Pushing Beyond Traditional School and Course Boundaries: High School and University Students Collaborate on an Interdisciplinary Project

Adrian Rusu [1], Amalia Rusu [2], Jennifer Kay [3], and Hong Zhang [4]

Abstract **- Aimed at promoting math, science, and technology education in elementary and secondary schools, K-12 students and their teachers have participated in a variety of outreach programs. However, the majority of these programs have focused on small-scope projects which do not expose students and teachers to a range of CSET fields, but rather to specific CSET technologies and concepts. In this paper we present a case study of a collaboration between Williamstown High School and Rowan University students, teachers, and faculty. In the context of a college-level Software Engineering course, we exposed all those involved to a variety of CSET disciplines and experts, including Mechanical Engineering, Electrical Engineering, Software Engineering, Computer Science, and Robotics. Student reaction to the project was positive, and several interesting findings are presented. Active or observing roles could be assigned to the participants, depending on the scope, goals, duration, and availability.**

Index Terms - Interdisciplinary project, K-12 education, Software engineering education, Undergraduate

INTRODUCTION

There are a myriad of outreach programs designed to promote math, science, and technology education to Kindergarten through 12th grade students. Special programs for K-12 teachers also exist to train them to provide new knowledge to their students (see for example [1, 2, 3]). These programs range from a day to a few weeks in duration (see for example [4, 5, 6, 7]) and even appear in the form of summer camps (see for example [8, 9, 10]). However, the majority of these outreach programs have focused on small-scope projects which do not expose students and teachers to a range of CSET fields, but rather to one or two specific CSET technologies and concepts.

Many high school students associate computer scientists and software engineers with the stereotype of the isolated "hacker geek" who spends all day alone coding. By exposing such students to the entire software development lifecycle, we can provide them with a better idea about what software careers involve. But why limit the example to software development? If we take care to choose a project that is interdisciplinary in nature, we can push beyond the traditional definition of "the computer science class" and in the process demonstrate how professionals from a variety of CSET disciplines can work as a team to produce a complete system.

In this paper we present a case study of a collaboration between high school and college students, teachers, and faculty. In the context of a college-level Software Engineering course, we exposed all those involved to a variety of CSET disciplines and experts, including Mechanical Engineering, Electrical Engineering, Software Engineering, Computer Science, and Robotics. We also present student reaction to the project and discuss possible challenges and additions.

OVERALL INTERDISCIPLINARY PROJECT IDEA

The project began in Summer 2006, when students from the Williamstown High School Academy of Science and Engineering built several underwater "Remotely Operated Vehicles" (ROVs). Each vehicle has three propellers that control left/right, forward/backward, and up/down movement. The vehicles were tethered to the surface and manually controlled through a series of four switches which operate the three propellers, powered by a battery (see Figure 1).

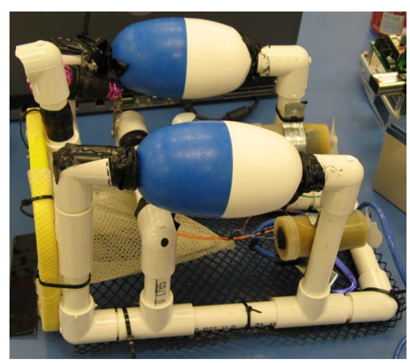

FIGURE 1
One of the Underwater ROVs built by Williamstown High School students during Summer 2006.

[1] Adrian Rusu, Department of Computer Science, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA, rusu@rowan.edu
[2] Amalia Rusu, Department of Software Engineering, Fairfield University, Fairfield, CT 06824, USA, arusu@mail.fairfield.edu
[3] Jennifer Kay, Department of Computer Science, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA, kay@rowan.edu
[4] Hong Zhang, Department of Mechanical Engineering, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA, zhang@rowan.edu

The project continued into Fall 2006, when Rowan University Computer Science and Engineering students and faculty worked together with a group of twenty five high school students enrolled in the programming class at Williamstown High School to develop hardware and software designed to eliminate the tether and enable the ROV to be computer-controlled from the surface. The overall concept is presented in Figure 2. The use of software control added new functionality and allows for future enhancements to the previously limited system.

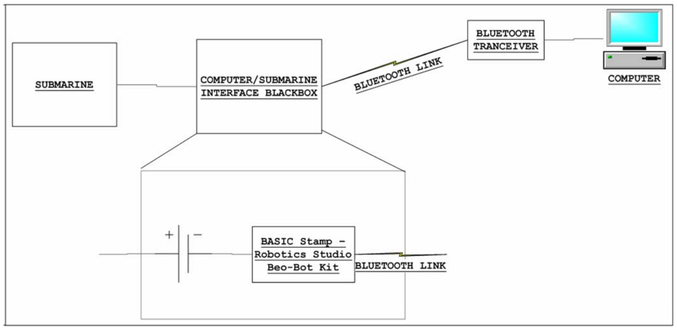

FIGURE 2
Overall project concept: a Teleoperated Submarine is controlled from the laptop across a wireless link.

SOFTWARE ENGINEERING AND COMPUTER SCIENCE COMPONENTS

Four Computer Science undergraduates and one Electrical and Computer Engineering undergraduate formed the software engineering team. These students were registered in the Software Engineering course in the Department of Computer Science at Rowan University. This semester-long introductory Software Engineering course is structured based on the approach introduced by Rusu [11]. In this approach, the instructor presents software engineering concepts and methodologies, and each student works in a team on a carefully-selected real-world project, interacts with a real-world customer, exercises every software engineering phase, and develops a software product which can be used, as is, by his/her team's customer. In this project, the customer was the high school representative and the users were high school students and teachers.

The students used a Rapid Prototyping lifecycle model and experienced all the phases of the software development process as they produced a fully operational software component (see Figure 3 for a screenshot of the final user interface). Current functionality of the software includes manual real-time directional controls:

- **Forward/Backward**: left and right propellers forward/backward in unison
- **Right**: left propeller forward, right propeller off – the Sub turns right gradually while maintaining forward motion
- **Left**: left propeller off, right propeller forward – the Sub turns left gradually while maintaining forward motion
- **Sharp right**: left propeller forward, right propeller backward
- **Sharp left:** right propeller forward, left propeller backward
- **Dive:** central propeller in reverse
- **Surface:** central propeller straight
- **Stop:** all propellers off

The software also allows the recording and playback of a manual course: if the Sub is placed in the same start location it will autonomously perform a repetition of the run under similar environmental conditions. Video feed recording through a computer-connected camera, as well as video playback of a recorded course are available.

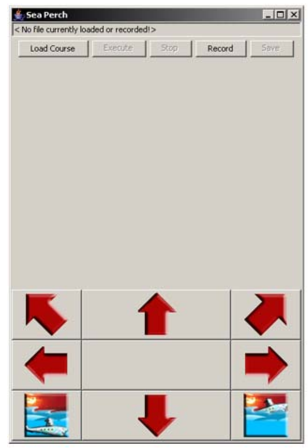

FIGURE 3
Final User Interface.

The requirements phase was split in two parts: requirements analysis and requirements specification. The Rowan students performed the requirements analysis through interaction with the high school teachers and students, as well as through the analysis of the existing system. Rowan students also developed a prototype to better understand and capture the requirements. During the requirements specification, students used the information gathered to produce the Software Requirements Specification (SRS) document. The SRS was explained to the high school teachers and students during a presentation session organized at the high school. Modifications and additions were performed based on the feedback received from the high school teachers and students.

To facilitate parallel development, as well as future additions, Rowan students chose an object-oriented design for the software, comprised of five major modules: *User Interface*, *Manual Control*, *Automatic Control*, *Video*, and *Communication* (see Figure 4). Each of these main modules was, in turn, comprised of several submodules. The User Interface module is responsible for setting up the graphical user interface, listening for input via keyboard or mouse click, and to allowing for embedded display of video feed. When a directional button or key is hit or clicked, it calls the corresponding Manual Control module. When a file containing a recorded run is loaded, it calls the Automatic Control module. The Manual Control module is responsible for determining which action is desired and sending the corresponding information to the Communication module. It is also responsible for recording direction control actions if the record button is pressed. The Automatic Control module is responsible for loading commands from a file and sending them to the Communication module. The Communication module opens an available COM port, establishes connection to the Basic Stamp through the use of the Bluetooth device, and converts the input from the Manual and Automatic modules into Basic Stamp signals which are being sent to the Basic Stamp. The detailed design document (DDD) was explained to the high school teachers and students during a presentation session organized at the high school. High school students also reviewed the DDD together with their teachers during follow-up sessions, and provided valuable feedback.

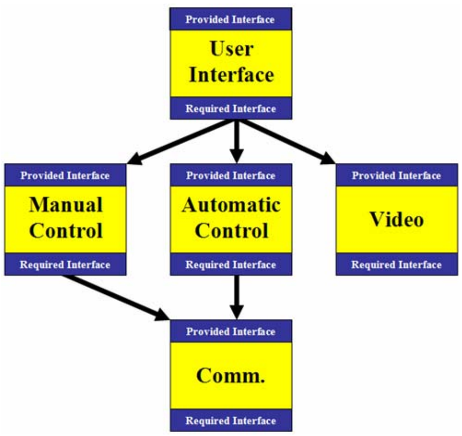

FIGURE 4
High level architecture of the Submarine system.

The Rowan students chose Java as the implementation language for the software to control the ROV for several reasons: it provides an easy means of interfacing with the hardware, it allows for easy access to computer communication ports for use with the Bluetooth, it incorporates easy access to development of user friendly user interfaces, as well as the embedding of video. Forward thinking, Java implementation, together with a modular design, allows for easy future expansions. Students established coding conventions to allow for a uniform development of the implementation.

Since reliability and robustness were two of the most important software qualities identified during the requirements analysis phase, the students performed an extensive testing phase. This included having team members review code and develop structural and specification-based test cases for modules not written by them to avoid blind spots and hind sight. The assertions developed during the implementation phase proved to be very helpful in correcting improper assumptions between modules, and allowed them to successfully perform all integration tests developed during the design phase. Finally, they performed acceptance tests using real commands and data provided by the high school teachers during the requirements analysis phase.

The Computer Science component, which was incorporated as part of the software design process, allowed the students to use the theory and fundamentals in the design of data structures, algorithms, and in the code they wrote.

The High School students played an active role in the Software Engineering part of the project, from eliciting requirements from their teachers, to designing "splash screens" (see Figure 5 for an example) for the software, and even writing a draft user manual.

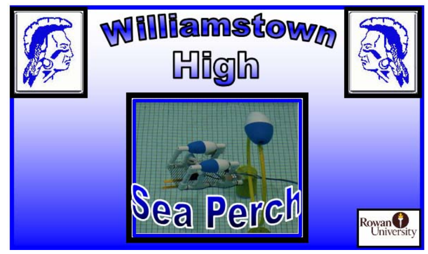

FIGURE 5
Sample "splash screen" produced by the High School students during the implementation phase.

ROBOTICS COMPONENT

The Robotics programming team consisted of a graduate Electrical and Computer Engineering student, an undergraduate Computer Science major (who was a member of the software engineering team), and one Computer Science faculty member with Robotics specialization. This team was responsible for programming the interface between the front end user interface and the Sub's motor controllers.

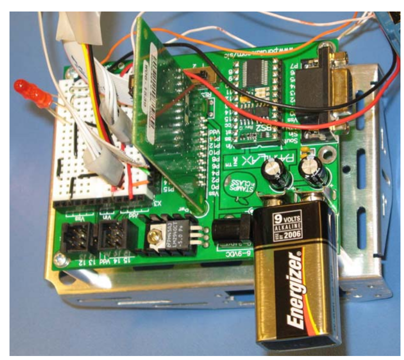

FIGURE 6
Basic Stamp with Bluetooth device.

The software was run on a Basic Stamp 2 microcontroller from Parallax Inc [12] programmed in PBASIC, a variant of the BASIC language. The Basic Stamp 2 received commands from the Java front end across a Bluetooth [13] wireless connection (see Figure 6). The commands received represented the generic movement commands for the Sub described earlier (Forward, Backward, Left, Right, etc.). The software converted these commands into speeds for the individual motors (in our system, there were only two speeds used: “full power” and “off”), which were sent over a serial connection to the Sub’s motor controllers.

ELECTRICAL AND MECHANICAL ENGINEERING COMPONENTS

The Electrical and Mechanical Engineering team consisted of one undergraduate Electrical and Computer Engineering student (who was a member of the software engineering team) and one member of the Mechanical Engineering faculty with a Mechatronics specialization.

There are several possible ways to connect the Basic Stamp to the Sub. For example, we could use a home-made H-bridge circuit based on either MOSFET or transistors, a series of relays, or an off-the-shelf controller. We chose to use commercially available amateur grade motor controllers for the project due to the simplicity of the interface and the tight time constraints. By testing the burst current of the motors with a multimeter, we found the maximum current was about 1.6A for each of the Sub’s engines. Hence, a controller that can handle 2A peak current would give us enough safety space for one engine. From past experience, the Pololu motor controller was chosen due to its long term partnership with Parallax microcontroller we were using. We also appreciated its simplicity and its intensive documentation available from both the company and Internet forums. For our project, we used three Pololu motor controllers, one for each of the three Sub engines.

One widely available version of the Pololu motor controller is SMC03A (see Figure 7). It can handle up to 3A current, which was more than sufficient for our needs. It communicates to the Basic Stamp 2 with one single serial pin (pin 3). It also provides us with the option to add feedback of the motor speed in the future.

FIGURE 7
Pololu 3A motor controller with feedback.

The physical connection of the controller to the computer and ROV is quite intuitive even to the high school students. First set the jumper to serial connection, then connect the corresponding pins of Basic Stamp to pins VCC, GND, and 3 of the controller, then finally secure the wires from the engine and battery to the white screw clamp terminal block. The programming in PBASIC is almost equally simple, though some fine tuning is required to get the comparable speeds for all motors. The visible variation of the frictions of thrusters leads to different speeds under the same current. Therefore, experiments have to be carefully conducted to find the right current to ensure that each motor has the same speed. Of course, a future version with speed feedback will eliminate the guesswork. This will require the addition of an optical encoder or a potentiometer to the circuit.

PRACTICAL IMPLEMENTATION TIPS

In this section we describe some additional implementation details that we hope will help others interested in pursuing similar projects.

Basic Stamp Hardware

We chose to purchase the Boe-Bot Robot kit [14]. The Boe-Bot kit includes a Basic Stamp 2 Chip that is mounted on a Board of Education development board [15]. The integrated breadboard makes interfacing with other hardware very straightforward. In addition to being able to connect to devices like the Pololu motor controller, we used some LEDs mounted on the breadboard to aid in debugging.

Programming the Basic Stamp

The PBASIC language (complete with GOTOs and GOSUBs) would not be our first choice of language pedagogically. However, while it is not elegant, it is sufficient for the needs of many programming projects, and the Boe-Bot kit is quite reasonably priced (at about $150 [14]). Programming the Basic Stamp chip involves creating the program on a remote computer and then downloading it to the Basic Stamp.

Interfacing with the Java Front End

Our own software on the Basic Stamp chip is quite straightforward: the system waits for data (corresponding to a generic movement command) to come in across the Bluetooth device using a SERIN command and then BRANCHes to the appropriate code that specifies the speeds (“full power” or “off”) to each of the ROV’s three motors and sends the appropriate message to the Pololu. Modifications to the functionality of the system thus need to be made both in the Java front-end code and the PBASIC code for the Basic Stamp.

Working Underwater

Many complications arise when working with an underwater ROV. The original Sub provided by the high school students was maneuvered through the water by a remote control unit attached to the sub via a tether. In an ideal world, the Basic Stamp chip and Bluetooth devices shown in Figure 6, together with the Pololu motor controllers and the battery would have been located on the Sub itself as envisioned in Figure 2, and the tether would no longer be required. However, because we did not have a waterproof housing for these units, we kept them out of the water, and connected them to the Sub via a tether (See Figure 8).

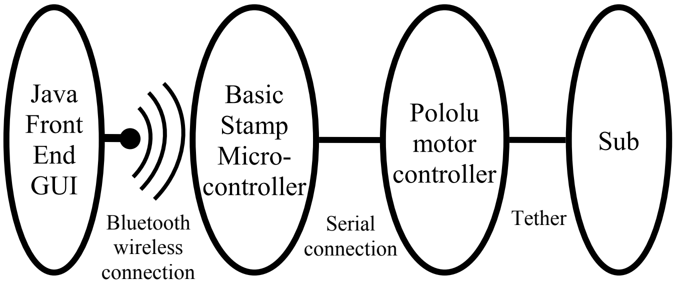

FIGURE 8
Hardware Setup.

Access to a clean body of water can also be a challenge. The high school students tested out their Subs in a backyard swimming pool during the summer. By the time the final project was ready to be demonstrated, outdoor pools had been drained and the availability of the university pool to such a project was extremely limited. Indeed, due to a hardware glitch on the day the university pool was reserved, the final project could not be demonstrated underwater, and the final (successful, but out-of-water) demonstration had to be made on dry land the next day. We encourage faculty working on underwater projects to require a final demonstration (for credit!) on land a day or two *before* a limited access resource like a swimming pool is scheduled.

Participation

Depending on the scope, goals, and their availability, various roles and degrees of participation could be established for each participant. In our case study, we split the high school students in groups of five, and each group was assigned for guidance and supervision to a college student. They were explained in detail each component, and actively participated in the Mechanical Engineering, Computer Science, and Software Engineering components of the project. High school teachers are currently using the Electrical Engineering and Robotics components to further them with interested students.

Project Demonstrations

In this class, as well as in previous sections of the Software Engineering course, we have designed the final project demonstrations to be significant affairs [11]. We invite all of those who have directly worked on the project (including all of the customers who interacted with our students). We also invite other individuals who have not had direct involvement in the project, but who are affiliated with the University and Department(s) like the department chair and faculty, as well as “high-ranking” individuals on the customer side. While we only have anecdotal evidence to support this, we have a strong belief that the knowledge that the final presentation is a major event provides a significant amount of extra motivation for the students involved.

For example, for this project the final demonstration was made to the twenty five high school students directly involved with the project throughout the semester, as well as seventeen high school students from other engineering classes, three high school teachers, the high school principal, and the Computer Science department chair. The university students were also subsequently recognized during a Monroe Township Board of Education meeting with certificates of appreciation.

HIGH SCHOOL STUDENT FEEDBACK

A comprehensive survey of the twenty five high school students involved with the project throughout the semester was performed after the final demonstration, which resulted in 19 returns.

Interesting findings include:

- 84% chose the proper order of the software development phases: requirements, design, implementation, testing.
- 74% of the high school students found interesting to watch University computer science and engineering students and faculty develop a complete system for their High School.
- Only one student was not interested in Computer Science in general, and in computer programming in particular.
- Only 16% were not interested in software engineering.
- Only two students were not interested in robotics.
- Only two students disagreed that studying computer science in college would be fun.
- 27% strongly agreed, 26% agreed, and 47% were neutral to the effect of the semester-long experience on their grown interest in studying computers.
- Only 16% would not recommend the experience to a friend.
- 68% agreed and only one strongly disagreed that they would like to continue learning about computers.

FUTURE ADDITIONS

The proposed project could also be extended to additional semesters. Potential enhancements include:

- Software enhancements:
  - Modify the human interface to enable speed control (the current hardware and motor controllers would work in such a system without additional modification).
  - Present dual video windows to allow the user to compare the video from a manual run to a live video feed from the Sub. This could show how accurately a recorded run of the course can be recreated, as well as how influential variables such as water movement are on the Sub.
  - Add the option of parameter-based autonomous movement, allowing the Sub to run a course automatically based on input parameters, without a pre-run.
  - Add Web accessibility, allowing the Sub to be controlled from a remote computer over the Internet.

- Design video analysis based autonomous control (build a computer vision module for the Sub to automatically identify obstacles and avoid them).
- Research and implement alternate wireless communication protocols.

- Hardware enhancements:
  - Connect a camera directly to the Sub to allow for autonomous control through video analysis techniques.
  - Make the system fully wireless by creating a waterproof housing for the basic stamp, motor controllers, and battery so they can be placed on the Sub.
  - Experiment with other wireless connections, such as sonar or acoustic technologies, if Bluetooth does not function well under water.
  - Allow for joystick control, instead of mouse or keyboard control.
  - Move the software to a PDA or cell phone with Bluetooth, and therefore control the Sub with a handheld device.
  - Incorporate speed feedback from the motors to ensure uniform speed as previously described.

CONCLUSION

In this paper we have presented a novel interdisciplinary project involving a collaboration between high school and college students. This collaboration truly benefited all the participants.

The high school student engineers and computer scientists were able to observe the progress of the hardware modifications and software development, as well as how the University engineers and computer scientists collaborated to produce the final product in a real world setting. The University students, in turn, had real world customers (with all the benefits and complexities that real customers provide) that were depending on them to succeed in the project. The Computer Science and Engineering high school teachers will be using the newly developed system to attract more students to their academies, as well as in classroom demonstrations and as a teaching aid. Finally, everyone involved was able to see the evolution from initial Sub concept, into a working Sub that was manually controlled, and then into a software based system that can continue to be expanded in numerous directions by high school and university students alike.

In addition to the exposure to the variety of CSET disciplines it covers, the complexity of the project allows high school teachers and college faculty much flexibility in assigning a more active role to the high school students in the development of the system, as considered fit.

ACKNOWLEDGMENT

We thank Williamstown High School Computer Science and Engineering teachers Mary Peacock, Jerry Votta, and Richard Kurczeski, as well as Principal Steven Stumpo for their support and dedication to the project. We also thank the Rowan University students who contributed to the success of this project: Christopher Dutkowski, Jacqueline Finger, Eric Kesterson, Eugene Langdon, Russell Posluszny, Ali Daneshmand, and all of the Williamstown High School Engineering and Computer Science academy students involved in the collaboration.

REFERENCES

[1] Morrison, D., Kranc, S., Smith, C., "Engaging High School Teachers in Engineering Research: Our Experience", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[2] Carlson, P., "Web Portals and Faculty Development", *Proceedings 32nd ASEE/IEEE Frontiers in Education Conference,* 2002.

[3] Blum, L, Cortina, T., "CS4HS: An Outreach Program for High School CS Teachers", *Proceedings 38th ACM Technical Symposium on Computer Science Education,* 2007.

[4] Cottrell, D., "Computer Aided Design for Introducing High School Students to Engineering", *Proceedings 32nd ASEE/IEEE Frontiers in Education Conference,* 2002.

[5] Collofello, J., Urban, J., Anderson-Rowland, M., Navabi, F., Banks, D., Roman, D., "COOL (Computer Outreach Opportunities for Learning) Development and Assessment", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[6] Spanias, A., Thrasyvoulou, T., Song, Y., Panayiotou, C., "Using J-DSP to Introduce Communications and Multimedia Technologies to High Schools", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[7] Huggard, M., Goldrick, C., "Incentivising Students to Pursue Computer Science Programmes", *Proceedings 36th ASEE/IEEE Frontiers in Education Conference,* 2006.

[8] Sullivan, J., Reamon, D., Louie, B., "Girls Embrace Technology: A Summer Internship for High School Girls", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[9] Pollock, L., McCoy, K., Carberry, S., Hundigopal, N., You, X., "Increasing High School Girls' Self Confidence and Awareness of CS Through a Positive Summer Experience", *Proceedings 35th ACM Technical Symposium on Computer Science Education,* 2004.

[10] Papalaskari, M., Hess, K., Kossman, D., Metzger, S., Phares, A., Styer, R., Titone, C., Way, T., Weinstein, R., Wunderlich, F., "PIVOTS: Service Learning at the Science, Theatre, and Magic Boundary", *Proceedings 36th ASEE/IEEE Frontiers in Education Conference,* 2006.

[11] Rusu, A., Webb, R., Shanline, D., Santiago, C., Luna, C., Kulak, M., "A Multiple-Projects-Multiple-Winners Approach for Teaching Software Engineering", *Proceedings 9th IASTED International Conference on Software Engineering and Applications*, ACTA Press, 2005.

[12] "Parallax, Inc-StoreFront Product Detail Page," http://www.parallax.com/detail.asp?product_id=BS2-IC, accessed March 2007.

[13] "Bluetooth.com | The Official Bluetooth ® Wireless Info Site," http://www.bluetooth.com/bluetooth/, accessed March 2007.

[14] "Boe-Bot," http://www.parallax.com/html_pages/robotics/boebot/boebot.asp, accessed March 2007.

[15] "Parallax, Inc [Board of Education Development Board]," http://www.parallax.com/detail.asp?product_id=28150, accessed March 2007.