11

Implementing Reconfigurable Wireless Sensor Networks: The Embedded Operating System Approach

Sanjay Misra and Emmanuel Eronu
Department of Computer Engineering , Federal University of Technology, Minna Nigeria

1. Introduction

Remote monitoring and control are vital in ensuring the efficiency and safety of entities beneficial to man and his environment. These entities cut across oil and gas, biomedical, healthcare, manufacturing, transportation, security, the Armed forces (Military, Navy, and Air force) etc. Wireless Sensor Networks (WSNs) are being used to effectively carry out the aforementioned task. Implementing WSN application might involve the deployment of hundreds or sometimes thousands of wireless sensor nodes to remote and inaccessible locations. A few examples of theses scenario can be related to the following: surveillance (e.g. international Border Monitoring, Littoral operations), environmental monitoring (e.g. Dam collapsing and flooding, earthquake early warning, avoiding Forest fire disaster), liquid (oil and water) and gas pipeline monitoring, etc. When operational needs changes or new functionalities are required in such scenario, reconfiguration of either the entire network or individual sensor nodes becomes inevitable. The inability to effect these changes could pose a serious challenge to the continued operation of the entire system. Other issues that could warrant the need for a reconfigurable WSN are bug fixes, regular code updates, update in response to security challenges, flexibility in adopting energy and performance efficient RF communication link or interface, efficient energy management, etc.

Several ways of carrying out dynamic reconfiguration exist, however employing a well-designed embedded operating system allows for a platform-independent implementation. The focus of discussion is what constitutes a well-designed platform-independent operating system for the wireless sensor network. These findings and other open research issues will form the bases of our presentation in this chapter. We also intend to highlight and discuss various research approaches adopted so far in realizing fully reconfigurable WSNs under severe size, limited processing capabilities and power consumption constraints within the context of embedded operating system.

The remainder of this chapter is organized as follows. Section 2 discusses key principles and implementation techniques of embedded operating Systems (EOS). The aim is to provide some form of background information on EOS and its advanced close relative: the real time operating system (REOS). Section 3 by extension further presents details of driving factors behind the need for a reconfigurable WSN. Some of the approaches adopted in

implementing reconfigurable WSN in notable operating systems as well as on going researches in this area are discussed in section 4. Design issues of reconfigurable EOS are discussed in section 5. Section 6 concludes the chapter.

2. The embedded operating system

An embedded operating system can be viewed as a collection of software modules meant to enhance an embedded system's efficiency, flexibility and robustness (Gomaa, 1993; Labroasse, 2002). To the designer and user it appears more like an extended machine and to the system, an indispensible resource manager. An extension of EOS, the real time embedded operating system (RTOS) is characterized by its ability to implement the aforementioned goals in real time. Hence its operational paradigms are deterministic and have guaranteed worst-case interrupt latency and context switching times (Barr, 1999).

RTOS is implemented around a multi-tasking kernel. The kernel controls when each task is to be executed. And it does this by allocating a time slice to each task (Walls, 1996). The key parts of an RTOS are: the scheduler, RTOS services, synchronization and messaging tools (Ibrahim, 2008). The scheduler which forms the heart of all RTOS is responsible for the selection of tasks to be executed. It does so by implementing any of the following known scheduling algorithms: cooperative (First in First Out), Round-robin, Earliest deadline First (EDF) and fixed priority scheduling (usually rate-monotonic scheduling (RM)). The RTOS services provide some support to the kernel in the likes of Interrupt handling, memory management, input-output services, device management and timing. The essence is to ensure that the RTOS runs efficiently. Whereas, synchronization and messaging tools are used to synchronize access to shared resources and inter tasks activities. Examples of these services and tools are: semaphores, event flags, mailboxes, pipes, message queues etc.

Reconfiguring WSNs can be achieved in two ways. One is a direct method where application tasks can easily be replaced or altered during design time and the other option is to remotely effect a replacement of the application tasks or alter its functions during runtime. In one case, the entire system will have to be put on hold, if not shut down completely while the other method, the system remains active while the changes are effected in real-time. Whichever method is adopted, an abstraction of the WSN's node hardware via EOS or REOS makes the reconfiguration process much easier to implement. In section 4, we briefly introduce the Processing Elements approach, however considering programming constraints; it is evident that the EOS approach is the most effective.

3. Factors behind the need for reconfigurable WSNs

In this section, we discuss some of the driving factors that are necessitating reconfigurability in WSNs. In our survey we categorize the factors in this respect: The need to achieve an Efficient Energy management sub system, Flexible All-Standard-Communication implementable subsystem and implementing dynamic and more secured security features.

3.1 Efficient energy management sub system

The need to manage the limited energy available to Wireless sensor node efficiently constitutes the majority of reason behind the demand for reconfigurable WSN. Energy harvesting techniques have been proposed (solar, vibration etc.) and in some cases implemented (Kompis & Sureka, 2010). However not all WSN applications are deployed in environments where they can take advantage of this option. For example, solar powered nodes will be difficult to sustain in locations where sunlight intensity and duration of availability are relatively low or non-existing (for example the arctic region or under the ocean). The solution, in most cases is to look inwards by adopting appropriate RF communication standards or routing protocol that will allow for long duration of system sustenance and operation.

In (Kompis & Sureka, 2010), energy consumption in nodes has been traced to three basic components namely: Sensing energy, Communication energy and Computation energy. The Sensing energy is dissipated when activating sensing circuitry in order to obtain data from the environment being monitored. The amount of energy consumed in this respect is proportional to the application requirement. Communication energy has to do with the energy consumed while relaying data or control commands either to neighboring nodes or base stations within the network. Similarly the Computational energy refers to that energy dissipated whenever the nodes' processing element (microprocessor / microcontroller/ system on chip) implements computational and logical operations.

Taking cognizance of these components especially the Communication energy and Computation energy components, a number of research works (Muralidhar & Rao, 2008; Kompis & Sureka, 2010) have devise ways of proffering solution by means of reconfiguring related sections of the wireless sensor nodes. Easily reconfigurable sub components or processes as identified in (Kompis & Sureka, 2010) are listed in table 1.0.

3.2 All-standard-communication implementable subsystem

Use of a particular communication standard from a range of many others in wireless sensor networks can be attributed to a number of factors (as depicted in figure 1) notably energy demand, transmission range, data rates, throughput etc. Adopting a particular standard is a function of the intended application's objectives. However when there are changes in application scenarios (context-prone), to use the same WSN node can only be possible if its communication interface can be adjusted accordingly. This important requirement constitutes one of the key issues in WSN reconfigurability demands.

A review of the work done in (Ramamurthy et al., 2004; Ramamurthy et al., 2005) depicts efforts geared towards realizing a generic reconfigurable wireless interface for the WSN node. They argue that realizing such an interface will allow for the deployment of the same WSN node in two distinct application scenarios namely: an Automotive Monitoring system and a Chemical Process control system (Ramamurthy et al., 2005). Exemplifying the second scenario, they proffer the use of RFID or Zigbee wireless communication standards (cheap and power efficient low-performance wireless technologies) in conveying sensed tire pressure, liquid level and corrosion data while the conveyance of highly active sensed data from encoder and Gyro would be better handled using high performance wireless technologies like Bluetooth or Wi-Fi.

Energy Consuming Component	Identified Sub Components/Processes	Proposed/Adopted Reconfiguration Paradigms
Computation	**Supply voltage/Operating frequencies** Implementing scaling of voltage and frequency thereby avoiding use of external oscillators	Reducing the operating supply voltage by changing the architecture of the system, for example through the use of pipelining. Implementing better algorithms and software optimization (e.g. Digital Signal Processing algorithms) that require fewer numbers of operations to perform a task such as filtering.
	ADC sampling rate Conversion of an analogue output from the sensor to digital equivalent which is proportional to the magnitude of voltage or current. Low efficiency factors can result in energy loss during conversion. Whenever voltage level across battery terminals decreases, conversion process draws increasing amount of increasing amounts of current from battery in order to maintain constant supply to the sensor component. Thereby leading to fast depletion of battery life (Kompis & Sureka, 2010; Khan & Vemuri, 2005).	Varying ADC sampling rate depending on the sensitivity or accuracy of data required.
	Peripheral utilization Operating certain peripheral (sensor interface, RF communication interface, etc.) when not needed has the tendency of depleting energy sources faster	Use of operating systems or middleware in isolating or switching off sub circuits that are not needed within at certain times
Communication	**Modulation scheme –** Various modulation schemes exist. A lot of computational power is always required when Implementing these schemes, depending on the algorithm employed. This normally translates to energy consumption within the nodes. Whenever there are cases of incessant retransmission, the energy source available to the nodes gets depleted easily. **Data rate -** Higher data rates results in high power consumptions. **Transmission range –** the transmission range to a very large extend depends on the power level of the transmitter which invariably is a function of energy consumption.	Techniques to reduce the number of retransmission necessary due to packet losses from faulty wireless links are being developed and implemented in various modulation schemes. However, having a reconfigurable RF communication interface can greatly assist in selecting the most appropriate energy friendly modulation scheme. Variation of data rates as much as implementing desired transmission range is also possible.

Table 1. Identified energy consuming components in WSN.

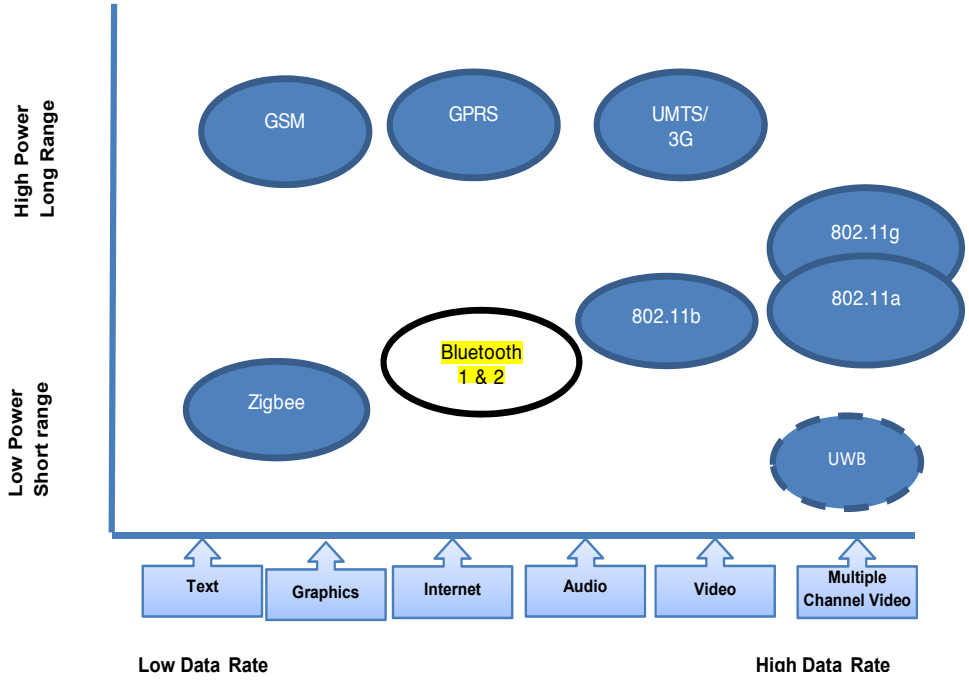


Fig. 1. Wireless communication standards (Muralidhar & Rao, 2008).

3.3 Implementing dynamic and more secured security feature

WSN by virtue of its mode of relaying messages wirelessly is open to attacks and risk. A sensor node in both isolated and non-isolated location can easily have its data or message compromised. Several attack scheme exist, among which are eavesdropping the message, injecting false messages, denial of service, data flooding, RF jamming etc. (Carpenter & Barrett, 2008; Moerschel et al., 2007). To a large extend, security requirements for WSN depends on the application where they are used. We must also recognize that providing

security in sensor network can be very demanding. WSNs are resource constrained systems, having low power, less memory and limited computational abilities.

A number of security protocols have being developed for WSNs, notably among others are TinySec (Karloff et al, 2004), MiniSec (Luk et al., 2007), and SPINS (Perrig et al., 2001). SPINS, MinSec and TinySec link layer security architecture implementation are based on secret key cryptography. They are designed to provide authenticated streaming broadcast, ensured data confidentiality, data authentication based on RC5 algorithm. Using smaller key sizes for transmission of less sensitive information translates into less energy consumption. The reverse is the case when highly sensitive information is to be relayed. Varying key sizes can only be possible when the system itself is reconfigurable. A number of today's cryptographic implementations are not disposed to key varying capabilities even for the most optimized hardware-based cryptographic accelerator platforms. Examples of theses platforms are those realized on ASICS, intended to improve performance and minimize energy consumption (Portilla et al., 2010). These places restriction on scalability as predictions about future distribution have to be done before the distribution takes place. These limitations inhibit the adoption of public-key algorithms and architectures, like digital signature or session key distribution, which has being effectively used in traditional networks (Portilla et al., 2010). Public key encryption schemes allows for established private keys to be used between nodes after deployment (Portilla et al., 2010). The benefits of employing asymmetric algorithms with variable key length in WSN technologies are enormous. However the challenge can be surmounted by using a reconfigurable hardware resource.

Secure protocols currently in use are for the network layer and data link layer (Yick et al., 2008). However no layer is exempted from attacks, hence the need to have a platform that allows for real time reconfiguration of the entire set of layers (physical to application), thereby ensuring a secure WSNs need to be explored.

4. WSN reconfiguration approaches

In this section, we explore the role of Processing Elements and Embedded Operating Systems (EOS) technologies as enablers and in some cases expediters in realizing reconfigurable WSNs. In the course of our discussion, we will highlight the features, prospects and challenges associated with these role players accordingly.

4.1 The processing element approach

Processing elements by definition refer to the component in the Wireless sensor node or mote that performs the actual data processing operations and controls the entire system activities. A typical example of processing elements employed in Wireless sensor nodes are microcontrollers, digital signal processors and Field programmable gate arrays, etc. It can be argued that the degree or extend of WSN reconfigurability depends largely on the nature of the processing elements they integrate (Portilla et al., 2010; Leligou et al., 2008). Using this criterion, (Portilla et al., 2010) classified WSN reconfigurable platforms into the following categories: single microprocessor, Field Programmable Gate Array (FPGA), a combination of both microcontroller and FPGA, System on Programmable Chip (SoPC) and finally

configurable hardware and microprocessor. There are basically three of these platforms: those built around Microcontrollers with detachable sensor and RF communication modules, Software based processor running on Field Programmable Gate Arrays (FPGA) and System on Programmable Chip (SoPC).

4.2 The embedded operating system approach

Traditionally operating systems were designed to provide virtual hardware platforms of processing elements in order to ease the design, development and deployment of application programs (Chen et al., 2010). The aim is to export virtual machines resembling hardware to user programs, thereby enhancing portability and flexibility.

Major issues facilitating WSN reconfiguration via Embedded Operating Systems (EOS) are the Programming Model, Communication Protocol Support and Resource sharing. We review three EOS (TinyOS, Contriki and AmbientRT) and discuss the various reconfiguration paradigms adopted as well as their associated challenges. Some of these findings are summarized in table 2 below.

4.2.1 TinyOS

TinyOS (Ramamurthy et al., 2005; Sugihara & Gupta, 2008; Omer & Kunz, 2011) originally developed by the University of California, Berkeley and Intel is one of the most popular operating systems for Wireless Sensor Networks(WSNs). Its programming model is characterized by its use of component modularization. The model is composed of monolithic abstraction layers which are broken-up into smaller, self-contained building blocks that interact with each other via interfaces. The use of distinct interfaces preserves the modularity of the solution and promotes reuse (Chen et al., 2010). TinyOS is implemented using nesC, an extension of C programming language. nesC is a flexible programming language that allows the EOS to tune every parameter for special application needs such as energy efficiency (Sugihara & Gupta, 2008).

Its support for reconfigurability can be attributed to the Encapsulation-by-modules feature in nesC that provides a unified interface. This provision frees the programmer from being conscious of whether any proposed functionality is being implemented in hardware or software (Mallikarjuna et al., 2009; Barr, 1999). More also, the introduction of Hardware Abstraction Architecture (HAA), a three-layer architecture supports WSN platform flexibility in several ways. First the bottom layer, Hardware Presentation layer (HPL) provides access to basic resources such as registers, interrupts and pins via nesC interfaces. the middle layer referred to as Hardware Abstraction Layer provides abstractions of the full capabilities of the underlying hardware. The top layer Hardware Independent Layer (HIL) presents abstractions that are hardware independent and therefore cross-platform.

4.2.2 Contriki

Contriki OS developed at SICS, is an open source, highly portable, networked, multi-tasking operating system for resource constrained systems like WSN node (Ramamurthy et al., 2004). It employs advanced reprogramming support in the form of Loadable modules as

Embedded Operating System	Example of Wireless Sensor Nodes where applicable	Reconfigurable features	Reconfigurable Implementation strategy	Reconfiguration Implementation Constraints
TinyOS	Mica, TmoteSky, BTnode, EYES	Reconfiguration is supported by accessing the boot loader found in the Atmel processor residing in the motes flash memory Provides thin abstraction over the external microcontroller pins using micros that enable setting and clearing of input/output pins as well as changing the pin's direction and function	Use is made of events programming paradigm to change the behavior of code running on a node. Supports platform flexibility through three layers of abstractions (Hardware Abstraction Architecture(HAA)) : these are Hardware Presentation Layer(HPL) , Hardware Abstraction Layer(HAL) and Hardware Independent Layer(HIL)	Exported hardware abstraction interface are strongly biased by features of the Atmel AVR microcontroller. Thereby hampering porting of the EOS to new platforms. Event-driven concurrency limits explicit state machine implementation
Contriki	ESB/2, TmoteSky	Advanced reprogramming support in the form of Loadable modules Uses Convergecast routing service to separation of communication services and implementation of various network protocols.	It provides an abstract programming interface that applications can use to perform actual transmission and routing of data message	Not Available
AmbientRT	µnode	Provides online reconfiguration and support for a modular data driven architecture. Employs Data Centric Entities (DCE) and Dynamic Loadable Module (DLM) paradigm to effect reconfiguration during runtime.	Where other embedded operating systems offer configuration only during compile time, AmbientRT dynamically adapts its functionality to create the most efficient configuration n for every situation.	Not Available

Table 2. Various reconfiguration paradigms employed by EOS.

well as an abstract programming interface that applications can use to perform actual transmission and routing of data message.

4.2.3 AmbientRT

AmbientRT is a real time operating system with the following capabilities, online reconfiguration, support for modular data driven architecture and real time scheduling. The data driven architecture enables it to dynamically reconfigure its functionalities. Its kernel can load and run modules dynamically. Modules are blocks of application software meant to achieve the application's goal.

The EOS implements dynamic reconfiguration via the concepts of Dynamic Loadable Module (DLM). This implies that the DLM can be loaded and executed anywhere in the program memory. With the module support, changes to an application can be done more efficiently. The implications are that only a section of the application software meant to perform a particular task has to be changed and not the complete application itself. This results in less traffic and thus less energy consumption. The DLM can easily be transferred to target WSN node through RF communication links where on arrival it is written to the program memory for subsequent execution.

5. Design issues and challenges

The EOS approach highlighted in section 4.2 is however without challenges. These challenges are attributed to WSNs' operational overhead and jitter. The overhead and jitter can be traced to among others the duration it takes the RTOS to execute its inherent basic system services and those of the application it is required to manage (Lee et al., 2003; Mooney & Blough, 2002). Relieving the processing unit execution time and resources by migrating some of the RTOS services to reconfigurable hardware platforms is one way of removing the overheard. These had being implemented in several research works (Stankovic & Ramamritham, 1989; Burleson et al.,1993; Lee et al., 2003; Mooney & Blough, 2002; Argon et al., 2006). The use of field programmable gate array (FPGA) in implementing some key EOS/RTOS services as system on programmable chip (SoPC) has been demonstrated (Adomat et al., 1996; Heron et al., 2001; Andrews et al., 2004). Argon et al., for example, designed and implemented a priority scheduler module as part of a multithread RTOS kernel (Argon et al., 2006). Their aim was to provide a modular and modifiable RTOS scheduling component that could perform all scheduling processing using little or no CPU processing time.

Another research work worth mentioning is that of Kohourt et al, as presented in (Kohourt et al., 2003). They were able to realize a Real-Time Task Manager (RTM) – much of a processor extension, that can reduce the performance drawbacks associated with RTOS bottlenecks (Task scheduling, time management and event management).

It is also of note that the tasks to be implemented by EOS/RTOS in WSNs are always an integral part of the application code. Hence effecting any change as a result of varying application needs, one is constrained to either directly or remotely replace the entire firmware image. However, envisioning a fully reconfigurable WSN, one should expect the changes to be effected at runtime and not design time. In comparison to operating systems

run on much larger systems like the desktop or enterprise networks, it is hoped that future research work will be able to address this issue.

6. Conclusion

In this chapter we have been able to highlight and discuss various research approaches adopted towards realizing fully reconfigurable WSNs under severe constrained resources. A critical look at the EOS approach suggests a more convenient and efficient paradigm however challenges posed by overheads and jitter has raised some concern. However, we have equally shown that implementing an EOS or RTOS as a function of a reconfigurable hardware (SoPC implemented using FPGAs) in conjunction with the traditional EOS services can greatly enhance the efficiency and flexibility of WSNs

7. References

Adomat, J., Furunäs, J., Lindh, L., & Stärner. J., (1996). Real-Time Kernel in Hardware RTU: A step towards deterministic and high performance real-time systems. *In Proceedings of Eighth Euromicro Workshop on Real-Time Systems*, pp. 164-168, 'Aquila, Italy.

Andrews, D., Niehaus, D., & Ashenden, P., (2004). Programming Models for Hybrid FPGA/CPU computational Components , IEEE Computer, 2004(1) : 118-120.

Barr, M. (1999*). Programming Embedded Systems in C and C++", First Edition*, O' Reilly.

Burleson et al., (1993). The Spring Scheduling Co-Processor: A Scheduling Accelerator, *Proceedings of the International Conference on Computer Design (ICCD)*, pp. 140-144..

Carpenter, T. & Barrett, J. (2008). *CWNA Certified Wireless Network Administrator Official Study Guide, Fourth Edition*, McGraw-Hill.

Chen, Y., Chein, T. & Chou, P. (2010). Enix: A Lightweight Dynamic Operating System for Tightly Constrained Wireless Sensor platforms, *In SenSys '10* , Zurich, Switzland.-

Chong, C. & Kumar, S.P. (2003). Sensor Networks: Evolution, Opportunities and Challenges, *Proceedings of the IEEE*, 91(8).

Gomaa, H., (1993). Software Design Methods for Concurrent and Real-time Systems, First edition, Addison-Wesley.

Heron, J.P., Woods, R., Sezer, S. & Turner R.H., (2001). Development of a Run-Time Reconfiguration System with Low Reconfiguration Overhead, *Journal of VLSI Signal Processing*, vol. 28, pp 97-113.

Hill, J., Horton, M., Kling, R. & Krishnamurthy, L. (2004). The Platforms Enabling Wireless Sensor Networks, *Communications of the ACM*, 47(6)

Ibrahim,D.,(2008). Advanced PIC Microcontroller projects in C, *Newness*

Karloff, C., Sastry, N. & Wagner, D. (2004). TinySec: A link Layer Security Architecture for Wireless Sensor Networks, *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys 2004*), Baltimore, MD.

Khan J. & Vemuri, R., (2005). Energy management in battery powered sensor networks with reconfigurable computing nodes, *in Proceedings of the International Conference on Field Programmable Logic and Applications (FPL '05*), vol. 2005, pp. 543–546, Tampere, Finland.

Kohout, P., Ganesh, B. & Bruse, J. (2003). Hardware Support for Real-time Operating Systems. *CODES-ISSS'03*, California, USA.

Kompis, C. & Sureka, P. (2010). Power Management Technologies to Enable Remote and Wireless Sensing, *Cyber Security White Paper*, Available from: www.libelium.com/libelium.../libelium-ktn-power_management.pdf.

Kulkarni, K., Sanyal, S., Al-Qaheri, H. & Sanyal, S. (2009). Dynamic Reconfiguration of Wireless Sensor Networks. *International Journal of Computer Science and Applications*: 6(4),pp 16-42.

Labroasse, J., (2002) MicroC/OS-II The Real Time Kernel, *Newnes*.

Lee, J., Mooney, V., Instrom, K., Daleby, T., Klevin, T., & Lindh, L., (2003). A comparison of the RTU Hardware RTOS with a Hardware/Software RTOS, *Proceedings of Asia and South Pacific Design Automation Conference*, Asia.

Leligou, H.C., Redondo, L., Zahariads, T., Retamosa, D.R., Karkazis, P., Papaefstathiou, I. & Voliotis, S., (2008). Reconfiguration in Wireless Sensor Networks. *ARTEMIS-2008-100032, SMART 2008.*

Luk, M., Mezzour, G., Perri, A., & Gligor, V. (2007). MiniSec: A Secure Sensor Network Communication Architecture, *IPSN'07*, Cambridge, Massachusetts, U.S.A.

Mallikarjuna A., Reddy V., Kumar, P., Janakiram, D. & Kumar, G .A. (2009). Operating Systems for Wireless Sensor Networks: A Survey Technical Report, International *Journal of Sensor Networks (IJSNet*), 5(4) : 236 – 255.

Moerschel, G., Dreger, R., Carpenter, T. (2007). *CWSP Certified Wireless Security Professional Official Study Guide, Second Edition* , McGraw-Hill.

Mooney, V., & Blough D.M.,(2002). A Hardware-Software Real-Time Operating System Framework for SOC's, *IEEE design and Test of Computers*, 2002(11):44-51.

Muralidhar, P. & Rao, C. (2008). Reconfigurable Wireless sensor network node based on NIOS core, *Processings of the 4th International Conference on Wireless Communication and Sensor Networks(WCSN'08),* Allahabad, India pp 67-72.

Omer, M. & Kunz, T. (2011). Operating Systems for Wireless Sensor Networks: A survey", *Sensors (an open access Journal)* , 11 (2011) : 5900-5930.

Perrig, A., Szewczyk, R., Wen, V., Cullerand D., Tygar, J.D. (2001). SPINS: Security Protocols for Sensor Networks, *Proceedings of 7th Annual International Conference on Mobile Computing and Networks*, Rome, Italy.

Portilla, J., Otero, A., Torre, Riesgo, T., Stecklina, O., Peter, S. & Langendorfer. P. (2010). Adaptable Security in Wireless Sensor Networks by using Reconfigurable ECC Hardware Coprocessors. *International Journal of Distributed Sensor Networks*.

Ramamurthy, H., Prabhu, B.S. & Gadh, R. (2004). Reconfigurable Wireless Interface for Networking Sensors (ReWINS), *Proceedings of IFIP TC6 , 9th International Conference on Personal Wireless Communication*. Netherlands.

Ramamurthy, H., Lal, D., Prabhu, B.S., & Gadh R. (2005). ReWINS: A Distributed Multi-RF Sensor Control Network for Industrial Automation, *IEEE wireless Telecommunication Symposium WTS 2005*, Pomona, California.

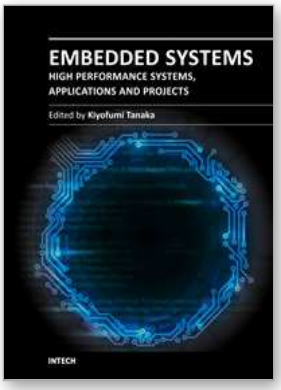
Stankovic, J.A. & Ramamritham, K, (1989). The Spring Kernel: A New Paradigm for Hard Realtime Operating Systems, *ACM Operating Systems Review*, 23(3), pp. 54-71.

Sugihara, R. & Gupta, R.K. (2008). Programming Models for Sensors Networks: A Survey, *ACM Transactions on Sensor Networks*, 4(2).

Walls, C., (1996). RTOS for Microcontroller Applications, *Electronic Engineering*, 68(831): 57-61.

Yick, J., Mukherjee B., & Ghosal, D. (2008). Wireless sensor network survey, *Journal of Computer Networks*, 52(2008):2292-2330.

Zuberi, K.M. & Shin, K.G, (1996). EMERALDS: A Microkernel for Embedded Real-Time Systems, *Proceedings of RTAS*, pp.241-249.



Embedded Systems - High Performance Systems, Applications and Projects

Edited by Dr. Kiyofumi Tanaka

ISBN 978-953-51-0350-9
Hard cover, 278 pages
Publisher InTech
Published online 16, March, 2012
Published in print edition March, 2012

Nowadays, embedded systems - computer systems that are embedded in various kinds of devices and play an important role of specific control functions, have permeated various scenes of industry. Therefore, we can hardly discuss our life or society from now onwards without referring to embedded systems. For wide-ranging embedded systems to continue their growth, a number of high-quality fundamental and applied researches are indispensable. This book contains 13 excellent chapters and addresses a wide spectrum of research topics of embedded systems, including parallel computing, communication architecture, application-specific systems, and embedded systems projects. Embedded systems can be made only after fusing miscellaneous technologies together. Various technologies condensed in this book as well as in the complementary book "Embedded Systems - Theory and Design Methodology", will be helpful to researchers and engineers around the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sanjay Misra and Emmanuel Eronu (2012). Implementing Reconfigurable Wireless Sensor Networks: The Embedded Operating System Approach, Embedded Systems - High Performance Systems, Applications and Projects, Dr. Kiyofumi Tanaka (Ed.), ISBN: 978-953-51-0350-9, InTech, Available from: http://www.intechopen.com/books/embedded-systems-high-performance-systems-applications-and-projects/implementing-dynamic-reconfigurable-wireless-sensor-networks-

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821