Building a Low-Cost Digital Garden as a Telecom Lab Exercise

In an interdisciplinary, semester-long class, undergraduate students learn how to build a low-cost, multihop wireless sensor network from first principles for a digital garden. This type of course better prepares electrical engineering graduates for the sensor-rich, pervasive computing era.

Aggelos Bletsas
Technical University of Crete

Aikaterini Vlachaki
University of Alberta

Eleftherios Kampianakis
Technical University of Crete

George Sklivanitis
State University of New York at Buffalo

John Kimionis and Konstadinos Tountas
Technical University of Crete

Megasthenis Asteris
University of Southern California, Los Angeles

Panagiotis Markopoulos
State University of New York at Buffalo

Modern digital communications, informatics, and electronics increasingly blend together in our daily lives. Cell phones, for example, have evolved into powerful, handheld computers with smart human-computer interfaces, broadband wireless connectivity, and a multitude of sensors (cameras, capacitive touchscreens, accelerometers, GPS receivers, and so on). Similarly, computation and information exchange has evolved from legacy mainframes and proprietary networks to handheld personal computers, grid computing, open-access broadband networks, and environmental sensor networks. Sensor networks bridge the analog world we live in to the digital infrastructure, computers, and networks developed to process and communicate any type of information.

As technology evolves in these interdisciplinary directions, it becomes increasingly important for electrical engineering departments to refresh their teaching curricula to reflect such knowledge diversity. Wireless sensor networks (WSNs), in addition to presenting an unquestionably fertile field for research, offer the perfect opportunity to demonstrate the integration of electronics, digital communications, networking, and computer science.

Here, we describe an educational experiment conducted in 2010 in the Electronic and Computer Engineering Department at the Technical University of Crete, Greece. The fundamental question was whether fourth-year undergraduate students could integrate diverse knowledge spanning electronics, telecommunications, and computer science to tackle difficult engineering problems and create a novel application, applying know-how and problem-solving skills developed throughout the course. The final term project was a custom, low-cost WSN for soil moisture and environmental temperature sensing. Other WSN applications would be appropriate for this course,[1–8] but the students built the digital garden application from first principles—without exploiting existing, commercial WSN technology.

Course Curriculum

The course contains a diverse curriculum of theoretical knowledge provided in the classroom

 Published by the IEEE CS ■ 1536-1268/13/$31.00

Course Duration and Prerequisites

There are different class times and prerequisites for each portion of the class.

Theory

The theoretical part spans the whole duration of the semester course, with two 1.5-hour lectures per week. Students should have mastered vector analysis and advanced integral calculus, signal and systems, digital communications, as well as introductory circuits and electronics. Theoretical exercises include simple questions or numerical problems, provided on a weekly basis. Material covered includes the first five chapters of Behzad Razavi's *RF Microelectronics* (Prentice Hall, 1998), the first two chapters of Kai Chang's *RF and Microwave Wireless Systems* (John Wiley & Sons, 2000), and the last chapter in George Sergiadis's *Synthesis of Telecom Modules* (University Studio Press, 2000 [in Greek]), as well as instructor's notes.

Programming Lab

The students should have mastered basic programming skills. Knowledge of operating systems programming is certainly useful but isn't required. Prior experience with embedded systems isn't assumed. This lab typically spans two three-hour labs (in two weeks). Additional attention and time should be spent, if needed, given the importance of embedded programming in the subsequent lab.

Communication Lab

Two three-hour labs are required for step-by-step explanation of the example code (roughly one three-hour lab for transmitter operation and one for receiver operation). The students are asked to browse the radio manual and answer a set of relevant questions before coming to class. Two additional three-hour labs are needed for practice and successful completion of the exercises. It's important for students to have deeply grasped the basic concepts of embedded programming during the preceding course labs.

and hands-on experiments with real-world (nonsimulated) equipment in the lab. (See the sidebar for course duration and prerequisites information.)

Radio Communication and Systems Engineering Theory

The theoretical part of the course commences with basic concepts of communication engineering and telecom electronics, including dynamic range (of analog-to-digital converters, for example), receiver sensitivity, and thermal noise, typically quantified by *noise figure* (NF). Students learn to calculate the NF of cascaded modules, where each module is characterized by its individual voltage (or power) gain and NF. Additionally, students learn to quantify nonlinear products of communication systems, with special emphasis on harmonic and intermodulation products. Similar to the NF case, students learn to quantify intermodulation products of cascaded systems in terms of the *third intercept point* (IP3). Finally, they learn about the fundamental tradeoff between minimization of NF and maximization of IP3—that is, between the thermal noise and the nonlinearity of telecom modules.

This set of knowledge is essential for any electrical or electronic engineer, given that NF and IP3 are macroscopic quality parameters that characterize many commercial communication modules. Unfortunately, such important information, although omnipresent in equipment manuals and heavily used by industry, isn't typically studied in classic electrical engineering (EE) undergraduate courses. Unlike other telecom lab curricula, this course follows an interdisciplinary approach and explains these concepts, providing simple examples from micro-electronics.

The course continues with receiver architectures, covering homodyne (zero-intermediate frequency) receiver front ends and their inherent sensitivity problems, heterodyne and superheterodyne architectures and their intrinsic channel selectivity-sensitivity tradeoff, and image-reject and digital-IF receivers (including modern embedded radios). Design schematics at the transistor level are also briefly explained.

Finally, the course provides an introduction to antenna and transmission line theory and an overview of basic performance metrics that enable students to examine industry manuals of relevant equipment (antennas, cables, or wave guides) and assess their quality. Given the breadth and depth of relevant theory, the course can't take the place of a class dedicated to antenna and microwave systems. However, by the end of the course, students can solve practical electromagnetic problems, understand the concept of transmission lines, use the appropriate tools (such as a Smith chart), understand guided waves and propagation in free space, and evaluate the voltage standing wave ratio of commercial antennas.[9] They also can answer questions such as, "When does a capacitor operate as a coil?" or "When does your UHF TV receiver jam (interfere with) your GSM mobile phone?" These simple questions require deep theoretical background information.

Embedded Programming Lab

Students typically attend two three-hour lab sessions, during which time they work in pairs and practice on a low-cost 8051 microcontroller unit (MCU) development board from Silabs

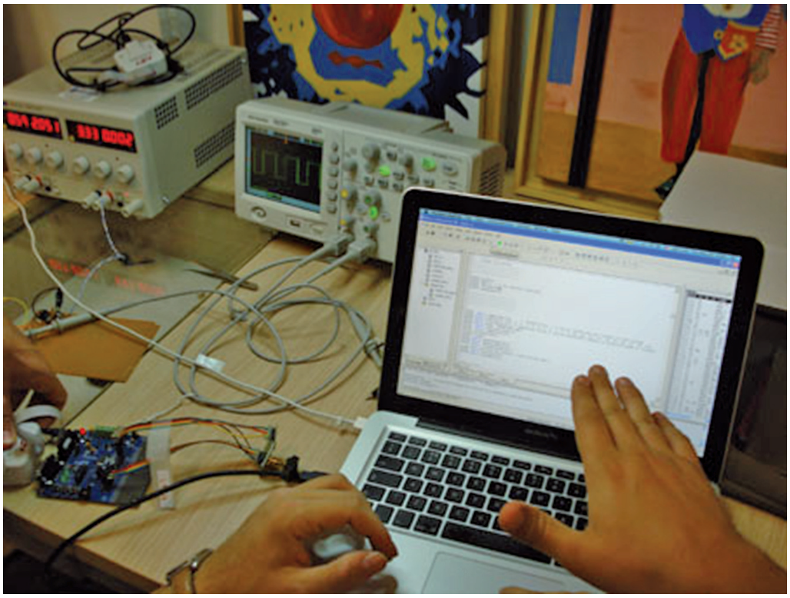


Figure 1. A typical lab bench includes a personal computer and the microcontroller unit (MCU) development board. A digital oscilloscope and voltage supply (also depicted) help with embedded development.

(see Figure 1). Specifically, they use the C8051F320DK unit, because the F320 family supports USB interfaces. The specific 8-bit MCU comes with a well-organized manual that thoroughly presents the internal modules and helps even inexperienced users quickly develop custom projects. The development kit also comes with plenty of software examples, both in C and Assembly.

Most fourth-year EE students have already had ample experience in digital design, though they might lack experimental skills in MCU programming. The lab follows a programming-through-examples approach and starts with introductory "hello-world" code—such as blinking the LEDs on the development kit. Hardware functionality is mainly controlled through special-purpose register values, and students learn how to

- efficiently read the MCU manual to initialize and fully exploit the MCU peripherals using plain old-fashioned C,
- setup the internal (or external) oscillator, and
- initialize the input and output ports (learning the difference between weak and strong pull-ups and between push-pull and open-drain ports).

Students also learn about the importance of timers and about the structure and significance of interrupt routines.

The students continue working with simple, readily available code examples related to MCU and development kit peripherals (such as buttons, LEDs, and serial ports). Part of their exercise is to build a debugging tool by interconnecting the MCU with their programming PC using the serial (or USB) connection. They can simplify software debugging and accelerate development by exploiting the PC screen as a useful output device for their MCU-based code.

The course instructor focuses on the MCU's 12-bit analog-to-digital conversion (ADC) and 10-bit digital-to-analog conversion (DAC) capabilities, presenting relevant software examples in conjunction with sensor designs. The existence of fully operational ADCs further justifies the specific choice of low-cost MCUs and provides a fertile software and hardware infrastructure for imaginative (student or professional) projects. Additional student exercises include interfacing low-cost, resistive, and capacitive sensors to the MCU's ADC and reading the recorded values on the connected PC.

By the end of the two lab sessions, students can quickly access the MCU manual and craft efficient code. More importantly, they know how to incorporate low-cost, programmable MCUs as part of complex projects.

Digital Communication Lab

In a subsequent set of three-hour lab sessions, each pair of students experiment with a programmable, embedded communication link, comprising two transceivers—one portable and one tabletop version. Figure 2 shows the portable transceiver, called *iCube* (which isn't related to the I-CubeX line of products from Infusion Systems). It consists of a Chipcon/TI CC2500EMK transceiver module, which the lab instructor carefully interfaces to a Silabs toolstick C8051F321 daughtercard. The daughtercard provides an MCU development kit environment that exploits most of the functionality of the F320/1 family in a small form factor, making it appropriate for portable applications. The tabletop transceiver consists of the same radio module, interfaced to the tabletop MCU development kit C8051F320DK, which the students heavily used during the preceding embedded programming lab.

There are several commercial transceivers with MCU logic and a radio module embedded in the same monolithic chip, but we didn't select such designs for three main reasons. First, the students should develop in-depth MCU programming skills, useful for a variety of electrical engineering applications not necessarily restricted to the wireless domain. Second, from a teaching perspective, it becomes instructive to visualize separately the control unit of each iCube radio from its RF front end. Finally, the overall course spirit is about the synthesis of diverse skills and technologies, so it becomes natural to directly interface MCUs to digitally controlled radios.

The students are given a fully functional software example that implements a point-to-point digital communication link with the iCube nodes. The instructor presents the software example, pointing out the radio module's inner details. More importantly, the students learn how to efficiently browse the radio's manual and identify the appropriate fine-tuning programming steps for desired functionality. Carefully reading an equipment manual (or a specification document) is an important skill for working engineers, yet it's not included in purely theoretical EE system courses.

The radios, although embedded, provide a plethora of control registers that adjust important point-to-point communication parameters. Students can control the following parameters:

- the carrier frequency and respective frequency channel (parameterized for frequency-hopping applications);
- the transmission power;
- the type of frequency-shift keying (FSK) modulation and respective frequency deviation (continuous-phase FSK, such as minimum-shift keying, is also supported);
- the receiver filter bandwidth;
- the number of preamble bits used for bit-level synchronization and number of bytes used for byte-level synchronization;
- the receive signal-strength-indication activation and automatic-gain control;
- test signals; and
- the cyclic redundancy check.

Figure 2. The portable transceiver iCube v0.1, developed for the course (on the right). It consists of a low-cost radio module interfaced to an MCU unit small-form-factor development kit. On the left are the AA-battery holder, the MCU toolstick USB programmer, and a radio module interfaced on an MCU development kit.

This list reveals the level of detail the students delve into when setting up their personal wireless links.

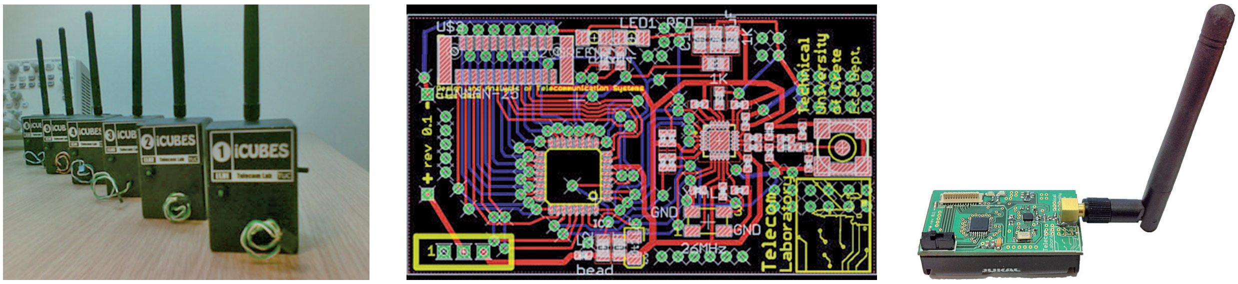
The example-based approach followed in this course is critical; the students greatly appreciated having a fully operational code example, which they can readily modify and experiment with, to develop an operational point-to-point communication link on each lab bench.

As a practice exercise, each pair of students implement a ping-pong exercise in which a set of two transceivers exchange packets consecutively and repeatedly and appropriately blink their LEDs in each successful packet transmission/reception. Additionally, the students implement a Morse code communication scheme by carefully using each tabletop transceiver's push buttons. They experiment with variable transmission power levels and record the corresponding received signal-strength levels, assessing and quantifying the intricacies of wireless propagation (such as fading). Finally, all students work together to tune their radios in different frequency channels, so that no interference degrades the simultaneous (in time) operation of all links.

Printed Circuit Board Design Lab

Over two weeks, the students attend yet another set of three-hour lab sessions. For this lab, the students design a two-layer printed circuit board (PCB), using Cadsoft's Eagle computer-aided design software tool (which has a free license for educational purposes, although access is limited to two-layer boards of relatively small dimensions).

The students are offered a step-by-step circuit design example tutorial, in conjunction with many Internet tutorials available online. Then, they must implement their own design and adhere to basic PCB design rules that limit electromagnetic compatibility (EMC) or minimize electromagnetic



(a) (b) (c)

Figure 3. The packaging and printed circuit board (PCB) design developed in class: (a) external packaging for portable iCubes, (b) the PCB design of iCube v0.2, and (c) the assembled iCube v0.2.

interference (EMI). Some design rules are as follows:[10]

- "Compact RF paths are better, but observe good RF isolation between pads and/or traces."
- "Make the number of return paths equal for both digital and RF."
- "Separate high-speed signals, such as clock signals, from low-speed signals. Separate digital from analog signals."
- "Keep connections to supply layers short. Avoid multipoint power supply connections. Use star-topology for signal distribution."
- "Keep ground planes as continuous as possible. Avoid vias (that is, vertical interconnect accesses) between bypass capacitors."

This list of rules is by no means complete, because EMC/EMI issues are generally complex, and success is heavily based on trial and error and experience. However, the students acquire the intuition behind the rules, with the help of the theoretical background taught throughout the course.

An example of the transceivers and their PCB design, produced through this lab, is depicted in Figure 3a and 3b, respectively. It is the design of *iCube v0.2*, consisting of C8051F320 MCU from Silabs and a CC2500 radio chip from Chipcon/Texas Instruments on a single board, with footprints for all required paraphernalia (switches, connectors, bypass capacitors, matching to antenna front-end circuit, LEDs and crystal oscillator). Figure 3c depicts the *iCube v0.2* which was assembled later; its size (excluding the antenna) is dominated by the battery holder which holds a pair of AA batteries.

Building the Wireless Sensor Network for a Digital Garden

Using the acquired course knowledge, students design and build a low-cost, multihop WSN that enables a *digital garden*.

The Sensor Electronics

For soil moisture measurement, students initially experimented with low-cost, capacitive humidity sensors from Honeywell (type HCH-1000-002). A simple circuit (see Figure 4a) was designed to interface the Honeywell sensors to the portable iCubes' MCU; a monolithic 555 timer circuit was used to convert changes in frequency to voltage pulses that triggered MCU interrupts. Despite its sensitive measurements, the overall sensor circuit prototype required a stable voltage supply, not available with the battery-operated iCubes. Moreover, each sensor consumed more than 1 mA of current, with an overall cost on the order of US$4 each.

As an alternative, the students developed resistive soil wetness sensors using simple material (plastic straws, plaster, and metal rack hangers), which directly measured soil wetness (see Figure 4b). You can find similar sensor designs on amateur gardening websites (such as www.cheapvegetablegardener.com/2009/03/how-to-make-cheap-soil-moisture-sensor.html). The students interfaced each sensor to the MCU's ADC through a voltage divider. Soil wetness is observed through the received signal's voltage drop. Resistive sensors significantly reduced current consumption compared to humidity sensors, extending the battery lifetime of each sensor-equipped iCube, at the cost of limited resolution (roughly three distinct wetness levels). Future work will use other, more complex soil-wetness sensing technologies (such as capacitive soil-wetness sensors[11]).

The environmental temperature was readily available from the corresponding thermistor embedded in each iCube's MCU. No other external temperature sensors were used. Finally, the students implemented a speaker circuit that could play music when wetness (or humidity) per sensor was satisfactory—in other words, plants could "sing" when watered (this was built for demonstration purposes in the lab—it wasn't deployed outdoors).

Wireless Relaying

The class addressed the problem of packet collision and the fundamental tradeoff between connectivity and

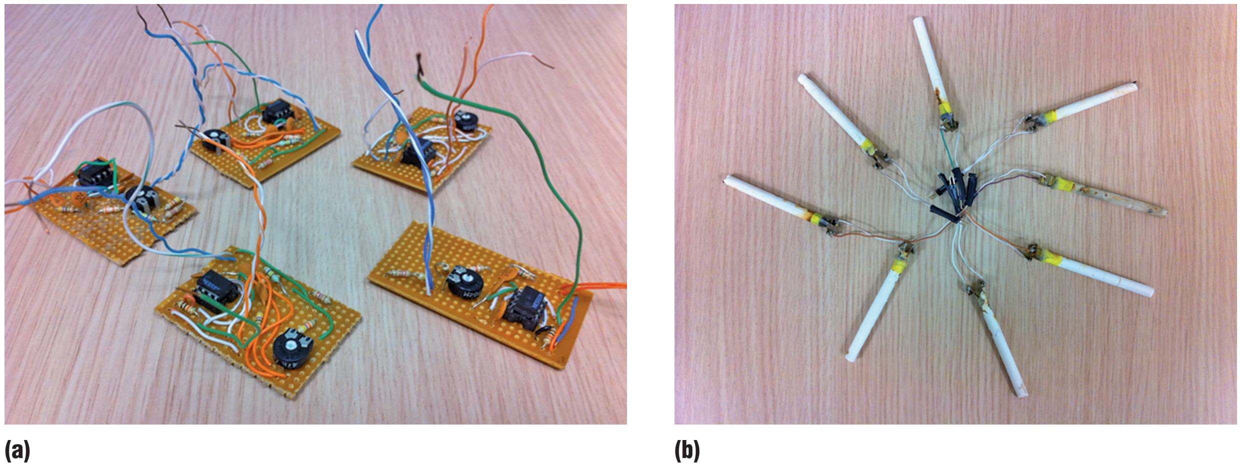


Figure 4. Humidity and wetness sensors tested in this project: (a) capacitive humidity sensor prototypes and (b) resistive wetness sensors.

energy consumption in multihop WSNs using the iCubes, which support multiple frequency channels around 2.4 GHz and can switch between channels in a relatively short time (on the order of microseconds). Furthermore, the set of plants to be networked and the corresponding wireless terminals were (naturally) immobile. To exploit fixed network topology (due to immobility) and address the collision problem, nodes were split in chain groups, with each chain using a specific radio channel. The node closest to the gateway chain used an additional channel, common across all chain groups.

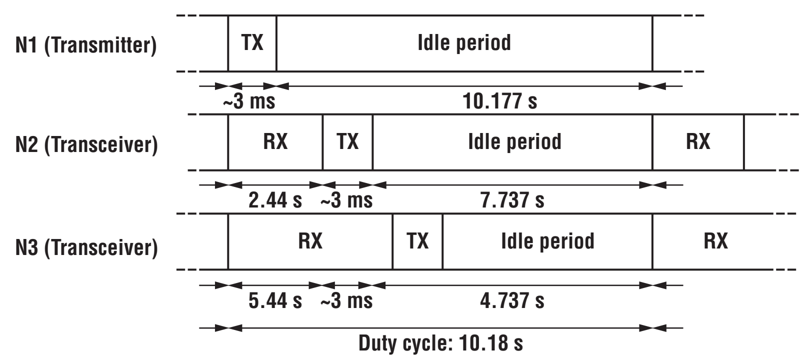


Figure 5. The duration of each state depends on each node's location in the chain. The students conceived and implemented the simple communication protocol during laboratory exercises.

Figure 5 depicts the operation in one chain with three nodes. Each chain node was initiated in "idle" (sleep) mode. When an internal timer expired, the node entered the reception (RX) mode. If the node received a valid packet (that is, received data without errors and a valid receiver identity address), the node would augment the packet with its own sensing information, switch to the transmission (TX) mode, and forward the packet to the next node in the chain before re-entering the idle mode. The internal timer would be set to an initial value dependent on the node's location in the chain—that is, the idle-mode duration depended on each node's location in the chain. In that way, each node could "wake up" on time and get ready to receive information from previous nodes in the chain.

Even though such *synchronized* operation reduced energy consumption and eliminated packet collision, no explicit time synchronization protocol among the nodes was required. Instead, synchronized operation was facilitated by carefully selecting the timer values and the RX state mode (as well as the idle mode) duration at each node, taking into account any instability of the MCU clock. The last chain node conveyed (to the gateway) sensing information from all nodes in the chain. A time-division multiple-access scheme, coordinated by the gateway, forwarded information from all chains to the gateway. The gateway was always in receive mode and powered either by a laptop computer or a power outlet (no batteries required).

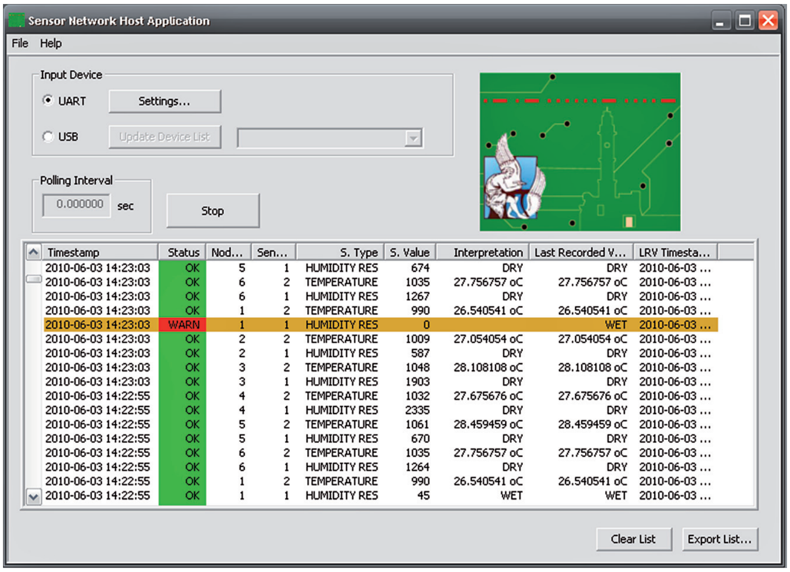


Figure 6. GUI software developed in class. The software runs on a laptop, connected through serial port (or USB) to the iCube gateway.

As a last exercise, students quantified the lifetime duration of each chain by taking into account each iCube's consumed milliampere-hours (mAh) at each state, the protocol's duty cycle, and the mAh capacity of the batteries used. Both the MCU and radio module used typically required a few mA.

The students designed and implemented the aforementioned simple communication/networking protocol. A number of additional engineering problems could be further explored at this point, such as optimized energy efficiency or energy consumption fairness (that is, all nodes should consume the same amount of minimum energy). Another direction could be a simple form of *cooperative diversity* reception,[12,13] since, depending on transmission power and internode ranges, more than one node in the network could listen to a single transmission.

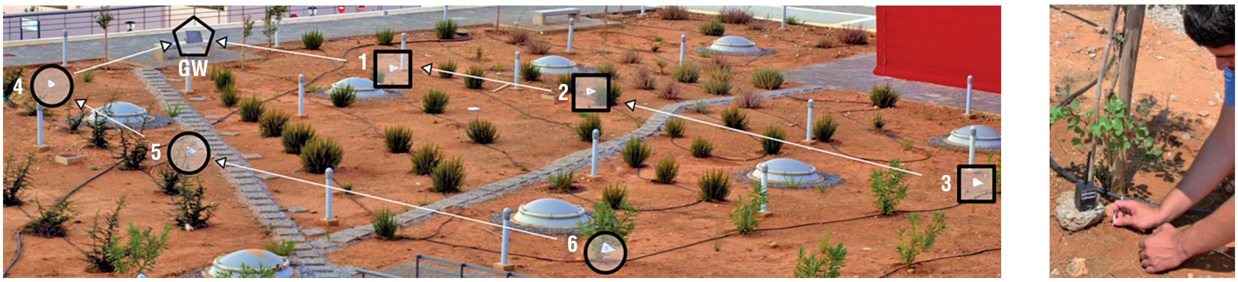
Packaging and GUI

Each portable iCube was packaged such that it could operate outdoors (see Figure 3a). Furthermore, new PCBs were designed to accommodate all modules in a single board and extend functionality (see Figures 3b and 3c). Finally, a desktop iCube was used as the gateway; sensed information was transported to a laptop computer through iCubes' serial port (or USB). A GUI software program was developed to display the data collected from the WSN in a user-friendly, aesthetically appealing form (see Figure 6).

Discussion

Students deployed a small-scale network of two chains and seven nodes in an outdoor environment for demonstration purposes (see Figure 7). Specifically, one chain consisted of nodes 1–3, and the second chain consisted of nodes 4–6, with gateway GW located closer to nodes 1 and 4.

As a demonstration run of the developed WSN, environmental temperature and soil wetness measurements were collected for approximately 15 hours (900 minutes). Figure 8 depicts the temperature measurements from the most distant sensors (3 and 6). The two plots closely coincide, even though the two plants were located several meters apart. This figure provides a concrete indication that the student-designed



(a) (b)

Figure 7. The wireless network deployed outdoors for demonstration purposes. It has (a) two chains (chain 1 in rectangles and chain 2 in circles) and seven nodes and (b) a soil wetness sensor. No special iCube mounting was performed.

WSN actually worked. Furthermore, these measurements motivate research on *distributed* source coding, by exploiting potential correlation among distributed sensor sources.[14] Thus, the students are offered additional food for thought.

Teaching Perspectives

The course instructor capitalized on his EE background and spent approximately four weeks on class preparation (lecture slides, homework assignments, software examples, and transceiver prototypes) before the beginning of the semester. Teaching required two 1.5-hour theory lectures and one three-hour lab per week. Grading was based on weekly lab exercises as well as a midterm and final to assess the theoretical part of the course and the students' understanding of manual specifications. In the final term project, teamwork was the ultimate goal—each member was allocated a specific task, and the instructor evaluated the quality of the task execution.

Students were encouraged to access the lab outside of class. In fact, most students coordinated through the course Web portal and managed to coexist in the lab during evenings, helping each other. During the first few weeks, they spent approximately three extra hours (beyond the three-hour lab class) and perhaps a little more than three hours when implementing the final project. Of course, the requirements of the class can be adjusted to be less demanding while students still get a flavor from all of its parts.

Educational Perspectives

Implementing a functioning WSN from its grassroots indirectly indicates the level of course success. However, we also distributed assessment questionnaires so that students could anonymously evaluate course material and provide constructive feedback.

Interestingly, the students unanimously endorsed the broad and diverse

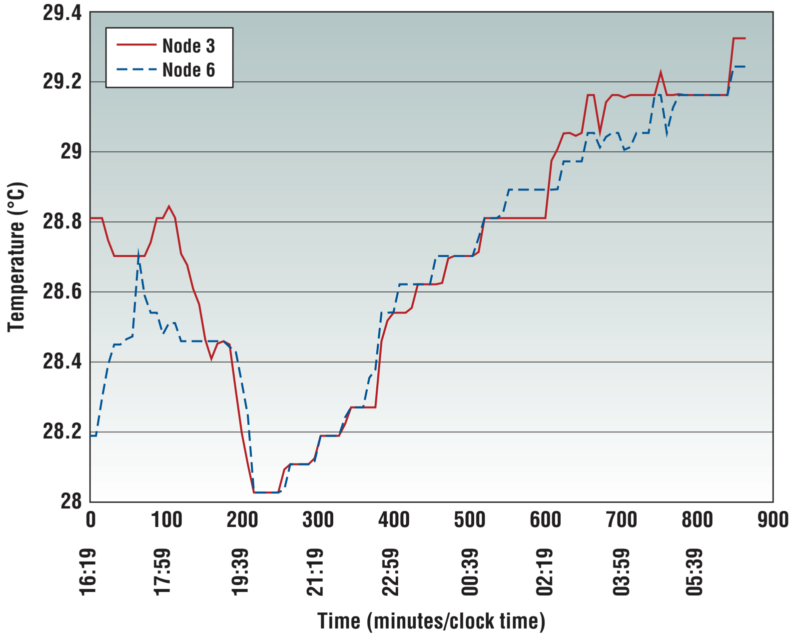


Figure 8. An example of two sensors' temperature measurements (transported through the network) as a function of time.

curriculum. Additionally, apart from breadth, most students (90 percent) approved the depth of the course material and appreciated the lab skills they developed throughout the semester. This was an important observation, given that some elements of the course, such as embedded programming (and a persistent reference to MCU and radio's manuals), required extensive laboratory experimentation and perhaps many testing iterations.

More importantly, 90 percent of the students highly appreciated the synthesis perspective of the course—that is, the ability to build a working system by intelligently resolving diverse, interdisciplinary subproblems. Furthermore, the students explicitly stated that such a synthesis feature was largely missing from the curriculum and significantly enhanced their problem-solving skills. (Note that only 20 percent of the students failed or dropped the class.)

Developing the specific WSN also revealed the substantial distance between classroom exercises and real-world system development. That became obvious during the sensor's development (for example, when the voltage drop at battery-operated iCubes was faster than anticipated) and during multihop relaying (when outdoor propagation modified operational range, compared to initial theoretical modeling and predictions).

One of the most important contributions of the course was the confidence that students gradually gained during the semester, through hard work on laboratory experimentation and tangible demonstrations. Such confidence is important for pedagogical reasons and might spark entrepreneurial ambitions among innovative students. The latter remains to be seen.

Overall Cost

Table 1 presents the minimum hardware cost per laboratory bench (supporting ideally a pair of students) and includes the iCubes, the MCU development kit, the programmer

TABLE 1
The minimum hardware cost per lab bench.

Type	Description	Quantity	Total cost (US$)
C8051F320DK	MCU development kit	1	100
634-TOOLSTICKBA	Toolstick programmer	1	14
iCube v0.1	A portable transceiver	2	116
iCube v0.2	A portable transceiver with custom printed circuit board design	2	60
Digital 60MHz Oscilloscope	Oscilloscope	1	900
Power supply	Variable voltage supply	1	300

for the toolstick MCU version, as well as a variable voltage supply and digital oscilloscope. The last two devices are useful in electronics and sensor development, as well as embedded programming debugging. However, they aren't necessary. Notice that the iCube second version (v0.2) is based on a custom PCB and significantly reduces each wireless terminal cost. The reported total cost per lab bench doesn't exceed the cost of a desktop PC. Furthermore, each lab bench should include a laptop PC, as well as a variety of low-cost sensors and a typical, low-cost educational kit for simple circuitry development (such as breadboards and resistors).

Thus, the hardware cost per lab bench is minimal. Software development costs (for example, regarding the embedded point-to-point link) depend on the user's expertise and can be limited. The only priceless resource is the student's imagination.

This educational experiment was successful; this work demonstrates that diverse curriculum, spanning telecommunications, electronics, and informatics can be taught in depth, facilitating novel applications built by students as course semester projects. Hopefully, the teaching methodology, curriculum, and low-cost materials presented here offer a useful teaching example to better prepare electrical and electronic engineering graduates for the real world.

REFERENCES

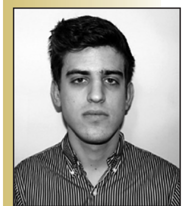
1. T. Wark et al., "Transforming Agricultural through Pervasive Wireless Sensor Networks," *IEEE Pervasive Computing*, vol. 6, no. 2, 2007, pp. 50–57.
2. R.D. Poor, "Embedded Networks: Pervasive, Low-Power, Wireless Connectivity," doctoral dissertation, Media Laboratory, Massachusetts Institute of Technology, 2008.
3. W. Zhang, G. Kantor, and S. Singh, "Integrated Wireless Sensor/Actuator Networks in an Agricultural Application," *Proc. ACM Conf. Embedded Networked Sensor Systems* (SenSys 04), ACM, 2004, p. 317.
4. R. Musăloiu-E. et al., "Life Under Your Feet: A Wireless Soil Ecology Sensor Network," *Proc. 3rd Workshop on Embedded Networked Sensors* (EmNets 06), ACM, 2006, pp. 51–55.
5. J.D. Lea-Cox et al., "A Low-Cost Multihop Wireless Sensor Network, Enabling Real-Time Management of Environmental Data for the Greenhouse and Nursery Industry," *Proc. Int'l Symp. High Technology for Greenhouse System Management* (Greensys 07), ISHS, 2007; www.actahort.org/books/801/801_58.htm.
6. J. McCulloch et al., "Wireless Sensor Network Deployment for Water Use Efficiency in Irrigation," *Proc. Workshop on Real-World Wireless Sensor Networks* (REAL WSN 08), ACM, 2008, pp. 46–50.

the AUTHORS

Aggelos Bletsas is an assistant professor in the Electronic and Computer Engineering Department at the Technical University of Crete. His research interests span the broad area of scalable wireless communication and networking, with emphasis on relay techniques, signal processing for communication, radio hardware/software implementations for wireless transceivers and low-cost sensor networks, RFID, time/frequency metrology, and bibliometrics. Bletsas received his PhD from the Media Laboratory at the Massachusetts Institute of Technology. Contact him at aggelos@telecom.tuc.gr.

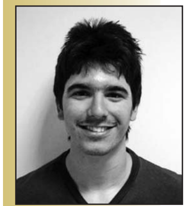
Aikaterini Vlachaki recently graduated from the Electronics and Computer Engineering Department, Technical University of Crete, and is now working on her MSc at University of Alberta, Canada. Her research interests include simulation and evaluation of WLAN deployments, traffic modeling, and optical networks. Vlachaki has received three undergraduate fellowship awards. Contact her at vlachaki@ualberta.ca.

Eleftherios Kampianakis is an MSc student at the Technical University of Crete in the Telecommunications Division. His research interests are wireless sensor networks for networking applications, low-cost hardware development transceivers, embedded programming, firmware and application development for WSN nodes, and the design and implementation of solar modules and panels. Kampianakis received his diploma degree in electronic and computer engineering from the Technical University of Crete. Contact him at ekabianakis@isc.tuc.gr.

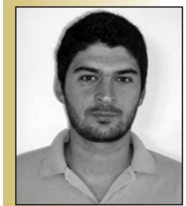


George Sklivanitis is a research assistant in the Communications and Signals Laboratory at the State University of New York at Buffalo, working toward his PhD in electrical engineering. His research interests are in the area of communications, signal processing, and networking with an emphasis on real-time optimization, online adaptive signal processing, underwater acoustic sensor networks and software-defined radio implementations. Sklivanitis received his diploma in electronic and computer engineering from the Technical University of Crete. He's a student member of the IEEE Communications and Signal Processing Societies. Contact him at gsklivan@buffalo.edu.

John Kimionis is an MSc candidate and research assistant at the Electronic and Computer Engineering Department at the Technical University of Crete. His research interests include backscatter radio and RFID, wireless sensor networks, software-defined radio for backscatter and sensor networks, microwave/RF engineering, and telecom hardware/embedded systems development. Kimionis received his diploma degree in electronic and computer engineering from the Technical University of Crete. Contact him at ikimionis@isc.tuc.gr.



Konstadinos Tountas is a fourth-year undergraduate student in the Electronic and Computer Engineering Department at the Technical University of Crete. His research interests focus on wireless communications and networking, radio hardware/software implementations for wireless transceivers and low-cost backscatter sensor networks and RFID. Contact him at ktountas@isc.tuc.gr.



Megasthenis Asteris is a PhD student at the Electrical Engineering Department of the University of Southern California, Los Angeles. His research interests focus on distributed storage and its applications. Asteris received his diploma in electronics and computer engineering (with excellence) from the Technical University of Crete. Contact him at asteris@usc.edu.

Panagiotis Markopoulos is a PhD student at the Department of Electrical Engineering at the State University of New York at Buffalo. His research interests span the areas of dimensionality reduction techniques, smart antennas and adaptive beamforming, space-time modulation and coding for MIMO wireless communications, low-complexity data detection, and radio hardware and software implementations for wireless transceivers and low-cost WSNs. Markopoulos received his diploma in electronic and computer engineering from the Technical University of Crete. Contact him at pmarkopo@buffalo.edu.

7. G. Vellidis et al., "A Real-Time Wireless Smart Sensor Array for Scheduling Irrigation," *Computers and Electronics in Agriculture*, vol. 61, no. 1, 2008, pp. 44–50.
8. A. Baggio, "Wireless Sensor Networks in Precision Agriculture," *Proc. Workshop on Real-World Wireless Sensor Networks* (REAL WSN 05), Swedish Inst. Computer Science (SICS),2005; www.sics.se/realwsn05/papers/baggio05wireless.pdf.
9. J.D. Kraus, *Antennas*,3rd ed., McGraw-Hill, 2001.
10. "Designer's Guide to LPRF," Texas Instruments, 2010; www.ti.com/lit/sg/slya020a/slya020a.pdf.
11. F. Kizito et al., "Frequency, Electrical Conductivity and Temperature Analysis of a Low-Cost Capacitance Soil Moisture Sensor," *J. Hydrology*, vol. 352, nos. 3–4, 2008, pp. 367–378.
12. A. Bletsas, "Intelligent Antenna Sharing in Cooperative Diversity Wireless Networks," doctoral dissertation, Media Laboratory, Massachusetts Inst. of Technology, 2005.
13. A. Bletsas and A. Lippman, "Implementing Cooperative Diversity Antenna Arrays with Commodity Hardware," *IEEE Comm.*, vol. 44, no. 12, 2006, pp. 33–40.
14. M. Gastpar, M. Vetterli, and P. Dragotti, "Sensing Reality and Communicating Bits: A Dangerous Liaison," *IEEE Signal Processing*, vol. 23, no. 4, 2006, pp. 70–83.