain trauma rehabilitation.

During the last few years there has been a significant increase in the number of various wearable health monitoring devices, ranging from simple pulse monitors [3,4], activity monitors [5,6], and portable Holter monitors [7], to sophisticated and expensive implantable sensors [8]. However, wider acceptance of the existing systems is still limited by the following important restrictions. Traditionally, personal medical monitoring systems, such as Holter monitors, have been used only to collect data. Data processing and analysis are performed offline, making such devices impractical for continual monitoring and early detection of medical disorders. Systems with multiple sensors for physical rehabilitation often feature unwieldy wires between the sensors and the monitoring system. These wires may limit the patient's activity and level of comfort and thus negatively influence the measured results [9]. In addition, individual sensors often operate as stand-alone systems and usually do not offer flexibility and integration with third-party devices. Finally, the existing systems are rarely made affordable.

Recent technology advances in integration and miniaturization of physical sensors, embedded microcontrollers, and radio interfaces on a single chip; wireless networking; and micro-fabrication have enabled a new generation of wireless sensor networks suitable for many applications. For example, they can be used for habitat monitoring [10], machine health monitoring and guidance, traffic pattern monitoring and navigation, plant monitoring in agriculture [11], and infrastructure monitoring. One of the most exciting application domains is health monitoring [12,13]. A number of physiological sensors that monitor vital signs, environmental sensors (temperature, humidity, and light), and a location sensor can all be integrated into a Wearable Wireless Body/Personal Area Network (WWBAN) [14]. The WWBAN consisting of inexpensive, lightweight, and miniature sensors can allow long-term, unobtrusive, ambulatory health monitoring with instantaneous feedback to the user about the current health status and real-time or near real-time updates of the user's medical records. Such a system can be used for computer-supervised rehabilitation for various conditions, and even early detection of medical conditions. For example, intelligent heart monitors can warn users about impending medical conditions [15] or provide information for a specialized service in the case of catastrophic events [16]. Accelerometer-

based monitoring of physical activity with feedback can improve the process of physical rehabilitation [17]. When integrated into a broader telemedical system with patients' medical records, the WWBAN promises a revolution in medical research through data mining of all gathered information. The large amount of collected physiological data will allow quantitative analysis of various conditions and patterns. Researchers will be able to quantify the contribution of each parameter to a given condition and explore synergy between different parameters, if an adequate number of patients is studied in this manner.

In this paper, we describe a general WWBAN architecture as well as our prototype WWBAN designed using Telos motes [18] and application-specific signal conditioning modules. The prototype consists of several motion sensors that monitor the user's overall activity and an ECG sensor for monitoring heart activity. This paper details our hardware and software platforms for medical monitoring, discusses open issues, and introduces our solutions for time-synchronization, efficient on-sensor signal processing, and an energy-efficient communication protocol.

Section 2 describes the general WWBAN architecture, its integration into a telemedical system, and possible deployment configurations. Section 3 describes our WWBAN prototype, including hardware design of the sensor platform ActiS, as well as corresponding software modules. Section 4 describes our implementation of a time-synchronization protocol. Section 5 describes power measurements and the resulting power profiles of our WWBAN prototype. Section 6 concludes the paper.

2. WWBAN architecture

WWBANs are a pivotal part of a multi-tier telemedicine system as illustrated in Fig. 1. Tier 1 encompasses a number of wireless medical sensor nodes that are integrated into a WWBAN. Each sensor node can sense, sample, and process one or more physiological signals. For example, an electrocardiogram sensor (ECG) can be used for monitoring heart activity, an electromyogram sensor (EMG) for monitoring muscle activity, an electroencephalogram sensor (EEG) for monitoring brain electrical activity, a blood pressure sensor for monitoring blood pressure, a tilt sensor for monitoring trunk position, and a breathing sensor for monitoring respiration; and motion sensors can be used to discriminate the user's status and estimate her or his level of activity.

Tier 2 encompasses the personal server (PS) application running on a Personal Digital Assistant (PDA), a cell phone, or a home personal computer. The PS is responsible for a number of tasks, providing a transparent interface to the wireless medical sensors, an interface to the user, and an interface to the medical server. The interface to the WWBAN includes the network configuration and management. The network configuration encompasses the following tasks: sensor node registration (type and number of sensors), initialization (e.g., specify sampling frequency

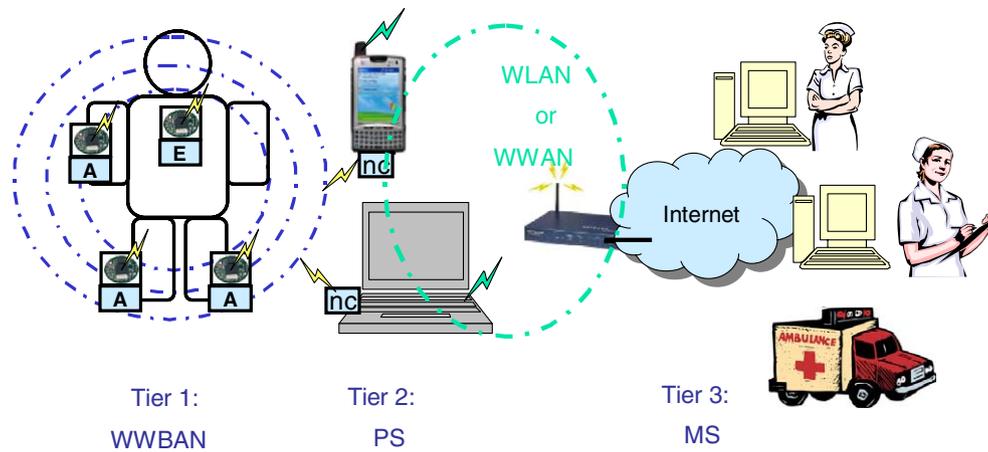


Fig. 1. WWBAN integrated into a telemedical system for health monitoring.

and mode of operation), customization (e.g., run user-specific calibration or user-specific signal processing procedure upload), and setup of a secure communication (key exchange). Once the WWBAN network is configured, the PS application manages the network, taking care of channel sharing, time synchronization, data retrieval and processing, and fusion of the data. Based on synergy of information from multiple medical sensors the PS application should determine the user's state and his or her health status and provide feedback through a user-friendly and intuitive graphical or audio user interface. Finally, if a communication channel to the medical server is available, the PS establishes a secure link to the medical server and sends reports that can be integrated into the user's medical record. However, if a link between the PS and the medical server is not available, the PS should be able to store the data locally and initiate data uploads when a link becomes available.

Tier 3 includes a medical server(s) accessed via the Internet. In addition to the medical server, the last tier may encompass other servers, such as informal caregivers, commercial health care providers, and even emergency servers. The medical server typically runs a service that sets up a communication channel to the user's PS, collects the reports from the user, and integrates the data into the user's medical record. The service can issue recommendations, and even issue alerts if reports seem to indicate an abnormal condition. More details about this architecture and services can be found in [14].

2.1. Deployment scenarios

Fig. 2 illustrates three typical scenarios using WWBAN. The configuration on the left can be deployed at home, in the workplace, or in hospitals. Wireless medical sensors attached to the user send data to a PDA, forming a short-range wireless network (e.g., IEEE 802.15.1 or 802.15.3/4). The PDA equipped with a WLAN interface (e.g., IEEE 802.11a/b/g) transmits the data to the home (central) server. The home server, already connected to

the Internet, can establish a secure channel to the medical server and send periodic updates for the user's medical record.

The modified configuration in the middle is optimized for home health care. The sensor network coordinator (nc in Fig. 2) is attached to the home personal server that runs the PS application. The medical sensor nodes and the network coordinator form a wireless personal area network. By excluding the PDA, we can reduce system cost. However, this setting is likely to require more energy spent for communication due to an increased RF output power and lower Quality of Service (QoS), requiring frequent retransmissions.

The configuration on the right illustrates ambulatory monitoring applicable any time and everywhere – the PS application runs on a Wireless Wide Area Network (WWAN) enabled PDA/cell phone (e.g., 2G, 2.5G, and 3G) that connects directly to the medical server. To illustrate, we will follow a user (let's call him Sam) who has a predisposition for heart disorders. Sam continues his normal daily activities, but now equipped with several non-invasive medical sensors applied in his clothes or as tiny patches on his skin. The PS application can discriminate Sam's state (walking, running, lying down, sitting, riding, etc.) using the data from the motion sensors, and can recognize an arrhythmic event, analyzing the data from the ECG sensor. If Sam's activity, heart rate, and personalized thresholds indicate an abnormal condition, he will receive a warning. In addition, a precise incident report can be sent to the medical server at Sam's hospital or doctor's office.

2.2. Requirements for wireless medical sensors

Wireless medical sensors should satisfy the main requirements such as *wearability*, *reliability*, *security*, and *interoperability*.

2.2.1. Wearability

To achieve non-invasive and unobtrusive continuous health monitoring, wireless medical sensors should be

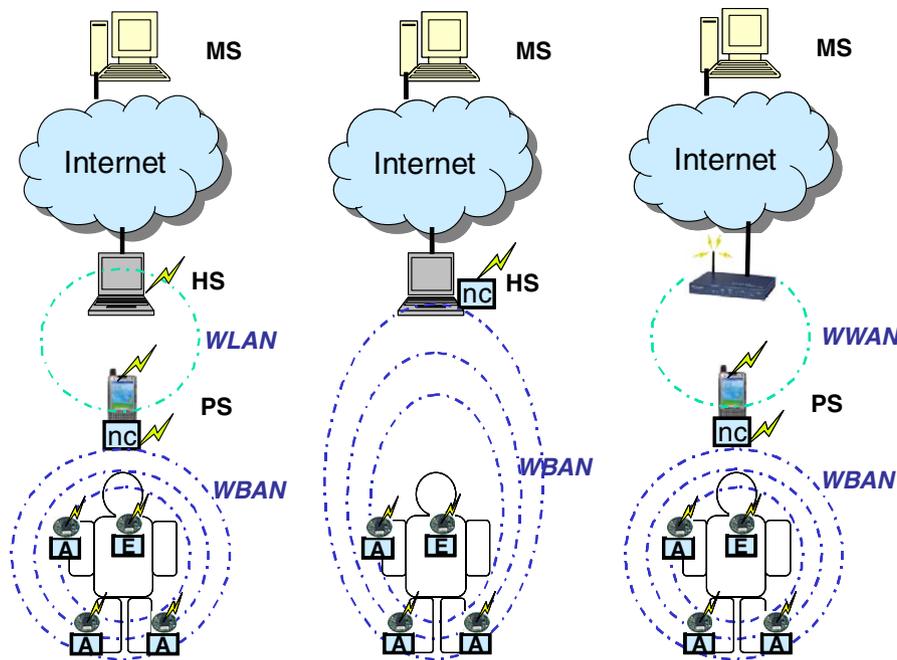


Fig. 2. WWBAN deployment scenarios. Legend: A, activity sensor; E, heart sensor; WBAN, wireless body area network; PS, personal server; WLAN, wireless local area network; WAN, wide area network; HS, home server; MS, medical server; nc, WBAN network coordinator.

lightweight and small. The size and weight of sensors is predominantly determined by the size and weight of batteries. But then, a battery's capacity is directly proportional to its size. We can expect that further technology advances in miniaturization of integrated circuits and batteries will help designers to improve medical sensor wearability and the user's level of comfort. Section 5 further discusses energy efficiency in WWBAN.

2.2.2. Reliable communication

Reliable communication in WWBANs is of utmost importance for medical applications that rely on WWBANs. The communication requirements of different medical sensors vary with required sampling rates, from less than 1 to 1000 Hz. One approach to improve reliability is to move beyond telemetry by performing on-sensor signal processing. For example, instead of transferring raw data from an ECG sensor, we can perform feature extraction on the sensor, and transfer only information about an event (e.g., QRS features and the corresponding timestamp of R-peak). In addition to reducing heavy demands for the communication channel, the reduced communication requirements save on total energy expenditures, and consequently increase battery life. A careful trade-off between communication and computation is crucial for optimal system design.

2.2.3. Security

Another important issue is overall system security. The problem of security arises at all three tiers of a WWBAN-based telemedical system. At the lowest level, wireless medical sensors must meet privacy requirements mandated by the law for all medical devices and must guar-

antee data integrity. Though key establishment, authentication, and data integrity are challenging tasks in resource constrained medical sensors, a relatively small number of nodes in a typical WWBAN and short communication ranges make these tasks achievable.

2.2.4. Interoperability

Wireless medical sensors should allow users to easily assemble a robust WWBAN depending on the user's state of health. Standards that specify interoperability of wireless medical sensors will promote vendor competition and eventually result in more affordable systems.

3. WWBAN prototype

In order to better understand various issues in designing a wearable wireless sensor network for health monitoring, we ventured into the development of a prototype system aimed to satisfy the above-mentioned requirements for small size, low power consumption, secure communication, and interoperability. Our WWBAN prototype consists of multiple ActiS sensor nodes that are based on a commonly used sensor platform and custom sensor boards [14,19]. The initial WWBAN setting includes a sensor node that monitors both ECG activity and the upper body trunk position and two motion sensors attached to the user's ankles to monitor activity. Such a WBAN allows one to assess metabolic rate and cumulative energy expenditure as valuable parameters in the management of many medical conditions and correlate that data with heart activity. Fig. 3 shows heart activity and acceleration data collected by our prototype during normal walking with a motion sensor attached to the right ankle.

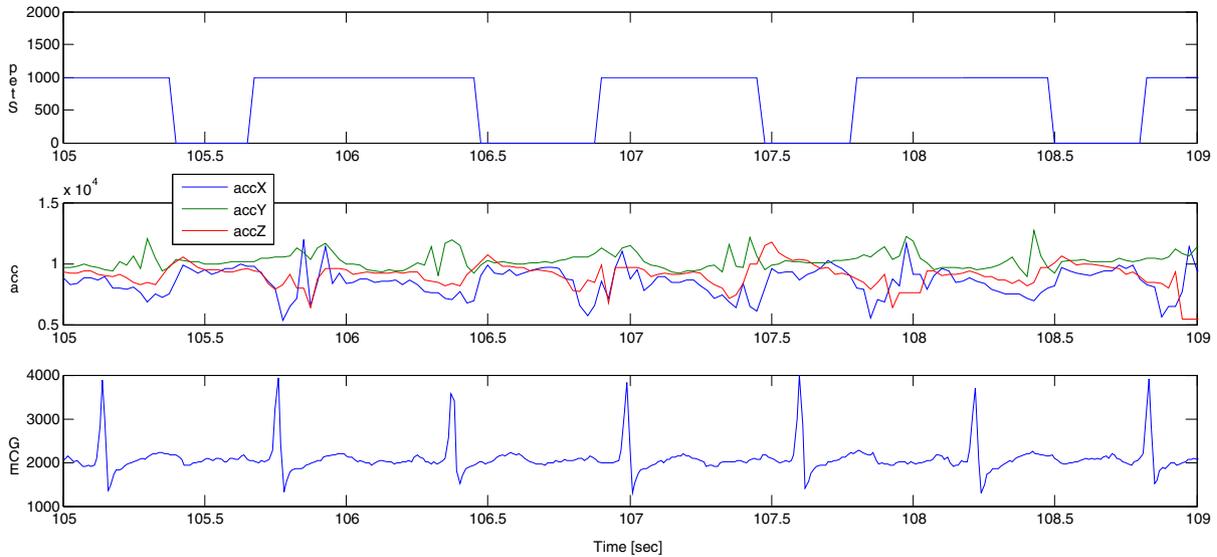


Fig. 3. ECG, acceleration, and foot switch data collected by the WWBAN prototype during normal walking.

3.1. Hardware platform

The ActiS sensor node features a hierarchical organization employed to offer a rich set of functions, benefit from the open software system support, and perform computation and communications tasks with minimal power consumption. Each ActiS node utilizes a commercially available wireless sensor platform Telos from Moteiv [18] and a custom intelligent signal processing daughter card attached to the Telos platform (Fig. 4). The daughter boards interface directly with physical sensors and perform data sampling and in some cases preliminary signal processing. The pre-processed data is then transferred to the Telos board. The Telos platform can support more sophisticated real-time analysis and can perform additional filtering, characterization, feature extraction, or pattern recognition. The Telos platform is also responsible for time synchronization, communication with the network coordinator, and secure data transmission.

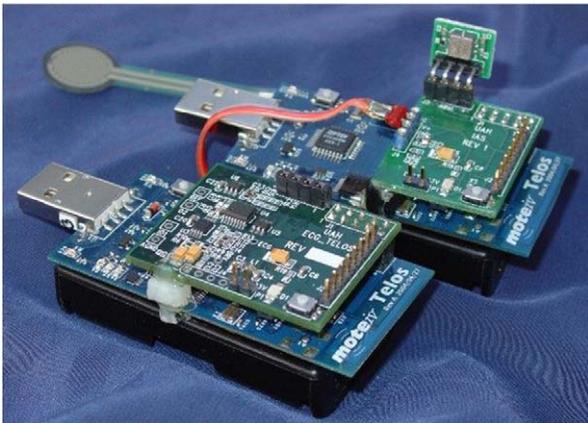


Fig. 4. Photo of two ActiS sensors with customized daughter boards.

Fig. 5 shows a block diagram of an ActiS with a Telos platform and a custom *Intelligent Signal Processing Module* (ISPM). Telos is powered by two AA batteries and features an ultra-low power Texas Instruments MSP430 microcontroller [20]; a Chipcon CC2420 radio interface in the 2.4 GHz band [21]; an integrated onboard antenna with 50 m range indoors/125 m range outdoors; a USB port for programming and communication; an external flash memory; and integrated humidity, temperature, and light sensors. The MSP430 microcontroller is based around a 16-bit RISC core integrated with RAM and flash memories, analog and digital peripherals, and a flexible clock subsystem. It supports several low-power operating modes and consumes as low as 1 μ A in a standby mode; it also has very fast wake up time of no more than 6 μ s. Telos Revision A features a MS430F149 microcontroller with 2 KB RAM and 60 KB flash memory; Telos Revision B (now TmoteSky) features a MSP430F1611 with 10 KB of RAM and 48 KB of flash memory. The CC2240 wireless transceiver is IEEE 802.15.4 compliant and has programmable output power, maximum data rate of 250 Kbs, and hardware support for error correction and encryption. The CC2240 is controlled by the MSP430 microcontroller through the Serial Peripheral Interface (SPI) port and a series of digital I/O lines with interrupt capabilities. The Telos platform features a 10-pin expansion connector with one UART (Universal Asynchronous Receiver Transmitter) and one I²C interface, two general-purpose I/O lines, and three analog input lines.

We developed two custom boards specifically for health monitoring applications, an ISPM and an IAS (Intelligent Activity Sensor). The ISPM board extends the capabilities of Telos by adding two perpendicular dual axis accelerometers (Analog Devices ADXL202), a bioamplifier with signal conditioning circuit, and a microcontroller MSP430F1232 (Fig. 5). The ISPM's two ADXL202 accelerometers cover all three axes of motion. One ADXL202 is

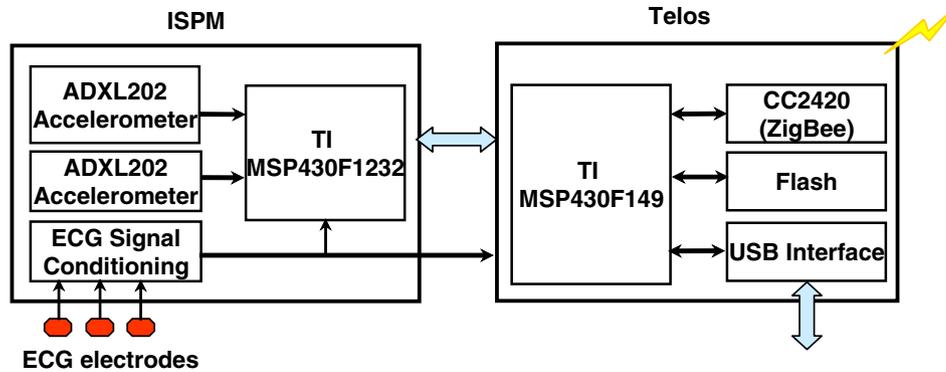


Fig. 5. Block diagram of an ActiS sensor node.

mounted directly on the ISPM board and collects data for the X and Y axes. The second ADXL202 is mounted on a card that extends vertically from the ISPM and collects acceleration data on the Z axis. The user's physiological state can be monitored using an on-board bioamplifier implemented with an instrumentation amplifier and signal conditioning circuit. The bioamplifier could be used for electromyogram or electrocardiogram monitoring. The output of the signal conditioning circuit is connected to the local microcontroller as well as to the microcontroller on the Telos board via the expansion connector. The ISPM has its own MSP430F1232 processor for sampling and low-level data processing, selected primarily for its compact size and excellent MIPS/mW ratio. Other features that were desirable for this design were the 10-bit ADC and the timer capture/compare registers that are used for acquisition of data from accelerometers. The MSP430F1232 also has a hardware UART that is used for communications with the Telos board. The IAS board is a stripped-down version of the ISPM with only accelerometer sensors and signal conditioning for a force-sensing resistor that can be used as a foot switch.

3.2. Software organization

The system software is implemented in a TinyOS environment [22]. TinyOS is a lightweight open source operating system for wireless embedded sensors. It is designed to use minimal resources, and its configuration is defined at compile time by combining components from the TinyOS library and custom-developed components. A TinyOS application is implemented as a set of component modules written in nesC [23]. The nesC language extends the C language with new constructs to facilitate the component architecture and multitasking. By adding direct language support for task synchronization and task management, it allows rapid development and minimizes resource usage. Fig. 6 shows a generalized WWBAN software architecture, and from top to bottom, it shows the network coordinator software, WWBAN node's Telos software, and WWBAN node's daughter card software.

3.2.1. Network coordinator

The network coordinator is also implemented on a Telos platform. It feeds the PS application through its USB connector and manages the WWBAN – transmits the messages from the PS that establish a session, assigns the individual sensor ID, distributes keys if secure data are encrypted, and assigns communication slots. The network coordinator autonomously emits beacon messages for time synchronization. After the initial setup, it receives data from individual sensors, aggregates the data, and forwards it to the PS application.

3.2.2. Telos software

The Telos application software is implemented as multiple TinyOS components encompassing the following high-level functions: wireless communication, extended flash storage, messaging software, board-to-board communications, and signal feature extraction. Telos serves as a master controller, and it requests data from the daughter sensor card every 40 ms (25 Hz) by raising an interrupt request line. The daughter sensor card sends preprocessed data via an asynchronous serial interface. The received data can also be processed and analyzed. For example, motion sensors can analyze acceleration signals to identify the moment when a step has been made. A step detection event and the corresponding time stamp are sent to the personal server. As an alternative, we can upload raw data from accelerometers at the price of increased power consumption. The processed data set can be stored in an external serial flash memory in the case of autonomous operation or if the wireless channel is not available.

It should be noted that the flash memory, CC2420 radio interface, and the daughter sensor card all share a single serial interface of the MSP30 on the Telos platform. This presented its own set of challenges since the Telos platform is tasked with reliable communications to multiple devices using this single serial interface. For example, to communicate with the daughter card, the software must configure the serial interface as a UART running at 115.2 kbps. Once sensor data is received, the serial interface is dynamically reconfigured for SPI at 500 kbps, allowing communications to both the on-board radio and flash. Because events are

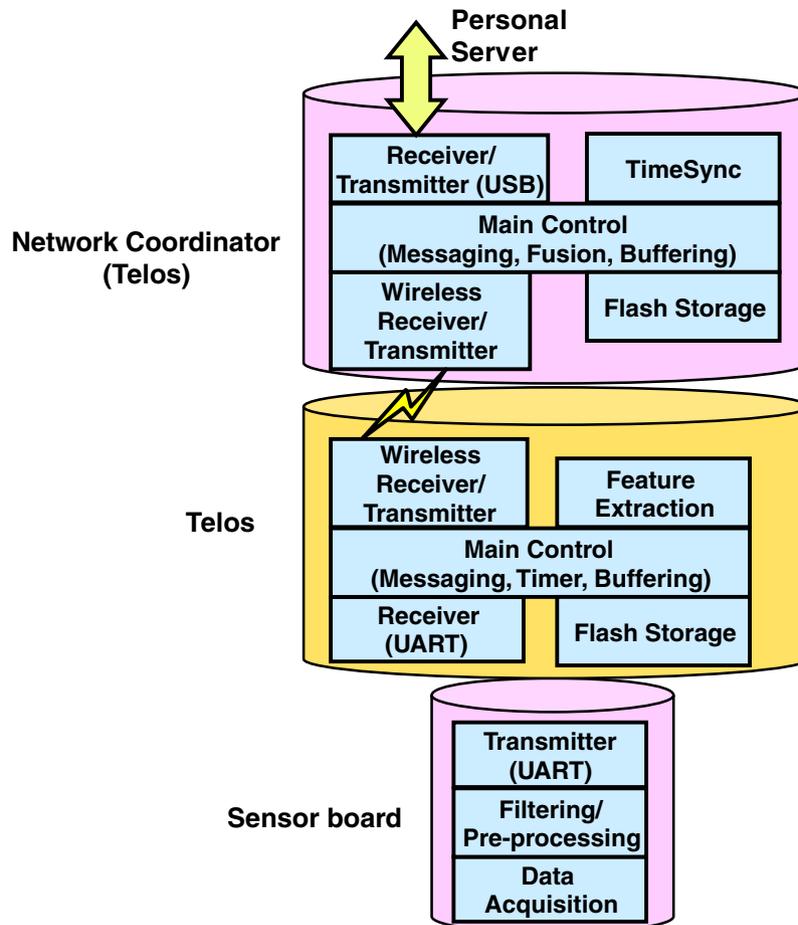


Fig. 6. WBAN software architecture.

recognized asynchronously, accurate event time stamps can be made, but often the messages must be buffered and queued for transmission when the serial interface is available.

3.2.3. Sensor software

The sensor boards handle acquisition of physiological signals and pre-processing. For example, the ISPM samples three independent accelerometer axes each at a rate of 200 Hz. The raw accelerometer data is filtered and pre-processed. The filtering includes moving an average filter to eliminate high frequency movement artifacts, and separation of low and high frequency components of the acceleration signal. Sensor orientation can be calculated as the angle between low frequency accelerometer components. User activity is estimated with a function based on the sum of the integrals of the AC components in each channel [17].

4. Time synchronization

Time synchronization is a common requirement for wireless sensor networks since it allows collective signal processing, sensor and source localization, data aggrega-

tion, and distributed sampling. In wireless body area networks, synchronized time stamps are critical for proper correlation of data coming from different sensors and for an efficient sharing of the communication channel. For example, our prototype needs to synchronize and time-stamp data from motion sensors and the heart sensor, and establish a protocol for sharing the communication channel. Precise time stamping is also important in the case of intermittent communication that can significantly postpone transmission of event messages.

A synchronization mechanism for a given application is determined by the following: (i) the degree of precision needed, (ii) the longevity of synchronization, that is, whether we need to stay synchronized all the time or just when needed, (iii) the resources available (clocks), and (iv) the power and time budget available for achieving time synchronization. A number of protocols and algorithms have been proposed and implemented to provide time synchronization in computer networks. However, they are often ill-suited for wireless sensor networks since they require significant computing resources and do not offer fault-tolerant solutions. Several protocols have been specifically developed for wireless sensor networks [24–26,28,29].

One of the key protocols for time synchronization in WSNs is the Flooding Time Synchronization Protocol (FTSP) developed at Vanderbilt University [29]. It features MAC layer time stamping for increased precision and skew compensation with linear regression to account for clock drift. The FTSP generates time synchronization with periodic time sync messages. The network can dynamically elect a root node. Whenever a node receives a time sync message, it rebroadcasts the message, thus flooding the network with time sync messages. The message itself contains a very precise timestamp of when the message was sent. The receiving node takes an additional local timestamp when it receives the message. Because the timestamps are taken deep in the radio stack, they eliminate non-deterministic error sources and only include highly deterministic events such as air propagation time, radio transmission, and radio reception time. Comparing the timestamps from the last several messages received, the node computes a simple linear regression to allow it to account for the offset difference in its clock from global time as well as the relative difference in frequency of local oscillators. This enables each node to maintain an accurate estimate of global time despite the fact that its local clock may be running slightly faster or slower than the global clock source.

We modified the original FTSP for WWBAN settings [30]. Our modification exploits the ZigBee [31] star network topology to further minimize resources needed for time synchronization. The prototype features a master–slave hierarchy where the network coordinator periodically transmits a beacon message to its slave nodes to maintain

the synchronized communication link; a slave node receives the beacon without re-transmission. This highly accurate time synchronization allows a slave node to disable its radio and enter a low power sleep mode, waking up just before the next message is due. In addition, we allowed an original implementation where a root can be dynamically chosen and the network flooded by the sync messages in the case that the network coordinator fails or is turned off.

For time synchronization to work, there must be a fixed point in time from which both sender and receiver can reference the timestamp in a given message. For a ZigBee message, this point is at the end of the Start of Frame Delimiter (SFD). Fig. 7 shows the interface between the Chipcon's CC2420 radio transceiver and the microcontroller at the top, and the IEEE 802.15.4 physical frame format and corresponding pin activity at the bottom. On the Telos platform, the wireless transceiver provides a signal to the microcontroller to indicate when the SFD byte has been received or transmitted. As soon as the SFD byte is transmitted, the radio transceiver raises the SFD pin that initiates a timer capture and an interrupt request on the microcontroller. This allows the microcontroller to get a timestamp immediately after the SFD byte is transmitted. This timestamp is then inserted into the current time synchronization message (Fig. 8). It should be noted that the message has already begun transmission when the timestamp is made and added to the message. In a similar way, the receiver makes a local timestamp when it receives the SFD and stores it with the time synchronization message (Fig. 8). Later, the microcontroller will compare the

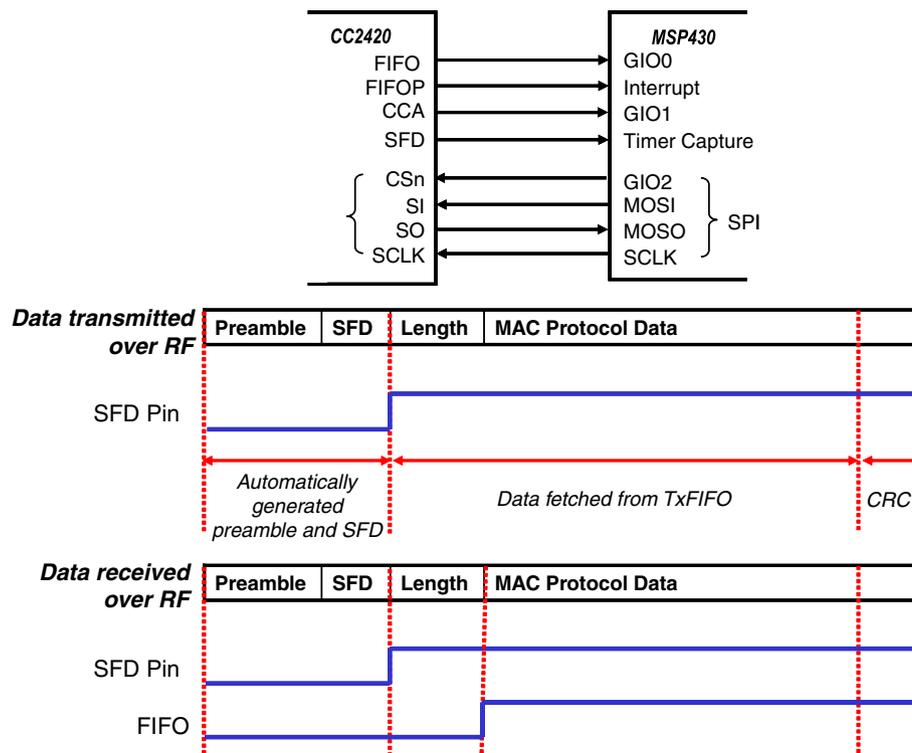


Fig. 7. Interfacing CC2420 RF transceiver (top). IEEE 802.15.4 frame format and pin activity (bottom).

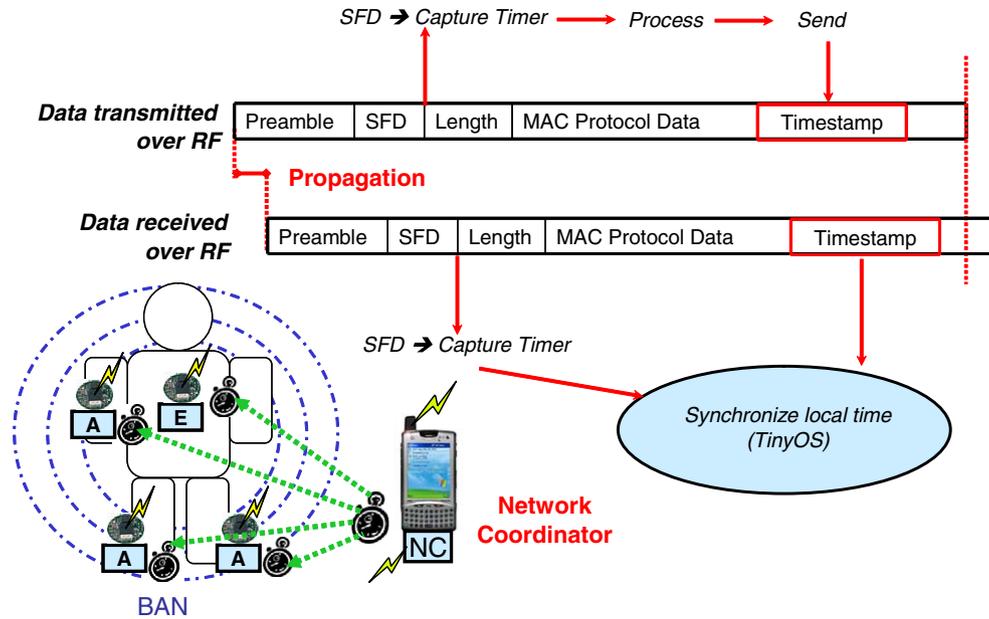


Fig. 8. Time synchronization in WBAN prototype.

two timestamps to determine the offset between the local time and the global time, and adjust the degree to which the local clock is running faster or slower than the global clock (the network coordinator global time).

This is a platform-specific implementation of the original FTSP mechanism, and it results in very accurate timestamps deep in the radio stack [30]. Unlike the original implementation of the FTSP on Mica2 boards where the processor directly controls the wireless bit-stream, the Telos's wireless controller offloads channel access from the main processor. In addition, the CC2420 provides flexible receive and transmit FIFO buffers and communicates with the main processor using a synchronous peripheral interface (SPI). Generally, when a message is being transmitted, the processor loads up the transmit FIFO with the entire message and then enables transmission. However, the FTSP messages contain a timestamp that is generated after the message has begun transmission. To implement this on the Telos platform, most of the message is placed in the FIFO and transmission is enabled. When the SFD interrupt occurs, the captured timer value is retrieved and converted to a global timestamp. The timestamp is inserted into the message and the rest of the message is placed in the FIFO. Assuming this can all be done quickly enough, the entire message is transmitted properly. If however the process is too slow, the FIFO will under-run and the message transmission will fail. Therefore, the speed must be checked for each individual application since ZigBee specifies a fairly speedy effective bit rate of 250 kbps.

The SPI interface between the microcontroller and the wireless controller on the Telos platform runs at 500 kbps, twice as fast as the message is transmitted over the radio. Experimental time measurements were taken to ensure adequate margin and instill confidence that our protocol can run reliably without FIFO under-run. It takes about

700 μ s to transmit a time sync message and about 150 μ s to calculate and insert the timestamp. It takes another 300 μ s to send the second part of the message over the SPI and finish filling the FIFO. All of this means that the entire message makes it into the FIFO with about 250 μ s to spare, which is adequate for our application.

Software support for time synchronization is implemented as a nesC interface that provides applications access to the time synchronization information. It can give the current global time, convert a local timestamp to global time, or calculate wait period until a future global time. This last facility is useful for an application that wants to do something at a specific global time. For example, we use it to put the radio transceiver in the sleep mode until the next communication window. If the next window will occur at a particular global time, the node can find out how long it will have to wait and then set a timer accordingly so it can wake up and enable the radio.

All timestamps and clocks were based on the 32768 Hz crystal on the Telos board. The crystal exhibits good short-term stability, an essential quality for our WWBAN prototype to work properly. Though the low frequency crystal does not allow a very high-resolution synchronization, the period of 30.5 μ s is quite satisfactory for typical WWBAN applications. It should be noted that the MSP340 microcontroller has a sophisticated clock module with a digitally controlled oscillator (DCO) with clock frequencies up to 5 MHz. Though a DCO's clock short-term stability is rather poor, an online modulation and clock stabilization can be employed if higher resolution is needed.

For testing of time synchronization protocol, we developed a test bed where the network coordinator and WWBAN slave sensors are all connected to a common wired signal connected to an MSP430's digital I/O port with timer capture capability. All nodes are configured to

report their respective global timestamps every time the common signal changes its state. For each event, the master's timestamp was compared to the slave's timestamp to determine the absolute error of the slave's timekeeping.

Testing the prototype indicated very good time synchronization. A number of experiments have been run varying the frequency of synchronization messages and test duration. In most cases the slave node's error was within ± 1 tick ($30.5 \mu\text{s}$), and an average error was approximately $0.1 * T_{\text{cycle}}$ or $3 \mu\text{s}$. It is expected that the more frequently time sync messages are sent, the better the network nodes will be able to maintain synchronization. Somewhat surprisingly, we discovered that increasing the time between messages improves time synchronization, because nodes are better able to estimate clock skew. However, the cost of larger inter-beacon periods is an increase in algorithm conversion time. Consequently, we implemented a hybrid scheme where messages are sent more often. During convergence, nodes will process every beacon. After a coarse convergence is achieved, nodes will begin to process every N-th beacon, allowing for a more precise skew estimation.

5. Energy efficiency

Energy consumption is a first class design constraint in wireless sensor networks since they are battery operated. To extend each node's lifetime, it is necessary to reduce power dissipation as much as possible; dissipation below $100 \mu\text{W}$ will enable operation on energy scavenged from the environment. Various design trade-offs between communication and on-sensor computation, collaborative protocols, and hierarchical network organization can yield significant energy savings. Once the sensor network is deployed, dynamic power management techniques can be employed in order to maximize battery life.

In WWBAN systems, reducing total power consumption is crucial for several reasons. Size and weight of sensors are predominantly determined by the size and weight of the batteries. On the other hand, a battery's capacity is directly proportional to its size. Consequently, WWBAN sensor nodes need to be extremely energy efficient, since reducing energy requirements will allow designers to use

smaller batteries. Smaller batteries will result in further miniaturization of physiological sensors and, in turn, an increased level of user's comfort. Second, an extended period of operation without battery changes is desirable, because frequent battery changes on multiple sensors are likely to hamper users' acceptance. In addition, longer battery life will decrease WWBAN operational costs.

We have designed a custom, application-specific protocol according to 802.15.4 recommendations [27]. In order to satisfy medical application requirements, the network protocol specifies a 1-s super frame cycle ($T_{\text{SFC}} = 1 \text{ s}$) and each slave node has its reserved time slot of 50 ms to transmit the data (Fig. 9). A super frame cycle starts with a beacon message sent by the network coordinator; the beacon message carries time synchronization information. Each sensor node wakes its radio interface up in a receive mode immediately before the next expected beacon.

Fig. 10 shows the power profiles recorded for a motion sensor using an environment for real-time power monitoring [32]. We can clearly identify three distinct states: Listen, Transmit, and Inactive modes. As described in Section 3, the ultra-low power microcontroller on the daughter board samples 3-axes of acceleration with frequency of 200 Hz. The data is filtered and buffered. The processor on the Telos board wakes up every 40 ms (25 Hz) and raises an interrupt requiring the data from the daughter card. In the Inactive mode the wireless transceiver on the Telos platform is turned off, and the average current drawn for the whole sensor is 1.53 mA. The motion sensor can be configured to send raw accelerometer data or detected steps. If raw accelerometer data is required, the amount information to be sent per one super frame (1 s) is 3 axes * 40 Hz * 2 bytes = 240 bytes, which is equivalent to 12 TinyOS packets. If the step detection is performed on the sensor only, information about that event and corresponding timestamp are sent (1 or 2 packets). The motion sensor draws on average 20.1 mA in the Transmit mode and 20.8 mA in the Listen mode. The main contributor to this figure is the wireless radio that draws 17.4 mA when it is transmitting and 19.7 mA when it is receiving. Based on these parameters, we can calculate the average current as follows:

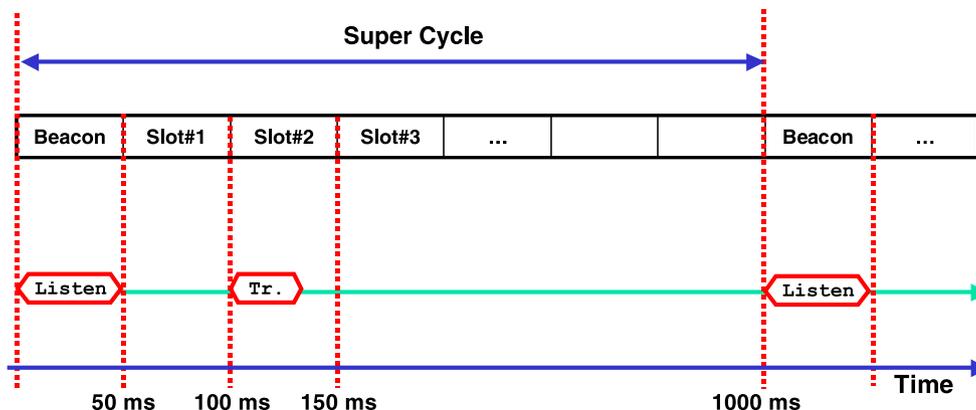


Fig. 9. Super frame cycle.

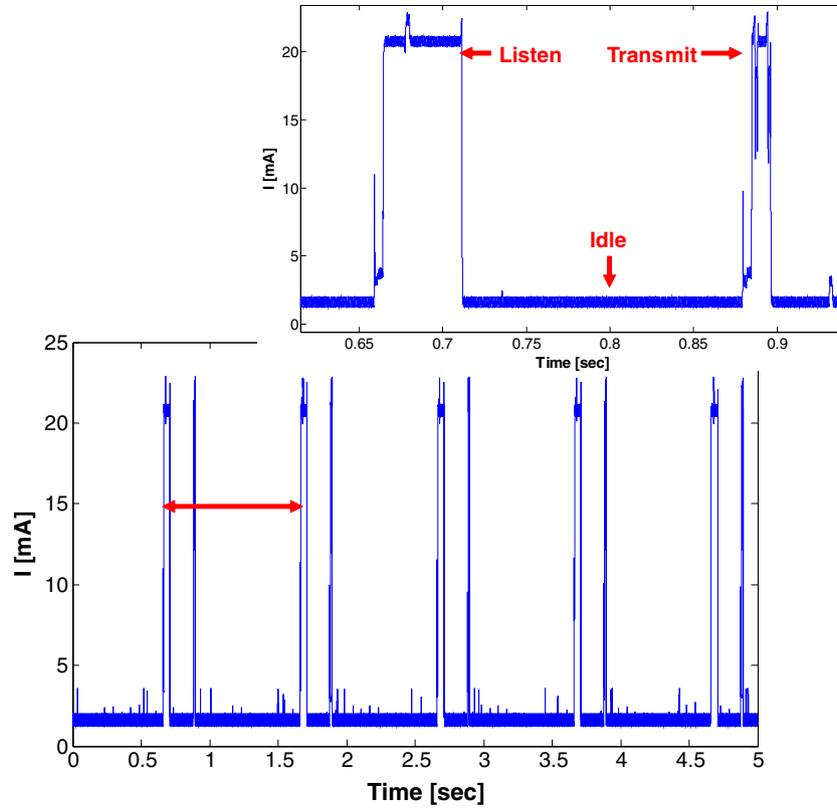


Fig. 10. Actis current consumption during 5 s of operation. Red arrow marks one super frame cycle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

$$I_{\text{avg}} \approx \frac{T_{\text{Listen}}}{T_{\text{SFC}}} \cdot I_{\text{Listen}} + \frac{T_{\text{Transmit}}}{T_{\text{SFC}}} \cdot I_{\text{Transmit}} + \frac{T_{\text{SFC}} - T_{\text{Listen}} - T_{\text{Transmit}}}{T_{\text{SFC}}} \cdot I_{\text{Inactive}}$$

The average current can be used to estimate battery life. If only two messages are sent per super frame cycle ($T_{\text{Listen}} = 50 \text{ ms}$, $T_{\text{Transmit}} = 15 \text{ ms}$), the average current is 2.77 mA. Two AA batteries on the Telos platform have 2900 mAh capacity, so the expected operating time of the motion sensor is 1046 h or over 6 weeks. However, with a tiny 120 mA rechargeable battery, the operating time will be slightly less than 2 days. Further optimizations are also possible: depending on the WWBAN deployment scenario, we could decrease the output power during transmission, a super cycle can be extended (the node would spend less time in the listen mode), or data can be stored locally in a compressed format and then later transmitted.

6. Conclusions

This paper demonstrates the use of WWBANs as a key infrastructure enabling unobtrusive, continual, ambulatory health monitoring. This new technology has potential to offer a wide range of benefits to patients, medical personnel, and society through continuous monitoring in the ambulatory setting, early detection of abnormal condi-

tions, supervised rehabilitation, and potential knowledge discovery through data mining of all gathered information.

We have described a general WWBAN architecture, important implementation issues, and our prototype WWBAN based on off-the-shelf wireless sensor platforms and custom-designed ECG and motion sensors. We have addressed several key technical issues such as sensor node hardware architecture, software architecture, network time synchronization, and energy conservation. Further efforts are necessary to improve QoS of wireless communication, reliability of sensor nodes, security, and standardization of interfaces and interoperability. In addition, further studies of different medical conditions in clinical and ambulatory settings are necessary to determine specific limitations and possible new applications of this technology.

Acknowledgments

We thank John Gober for his help in the ISPM and IAS printed circuit board development and Dennis Cox for his help with time synchronization protocol. We also thank Piet de Groen from Mayo Clinic for inspiring exchange of ideas about WWBAN.

References

- [1] National Coalition on Health Care, in: <<http://www.nchc.org/facts/cost.shtml/>>, Accessed: June 2005.

- [2] P. Binkley, Predicting the potential of wearable technology, *IEEE Engineering in Medicine and Biology Magazine* 22 (3) (2003) 23–24.
- [3] Seiko Pulse Graph Wrist Watch, Available at: <<http://www.seiko-pgt.or.jp/>> (Japanese), Accessed: July 2005.
- [4] Polar, Available at: <<http://www.polarusa.com/>>, Accessed: July 2005.
- [5] Digi-Walker step counter, Available at: <<http://www.digiwalker.com/>>, Accessed: July 2005.
- [6] Actigraph, Available at: <<http://www.mtiactigraph.com/>>, Accessed: July 2005.
- [7] Holter Systems, Med-electronics Inc., Available at: <<http://med-electronics.com/>>, Accessed: July 2005.
- [8] Cardio Micro Sensor, Available at: <<http://www.cardiomems.com/>>, Accessed: July 2005.
- [9] T. Martin, E. Jovanov, D. Raskovic, Issues in wearable computing for medical monitoring applications: a case study of a wearable ECG monitoring device, in: *International Symposium on Wearable Computers ISWC 2000*, Atlanta, October 2000.
- [10] R. Szcwcyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, D. Estrin, Habitat monitoring with sensor networks, *Commun. ACM* 47 (6) (2004) 34–40.
- [11] J. Burrell, T. Brooke, R. Beckwith, Vineyard computing: sensor networks in agricultural production, *IEEE Pervasive Computing* 3 (1) (2004) 38–45.
- [12] E. Jovanov, J. Price, D. Raskovic, K. Kavi, T. Martin, R. Adhami, Wireless personal area networks in telemedical environment, in: *Proceedings of the Third International Conference on Information Technology in Biomedicine (ITAB-ITIS2000)*, Arlington, VA, 2000, pp. 22–27.
- [13] E. Jovanov, A. Lords, D. Raskovic, P. Cox, R. Adhami, F. Andrasik, Stress monitoring using a distributed wireless intelligent sensor system, *IEEE Engineering in Medicine and Biology Magazine* 22 (3) (2003) 49–55.
- [14] E. Jovanov, A. Milenkovic, C. Otto, P.C. de Groen, A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation, *Journal of NeuroEngineering and Rehabilitation*, 2:6, March 1, 2005. Available at: <<http://www.jneuroengrehab.com/content/2/1/6/>>.
- [15] J. Welch, F. Guilak, S.D. Baker, A wireless ECG smart sensor for broad application in life threatening event detection, in: *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Francisco, CA, September 2004, pp. 3447–3449.
- [16] D. Malan, T.R.F. Fulford-Jones, M. Welsh, S. Moulton, CodeBlue: an ad hoc sensor network infrastructure for emergency medical care, in: *Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004)*, Boston, MA, June, 2004, pp. 12–14.
- [17] M.J. Mathie, B.G. Celler, A system for monitoring posture and physical activity using accelerometers, in: *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2001, pp. 3654–3657.
- [18] Moteiv: Wireless Sensor Networks, Available at: <<http://www.moteiv.com/>>, Accessed: June 2005.
- [19] C. Otto, J.P. Gober, R.W. McMurtrey, A. Milenkovic, Emil Jovanov, An implementation of hierarchical signal processing on a wireless sensor in TinyOS environment, in: *Proceedings of the 43rd ACM Southeastern Conference*, Kennesaw, GA, vol. 2, March 2005, pp. 49–53.
- [20] MSP430 MCUs, Available at: <<http://www.ti.com/msp430/>>, Accessed: January 2005.
- [21] Chipcon RF Transceivers, Available at: <<http://www.chipcon.com/>>, Accessed: June 2005.
- [22] TinyOS: an open-source operating system for sensor networks, Available at: <<http://www.tinyos.net/>>, Accessed: June 2005.
- [23] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, The nesC language: a holistic approach to networked embedded systems, in: *Proceedings of Programming Language Design and Implementation (PLDI) 2003*, June 2003.
- [24] J. Elson, D. Estrin, Time synchronization for wireless sensor networks, in: *Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS '01)*, pp. 1965–1970.
- [25] J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcasts, in: *Proceedings of the fifth symposium OSDI '02*, pp. 147–163.
- [26] S. Ganeriwal, R. Kumar, M.B. Srivastava, Timing-Sync Protocol for Sensor Networks, in: *Proceedings of the 1st International Conference on Embedded Network Sensor Systems (SenSys '03)*, pp. 138–149.
- [27] 802.15.4-2003 IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), 2003.
- [28] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, Robust multi-hop time synchronization in sensor networks, in: *Proceedings of the International Conference on Wireless Networks (ICWN '04)*, vol. 1, pp. 454–460.
- [29] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, The flooding time synchronization protocol, in: *Proceedings of the 2nd International Conference on Embedded Network Sensor Systems (SenSys '04)*, pp. 39–49.
- [30] D. Cox, E. Jovanov, A. Milenkovic, Time synchronization for ZigBee networks, in: *Proceedings of the 37th IEEE Southeastern Symposium on System Theory (SSST'05)*, Tuskegee, AL, March 2005, pp. 135–138.
- [31] ZigBee Alliance, Available at: <<http://www.zigbee.org/>>, Accessed June 2005.
- [32] A. Milenkovic, M. Milenkovic, E. Jovanov, D. Hite, An environment for runtime power monitoring of wireless sensor network platforms, in: *Proceedings of the 37th IEEE Southeastern Symposium on System Theory (SSST'05)*, Tuskegee, AL, March 2005, pp. 406–410.



Aleksandar Milenkovic is an assistant professor in the Department of Electrical and Computer Engineering at the University of Alabama in Huntsville. He currently directs the LaCASA Laboratory (<http://www.ece.uah.edu/~lacasa/>). His research interests include advanced architectures for the next generation computing devices, low-power VLSI, reconfigurable computing, embedded systems, and wireless sensor networks. Dr. Milenkovic received his Dipl. Ing., MSc, and PhD degrees in Computer Engineering and Science from the University of Belgrade, Serbia, in 1994, 1997, and 1999, respectively.



Chris Otto is an embedded software engineer at Lewis Innovative Technologies, Inc. in Huntsville, Alabama. His interests include wireless sensor networks, real-time systems, embedded systems and packet voice technologies. Mr. Otto received his BS in Computer Engineering at the University of Alabama in Huntsville in 1999 and is expected to receive his MS in 2006.



Dr. Emil Jovanov is an Associate Professor in the Electrical and Computer Engineering Department at the University of Alabama in Huntsville. He received the Dipl. Ing., MSc, and PhD degrees in electrical and computer engineering from the University of Belgrade. Dr. Jovanov has been developing wireless intelligent sensors for personal health monitoring and mobile computing for more than five years. He was principal investigator or co-investigator on several grants from NSF and industry in the field of wireless

sensor networks, wireless intelligent sensors, and wearable health monitors. His research interests include ubiquitous and mobile computing, biomedical signal processing, and telemedical health monitoring. Dr. Jovanov serves as an Associate Editor of IEEE Transaction on Information Technology in Biomedicine and Applied Psychophysiology and Biofeedback.