A Laboratory Setup and Teaching Methodology for Wireless and Mobile Embedded Systems

Jean-Samuel Chenard, *Student Member, IEEE*, Zeljko Zilic, *Senior Member, IEEE*, and Milos Prokic

***Abstract*—Increasingly, electrical and computer engineers are making their careers in designing wireless embedded systems. This paper presents a teaching methodology and the associated laboratory setup designed to meet the needs in teaching wireless embedded systems. The courses allow the students not only to apply their previous knowledge of digital system design, computer architecture, electronic circuits, wireless networking, and software engineering, but experience actual systems engineering by designing and implementing a large-scale team project within a semester. A flexible hardware platform was developed and was accompanied by teaching methodologies that allow quick completion of ambitious course projects in this area.**

***Index Terms*—Digital circuits, energy conservation, engineering education, microcontrollers, networks, radio communication, software engineering, software tools.**

I. Introduction

THE proliferation of wireless and mobile systems in sectors ranging from home and industry automation to health care and entertainment is only in its beginnings. In the next few years, a plethora of activity will be given to cornering these new markets, and an army of well-trained engineers will therefore be a highly needed resource. The curriculum development necessary to accommodate this proliferation in wireless and mobile computing is barely visible in computer engineering programs. While the December 2004 IEEE/Association for Computing Machinery Computer Engineering Task Force includes wireless and mobile computing, and some universities, such as the University of California, Los Angeles, and Virginia Polytechnic Institute and State University, Blacksburg [1], have created various courses, there is a lack of relevant lab courses offered from the perspective of extending embedded system design lab practice. This paper describes a laboratory setup designed inhouse to fulfill this need for a trained workforce. The laboratory is based on the microprocessor systems kit *McGumps* [2], [3] of McGill University, Montréal, QC, Canada. The kit, designed to be expandable and relatively long-lived, has served well for three years and has accommodated various add-ons.

The range of skills needed in industry include teamwork, project management, and the exposure to realistically-sized projects, in addition to domain-specific practical knowledge. The proposed initiative creates favorable conditions to developing such skills in wireless embedded systems, going beyond any teaching aids currently on the market. Under this methodology, teams of four students were able to cooperate and effectively successfully complete final projects on the motherboard that included liquid crystal display (LCD)-based graphical user interface, touchscreen and wireless links, within the span of a single semester.

The teaching goals are achieved here by a combination of teaching methodology, laboratory aids, and curriculum additions. The cognitive approaches used in the teaching methodology are presented in Section II. Some of the considerations used during the development of the material are discussed in Section III to help other universities should they decide to build their own teaching kits. The laboratory kit details are covered in Section V, which presents the key features of the teaching kit. Section IV details the wireless interfaces design decisions and targeted skills development. The grading scheme and peer evaluation methods used are covered in Section VI. The improvements offered by this new platform compared to the earlier laboratory setup conclude the paper.

II. Underlying Teaching Approaches

These courses are aimed at more senior undergraduates, as a way for them to synthesize their knowledge acquired from multiple prerequisite courses, and to facilitate their ability to realize ambitious real-world projects in a short time. These students will be able to create better impressions in subsequent job interviews. The following pedagogic approaches were used to maximize the teaching potential of the lab and to leave students with a positive outlook towards working with embedded systems.

A. A Scaffolding Approach

The laboratory is structured to offer formal lectures (with quizzes) and directed exercises during approximately the first third of the semester. Exercises are made progressively more complex as students begin to master the microprocessor core instruction set, board hardware, and development tools. This instructional scaffolding approach to education offers the benefit that students feel guided in their initial contact with the laboratory material. For most of them, this course provides their first experience of working directly with a complex circuit board and this approach reduces their frustration and anxiety. The first exercises are done individually and marked so as not to weight too heavily in the final grade. These exercises mostly focus on the CPU instruction set architecture and highlight the main elements of the microcontroller. Basic embedded programming

Manuscript received July 2, 2007; revised September 10, 2007. Published August 6, 2008 (projected). This work was supported in part by the FQRNT and NSERC agencies, the McGill Engineering Equipment Fund, McGill Teaching and Learning Improvement Fund, and the Wighton Fellowship.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada (e-mail: jsamch@macs.ece.mcgill.ca).

Digital Object Identifier 10.1109/TE.2008.919690

TABLE I
INTRODUCTORY COURSE MILESTONES TIMETABLE

Week	Milestones
1,2	Assembly language, C and development tools
3	Basic interrupts (timers), input/output and CPLD
4	UART interfacing, simple user interfaces
5	Advanced interrupt processing (multiple sources)
6	SPI bus, A/D converters
7,8	Project peripherals (LCD, RF)
9-12	Team project

skills such as assembly and simple C programs are used to introduce the hardware platform through further simple exercises. At this stage, teaching assistants perform detailed demonstrations and step-by-step tutorials to ensure everyone has a full understanding of the tools needed during the semester. New hardware is introduced in a similar manner. Table I summarizes a typical introductory course schedule.

B. Collaborative Learning

Once students have individually mastered the basic skill set, they are asked to form teams of two, which will be maintained throughout the semester. Students individually select their partner, but if this has not been done by a certain deadline, the pair is instructor-formed. Groups of three, generally requiring more complex interpersonal dynamics, are allowed only when there is an odd number of registered students. This collaborative learning process [4] gives students the possibility of sharing their thoughts as they approach the solutions to the assigned exercises and also as they practice proper "code sharing" techniques, such as commenting and breaking code in modules. Teaching assistants are instructed to quiz either of the team members on the solution, without regard which of the students "wrote the code." This technique requires an understanding of the solution from both in the pair.

In the final part of the semester, two pairs are merged to form a bigger group to tackle the final project. This larger group offers students the opportunity to discuss and define their individual objectives as aligned with the group goals.

C. Student-Centered Learning

As the semester advances, assignment requirements are made progressively less detailed and students are expected to "fill-in" the gaps in the specifications by proposing their own implementation solutions. As each main exercise of a total of four requires a written report, students must justify their decisions. This approach has its roots in *student-centered learning* [5], whereby the hardware kit and assignments are designed to highlight the benefit of well-known design methods (fast interrupt processing or mutual exclusion, for example) by making sure that a solution that is built without taking these methods into account would be inefficient (and very likely to show problems during the demonstration). This feature has the effect of reinforcing the use of proper design methods, as previously covered formally in class.

III. DEFINING THE TEACHING KIT CORE ARCHITECTURE

Before developing a new laboratory kit to replace the existing laboratory equipment, several pilot projects were undertaken to study and possibly adopt existing kits. Existing kits on the market can be classified in three broad categories described later. Two boards belonging to the each of the categories were studied for at least two semesters using volunteers in the course.

1) Soft Core Microprocessors: The first type of boards included systems that employ programmable logic to implement "soft" microprocessors, such as Altera NIOS and Xilinx Microblaze. The experience in the classroom showed that while these kinds of kits are educational in understanding of the inner workings of the microprocessor, the time-consuming compilation and download into the programmable logic, as well as the relative lack of software robustness, were the main impediment to realizing the teaching goals. Furthermore, the complex implementation of an RTL-level microprocessor, combined with the student's relatively immature understanding of computer architectures, makes the task overwhelming in the context of a single-term undergraduate class focusing on system-level development. Recent developments in this area offer hope that such systems could eventually be used in an introductory class [6].

2) High-End Microprocessors: The second type of boards includes the current high-end microprocessors that excel in performance and efficiency in executing the application. Examples studied include several boards employing the ARM (Intel PXA) and two boards involving the PowerPC. While these boards were excellent for advanced studies and the development of more elaborate projects, they were too complex for a first course on microprocessors. Further, the software and the development setup of such boards requires a learning curve that makes them unsuitable for single-semester courses, especially if the course is to focus on low-level hardware operations (interrupts, use of hardware peripherals, low-memory and low-power).

3) Simple, Low-End Microcontrollers: In this category, some examples studied included earlier Motorola and Atmel processor boards. In some instances, these boards are based on one of the very first generations of microprocessors dating back from the early 1970s. In these cases, the supporting software, a must in this type of laboratories, is also several generations out of date. In some cases, the company that produced this software has ceased to exist. While students can still benefit from the course because the equipment is simple, improvements are needed if the course is to maintain its relevance to modern designs. Finally, the lack of on-chip FLASH memory on those older systems makes them increasingly unproductive to use.

Newer microcontrollers include a modern 16-b or 32-b microprocessor core with a register organization that facilitates high-level programming. However, in all cases, commercial board offerings were found to be very specific in addressing the needs of people "evaluating" the microcontroller, but limited in their useability as an expandable teaching platform.

A. Teaching Kit Microcontroller Selection

The course structure aims for students to be programming the microcontrollers within approximately two weeks after the introductory class. The course can then quickly cover sufficient background to allow students to plan their final projects and

work on the various low-level subprojects. Due to those constraints, the selection focused on two simple, modern architectures, namely the MSP430 [7] from Texas Instruments and ARM7-TDMI-S (using the NXP LPC2000 [8]) from ARM.

The final choice of microcontroller for the basic course clearly had to be the MSP430. This microcontroller's simple, well-integrated and documented microarchitecture, broad device availability and modern reduced instruction set computer (RISC) and orthogonal register file offer a good example of a typical microprocessor core that is realistic in low-end embedded equipment, yet educational, as students can for most part grasp how its architectural blocks are implemented and interfaced. With no instruction and data caches, the MSP430 family provides predictable timing, simplifying the performance estimates of assembly code by students.

A more advanced kit (McZub) was developed using the NXP LPC2000 that so far has been used for advanced wireless systems training, where students have already shown strong interest in embedded systems. The main benefits of this microcontroller lie in its higher execution rate (60 MHz versus 8 MHz for the MSP430), 32-b architecture and in the use of instruction and data caches. Thus, the embedded software can be designed with more layers, which provides more flexibility, while maintaining proper real-time operation.

B. Maintaining Flexibility

The interfaces and peripherals that form part of the teaching kit have a great impact on the course. Such interfaces require the inclusion of the increasingly complex hardware. Classical "wirewrapping" of discrete components by connecting them via wires wrapped around IC pins can offer the option to add-in some peripherals to a board. While this approach was adequate some decades ago, the amount of wiring, low reliability of the process and higher bus speeds will cause for major downtimes in the lab. Thus, wirewrapping can be used only for expansion of simple, low-speed interfaces.

A subsystem for prototyping larger hardware components by programmable logic devices was designed in a way featured in virtually no similar teaching kit. The setup is flexible and adjustable to fast-changing sets of new communication protocols and computer interfacing standards. For example, modern PCs contain dozens of new standard interfaces [(e.g., universal serial bus (USB), Ethernet, WiFi, Firewire] that were not present a decade ago. Graduating students can be prepared to tackle the design of new and emerging computer standards because the equipment for facilitating such learning has the flexibility to permit this. The programmable logic chip offers a reliable way to virtually "wirewrap" the expansion boards. The only difference is in the handling of bidirectional signals, which is solved by using the tristate drivers.

C. Proposed Teaching Kit Architecture

The beginning courses on embedded systems programming require a simpler low-end microcontroller, such the MSP430, augmented with expansion peripherals and prototyping area, for example through complex programmable logic devices (CPLDs) on the board. Further expansion is facilitated by using "daughtercards." The expansion connector pinout was designed

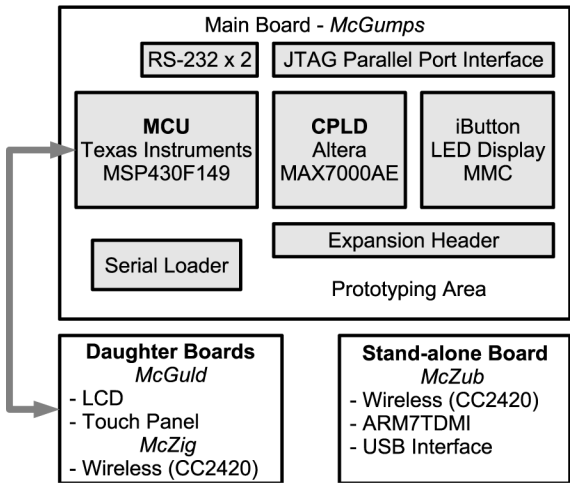

Fig. 1. Teaching platform overview.

to provide power to the "daughtercard," and connections to specific pins on the MSP430 processor for hardware-assisted interfaces and analog links. This design accommodates a wide variety of possible expansion cards. Furthermore, both the processor and the CPLD permanently store their configuration in nonvolatile FLASH memory, so students can simply plug-in the kit and everything runs without their having to attach any computer or device programmer. The board enables the use of the simple methods to program the FLASH memory on the processor core and the CPLD using the same cable, by integrating both device programmers on-board.

The resulting teaching platform and peripheral boards are shown in Fig. 1. In some semesters, multiple McGumps boards were linked together through the expansion connector to allow small multiprocessor systems. Rice University, Houston, TX, also selected the same processor architecture for their course [9] on embedded sytems.

The more advanced McZub ARM-based board has an integrated RF transceiver and USB interface, and can easily be used to act as a proxy between a PC and the wireless interface. The board's larger memory also allows it to manage more wireless nodes than the MSP430.

IV. Laboratory and Wireless Skills Development

The simplifications achieved through the use of the McGumps main board freed a few weeks in the course schedule, which was used to introduce new technical content. Wireless interfaces were added to the teaching kit to offer interesting additions to the course material. The wireless extensions required careful engineering to create reliable networking software around which students can design their final project.

A. Wireless Networking

Since students typically show great interest in using the latest technology in their learning, adding low-cost wireless interfaces already had a strong appeal. Furthermore, recent advances in radio-frequency integrated circuits (RFIC) have pushed very low cost transceivers into the marketplace. For a few dollars, a radio link can be added to an embedded system, tremendously expanding its capabilities. Including such a link in a course on

embedded systems allows the creation of interesting projects, while keeping the overall course difficulty at a manageable level.

Over several semesters, both the Nordic nRF2401 and the Texas Instruments CC2420 transceivers were used. Both transceivers offer relatively simple means of transmitting digital data wirelessly. The Chipcon IC supports the Zigbee standard, which, judging by the current industrial involvement, will be widely used in the near future. In the laboratory, it is preferable not to use any prebuilt networking protocol stacks, but rather let the students implement their own, which allows them to learn concepts such as increasing robustness to interference, node management, and discovery. The added freedom, and the consequent potential for students to make mistakes, makes the experience all the more enriching. More complex communication systems such as Bluetooth or 802.11b were considered, but these could overwhelm students because of the complex configuration steps required to operate the transceiver. If, instead, students were to use a prebuilt software stack, they would learn little more than *socket programming*, which can be easily taught using standard personal computers.

The selection of RFIC transceivers allows students to send and receive radio packets after only a few days of work. Most of their problems relate to their lack of understanding of the serial peripheral interface used to communicate with the RFIC. Students also quickly experience the intricacies of designing wireless systems operating in the ISM band by having to deal with corrupted data, dropped packets, unwanted reception (from other teams), and other similar challenges of wireless system design.

The original IEEE 802.15.4 networking stack was also designed to be suitable for modification by students. This stack's well-documented programming interface and portable code allows it to run on both teaching platforms. Students have built complex systems out of this wireless networking software. Some examples include distributed temperature monitoring and control, a conference management system, and a distributed file server.

B. Software Engineering

Since wireless systems are quite complex even with the simplest transceivers, a good software engineering approach is essential. The use of the C programming language instead of assembly language is a must due to the complexity of the final project. Some exceptions may arise when students wish to manipulate specific processor registers or maximize performance of a particularly critical section of code. Students are taught in the class to mix C and assembly properly in the same project. Throughout the semester, students are shown how to prototype and test their code by writing C stub functions and running tests on their personal computers before its integration on the microcontroller.

The McGumps kit was designed with particular concerns for the design tools that students would use in the laboratory. The main integrated development environment chosen was Rowley Associates CrossStudio for MSP430 for its very intuitive graphical user interface and straightforward build process. The same integrated development environment is also available for the ARM-based processors as a possible upgrade path. One of the requirement for this choice of processor was that it had to be supported by the GNU GCC compiler in the event that students wished to build the software without commercial tools. The MSPGCC toolchain offers this option and is compatible with this teaching platform and in-circuit programmer.

C. User Interface Design

Every semester, students need to construct robust user interfaces. This usability requirement becomes important in mobile wireless systems. Towards this goal, several printed circuit boards were developed, such as the McGuld "daughtercard" that includes a touchscreen panel and low-power graphical display. Students often relate to those interfaces as they do in their everyday use of personal digital assistants or advanced cellular phones. Through these add-on modules to McGumps, they have a chance to obtain hands-on experience in designing the low-level drivers for such hardware, starting from the LCD module and controller specification data sheets.

Students were often observed to underestimate the complexity of building good user interfaces. As the developers, they know the inner working of their prototype too well and this often leads to cryptic interfaces. In several instances, peers (students outside the team) were used to review the "look-and-feel" of their classmates user interface. This has yielded good feedback and noticeable improvements in the way students approached their final project and in their system's interaction with the end-user.

D. Digital Systems Design

In a low-power wireless system, some events require precise timing measurements, such as beacon frames used to synchronize low-power modes among nodes. In such cases, the problem can be solved by using either a microcontroller peripheral or the kit's programmable logic device.

The benefit of simultaneously using hardware description languages (VHDL) and sequentially executed languages ("C" or assembly) is that this helps students understand the fundamental differences between those two description styles. While students taking the laboratory are familiar with both, it is well known [10] that students do initially experience difficulty making a distinction between the two. They have little experience of partitioning a particular design into its programmable logic and executed code subsets. A FLASH-based CPLD added to the McGumps was critical in enabling projects that require students to simultaneously code VHDL and C, such as the example mentioned previously. This dichotomy is emphasized very early in the course by simultaneously introducing the MSP430 assembly language and a VHDL-based peripheral module to solve a simple design problem.

In addition to forcing a review of the students' VHDL coding skills, the CPLD also highlights the benefits of hardware-assisted parallelism and some of the elements specific to reprogramming hardware in an embedded system. The CPLD also requires students to understand clearly the board-level schematics to define pin directions and to diagnose and fix signal contention.

The Altera MAX CPLD [11] was chosen because of its compatibility with the 3.3V I/O of the MSP430, and also for its

ability to support 5V interfaces (so as to re-use some older external ICs) and its hot–pluggable capability. The last of these features is important as it is possible to disconnect the power rail of the CPLD to determine quickly if the CPLD is the cause of a bus conflict. Some programmable devices cannot tolerate inputs on their I/O pins without being powered.

E. Electronic Circuits

One of the main objectives was to let student design simple electronic circuits on the teaching kit. Teachers at Harvey Mudd College, Claremont, CA, had success letting students hand-assemble their own programmable-logic and microcontroller boards [12]. While the authors feel that students still need to design and build some circuitry, having a working "core" platform right from the first day of class can save time and frustration.

Wirewrapping headers were kept in the teaching kit as this lets students "make mistakes" and learn the process of debugging circuits. Some simple devices such as real-time clocks, D/A converters or serial memory are still available in dual inline (DIP) packages which can be inserted in wire-wrapping sockets; such devices are used each semester for a portion of the design. Intermittent faults, shorts and open circuits are quite common with wire-wrapped circuits, and despite frustrating a few students, it nevertheless taught them a few key lessons of having a good debug methodology. Modern mass-produced electrical circuits are not immune to those faults, so students should encounter them more frequently in their learning environment. However, for complex expansion boards with multiple surface-mount devices and tiny components, fully assembled and tested daughterboards are provided.

V. Designing the Teaching Kit

A. Rationale for Inhouse Development

Traditionally, the microprocessor system laboratory is conducted on purchased systems that allow basic development of software and rudimentary computer hardware exercises. Building the equipment inhouse does introduce some complexity, such as the manufacturing and sourcing of components, but facilitates matching the equipment with the desired teaching goals. Complexity can be reduced by first using parts that can be sourced from only a few distributors and then outsourcing the production and assembly of the printed circuit boards to a specialized shop. A preproduction prototype was produced by a group of graduate students in the previous semester and some of the exercises were given to undergraduate students willing to go "off the beaten track" and try the future teaching kit. This approach allowed the design team to iron out bugs in the design and clean up the documentation.

The cost of producing the kits inhouse is similar to that of those bought ready made. The cost savings from inhouse development is offset by the higher manufacturing cost of producing a smaller quantity.

Inhouse design has several key advantages that help integrate the teaching kit in the existing curriculum. The most notable is the vendor-independency of this approach. The CPU, CPLD, toolchain, and external peripherals could be selected such that they would exactly match the software and hardware configuration of the lab computers. The inhouse design also offered the possibility of incorporating several layers of hardware and software protection, to avoid damage to the equipment from common mistakes made by students. The inhouse development of the laboratory equipment additionally improved the ability to reach teaching goals by making the underlying technology more understandable and approachable to students. Having all the design documents also makes the laboratory equipment much easier to maintain and upgrade. Students have access to the full printed circuit libraries and bills-of-materials used to produce the teaching kits. University technicians were involved in the feasibility analysis of part replacement and upgrades, ensuring that the kits can be used for many semesters and that the equipment is independent of vendor lock-in or end-of-life issues that could affect commercial teaching kits.

B. Printed Circuit Boards

The relatively low-speed of the processor and the large board size were selected to achieve teaching objectives, rather than manufacturing and production cost savings. The main philosophy was to *hide nothing* from students. The layout was designed such that probe points (vias or pads) were all accessible from the top, to ensure that students could debug their circuit without the need to turn the board over. Only the radio-frequency sections on the McZig and McZub boards did not have probe points designed into the boards, due to the particular layout rules of the RF design. The RF interfaces on the McZig and McZub boards were designed using 2.5D and 3D electromagnetic simulation tools. The design files for the RF interfaces and the antenna radiation patterns simulations are available for students to study and modify should they decide to focus their studies on RF design.

All the prototypes were hand-assembled using relatively simple tools. A small-tip soldering iron, a binocular microscope and a small reflow oven (toaster oven) were the only equipment required to assemble the prototypes. The department technical staff was involved during the design phases of the project, and care was taken that they had the required information to maintain the kits and had access to extra blank circuit boards and leftover spare parts from the production run.

The final teaching kit with the cables and carrying box is illustrated in Fig. 2.

VI. Student Performance Evaluation and Grading

Students are evaluated on how they refine their specification and how they meet their own project objectives. Short milestone presentations ensure constant progress along the fixed deadlines. After the final project is completed, a "peer evaluation" of team members is carried out to ensure a weighted distribution of some percentage of the final grades based on each member's effort. The explicit training in successful teamwork constitutes an excellent conclusion to their undergraduate education in electrical and computer engineering.

A. Student-Defined Feature Set

Having students select the level of complexity and feature set of their final project work creates a certain competitive atmos-

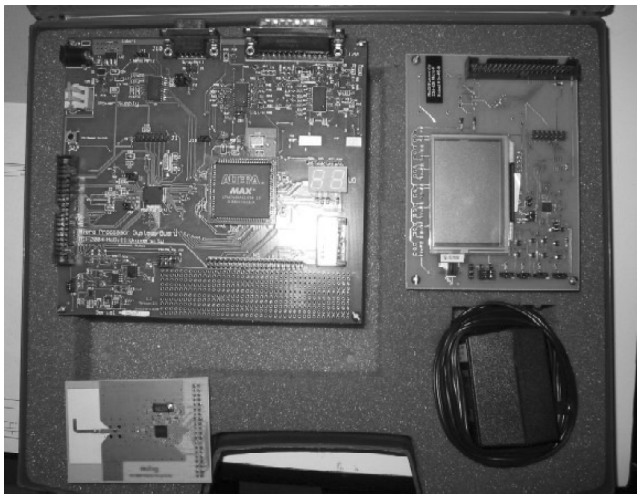

Fig. 2. Teaching kit photograph showing the McGumps main board (upper left), McGuld LCD with touch panel (upper right), and the McZig RF interface (lower left).

TABLE II
EVALUATION TABLE FOR STUDENT-DEFINED FEATURE SET

Feature	Delivered	Not Delivered
Specified	Great (++)	Bad (-)
Not Specified	Good (+)	Fair

phere in the lab, and usually elicits quite advanced and often surprisingly complete solutions to the given problems. From the received feedback, most students prefer this approach as it offers them an environment that is similar to the one they expect to later encounter in the industry.

As part of the teaching approach, students are provides with a broad set of specifications. Each team is left to decide on their final project specifications details. Since over-specifying and under-delivering is something frowned upon in the industry, students are encouraged to realistically predict the outcome. The simple evaluation criteria are given in Table II. Teams need to establish the feature set of their project early on and their initial estimates are reexamined at the end of the project.

B. Curbing Plagiarism

Increased opportunities for plagiarism are being created by technology improvements in file sharing and broad access to the Internet. Computer files with solutions can be distributed easier than ever, so thorough checks are a necessity. In the case of this laboratory kit, where the daughterboards are tailored to the individual projects, and with no exercises or projects are being exactly repeated between semesters, the opportunities for plagiarism between semesters are virtually eliminated.

VII. TEACHING KIT EVALUATION

With new equipment having a number of different hardware and software modules, it is easy to "localize" the skills taught by the type of resource used. A database is maintained of major "orthogonal" skillsets and measures of the quantity and quality (e.g., grades assigned) of activities undertaken to reach each objective. Then, since the equipment is flexible and programmable, teachers can adjust the type of equipment modules used. Further, as the old equipment often failed and caused downtime (time wasted instead of used for reaching teaching objectives), the authors are putting in place a Web-based system that will track the downtime and problems caused by new equipment.

A. Feedback From Students

The laboratory material is rated every semester by students. Before the introduction of the McGumps kit, the ratings for the lab material for the 3 semesters before were around 4.2/5.0. After the introduction of the McGumps kit, ratings climbed up and above 4.8/5.0. Furthermore, when the McGumps kit was introduced, 94% of the students polled in the end of semester faculty survey found the laboratory to be "very valuable" to their studies (score of 5/5).

The evaluations also showed that 73% of the students found the difficulty to be "about right" and 23% found it to be "difficult." Only 3% and 1% found it "too hard" and "too easy," respectively.

B. Notable Improvements in the Teaching Results

The previous teaching kit required approximately four weeks to get students up to speed with the microcontroller and its development environment. Complex downloading procedures to update the firmware on the board were difficult to teach in a short time. The complex instruction set computer (CISC) "flavor" of the old microcontroller also required students to spend a lot of time learning a complex assembly language to perform simple tasks. The previous platform also used an embedded monitor to debug the firmware.

The new kit uses the joint test action group (JTAG) and hardware breakpoints that can be set directly from the development environment, making it much more intuitive for students. The main benefit from this tight integration between the development environment and the microcontroller is in saving of more than two weeks of time early in the course. An additional benefit is a much more rapid "cycle" between code change and updates (five seconds in the new setup versus approximately 45 seconds in the old). The faster and more transparent debugging interfaces make a significant difference in learning and visualizing the effects of the code. The permanent retention of both the CPU code and CPLD logic also speeds up the lab demonstrations as no time is spent waiting for students to setup.

The time saved early in the semester allows student to become very productive and master a core set of skills—assembly and "C" programming on embedded systems, interrupt handling, modular code design, hardware/software partitioning, and I/O interfaces—by the middle of the semester. This gives them more time to learn the peripheral modules used in the term and concentrate on their final project.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a laboratory kit that is used to teach embedded systems through the use of simple and power-efficient wireless interfaces and various add-on "daughtercards." The in-house development of the kit proved to be a viable, economical and beneficial approach, bringing the ability to integrate and control all the aspects of the final product. Students consistently showed strong interest in the various projects built with

the teaching hardware which has helped in the continuous improvement of the teaching platform for more than three years. The new kit allowed a more rapid ramp-up in students embedded programming skills, which results in them having more time to integrate complex peripherals.

With respect to future upgrades, wireless modules that use flexible transceivers which allow a finer control on the physical layers are being evaluated. The integration of reprogrammable mixed-signal array controllers [13] is also being studied to expand further the prototyping abilities of this teaching kit.

ACKNOWLEDGMENT

The authors would like to thank Ph.D. student G. Gauthier of the Department of Educational and Counselling Psychology, McGill University, for her assistance in clarifying the cognitive learning principles underlying the teaching methodology. The authors would also like to thank B. Thomson for his insight and help during the design of the prototypes and for his support during board production.

REFERENCES

[1] S. F. Midkiff, "An experiential course in wireless networks and mobile systems," *IEEE Pervasive Comput.*, vol. 4, no. 1, pp. 9–13, Jan. 2005.

[2] J.-S. Chenard, U. Khalid, M. Prokic, R. Zhang, K.-L. Lim, A. Chattopadhyay, and Z. Zilic, "Expandable and robust laboratory for microprocessor systems," in *Proc. IEEE Int. Conf. Microelectronic Systems Education*, Anaheim, CA, Jun. 2005, pp. 65–66.

[3] Z. Zilic, J.-S. Chenard, and M. Prokic, "A laboratory for wireless and mobile embedded systems," in *Proc. IEEE Int. Conf. Microelectronic Systems Education*, San Diego, CA, Jun. 2007, pp. 103–104.

[4] A. A. Gokhale, "Collaborative learning enhances critical thinking," *J. Technol. Educ.*, vol. 7, no. 1, pp. 22–30, 1995.

[5] R. S. Dannelly, P. Michaud, H. Patterson-McNeill, and C. W. Steidley, "A multidisciplinary student-centered laboratory," in *Proc. 32nd Frontiers in Education Conf.*, Boston, MA, Nov. 2002, pp. T2D–6.

[6] T. S. Hall and J. O. Hamblen, "Using an FPGA processor core and embedded linux for senior design projects," in *Proc. IEEE Int. Conf. Microelectronic Systems Education*, San Diego, CA, Jun. 2007, pp. 33–34.

[7] Texas Instruments MSP430 Microcontrollers [Online]. Available: http://www.ti.com/msp430

[8] NXP LPC2000 Microcontrollers [Online]. Available: http://www.standardics.nxp.com/products/lpc2000

[9] P. Frantz, C. Garnier, E. Welsh, and A. Valenzuela, "Microcontroller and embedded systems laboratory," 2005 [Online]. Available: http://cnx.rice.edu/content/col10215/latest

[10] R. J. Duckworth, "Embedded systems design with FPGAs using HDLs (Lessons learned and pitfalls to be avoided)," in *Proc. IEEE Int. Conf. Microelectronic Systems Education*, Anaheim, CA, Jun. 2005, pp. 35–36.

[11] MAX 7000A Programmable Logic Device Data Sheet (Version 4.5), Altera Corporation, San Jose, CA, 2003 [Online]. Available: http://www.altera.com/literature/ds/m7000a.pdf

[12] S. Harris and D. Harris, "Inexpensive student-assembled FPGA/microcontroller board," in *Proc. IEEE Int. Conf. Microelectronic Systems Education*, Anaheim, CA, Jun. 2005, pp. 101–102.

[13] Cypress PSoC Mixed-Signal Controllers [Online]. Available: http://www.cypress.com/psoc

Jean-Samuel Chenard (S'99) was born in Amos, QC, Canada. He received the B.Eng and M.Eng. degrees from McGill University, Montréal, QC, Canada, in 1999 and 2005, respectively. He is working towards the Ph.D. degree in the Integrated Microsystems Laboratory at McGill.

He has been a Hardware Intern for iMPath Networks and a Consultant for EXFO Protocols. From 2000 to 2003, he was an ASIC Engineer for Marconi plc, designing broadband switching equipment. His research interests include embedded wireless systems, flexible radio, reprogrammable computing, and system-level debugging methodologies.

Zeljko Zilic (S'93–M'97–SM'07) received the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1997.

He is currently an Associate Professor at McGill University, Montréal, QC, Canada. From 1997 to 1998, he was a Member of the Technical Staff, FPGA Division, Microelectronics Group, Lucent Technologies. His research involves various aspects of system design, test, and verification. He has authored more than 100 research papers, holds four patents in the area of clock and power management, and coauthored *Verification by Error Modeling* (New York: Springer) in 2003 and *Generating Hardware Assertion Checkers* (New York: Springer) in 2008.

Dr. Zilic has served as a Member of Technical Program Committees of the Association for Computing Machinery International Symposium on FPGAs, IEEE International Test Conference, Midwest Circuits and Systems Symposium and Electronic Circuits and Systems. He is on the editorial board of the *Journal of Multiple-Valued Logic and Soft Computing* and the *International Journal of Software and Information Technologies*. He is a recepient of a Chercheur Stratégique research chair from the province of Québec. He received the Myril B. Reed Best Paper Award from the IEEE International Midwest Symposium on Circuits and Systems in 2001, a Best Paper Award at the Design and Verification Conference in 2005, and several honorary mention awards. For his undergraduate teaching, the National Council of Deans of Engineering and Applied Science and Sandford Fleming Foundation awarded him with the Wighton Fellowship in 2006.

Milos Prokic was born in Belgrade, Yugoslavia. He received the B.Eng. and M.Eng. degrees with honors in electrical engineering from McGill University, Montréal, QC, Canada, in 2003 and 2005, respectively.

He is currently managing the Business Integration Department, Adaltis, Montréal, QC, Canada, and Rome, Italy.