An Activity-Based Sensor Networks Course for Undergraduates with Sun SPOT Devices

Damon Tyman
Portland State University
Portland, OR
damont@pdx.edu

Nirupama Bulusu
Portland State University
Portland, OR
nbulusu@cs.pdx.edu

Jens Mache
Lewis and Clark College
Portland, OR
jmache@lclark.edu

ABSTRACT

Wireless sensor networks are revolutionizing the instrumentation of the physical world, across scientific, industrial and military applications. In this paper, we describe our efforts developing and classroom-testing hands-on materials for use in undergraduate-accessible courses on sensor networks. In Winter 2008 at Portland State University, we introduced an in-class laboratory component to a sensor networks course that had previously been entirely lecture-based. For the laboratory exercises, we utilized Sun's Java-programmable Sun SPOT [7] sensor network technology. We found the Sun SPOT based laboratory activities to be quite powerful as a teaching and excitement-fostering tool.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real Time and Embedded systems, Microprocessor/Microcomputer Applications, Signal Processing Systems; C.2.1 [**Network Architecture and Design**]: Wireless Communication; K.3.2 [**Computer and Information Science Education**]: Computer Science

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Wireless sensor networks, Java, Sun SPOTs, embedded computing.

1. INTRODUCTION

Wireless sensor networks were named by MIT Technology Review in 2003 as "one of the ten technologies that will change the world in the 21st century." They are increasingly being utilized to monitor the physical world across a wide array of applications, ranging from habitat monitoring to reducing energy consumption in data centers. Though interest in wireless sensor networks has been increasing amongst computer scientists in recent years, institutions that offer courses dedicated to the topic are still in the minority. Additionally, what courses are offered tend mostly to be designed for and taught only at the graduate level.

Our aim is to make our introductory wireless sensor networks class accessible to as many individuals and groups as possible, while still offering a thorough and engaging survey of the field. Primarily, we want the course to be open to undergraduates. By fostering student excitement early, we believe we increase the likelihood that they finish a computer science degree and consider graduate school. We hope that in the future, more schools will include similar undergraduate sensor network courses.

One of the challenges of teaching sensor networks to undergraduates is the lack of pre-existing educational materials appropriate for this level. There are textbooks on the subject, but most focus on a specific research problem or sub-domain. A few do provide solid introductions to all the most important sensor network concepts, but if used in a class, the instructor must still carefully select what topics to cover, the order of coverage, and the level of detail for each concept. Due to the rapid rate at which research in this area has been advancing recently, a textbook from a few years ago may already be out-of-date in regards to many of the latest techniques and algorithms favored for certain important tasks. Thus, only those instructors firmly immersed in the sensor network research community are properly equipped to take on a class of this kind.

Figuring out how to properly incorporate a hands-on in-class laboratory component to a course can also be a formidable challenge. Since wireless sensor networks are inherently a more hands-on technology than most others encountered in computer science, we figured adding this aspect to the in-class experience would help increase student excitement by requiring students to do things and witness the results themselves, rather than just reading about what should happen. Also, we think properly designed laboratory exercises are an invaluable way to reinforce the lecture concepts because they allow students to study each key concept from a theoretical, research-based perspective (in lecture) as well as experiencing related practical implications or implementation challenges on their own.

Unfortunately, most embedded sensors run special operating systems, like TinyOS [9], and require learning new programming languages such as NesC [6]. With the amount of effort required for students to become proficient in programming TinyOS and NesC, a course that chooses to familiarize students with them would need to allow for a long learning period and make significant sacrifices in lecture concepts covered. Furthermore, students' abilities are not likely to be such that laboratory exercises based on lecture concepts would be reasonable.

The reasons described above motivated us to choose Sun Microsystems' relatively new Java-programmable sensor network devices, called Sun SPOTs [7]. With experience programming in Java, one is able to get started programming the Sun SPOTs quite quickly compared to motes that run TinyOS. We found this significantly gentler learning curve, combined with durable packaging and nifty features (8 programmable tri-color LED's, onboard accelerometer, 2 programmable switches to name a few), made these perfect teaching tools for use in the laboratory-based part of our class. With the Sun SPOTs, students were able to gain programming proficiency quick enough to complete exercises that accompany and exemplify the material covered in lecture.

As we are attempting to accommodate a somewhat broader group of students than is customary for this kind of course, one of our goals is to identify and supply relevant background material wherever possible for laboratory exercises (labs) or homework assignments. Each lab write-up begins with pointers to pre-lab readings and background material so that all students, each with their own varying backgrounds and strengths, can get up to speed.

2. EXERCISE DESCRIPTIONS

In this section we describe the laboratory exercises that we developed for the winter 2008 course. The first few labs were designed to introduce the students to the Sun SPOTs and get them started learning how they are programmed. After that, each week's lab was designed to correspond with the lecture topic for that week. For each lab, a detailed write-up introduces the goals of the lab, walks students through the steps for completion, providing hints and notes along the way, and offers challenges for further investigation at the end. While the beginning four labs are mostly tutorial style, taking students step-by-step through a process, the later labs tend to consist more of simple instructions as to what application(s) to build or measurements to make. The lab write-ups from the course are available at [11].

2.1 Labs 1, 2: Introduction

Labs 1 and 2 served as the first time that students had seen or worked with the Sun SPOTs. The first lab is designed merely for the purpose of introducing students to the physical devices and their capabilities. The second lab goes further in showcasing what can be achieved with this technology; it also teaches students to deploy new applications to the SPOTs and manage them from the software. It is important to note that for all these exercises students had access to computers with all the relevant software pre-loaded. Requiring installation to be completely by the students as a first step, though a relatively simple process with Sun's SPOT Manager would have detracted from students' learning experiences and delayed their exposure to the SPOT capabilities. Over the course of the term, many students ended up installing the necessary components on their own computers anyway, so presenting the process in class would not have been productive.

Sun provides with the Sun SPOT Software Development Kit (SDK) many demo applications that showcase various possibilities for SPOT applications. One such demo, called the Ectoplasmic Bouncing Ball, comes preloaded on the SPOTs when they are first purchased. We think that this serves as a great introduction to the SPOTs and a way to familiarize people with interacting with them, so our first lab is based around this demo. By experimenting with it, students learn the layout of SPOTs. They are shown something fun and cool that can be done with the devices. In most cases, when people encounter this for the first time, they are quite impressed, and it works well for generating excitement about the capabilities of the technology. In winter 2008, we accompanied the first lab with some Java review questions based off some of the SPOT application code. These questions turned out to be somewhat more difficult than had been intended, but were useful in revealing what level students' Java programming skills were at.

In the second lab, we cover the process of deploying applications to the Sun SPOTs. We show how to do this through NetBeans (with the SPOT modules installed), but also make it clear where to look to learn how to do it over the command line. We have students deploy and experiment with the Telemetry demo. With this demo, students also learn how to use a SPOT base station for the first time. The Telemetry demo (supplied with the SDK) consists of both a host application, run on a PC with base station attached, and a SPOT application, run on a remote SPOT. The host application displays a graph that reports the accelerometer data from the remote SPOT. Thus, as one causes the remote SPOT to experience acceleration, the change in acceleration reading will be displayed on the graph in real time. At the end of this lab, students should be comfortable working with the SPOTs and associated development tools.

One thing we encountered while doing this lab in class is that sometimes one student's host application will communicate with another student's remote SPOT. As a take-home assignment, students were required to examine the application code and modify it so that a student's host application will only connect to the designated (hard-coded) remote SPOT. This served as a great way to introduce working with SPOT application code without requiring significant programming effort.

2.2 Labs 3, 4: Communication

Labs 3 and 4 make use of provided code snippets to illustrate how certain communication tasks can be performed with the SPOT API. Along the way, students are taught both the basics of developing SPOT applications and the specifics of two communication protocols. After students get the example applications working for which the code is given in the write-ups, challenges are provided in both labs that require students to add or modify code to demonstrate their understanding of applications.

In Lab 3: Radio Communication, students are taught how to code basic radio communication between remote SPOTs. This is an essential task in almost all wireless sensor network applications. Of the two protocols provided by the API, radiostream and radiogram (similar to TCP and UDP), the example application in lab 3 makes use of the radiogram protocol. The existence of two protocols and a reference to radiostream are, however, made clear in the lab write-up. The example application has each remote SPOT keep a number that is regularly incremented and broadcasted over the radio. At the same time, each SPOT also displays the numbers it receives over the radio from other SPOTs with its LEDs. As a homework following this lab, students were asked to test the packet loss rate at various distances between SPOTs. This homework presented both experimental and programming challenges and demonstrated one of the fundamental characteristics of sensor network wireless communication (over 802.15.4).

Lab 4: HTTP Access follows a similar format wherein code snippets are presented and then explained. The student is taught to construct an application from these code snippets. Lab 4 teaches students how the socket proxy application can be run to use a basestation to supply remote SPOTs with access to the Internet through the connected host PC. The remote SPOTs run an application that gets an HTML page from the Internet and parses it, flashing its LEDs whenever certain tags are encountered. At this point in the term, students demonstrated a readiness to begin building their own applications with minimal code provided.

2.3 Lab 5: Localization

Network localization refers to the computation of the positions of sensor nodes relative to some external frame of reference. Although the Global Positioning System is widely available, its use is often limited by cost and practical deployment constraints (indoors, underwater). Researchers have been developing and testing a number of algorithms and systems to localize devices at unknown positions from ranging constraints (acoustic, vision, radio etc.) to devices with known positions.

In Lab 5, students were required to program a remote SPOT to localize itself using a simple algorithm and coordinates received over the radio from fixed beacons. Six beacons were preprogrammed and placed throughout the room where the lab took place. The trivial code for the beacons is also present in the write-up for those that are curious. Students were to program their remote node to listen for beacon broadcasts and use the RSSI (Received Signal Strength Indication) value associated with each beacon as a weight in a weighted centroid computation [1]. In this way, each node's computed coordinate is merely the weighted average of the beacon coordinates where the RSSI values associated with the beacons are used as weights. Nodes were to calculate new coordinates at regular intervals. Students were definitely up to the challenge of this lab, and seemed to enjoy doing it; see Figure 2 and Table 1.

2.4 Lab 6: Power Consumption

Wireless sensor networks are expected to be deeply embedded in the physical world and operate unattended and un-tethered for several years, running on small batteries, and occasional availability of energy harvesting sources such as wind and solar. Energy conservation and management of harvested energy are dominant design drivers of sensor networks.

For a lab, we decided to have students investigate the power consumption characteristics of the SPOTs. Through the API, one is able to get a reading either of the current being drawn from the battery at a specific point in time (close to the time of the API call) or the maximum current that was drawn between two method calls. With instructions on how to do this, the lab called for students to measure the maximum current drawn during a number of different scenarios including radio transmit, radio receive, and intensive computation.

Besides measuring power consumption (as best as one can through API calls) at various times, the lab asked students to investigate the battery-saving deep sleep mode available on the SPOTs. Not only were they to attempt putting a SPOT into deep sleep mode, but also they were asked to answer a number of questions to test their understanding of the requirements for and characteristics of deep sleep. It would be nice in future versions of this lab to measure power consumption with an oscilloscope and/or find some device to attach to SPOTs to achieve some sort of energy harvesting.

2.5 Lab 7: Security

Network security is an extremely vast topic, even as it applies to wireless sensor networks. Because of sharp memory, energy and bandwidth constraints, symmetric key cryptography is widely preferred in embedded sensor networks.

In the security lecture, we discussed the specifics of the Eschenauer-Gligor Key Management Scheme [3] and decided to use this lab as a reinforcement of how the scheme is designed to work. In this key management scheme, there is generally a master pool of security keys from which all the keys stored on each node are drawn. The number of keys stored on each node, however, is much smaller than the cardinality of the master key pool. Furthermore, each node could have a unique set of keys, and there is no assurance that any two nodes share any keys. However, the scheme is based on the principle that each node need only have a fraction of the total keys for there to be a decent chance of any two nodes being able to communicate (meaning they share at least one key), similar to the birthday problem. With a pool size of 10,000 keys, each node having only 70 keys, will result in a roughly 50% chance that any two nodes can communicate.

For our lab, we aimed to simulate this key management scheme. A key manager was programmed which would maintain a pool of keys and return a specified number of keys when requested to do so. The code to make such a request and obtain a set of keys was provided for students. It was the job of the students to program two nodes that would request sets of keys of varying amounts and for each amount of keys, repeatedly request new keys and measure how often the two nodes shared at least one key. Thus, students were supposed to empirically determine how many keys were necessary, given our pool size of 1000, for two nodes to have a roughly 50% chance of sharing a key.

2.6 Lab 8: TinyOS Tools

While we were very happy with our decision to use the Sun SPOTs for most of our laboratory exercises, we also reasoned that due to the continuing importance of the TinyOS operating system in sensor network research, it would perhaps be good to at least introduce students to the TinyOS operating system, the NesC language, and the MICAz mote hardware [2].

Unfortunately, we discovered that it is difficult to do so in a single class period. Originally, we had hoped to have this lab on TinyDB, however we had trouble getting that to work in a limited time frame so we chose to do a TinyOS introduction instead. Owing to the time constraint of a single class and the vastness of TinyOS as a topic, we decided to make this lab period more open-ended than preceding labs. In actuality, this seems to have back-fired and made students feel less guided rather than empowering them with the freedom to work on their own.

The suggested paths to take for this lab were to read some of the TinyOS tutorials to learn the basics of NesC, learn to deploy applications onto the Micaz's, test various demo programs, and to experiment with the TinyOS simulator, TOSSIM. While students almost unanimously like this lab the least, we believe that providing an introduction to alternative technologies is still a positive thing. In the future, we may make this into two structured laboratory exercises.

2.7 Final Assignment: Contour Tracking

After all the in-class labs had been completed, students were given a final exercise to complete on their own, a contour tracking assignment. Contour tracking is one of the canonical tasks that must be performed across many different sensor networks. For example, a sensor network could be deployed to report the boundary of a chemical spill, forest fire or report temperature and humidity iso-contours within a habitat. Implementing a contour tracking application requires integrating all previously taught sensor networking concepts. For this project, students were tasked with programming a grid of Sun SPOTs (usually nine) to identify a light contour. Thus, when a flashlight was placed amongst the SPOTs, the SPOTs would record light values and communicate amongst themselves to determine where the light contour occurred. Once the contour was determined, this was indicated by use of LEDs or printed message on a host computer. Figure 1 demonstrates how this is to work. Given the preparation from the other lab exercises leading up to this, students were fully capable of completing this assignment in less than two weeks at the end of the term. One student solution was shown at the demonstration session of ACM SIGCOMM 2008 conference [10].

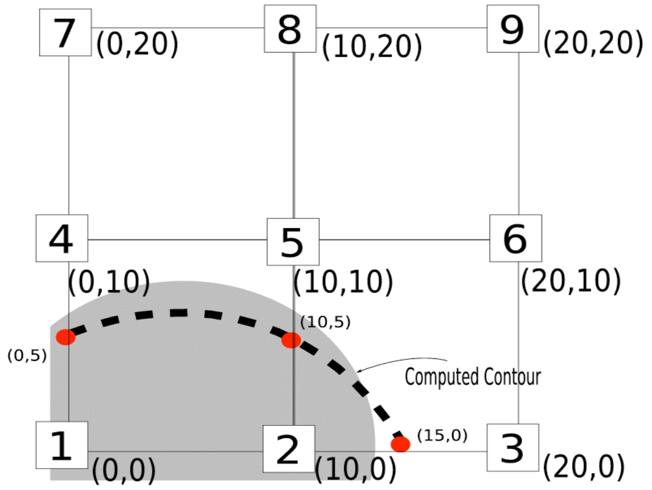

Figure 1: Illustration of the algorithm for contour tracking used in the assignment

3. FEEDBACK

To better understand the student experience with sensor network programming, we did our best to collect feedback from the students throughout the course of the winter 2008 wireless sensor networks class. For each lab, we made available feedback forms that allowed people to rank the interest and difficulty of the labs as well as provide comments of any sort desired. These have been quite useful in helping us determine what changes to make for the next iteration of the course. They were also reassuring in proving that the overall response to the course structure and laboratory exercises was generally very positive.

Besides the general feedback forms that were available throughout the term, we also distributed a special feedback form at the end of the course that requested that students rate every lab in terms of how interesting it was and how well it reinforced the related lecture. The results from nine of these completed forms (one student failed to complete the form) are shown in Figure 2. Additionally, we asked students to rank the labs in order from favorite to least favorite. Table 1 lists the results from the nine responses to this part of the form. The first four columns of the table show responses from undergraduate students.

Table 1: Student ranking of lab preferences

Lab\Student	1	2	3	4	5	6	7	8	9	Mean
Introduction	5	5	4	6	5	5	5	4	4	4.78
Communication	4	3	5	4	4	4	4	2	1	3.44
Localization	1	2	1	1	1	3	1	2	2	1.56
Power	3	4	2	2	2	2	2	1	3	2.33
Security	2	1	3	3	3	1	3	4	5	2.78
Tools	6	6	6	5	6	6	6	6	6	5.89

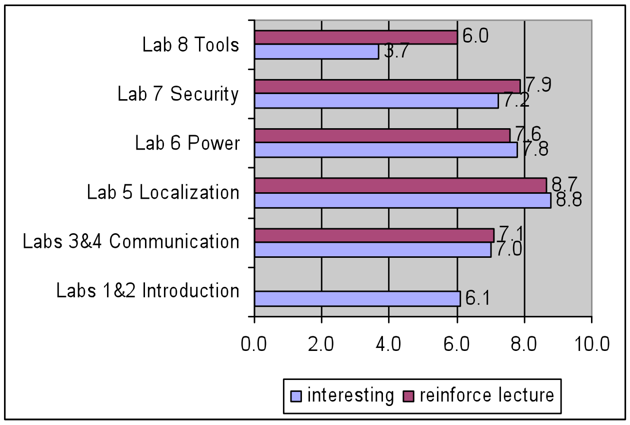

Figure 2: Student lab ratings (0-low, 10-high)

As can be seen from Table 1 and Figure 2, students generally found all labs except for Lab 8 to be interesting, and confirmed that the labs were effective at reinforcing lecture. Students most preferred the localization, power, and security labs. In all, five of the students (half of the class) from the winter sensor networks course also took the subsequent spring advanced sensor networks course (this course does not require having taken the one taught in winter). Three of these students utilized Sun SPOTs for their class projects, and another originally planned to. In contrast, none of the students that only took the spring course chose to make use of Sun SPOTs for their projects.

In addition to the quantitative feedback, the students also gave us verbal feedback. Unanimously, the students agreed that the labs were very useful and probably the best part of the course. In the words of one student, "The course is very lab oriented and it helps in understanding concepts. The lab instructions are very clearly defined. We both learned about the concepts pertaining to the subject matter, and also got to practice working with the technology hands-on." On the flip side, students requested more orientation about techniques for debugging the Sun SPOTs and possibly further simplifying the Demo code provided by Sun.

4. PLANNED CHANGES

Based on feedback from students, lessons learned, and things we were unable to accomplish the first time around, we have a number of planned changes and additions to the labs for next winter's course. For one, students rated the first few labs as too easy and seemed to be ready to start developing their own applications sooner than we required. For the next time, we plan to combine the first four labs into one to three labs and change the initial pace.

For the localization lab, we are planning to add coverage of a more mathematically significant algorithm, and possibly even accelerometer-based localization techniques. A new lab will be introduced to cover experimentation with the recently released SunSPOT Emulator. The background reading listings are being expanded for all labs, including references to relevant sections of the Sun's new SunSPOT online certification course [8]. Another lab that is being planned will be centered on sensor networks with heterogeneity. The TinyOS lab will be completely redone, and may be turned into a two-part lab. Time and equipment allowing, the power lab may be enhanced with oscilloscope power readings or energy harvesting capabilities.

The latest version of the Sun SPOT SDK, Blue, introduces several new demo applications we plan to evaluate for our course. Of particular note is one that demonstrates database interaction between Sun SPOTs and JavaDB (now included with the SDK).

5. RELATED WORK

Research on teaching sensor networks, especially to undergraduates, is extremely limited. References [4][5] present TinyOS-based laboratory exercises suitable for undergraduates. Described in [4] are the efforts of four undergraduates to get current TinyOS tutorials working on MicaZ motes in a short time frame. The difficulties faced by the research students are presented along with a discussion about the appropriateness of using the materials in an undergraduate course.

In [12], Yang *et al* describe projects for teaching wireless sensor networks using TinyOS with Mica2 and TelosB motes. They present two exercises: one on data collection utilizing code from the MoteWorks development package, and one on detecting human presence utilizing code from EasySen. In contrast to prior work, our focus has been on using the Java programmable Sun SPOT devices, rather than TinyOS based motes, to make sensor networks more accessible to undergraduates in their junior years.

6. CONCLUSION

Overall, we found our first hands-on laboratory-based undergraduate-accessible wireless sensor networks course to be a significant success. Students responded well to working with the Sun SPOT devices, and seemed more engaged to pursue knowledge on their own because of them. Students liked most of the labs that required that they develop meaningful applications on their own and that were relevant to the lecture concepts.

We feel that the gentler learning curve of the SPOTs was a necessity for developing labs that reinforce important concepts from the lecture, and that this would have been impossible if students were required first to learn TinyOS and NesC. At the end of the 10-week course, students had become quite proficient in developing reasonably complex sensor network applications for the Sun SPOTs (as demonstrated by the success of the contour tracking assignment), and many seemed eager to continue to pursue knowledge in this area. Since this was our first time teaching a Java based sensor network programming course, we are optimistic for significantly improved student productivity in the next reincarnation of the course.

7. ACKNOWLEDGMENTS

We would like to thank Jeanette Palmiter, as well as Suman Nath from Microsoft Research, and Mark Yarvis and Chieh-Yih Wan from Intel for serving on our course advisory board and for helpful discussion. This work is supported by the National Science Foundation through NSF grants CNS-0720914 & CNS-0720545 and by an equipment donation from Sun Microsystems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Blumenthal, J., Grossman, R., Golatowski, F., and Timmerman, D. Weighted Centroid Localization in Zigbee-based Sensor Networks. In Proceedings of IEEE International Symposium on Intelligent Signal Processing (Alcala de Henares, October 03 - 05 2007). WISP '07. IEEE, 1-6. DOI= 10.1109/WISP.2007.4447528

[2] Crossbow MICAz motes. http://www.xbow.com/products/product_pdf_files/wireless_pdf/6020-0060-01_a_micaz.pdf.

[3] Eschenauer, L. and Gligor, V. D. A Key-Management Scheme for Distributed Sensor Networks. In Proceedings of the 9th ACM Conference on Computer and Communications Security (Washington, DC, USA, November 17-21 2002). CCS '02. ACM Press, New York, NY, 41-47. DOI= http://doi.acm.org/10.1145/586110.586117

[4] Mache, J., Allick, C., Charnas, J., Hickman, A., and Tyman, D. Sensor Network Lab Exercises Using TinyOS and MicaZ Motes. In Proceedings of the International Conference on Pervasive Systems and Computing (Las Vegas, NV, June 2006). PSC '06. CSREA Press, 154.

[5] Mache, J., Dean, E., and Imber. K. Sensor Network Lab Exercises Using TinyOS and MicaZ Motes, Part II. In Proceedings of the International Conference on Wireless Networks (Las Vegas, NV, June 25-28 2007). ICWN '07. CSREA Press, 464-467.

[6] NesC. http://nescc.sourceforge.net/.

[7] Sun SPOTs. http://www.sunspotworld.com/.

[8] Sun Student Courses: SPOTs-101. http://www.sunstudentcourses.com/course/view.php?id=12.

[9] TinyOS. http://www.tinyos.net/.

[10] Tyman, D., Dua, A., Mache, J., and Bulusu, N. "Contour Tracking: A Comprehensive Student Project for Sensor Network Education." In Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (Seattle, WA, August 17-22 2008). SIGCOMM '08. ACM Press, New York, NY, 519.

[11] Portland State University Wireless Sensor Network Course Labs. http://sys.cs.pdx.edu/home/sunspots.

[12] Yang, T. A., Jain, D., and Sun, B. Development of Emulation-Based Projects for Teaching Sensor Networks. *J. Comput. Small Coll.* 24, 2 (Dec. 2008), 64-71. ACM Press, New York, NY, USA.