Webit&NEU: An Embedded Device for the Internet of Things

Jialiang Wang, Hai Zhao, Jiuqiang Xu, and Yuanguo Bi

College of Information Science & Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Yuanguo Bi; biyuanguo@ise.neu.edu.cn

Received 16 June 2013; Revised 13 December 2013; Accepted 19 December 2013; Published 14 January 2014

Academic Editor: Naveen Chilamkurti

Copyright © 2014 Jialiang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) is regarded as the future generation Internet, which ranges from radio frequency identification (RFID) to the ubiquitous computing systems such as wireless sensor networks and mobile ad hoc networks. With the rapid development of IoT, designing an effective low-cost embedded terminal device for the IoT become very necessary. A new embedded device, Webit&NEU, and its reduced embedded real-time operating system used for IoT are implemented by our China Liaoning Province Embedded Technique Key Laboratory in this paper. Besides, related modules in terms of RFID technique, wireless communication, and network protocol are also provided in this paper. Compared with several current solutions of connecting devices and Internet, it has the advantages of good real-time performance, light weight, and low cost. Besides, this paper also puts forward a localization algorithm for the Webit&NEU, and experimental test results in terms of real-time system ability, network communication performance, and localization algorithm show that Webit&NEU can work well and meet the actual requirements of the IoT.

1. Introduction

The IoT is a new-emerging technique. After the age of Internet and wireless sensor network, people are gradually entering the era of IoT. The IoT refers to uniquely identifiable objects (things) and their virtual representations in an Internet-like structure [1]. In fact, the IoT includes technologies such as RFID, sensors, and smart devices. The concept of the IoT has become more and more popular now. RFID is often regarded as a prerequisite for the IoT [2, 3]. Supposing all objects of daily life were equipped with radio tags, thus they could be identified, tacked, and managed by computers, and then they will certainly bring great convenience to people's lives.

Embedded Internet device Webit2.0 can make the standard industry device have the ability of accessing the Internet, and it has successfully been embedded to fieldbus, industrial devices, and so forth, so people can feel free to access those devices through Internet. Webit2.0 has been successfully developed by our China Liaoning Province Embedded Technique Key Laboratory and has been large-scale manufactured by China Shenyang Neu-era Information Technology Stock Co., Ltd [4]. It is worth noting that Webit2.0 has obtained the China's product patent (nos. ZL 00 2 52482.1 and ZL 01 2 29725.9).

The design and implementation of embedded device Webit&NEU are based on Webit2.0. Besides the basic functions of Webit2.0, new functions such as RFID data processing and wireless date communication are also added to the Webit&NEU, and its microcontroller is also updated to Atmega128L so that it can well be used in IoT.

This paper mainly makes three novel research contributions.

(1) It implements a light-weight and low-cost embedded device Webit&NEU used for the IoT and its reduced operating systems kernel Webit&NEU OS written in AVR assembly language.

(2) It implements data collection module, wireless communication module, and network protocol module efficiently.

(3) It proposes an improved localization algorithm for the practical application of Webit&NEU.

This paper is organized as follows. The second section presents the current embedded system communication
solutions and related analysis. The third section describes the design and implementation of Webit&NEU hardware system, its light-weight operating system kernel, and related performance analysis. Besides, the wireless communication module between different devices, RFID data collection module, and the TCP/IP protocol module are provided in this section. The fourth section proposed an improved localization algorithm for Webit&NEU, and performance comparison in terms of localization error and localization time is described in this section. Concluding remarks are given at the end of the paper.

2. Current Embedded System Communication Solutions

The current common methods used to collect, communicate, and manage data are shown as follows [5, 6].

(1) RS485 is adopted in consideration of its good capability of long distance communication. The implementation includes connecting all RFID readers by concentrator and linking them by server. In this method, the server handles the data and also supports Internet interface so as to make users access them remotely as Figure 1 shows.

(2) Handheld RFID is used reader to read and write RFID card, and then system sends data to nearby server by the method of wireless transmission and implements resources sharing by connecting to Internet. This method is adopted by some applications having high portable requirements as Figure 2 shows.

(3) RFID and PC are usually connected by RS232 in this method. As the limit of communication capability for RS232, it is suitable for short distance communication. PC filters and handles these data collected by RFID reader and then sends these data to the server as Figure 3 shows.

For most of the RFID products, the connection method between different devices is usually implemented by USB or serial port [7], which is not only inconvenient to expand, but also is unsuitable for remote data transmission, so this is the bottleneck for these devices to be applied to the IoT. So we engaged to develop embedded device features: high efficiency, light weight, and low cost used in the IoT.

3. Design and Implementation of Webit&NEU

3.1. Design of Webit&NEU Hardware. Webit&NEU uses the basic structure of Webit2.0. It can manage the data collected from the RFID reader and CC2530 and store the necessary data to memory. Besides it can assemble HTTP packet according to different applications and then send them to Internet by the RTL8019AS.

The product appearance of Webit2.0 is shown as Figure 4 shows (compared with one yuan coin).

Its corresponding interface parts are described as below:

(1) 14-bit full-duplex User I/O (TTL electrical level),
(2) power supply interface (DC 9V),
(3) network interface: IEEE802.3/RJ-45,
(4) network data signal lights,
(5) in-system programming interface.

And its technique feature and performance are provided, respectively, as below.

Technique feature:

(i) no gateway and PC needed,
(ii) independent design of hardware and software architecture,
(iii) independent intelligent,
(iv) independent function of network server,
(v) independent addressability,
(vi) embedded design,
(vii) low cost,
(viii) small in size.
Performance of Webit2.0:

(i) Atmel AVR RISC processor,
(ii) customized definition Web page,
(iii) customized definition CGI programming,
(iv) duplexing I/O interface with 14 bit TTL electrical level,
(v) UART interface of TTL electrical level supports up to 115200 bps,
(vi) 10 M Ethernet interface (RJ-45),
(vii) in-system Programmability (ISP),
(viii) protocol supported: ARP, ICMP, IP, TCP, and HTTP,
(ix) Network interface: IEEE 802.3/RJ-45.

Webit&NEU can not only implement wireless communication with each other by CC2530, but also send the data collected from RFID reader to the Internet through RJ-45 interface as Figure 5 shows. Thus while devices are embedded with the Webit&NEU, then they can be used to smartly identify, locate, track, monitor, and manage these devices having RFID tag. The Webit&NEU well inherits good feature such as real-time ability from Webit2.0, and the most important is that the design goal for Webit&NEU is the implementation of light weight and low cost.

RFID reader collects data from devices having tags so as to identify devices [8]. The RFID data collection module, CC2530 wireless communication module, and Webit&NEU microprocessor module form the whole Webit&NEU hardware, and architecture illustration between Webit&NEU microprocessor module and RFID function module is shown as Figure 6 shows.

The Webit&NEU microprocessor controls the RFID reader to achieve the function of data collection and management, and it mainly includes the below 4 procedures.

(1) **Initialization.** Webit&NEU microprocessor implements the initialization of RFID reader by sending specific command (01H) to it.

(2) **Sending Commands.** Webit&NEU microprocessor sends a request command (02H) to RFID reader so as to indicate that it will send commands, and RFID reader will return a response command (10H) to the microprocessor to show it has already prepared for receiving commands. After executing the above procedures, the microprocessor of Webit&NEU begins to send commands.

(3) **Data Verification.** This is used to make sure the data received are correct. It is implemented by the method of XOR operation. If this procedure is correct, then the system will execute data handling program. If not, the system will set the Flag in response frame as 1, so as to imply RFID reader that the data is wrong and order RFID reader to send it again.

(4) **Data Filtering.** While microprocessor of Webit&NEU receives data from RFID reader, it will judge whether the data is anticipated according to message ID, number of data bytes, and data verification. If not, then the system will filter the wrong and incomplete data.

3.2. The Reduced Webit&NEU Operating System Kernel

3.2.1. The Design and Implementation of Webit&NEU OS. A real-time operating system must have the ability to respond to external events timely. In order to coordinate and execute tasks effectively, Webit&NEU OS system provides the following services: interrupt handling, intertask communication, task synchronization, memory management, timing services, task priority assignment, and so on.

All of these functions implemented by the unique design of system call can easily execute real-time applications. System call is a program supported to user by system, while users need to implement certain system functions; what they need to do is to call the accordingly system call in their application program, so it is convenient for users to use the system kernel.

All the system calls are written in AVR assembly language. As one of the important advantages for AVR assembly
language is that it can easily and flexibly operate a certain storage location in memory, even its certain bit of the storage location, so system can handle task, message, interruption, timing, and so on effectively.

So the Webit&NEUOS system can well be used to effectively manage data collection, wireless communication in the IoT [9, 10], and it has successfully been developed by our China Liaoning Province Embedded Technique Key Laboratory [11].

The system calls constitute the basic operating system and their brief descriptions are provided as follows.

(1) Task management services:
   (i) CREATE_TASK: creating and scheduling a task dynamically,
   (ii) DELETE_TASK: deleting certain specific task,
   (iii) GET_FUNCTION_ID: obtaining task function ID (value ranges from 00 to 0FFH),
   (iv) SUSPEND: suspending a task being executed.

(2) Intertask communication services:
   (i) ALLOCATE: allocating buffer space where a task creates a message to be sent,
   (ii) SEND_MESSAGE: sending certain message to specific task,
   (iii) WAIT_MESSAGE: allowing a task to wait for a message to come,
   (iv) DEALLOCATE: returning certain specific buffer to the system buffer.

(3) Internal memory management services:
   (i) GET_MEM: getting the address of certain memory having specific length is currently available in the system,
   (ii) RELEASE_MEM: returning certain memory with specific length to the system memory pool.

(4) Interrupt-handling services:
   (i) DISABLE_INTERRUPT: disabling some interruptions,
   (ii) ENABLE_INTERRUPT: reenabling disabled interruptions,
   (iii) WAIT_3: synchronizing tasks with interruptions.

(5) Timing services:
   (i) SET_INTERVAL: setting a time interval after the interval event occurred,
   (ii) WAIT: waiting for interval event or timeout to occur.

The design and implementation of data variable units and tables in the memory block bring great convenience for system to handle task information and its responding priority, interruption, message, timing, and so on. Besides, while designing the system calls, the data storage units and tables are all stored in the internal SRAM, so the speed in terms of task switching and information handling is improved to some extent.

By using the above system calls, system can well synchronize and schedule multitasks in the IoT. Although only one task can be executed on the processor at any given time, the multiplexing of all tasks makes it appear as if all of the tasks are running simultaneously.

3.2.2. The Performance Test of Real-Time Ability about Webit&NEUOS. In order to evaluate the real-time performance of Webit&NEUOS, we mainly tested two performance parameters of task switching time and max interruption inhabit time [12, 13] and then compared them with Webit5.0 OS (The fifth-generation operating system designed for Webit2.0) and TinyOS 2.1.2. We chose the TinyOS 2.1.2 as the comparison system because it also has relatively small kernel code size compared with the other operating systems.

TinyOS is an open source, BSD-licensed operating system designed for low-power wireless devices, such as those used in sensor networks, ubiquitous computing, personal area networks, smart buildings, and smart meters. It is written in the nesC programming language as a set of cooperating tasks and processes. TinyOS started as a collaboration between the University of California, Berkeley, in cooperation with Intel Research and Crossbow Technology. A worldwide community from academia and industry uses, develops, and supports the operating system as well as its associated tools. TinyOS 2.1.2 is officially released on August 20, 2012. TinyOS 2.1.2 includes support for updated msp430-gcc (4.6.3) and avr-gcc.
(4.1.2). A complete 6lowpan/RPL IPv6 stack. Support for the ucmini platform and ATmega128RFA1 chip [14].

We used the tools of logic analyzer (TLA603), arbitrary waveform generator (AWG2021), and digital storage oscilloscope (TDS1012). In the test programs, we added some control codes by transforming the value of high and low levels of pulses through I/O interface. For each experiment below, we tested 20 times totally, kept down the maximum and minimum, and then calculated their average as Figures 7 and 8 show.

For the Webit&NEU OS, the reduction of task switching time and max interruption inhibit time is 8.41% and 13.44% separately compared with Webit 5.0 OS, which means Webit&NEU can not only process task switching more quickly, but also fast respond to external interruptions.

Compared with TinyOS 2.1.2, we can know that the task switching time of Webit&NEU OS is shorter, mainly because of the special design structure of TASK READY_TAB (16 memory units ranging from 0283H to 0292H are used to store related information of every ready task). The low 4 bit stores the task priority, while the high 4 bit of each unit stores the task ITD. When a task is inserted into TASK READY_TAB, it will be put in suitable place according to its priority so as to make sure all tasks in the task ready table are sorted by task priority. But for the max interruption inhibit time, TinyOS 2.1.2 is better, because the two categories of interruption handling program (COMMON SERVER and TIMER0 SERVER) take longer time for the Webit&NEU OS. But compared with Webit5.0 OS, the Webit&NEU OS is better in terms of both task switching time and max interruption inhibit time.

3.3. RFID Data Collection Module. The RFID terminal function module of Webit&NEU can be summarized as below.

The data collected from the RFID reader are sent to specific server by Internet so as to make the devices networked. In order to implement these functions, the architecture of terminal function module is mainly composed of network driver, data collection, and device driver module. The three modules and their composition are shown in Figure 9.

Network driver module is the key to ensure RFID terminal devices are networked. In order to implement the connection between devices and Internet, the TCP/IP protocol is prerequisite. This system also implements the function of Web server so as to make clients access these devices by browser.

RFID data collection module is the input data part for the whole system. This module implements communication between different devices by RFID technology. In order to guarantee the correction and security of data collection and transmission, firstly there should be a procession of establishing link before data frames transmission, and system uses handshake principle to implement synchronization between sender and receiver. Secondly, the data frames used to communicate should include not only the actual data, but also the necessary control and check bit, which means data should be verified so as to guarantee the data correction, so sometimes the data retransmission operation is required. Finally, the system filters the control and check bit and then obtains the effective data.

Devices driver module is the basic part of the RFID terminal devices. This layer masks the complexity of microcontroller underlying hardware and supports simple interface to applications. This module includes Webit&NEU system initiation, serial port driver, and network driver.

3.4. Communication between Different Webit&NEU Devices. The basic communication mechanism for the Webit&NEU is sending and receiving the format message through CC2530 during different Webit&NEU devices. The implemented format of data packet includes link types, message length, bit identification, source/destination task, order/response identification and data as Figure 10 shows, which aims to implement wireless communication efficiently [15].

The first 4 bytes of the data packet is preamble which is used to implement the packet synchronization, and the
3.5. Implementation of TCP/IP Protocol. As the storage resource of the ATmega128L is relatively limited, so this paper aims to implement a reduced TCP/IP protocol so as to meet the basic requirements in the practical application. The design of TCP/IP protocol stack is using hierarchical structure. About link layer, it refers to the network card driver so as to implement the functions of receiving and sending Internet data frame, as this procedure depends on hardware address. Considering that IP protocol in the network layer is IP address, a dynamic translation between these two addresses is needed.

System judges the type of the data packet if it is ARP request (value is 0806H), and then the system will call the ARP response module; if it is IP packet (value is 0800H), system will then upload the packet to network layer and then call the IP handling module.

The system executes correspondingly operations according to the protocol value of IP packet existent in IP Input function; if the value is 0x01, the system will turn to ICMP execution module; if the value is 0x06, the system will turn to the UDP handling module; if the value is 0x11, the system will turn to the UDP processing module.

For the TCP protocol in the transport layer, if the port of TCP is 0x80, the system will send the data to HTTP server, and the HTTP server will respond to it after executing related operations. The whole data process flow in the TCP/IP is shown in Figure 11.

In order to evaluate the network performance, we tested the parameters 20 times in terms of network response time, network upload rate and download rate. The maximum, minimum, and average values are described respectively in Figures 12, 13 and 14 shows. Network response time, network upload rate, and download rate are very important for evaluating the network performance, which can well reflect the network transmission velocity and real-time ability.

The experimental test results show that Webit&NEU OS performs better than Webit5.0 OS in terms of network response time and network upload and download rate. Compared with Webit&NEU OS and TinyOS 2.1.2 systems, Webit&NEU OS has little quicker network response time than TinyOS 2.1.2, but the network upload rate and download rate are slower than TinyOS 2.1.2. In general, the network
Figure 11: Process flow of data packet.

Figure 12: Experimental test results about network response time (μs).

Figure 13: Experimental test results about network upload rate (kbps).

Figure 14: Experimental test results about network download rate (kbps).

The performance of Webit&NEU can meet the basic requirement of data transmission so as to make sure that Webit&NEU can well be applied to the IoT [6, 7].

4. Localization Algorithm Designed for Webit&NEU

One of the most important functions for this system is to manage the devices efficiently, so obtaining the position of certain device is very necessary for Webit&NEU. The development of Webit&NEU is devoted to make it well used in the practical application; thus, the research of their localization is very important, because it directly affected the communication ability and efficiency.

Based on the traditional localization algorithm TL (used in the Webit2.0), we propose an improved localization algorithm ETL for Webit&NEU used in the IoT. In below discussion about localization algorithm, each Webit&NEU is regarded as a node.

The localization of nodes in the two-dimensional space can directly be calculated by their distance between nodes. If only knowing the coordinates of the two reference nodes O₁ and O₂, it may cause uncertainty while calculating the unknown reference node as Figure 15(a) shows; this is because the two circles form two intersections of P₁ and P₂, so it can not determine the accurate localization about the unknown node. Thus, at least three reference points are needed to determine the localization of the unknown node in two-dimensional space as Figure 15(b) shows.

Calculating the distance between nodes above is under the assumption that the distance measurement is carried out with absolute precision; however, the measurement error is inevitable in the practical application environment. Supposing the error range is (0, ±s), while the practical distance of two nodes is r, then the measured distance range is (r, r ± s).
Due to the existence of error, the localization error will not form a point, but a small area as the shaded area shows in Figure 15(c). The localization error area caused by the three reference points of \( P_1, P_2, \) and \( P_3 \) is the region formed by the six arcs of \( F_1, F_2, F_3, F_4, F_5, \) and \( F_6 \) as Figure 16 shows.

While the error value of \( s \) is small enough, then the error value can be approximated as the region formed by the hexagon of \( A, B, C, D, E, F \). Consider the following:

\[
\text{area}\ (Q_6AQ_1P) = 2\text{area}\ (Q_6AP) = 2 \left( \frac{1}{2} \cdot e \cdot \frac{\beta_{1,2}}{2} \right)\]

\[
\therefore \text{area}\ (ABCDEF) = 2e^2 \left( \tan \frac{\beta_{1,2}}{2} + \tan \frac{\beta_{2,3}}{2} + \tan \frac{\beta_{3,1}}{2} \right)\]

\[
\therefore \ (\tan \beta)^{''} = 2 \tan \beta \cdot \sec^2 \beta \geq 0 \quad \left( 0 \leq \beta \leq \frac{\pi}{2} \right)\]

\[
\therefore \text{area}\ (ABCDEF) = 6e^2 \frac{1}{3} \left( \tan \frac{\beta_{1,2}}{2} + \tan \frac{\beta_{2,3}}{2} + \tan \frac{\beta_{3,1}}{2} \right)\]

\[
\geq 6e^2 \tan \frac{\beta_{1,2} + \beta_{2,3} + \beta_{3,1}}{6} = 6e^2 \tan \frac{\pi}{6} = 2\sqrt{3}e^2. \quad (1)
\]

Algorithm 1 (proposed ETL algorithm).

Step 1. Calculate distance \( S_{ij} (i \neq j) \) between any two nodes and store them to database.

Step 2. Every node sends broadcast packet periodically, and each broadcast packet should include a message as the format of \( \{ID, T_{Send}(a, b)\} \). \( ID \) is the identification of unknown node, \( T_{Send} \) is the time of sending message, and \( (a, b) \) is its coordinate.

Step 3. After receiving the broadcast packet, system calculates the distance \( L_i \) according to the time of sending message \( (T_{Send}) \) and receiving message \( (T_{Rec}) \) and then sends these dates in the database.

Step 4. While database receives \( N \) messages totally, for each set of \( C_N^3 \), use the formula of \( \cos \alpha_{ij} = (L_i^2 + L_j^2 + S_{ij}^2) / 2L_iL_j \) to calculate the angle between unknown node and two other reference nodes.

Step 5. Use three nodes whose angle exists in the range of \( \pi/6 - \lambda < \alpha_{1,2}, \alpha_{2,3}, \alpha_{3,1} \leq \pi/6 + \lambda \) to calculate the position of...
unknown node. For each eligible set in $C_3$, system calculates corresponding position of unknown node {$H_1, H_2, \ldots$}.

**Step 6.** For all the position of unknown node {$H_1, H_2, \ldots$}, system calculates average value $H_{avg}$.

For TL algorithm used in Webit2.0, it gets the position by rough calculation firstly and then obtains the final value by repeated similar procedure. Compared with TL algorithm, the ETL algorithm we proposed avoids some repeatable process by the above specific filter. The subsequent experiments will evaluate the performance of localization error and localization time for the ETL algorithm.

(a) **The Test and Analysis of Localization Error.** In order to locate the position of unknown devices, the two localization algorithms are both implemented. We compared the testing results by executing the Webit&NEU OS. The experimental region adopted is the indoor environment whose area is $4m \times 6m$. There are totally 17 Webit&NEU devices as the reference node, which are placed by the method of the above localization algorithm, and these devices form 20 equilateral triangles totally in this localization area.

As the Figure 17 shows, while the number of Webit&NEU devices is 3, the localization error caused by the two algorithms is close, but, with the increase of the Webit&NEU devices number, the localization error of ETL algorithm decreases more quickly than that of TL algorithm. While the number of Webit&NEU devices reaches to 6, the localization error decreases smoothly for ETL algorithm, but correspondingly while the number of Webit&NEU devices reaches to 12, the localization error begins to decrease smoothly for TL algorithm.

The comparison of the two algorithms verifies that the ETL algorithm causes less localization error than TL algorithm; thus ETL algorithm is better to be used in the IoT because of its good localization accuracy.

(b) **The Test and Analysis of Real-Time Ability.** In order to test the real-time ability ETL algorithm, we also implement ETL algorithm and TL algorithm separately based on Webit&NEU OS and evaluate the real-time localization ability while the number of Webit&NEU devices is increased continuously, and the experimental results measured are shown in Figure 18.

As we can see from the above results, while the number of Webit&NEU devices is 3, 4, and 5, the localization time for these two localization algorithm is almost the same, but with the increase of the number of Webit&NEU devices, the localization time of ETL algorithm increases exponentially while ETL algorithm increases correspondingly smoothly.

From the above analysis we can know that ETL algorithm is better than TL algorithm in terms of good-time ability and higher localization accuracy [17, 18]. Due to these features of good real-time ability and less localization error, thus ETL algorithm can be better executed for Webit&NEU in the practical application [19].

5. **Conclusions**

Webit&NEU is a new device designed to be used in the IoT in order to implement smart management of devices. This paper mainly introduces its hardware architecture, software system, wireless communication module, RFID module, and network protocol module. In order to evaluate its real application value, we improve traditional localization algorithm used for Webit2.0 and propose an ETL algorithm, and the experimental results show that ETL algorithm has good real-time ability and less localization error than the traditional one. Besides, experimental results in terms of task switching time, max interruption inhibit time, network response time, network
upload rate, and network download show that it works well and can also meet the basic application requirements for the IoT. The Webit&NEU OS can well be executed in some experimental tests, but it still will be continuously optimized so as to better be applied to IoT.

The design and implementation of embedded device Webit&NEU will provide a good solution for the IoT. The widely application of Webit&NEU will undoubtedly bring great convenience for people to implement smart management of devices. At the meantime, the design aim of light weight and low cost will certainly facilitate the widely usage of Webit&NEU.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

This work is partly supported by the National Natural Science Foundation of China (no. 61101121), Fundamental Research Funds for the Central Universities of China under Grant no. N110404002, Key Laboratory Project Funds of Shenyang Ligong University under Grant no. 4771004ksf03, Educational Committee of Liaoning Province Science and Technology Research Projects under Grant no. L2013096, and Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (no. 708026). The authors thank Jan Vitek, Ales Plsek, and Lei Zhao for their help during studying at Purdue University from September 2010 to September 2012. The authors also thank the anonymous reviewers for their valuable comments.

**References**

[1] http://en.wikipedia.org/wiki/Internet-of-Things.

[2] G. D. Abowd and E. D. Mynatt, "Charting past, present and future research on ubiquitous computing," *ACM Transaction on Computer-Human Interaction*, vol. 7, no. 1, pp. 29–58, 2002.

[3] G. Y. Xu, Y. C. Shi, and W. K. Xie, "Pervasive computing," *Computer Journal*, vol. 26, no. 9, pp. 1042–1052, 2003.

[4] Z. Hai and C. Yan, *Pervasive Computing*, Northeastern University Press, Shenyang, China, 2005.

[5] A. H. Ho, Y. H. Ho, and K. A. Hu, "Handling high mobility in next-generation wireless ad hoc networks," *International Journal of Communication Systems*, vol. 23, no. 9-10, pp. 1078–1092, 2010.

[6] S.-R. Yang and C.-W. Leong, "A conformance test tool for next generation network applications and systems," *International Journal of Communication Systems*, vol. 23, no. 6-7, pp. 708–731, 2010.

[7] N. Bagherzadeh and M. Matsuura, "Performance impact of task-to-task communication protocol in network-on-chip," *Journal of Circuits, Systems and Computers*, vol. 18, no. 2, pp. 283–294, 2009.

[8] K. Cho, S. Pack, T. T. Kwon, and Y. Choi, "An extensible and ubiquitous RFID management framework over next-generation network," *International Journal of Communication Systems*, vol. 23, no. 9-10, pp. 1093–1110, 2010.

[9] R. Pellizzoni and M. Caccamo, "Impact of peripheral-processor interference on WCET analysis of real-time embedded systems," *IEEE Transactions on Computers*, vol. 59, no. 3, pp. 400–415, 2010.

[10] J. Wu, J.-C. Liu, and W. Zhao, "A general framework for parameterized schedulability bound analysis of real-time systems," *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 776–783, 2010.

[11] Intel Corporation, *IRMX 51 iDCX 51 Distributed Control Executive User’s Guide for Release 2.0*, Order Number 460367, Intel Corporation, Santa Clara, Calif, USA, 1987.

[12] W. Dong, C. Chen, X. Liu, K. Zheng, R. Chu, and J. Bu, "FIT: a flexible, lightweight, and real-time scheduling system for wireless sensor platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 1, pp. 126–138, 2010.

[13] J. Whitham and N. Audsley, "Time-predictable out-of-order execution for hard real-time systems," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1210–1223, 2010.

[14] http://www.tinyos.net/.

[15] L. Sun, J. Li, Y. Chen, and H. Zhu, *Wireless Sensor Network*, Tsinghua University Press, Beijing, China, 2005.

[16] Y. Zhou, "The optimizing selection and error analysis of location reference nodes in smart space," Northeastern University, pp. 41–49, 2008.

[17] G. Cabri, L. Leonardi, M. Mamei, and F. Zambonelli, "Location-dependent services for mobile users," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 33, no. 6, pp. 667–681, 2003.

[18] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[19] L. Yan, Y. Zhang, L. T. Yang, and H. Ning, *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems (Wireless Networks and Mobile Communications)*, Auerbach Publications, 2008.
Submit your manuscripts at
http://www.hindawi.com