



Teaching Systems and Robotics in a Four-Week Summer Short Course

Dr. Andrew Danowitz, California Polytechnic State University, San Luis Obispo

Andrew Danowitz received his PhD in Electrical Engineering from Stanford University in 2014, and is currently an Assistant Professor of Computer Engineering at California Polytechnic State University in San Luis Obispo. His engineering education interests include student mental health, retention, and motivation.

Dr. Bridget Benson, California Polytechnic State University, San Luis Obispo

Bridget Benson received a Bachelor's degree in Computer Engineering at California Polytechnic State University San Luis Obispo in 2005, a Master's degree in Electrical and Computer Engineering at the University of California Santa Barbara in 2007 and a PhD degree in the Computer Science and Engineering at the University of California San Diego in 2010. She is currently an Assistant Professor in the Electrical Engineering Department at California Polytechnic State University San Luis Obispo. Her research interests span engineering education, internationalization, and embedded systems.

Prof. Jeremy Edmonds, California Polytechnic State University, San Luis Obispo

Jeremy Edmonds has extensive professional experience across a variety of industries such as information security and cryptography, cryogenic engineering, military and aviation power systems and telemetry, and robotic ultrasonic scanning research for the oil and gas industry. He owns and manages a private startup company for design, consulting, and services, including the utilization of computer vision for investigations and computer forensics services. Jeremy has previously taught college courses such as analog circuits, troubleshooting, robotics, and various other specialties. His primary passion and focus is helping aspiring professionals learn and easily understand complex concepts, instilling student confidence, and giving students the ability to put their knowledge to practical use.

Teaching Systems and Robotics in a Four Week Summer Short Course

Abstract

This paper describes a four-week summer short-course designed to introduce students with limited hands-on technical experience to the low-level details of embedded systems and robotics. Students start the course using a Raspberry Pi 3 to learn the basics of Linux and programming, and end the course by competing in a capture-the-flag type competition with the web-configurable GPS-guided autonomous robots they designed and tested in the course. Throughout the course, students are introduced to programming languages including Python and PHP, advanced programming concepts such as using sockets for inter-process communication, data interchange formats such as JSON, basic API development, system concepts such as I2C and UART serial interfaces, PWM motor control, and sensor fusion to improve robotic navigation and localization. This course was offered to students for the first time in the summer of 2016, and though formal feedback collection was limited, informal feedback indicated that students found the course to be challenging, engaging, and beneficial to their overall understanding of engineering.

The paper walks the reader through the background of this course. It then discusses the weekly lesson plans, supplemental material provided to the students, and our general strategy for teaching the course's programming and system design concepts in such an accelerated time frame. Finally, the paper discusses the student and instructor reactions to the course, lessons learned, and suggestions for future offerings. The material developed for this course will be posted online so that other educators may use it in their teaching.

Introduction

Autonomous vehicles and robotics are perennial hot-topics in the field of engineering. Robotics are frequently used as a teaching tool at the K-12 level to draw students into STEM fields [1, 2] and Robotics Summer Camps and extra-curricular activities have even been created for K-12 students [3, 4, 5, 6]. In higher education, although elements of robotics programs are found in most engineering disciplines, including Aerospace, Mechanical, Industrial, Electrical and Computer Engineering, as well as Computer Science, the focus of robotics courses is often narrowly tailored. While a Computer Science Robotics program may focus on the high-level algorithms used for image recognition and navigation, a mechanical engineering program may primarily focus on the manipulation of servos and motors to complete specific tasks. For college students looking to become involved in robotics, however, it can be difficult to find an introductory course that empowers them with the knowledge to construct and experiment with their own autonomous robots. Also, given that a number of college students have had K-12 robotics experience, it can be difficult for instructors to properly scope a general robotics course that can keep students of all backgrounds engaged and interested.

To remedy this situation, we have developed a 4-week intensive summer short course to give engineering students from all disciplines and backgrounds the basic skills needed to construct and test their own web-connected autonomous vehicles. The course differs from previously mentioned summer robotics programs by tailoring the course to undergraduate engineering

students rather than to K-12 students. The rest of this paper is dedicated to explaining the development of the course, and student reactions to the material presented.

The following section presents some background for the course, and how it was originally developed for our institution's Cal Poly's new international student summer program. Section Course Design describes our strategies for providing students with varying skill levels in programming and embedded systems a meaningful robotics experience in only four weeks. Section Student Reactions discusses data we collected from students about the course and perceived impacts on their engineering abilities. Finally, Conclusions discusses the lessons we learned from the first offering of the course, and our proposed improvements for future offerings.

International Summer School Program

This course was initially developed and piloted as an offering for our institution's new international summer school program. The summer school program is designed to bring engineering students from all over the world to California Polytechnic State University for a series of short-courses in topics of special engineering interest (automation, building design and construction, etc.). The short courses are designed and offered under the supervision of Cal Poly's faculty, and the program was marketed to students as a way to experience life and learning in the U.S. while honing their hands-on technical skills. These courses are not graded or offered for college credit, however, and the cost is roughly \$3000 plus travel expenses, potentially making it difficult for students from certain countries and socio-economic backgrounds to attend.

While the program provides a solid learning opportunity for international students, the way it is implemented presented two difficult challenges for designing our hands-on robotics course.

The first challenge we faced is that each course must cater to students from a wide range of educational backgrounds. To attend the summer program, students simply need to be fluent in spoken and written English, and enrolled in an undergraduate engineering program in their home country: there are no set requirements for academic level or achievement. Additionally, admitted students are eligible to take any set of short courses, regardless of their engineering specialty. As a result, our robotics short course had to be accessible to students who had never worked with embedded systems before while at the same time it needed to engage and challenge those students who already had robotics project experience.

The second major challenge we faced is that all of the presented material had to be interesting and engaging enough to keep students coming to class. The courses were not graded and students received no academic credit for taking them, so short course instructors had no prescriptive means of enforcing attendance or study policies. Also, since this was the first trip to the U.S. for many of our participants, we had to capture the interest of students who might be more inclined to enjoy the non-academic aspects of American college life.

While these two factors made the process of course design challenging, as discussed in Course Design and Student Reactions, we feel that we managed to create a robotics course that met the needs of our international summer students. We also believe that the challenges of catering to students of different backgrounds and skill levels and keeping students motivated in their

academic pursuits are not unique to our international summer program, and believe that this course would be of significant benefit to all students looking to get involved in the field of autonomous robotics.

Course Design and Implementation

As the course was developed to be a fast paced, practical hands-on experience for students of varying backgrounds, the system board and sensor used needed to meet specific criteria. First, we wanted to rely on off the shelf hardware supported by large communities of hobbyists, which would allow students the benefit of finding help and examples online both during and after the course. For our computing and sensing hardware, we also wanted devices that were relatively affordable, so that students could easily purchase their own components to tinker with after the course. Finally, we wanted the sensors and development platform to be relatively flexible and fully featured, so that the same basic robotic hardware we used in the course could be used for drones, land-based robots, and aquatic robotics. Based off these criteria, we came up with the list of parts shown in Table 1.

Table 1. Part selection for robotics course. Prices are current as of March 2017 [7, 8, 9].

Part	Description	Vendor	Price
Raspberry Pi 3	Main system board	Adafruit.com	\$39.95 USD
8 GB SD card	Store OS and files for Raspberry Pi	Amazon.com	\$5.99 USD
Adafruit Ultimate GPS Breakout Board	GPS module for location sensing	Adafruit.com	\$39.95 USD
Adafruit 9 DOF Absolute Orientation IMU Fusion Breakout BNO055	Accelerometer/Gyro/Compass module for fine-grained localization	Adafruit.com	\$14.95 USD
PiCamera	Raspberry Pi Camera	Adafruit.com	\$29.95 USD
USB to TTL Cable	USB to UART to connect sensors to board	Adafruit.com	\$9.95 USD
Dagu Wild Thumper 4WD All Terrain Chassis with 34:1 Motors	Robot Chassis	Pololu.com	\$174.95 USD
Pololu Qik 2s12v10 Dual Serial Motor Controller	Robot motor controller	Pololu.com	\$74.95 USD
2 Pack 7.2v 5000mAh NiMh Rechargeable RC Battery Packs	Robot power supply	Amazon.com	\$43.98 USD
Pololu 5V Step Up/Step Down Voltage Regulator S18V20F5	Converts robot power supply to 5V for Raspberry Pi	Pololu.com	\$14.95 USD
		TOTAL PER ROBOT	\$449.57 USD

We chose the Raspberry Pi over other popular choices like Arduino to give the students experience working with a fully featured Linux operating system (Raspbian [10]). Also, with Raspberry Pi, all development could be done directly on the board, so students could interact with the sensors in real time. Finally, the built-in wireless options on the Pi gave us the opportunity to teach students how to control their robots through the internet or smartphone using basic web-technologies.

For sensor selection, we settled on a GPS and nine degree of freedom IMU since these types of instruments are generally used in any sort of terrestrial or aeronautical autonomous vehicle. We selected parts from Adafruit since they were well documented with lots of tutorials and sample code available. Preference was also given to these sensors since they supported an array of options for accelerating the student development process. For example, the IMU can output raw sensor data, or ready-processed heading data in a number of coordinate systems. These choices of versatile, well-documented hardware opened opportunities for students to get the hardware working quickly but also left room for students to use creativity and make unique design choices.

For a robotic chassis and drivetrain, we selected a chassis and motors from Dagu and Polulu and a compatible motor driver. This offered a simple UART interface for controlling the motors and was rugged enough to drive around grass fields on campus, and survive a few crashes during the student learning process. The assembled robots are shown in Figure 1.

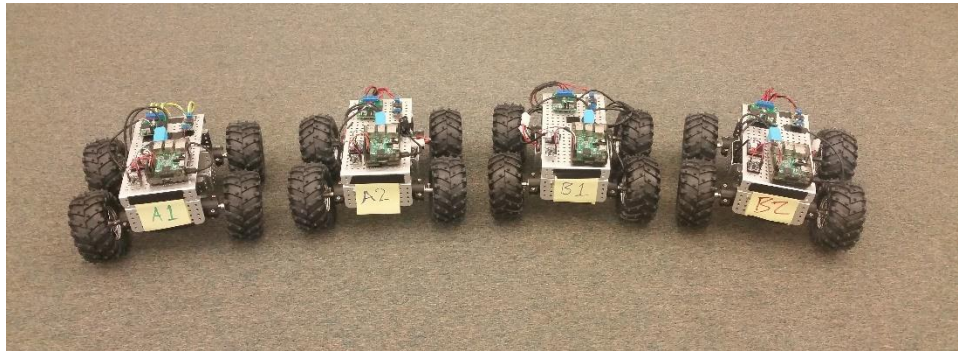


Figure 1. Four fully assembled course robots.

At the start of the course, each pair of students was entrusted with a Raspberry Pi and the whole host of sensors. Due to budget constraints, however, we were only able to afford four robot chassis and drive-trains, requiring students to take turns testing out new algorithms when we got into the prototyping phase of the course.

Course Demographics

For the pilot course, students largely came from two Egyptian universities. In future years, however, recruitment will be broadened to include a number of other countries.

In total, we were able to offer two sections of our course with roughly 24 students per section. The course met five days a week for four hours per day per section, and students were highly encouraged (although not required) to continue to work on course material outside of the

classroom. Each section was supervised by one of the three course instructors, although a second instructor would occasionally volunteer to attend to help with the day’s activities.

All students were fluent in English; however, all came in with highly variable knowledge of embedded systems. Some students had no programming experience beyond basic C for microcontrollers, while others were already familiar with the Raspberry Pi and Linux platform. To accommodate these varying backgrounds, we allowed students to form themselves in groups of 2-4, and we encouraged collaboration among different groups to complete the labs.

Course Breakdown

In order to support the wide range of background and skill level of the students, the course schedule was designed into four phases which fit roughly into one week each. These phases and topics are summarized in Table 2. For each of these segments, we provided laboratory assignments that would allow the students to see results early on in the learning process while introducing concepts that would allow the more advanced students to customize their systems.

Table 2. Course Schedule and Outline.

Week	Unit	Topics
1	Intro to Linux Programming	Introduction to Raspbian
		Introduction to PHP/HTML web programming
		Navigation Calculations
2	Interfacing Sensors	Introduction to Python
		Pulling and parsing data from GPS sensor
		Pulling and parsing data from IMU
		Python-PHP communication through sockets
3	Controlling the Robot	Introduction to AJAX
		Assembling robot chassis
		Closed loop control
4	Final Competition	Navigation algorithms
		Autonomous Capture the Virtual Flag

In the first week of the course, students were familiarized with the Linux environment of the Raspberry Pi and given a basic introduction to PHP/HTML web programming. We started with web-programming to give students a simple way to develop an interface to control their robots through their mobile devices. We selected PHP as our web integration language of choice since it is a common server-side web language with many available online tutorials [11] and makes it very easy to start creating very basic interactive webpages. We also used this segment to familiarize students with editing and running code in a Linux environment.

The second week of the course focused on familiarizing students with the basic sensors they need for many autonomous robotics applications: a GPS unit to determine position, and an IMU to determine heading and acceleration. The sensors we selected had substantial Python library support for the Raspberry Pi [12, 13], so we used this week as an opportunity to introduce students to the Python programming language. We also instructed students in the basics of Linux

sockets and the JSON interchange format so they could transfer data between their Python code and their PHP front-end. To make this task more manageable, we provided students with example Python code that opened an internal socket and listened for commands.



Figure 2. Students developing code.

During week three, students began the assembly of the final robotic platforms and were introduced to mobile autonomous robotic topics such as mobile robotic kinematics, motion control and feedback, perception, localization, and path planning. Common algorithms were introduced and explained, and students were tasked with lab objectives to begin merging the code they had previously developed for the sensors and hardware into systems capable of basic autonomous functionality. Throughout the week, students were challenged to implement what they were learning in the lectures into the robotic platforms.

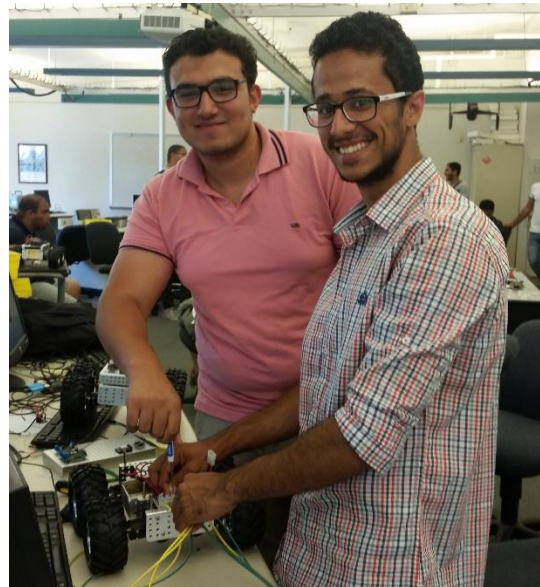


Figure 3. Students assembling the robot chassis.

Final Project

In the final week of the course, students from each section worked together on the development of robots to compete in an autonomous capture the flag game. Each section was responsible for distributing tasks among its members and assigning responsibilities, as illustrated in Figure 4. The two sections were also responsible for developing and agreeing on a common web API for the robots to share data.



Figure 4. Student team planning meeting, highlighting student leadership.

Lectures in the final week were given opportunistically to assist the students in understanding how to accomplish tasks, how to problem solve, and how to make design decisions and come to solutions. Instructional time was primarily spent guiding students through the processes of the implementation of algorithms, and working through the difficulties and pitfalls of real hands-on development.

Because students had the freedom to set their own goals, they gained experience in scheduling timelines, teamwork, compromise, and avoiding design creep. One very noteworthy event was by the end of the week, both teams realized that they would not be able to complete the project in time. In order to meet the goal of project completion, both opposing teams worked together and shared code that the other team needed.

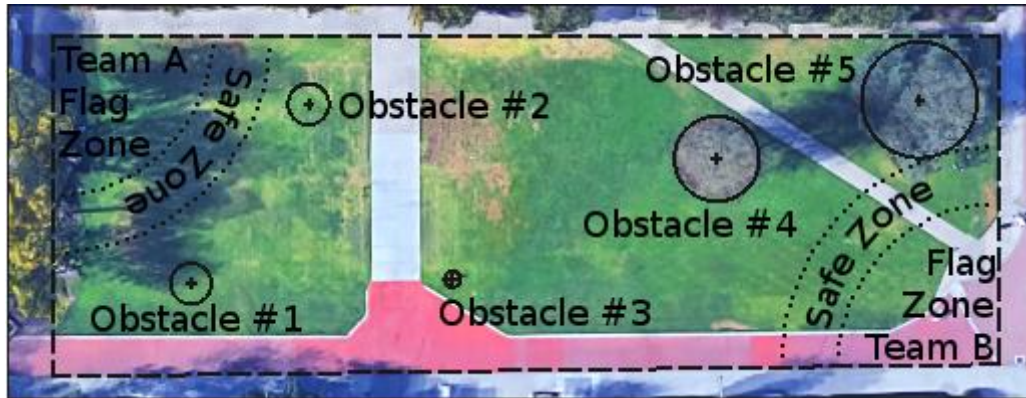


Figure 5. Boundary map for final competition.

At the end of week four, both teams had developed two robots that could autonomously compete in the capture the virtual flag competition. The robots each communicated their GPS and IMU data to the other robots, as well as game state information such as which robot has the flag, if the robots should return to starting positions, if the game is currently being played, and various other information. Both teams developed their own web interfaces to monitor and send commands to their robots. Both teams also developed their robotic code using their own design choices based on the various techniques and algorithms and other ideas covered in the class.

The competition itself was very informal. The students kept track of the score and declared a winner. The students took time to do a presentation of their work, and one student demonstrated a smartphone app he wrote that allowed him to control the location of the robot by clicking a desired location on an interactive map. While neither team's robot was very efficient at the capture the flag task, we viewed the exercise, and the class as a whole, as a success.

Student Reactions

Informal feedback (talking directly with students on the day they completed the course) indicated that students found the course to be challenging, engaging, and beneficial to their overall understanding of engineering. Anecdotally, many students thanked us for providing them with such a great experience. To obtain more formalized feedback, we emailed the students a link to a custom Survey Monkey survey roughly after the course was completed. Out of the 40 students who took the course, only 12 responded to the survey, which may be attributed in part to the fact that many students had extensive travel plans after the summer camp and therefore did not see the survey until much later. Although this formal feedback is limited, it still provides a useful evaluation of the course.

Figure 6 illustrates the average responses based on a five point Likert scale (with, depending on the question, 1 being too little, too slow, very uncomfortable, no effect, or poor, and 5 being too much, too fast, very comfortable, significant improvement, or excellent respectively) to six of the survey questions. From the figure, we can note that students found the amount of the material a tad too much (average 3.6) and the pace of material on the fast side (average 4.0) and reported spending anywhere from 0-36 (with an average of 9.5) hours per week on the course outside of class. However, they also reported a significant increase in their ability to put theoretical knowledge to practical use (3.0 before the course and 3.8 after the course) with the hands-on

activities also helping to improve their understanding of theoretical concepts (average 3.6). Also, overall, the students rated the course as Very Good (4.0) with no student rating the course below a 3 (Good). When asked if the students would recommend the course to their friends (not shown in the figure), 100% said yes (with response choices of yes or no).

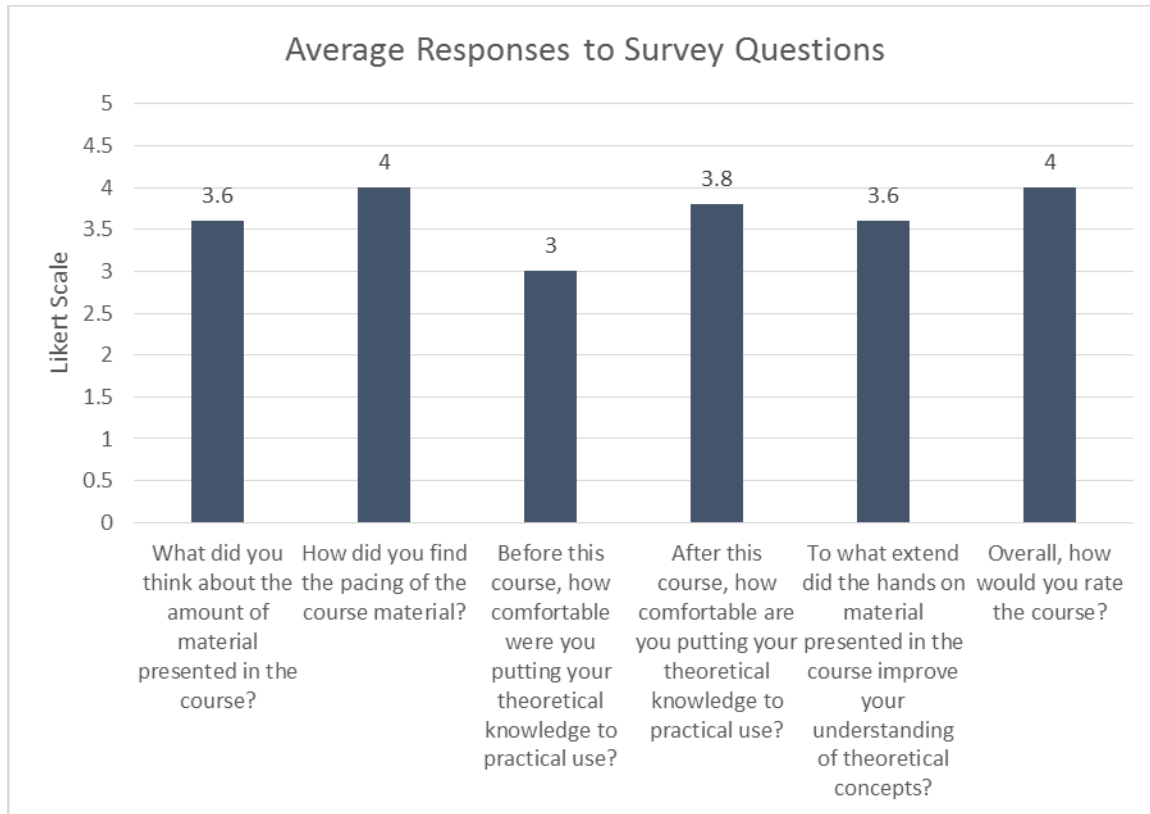


Figure 6. Average Responses to Survey Monkey survey questions.

In the open-ended questions of the survey, students reported their top three reasons for taking the course were:

1. To experience the American (and other) culture(s) and a different education method
2. To improve their practical / hands on skills
3. To learn more about robotics specifically.

To that effect, they also reported the top three ways to improve the class would be to:

1. Have students from multiple different countries in the class
2. Either slow down significantly or spend less time on the tougher theoretical concepts (especially using sockets for inter-process communication)
3. Spend more time on the robot itself

Some very telling quotes from the survey included:

"I believe I soaked as much knowledge as possible from all of the professors and I'm grateful for that."

"If I learn a new thing every day I feel satisfied, which I did on most days."

"Since the course, I haven't stopped teaching myself python and I'm improving a lot. I'm also putting some knowledge that I learned into my graduation project."

Based on the informal and formal feedback, we would state that the course was successful in providing the students with a meaningful introductory, yet comprehensive robotics experience.

Conclusions

We developed a four-week Robotics short course for International students from varying engineering backgrounds and various skill levels. Based on informal and formal feedback, we deemed the course successful in its ability to give these students an appropriate introduction to a complete robotics design experience. Although we were pleased with the outcome, we plan to implement several improvements for future offerings.

First, we will either increase the length of the course or reduce its content. Both the instructors and the students (based on survey responses) found the amount of material presented in the course was a bit too much for a four week period. A six week course would be more ideal to cover all of the desired material, but if that were not possible, the amount of material in the course could be reduced. One way to reduce the material would be to provide more cook-book solutions and starter code for all of the lab exercises, so that students could spend more time on making small modifications and tweaks to the code rather than developing code from scratch.

Second, we will expand the additional exercises offered the advanced students. One of the main strengths of our course was that it was able to keep the interest of students of varying backgrounds and skill levels primarily through the use of additional exercises for advanced students to extend the hardware and software capabilities of the robot. Expanding on the varying levels of exercises is crucial for keeping all of the students engaged.

Finally, we plan to focus more on the robot (the complete system). Students were more interested in making their robot do specific things rather than getting bogged down into the domain-specific details of its design (i.e. Linux OS, socket communications, mechanical parts selection, etc.). By giving students more cook-book solutions and starter code for all lab exercises, students can spend more time on the system aspects of the robot and its functions.

As we believe the course can provide a significant benefit to all students looking to get involved in the field of autonomous robots, we plan to offer the course again summer 2017 and hope to draw from its contents to create a technical elective course for our own undergraduate students.

Bibliography

- [1] F. B. V. Benitti, "Exploring the educational potential of robotics in schools: A systematic review.," *Computers & Education*, vol. 58, no. 3, pp. 978-988, 2012.

- [2] A. Melchior, F. Cohen, T. Cutter, T. Leavitt and N. Manchester, "More than robots: An evaluation of the first robotics competition participant and institutional impacts," *Heller School for Social Policy and Management, Brandeis University*, 2005.
- [3] G. Nugent, B. Barker, N. Grandgenett and V. Adamchuk, "The use of digital manipulatives in k-12: Robotics, gps/gis and programming," in *Frontiers in Education Conference*, 2009.
- [4] S. Grover, "Robotics and Engineering for Middle and High School Students to Develop Computational Thinking," in *Annual Meeting of the American Educational Research Association*, 2011.
- [5] W. W. Lau, G. Ngai, S. C. Chan and J. C. Cheung, "Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students," *SIGCSE Bull.*, vol. 41, no. 1, pp. 504-508, March 2009.
- [6] 15 January 2017. [Online]. Available: <http://www.firstinspires.org/>.
- [7] Adafruit Industries, "Adafruit Industries, Unique & Fun DIY Electronics and Kits," [Online]. Available: <https://www.adafruit.com/>. [Accessed 4 March 2017].
- [8] Amazon.com Inc., "Amazon.com: Online Shopping for Electronics, Apparel, Computer, Books, DVDs & More," [Online]. Available: amazon.com. [Accessed 4 March 2017].
- [9] Pololu Corporation, "Pololu Robotics and Electronics," [Online]. Available: <https://www.pololu.com/>. [Accessed 4 March 2017].
- [10] "Frontpage - Raspbian," [Online]. Available: <http://www.raspbian.org>.
- [11] w3schools.com, "PHP 5 Tutorial," [Online]. Available: <http://www.w3schools.com/php/>.
- [12] Adafruit, "BNO055 Absolute Orientation Sensor with Raspberry Pi & BeagleBone Black," [Online]. Available: <https://learn.adafruit.com/bno055-absolute-orientation-sensor-with-raspberry-pi-and-beaglebone-black>.
- [13] Adafruit, "Adafruit Ultimate GPS on the Raspberry Pi," [Online]. Available: <https://learn.adafruit.com/adafruit-ultimate-gps-on-the-raspberry-pi/setting-everything-up?view=all>.