1-1-2016

Connecting Hardware and Software in a Middle School Engineering Outreach Effort-RTP

Noah Salzman
Boise State University

Sin Ming Loo
Boise State University

ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #17232

Connecting Hardware and Software in a Middle School Engineering Outreach Effort-RTP

Dr. Noah Salzman, Boise State University

Noah Salzman is an Assistant Professor at Boise State University, where he is a member of the Electrical and Computer Engineering Department and IDoTeach, a pre-service STEM teacher preparation program. His work focuses on the transition from pre-college to university engineering programs, how exposure to engineering prior to matriculation affects the experiences of engineering students, and engineering in the K-12 classroom. He has worked as a high school science, mathematics, and engineering and technology teacher, as well as several years of electrical and mechanical engineering design experience as a practicing engineer. He received his Bachelor of Science degree in Engineering from Swarthmore College, his Master's of Education degree from the University of Massachusetts, and a Master's of Science in Mechanical Engineering and Doctorate in Engineering Education from Purdue University.

Prof. Sin Ming Loo, Boise State University

Connecting Hardware and Software in a Middle School Engineering Outreach Effort-RTP

Recent years have seen tremendous growth in outreach programs aimed at bringing computer programming to children and young adults via in-class and extracurricular coding activities. Programs such as the Hour of Code and Girls who Code have introduced millions of young people to programming around the world. For this study, we explored how combining programming with interactive electronics hardware can create a more engaging and dynamic learning environment for some students than what programming alone can achieve.

In this paper, we describe an electrical engineering outreach effort in collaboration with the technology and engineering teacher at a local middle school. Beginning with an introduction to programming via the Hour of Code, we progressed to lessons utilizing the Sparkfun Electronics Digital Sandbox, an Arduino-compatible microcontroller board with numerous built-in sensors and outputs. Under the guidance of both a professor of electrical and computer engineering and their own technology teacher, the students learned about the relationship between electronics hardware and software via a series of hands-on activities that culminated in a final design project.

To understand the experiences of the students who participated in these activities and develop insights into the relationship between hardware and software and students' learning outcomes, we administered a survey and conducted a focus group with the students. The students described an overall positive experience, and also appreciated the ability to connect coding with the interactivity provided by the microcontroller board. The students described deriving significant satisfaction out of relatively simple tasks like programming an LED light to blink or change color. The students also overwhelmingly felt that learning about the interconnections between hardware and software gave them an understanding and better appreciation of the complexity of the electronics and computer software they interact with on a daily basis. The students generally found the programming to be the most challenging part of the activity but also rewarding, but tended to indicate activities utilizing hardware as the most engaging activity they encountered.

Overall, the results of this study suggest that combined hardware and software educational activities can engage a wide number of students, help students understand the interconnectedness of these areas, and create a positive learning environment.

Introduction

In their 2010 report *Running on Empty*[1], the Association for Computing Machinery and the Computer Science Teachers Association describe a critical need to increase K-12 students' exposure to computer science and programming concepts to help these students learn the skills that they need to be successful in the 21st century. This call has been echoed by leaders from industry and academia, and has resulted in the development of numerous successful initiatives designed to introduce students to the basics of computer science and programming such as the Hour of Code[2] and Girls Who Code[3].

Similarly, multiple reports from the National Academy of Engineering and others have asserted the need for exposing K-12 students to engineering to help them develop 21st century skills, improve science and mathematics achievement, develop technological literacy, and inspire and prepare students to pursue careers in engineering[4–6]. This has resulted in the rapid growth of K-12 engineering curricula like Project Lead The Way, the International Technology and Engineering Education Association's Engineering byDesign, and extracurricular programs like FIRST Robotics.

Many of these engineering initiatives included significant programming components. Robotics competitions typically involve varying degrees of programming to control the robots and allow them to operate autonomously, while many K-12 engineering curricula and outreach programs include programming as part of students' explorations of embedded electronics design and development, many of which utilize the popular Arduino programming environment. Within the context of engineering, these activities provide students the opportunity to develop computational thinking skills that numerous experts recognize as an important learning outcome[7].

Despite the popularity of initiatives to develop computer science and programming skills and engineering activities that incorporate programming elements, relatively little is known about the relationship between programming and more hands-on electronics activities. To address this gap, we developed and assessed an outreach program at a local middle school focused on addressing the following research questions:

1) How does the combination of programming and electronics exercises and design activities affect student engagement when learning programming concepts?
2) How does learning about the relationship between programming and electronics influence students' understanding of commonly encountered technologies like smartphones and computers?

Method

The outreach activities described in this paper took place in the Spring of 2015 at South Middle School, an economically and racially diverse school serving the residents of Boise, Idaho. The activities were carried out in two 8th grade and one 9th grade technology classes. We worked with the school's Technology and Engineering teacher to integrate the activities in class two days a week over a period of 12 weeks.

The activities that we led with the students consisted of two parts. We first acquainted the students with basic programming concepts via The Hour of Code[2], an online introduction to programming developed in partnership with industry and education professionals that has introduced millions of students to the basics of writing code via simple exercises that utilize familiar popular culture references such as Minecraft or the characters from the Disney movie *Frozen* to engage students[8–10].

Following the Hour of Code, we dedicated several weeks to building on basic programming concepts and helping students understand the relationship between programming or software and hardware utilizing the Digital Sandbox, an Arduino-compatible development board with a series of accompanying educational activities developed by Sparkfun, Inc.[11] and shown in Figure 1. The Digital Sandbox combines a microcontroller with numerous input and output peripherals including LEDs, buttons, an analog slider input, a light sensor, and a microphone.

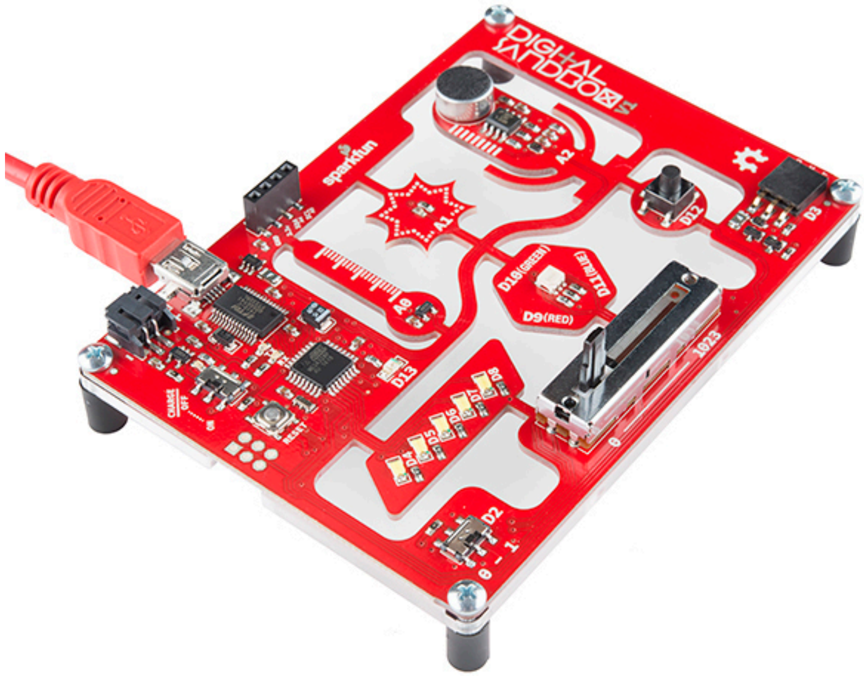

Figure 1: The Digital Sandbox development board. From https://cdn.sparkfun.com//assets/parts/9/3/5/0/12651-03.jpg

We chose this particular board because it allows students to easily experiment with hardware and software interfacing without requiring them to learn to wire components together and with a relatively low risk of damaging the hardware that can lead to frustration. The students learned to program the board using ArduBlock, an add-on to the Arduino microcontroller development environment that allows users to write programs using graphical programming blocks assembled like puzzle pieces as illustrated in Figure 2. The students completed a series of increasingly complicated activities using the Digital Sandbox and ArduBlock, ranging from blinking and dimming an LED using different loop structures to more complicated programs integrating multiple loops and if/then statements that utilized multiple input sensors and outputs. These activities culminated in a final open-ended design project where students developed their own programs that utilized multiple input and output elements of the Digital Sandbox development board. Example projects included security systems that triggered an alert based on sound or light activity, clapping to turn a light on, and a music note player,.

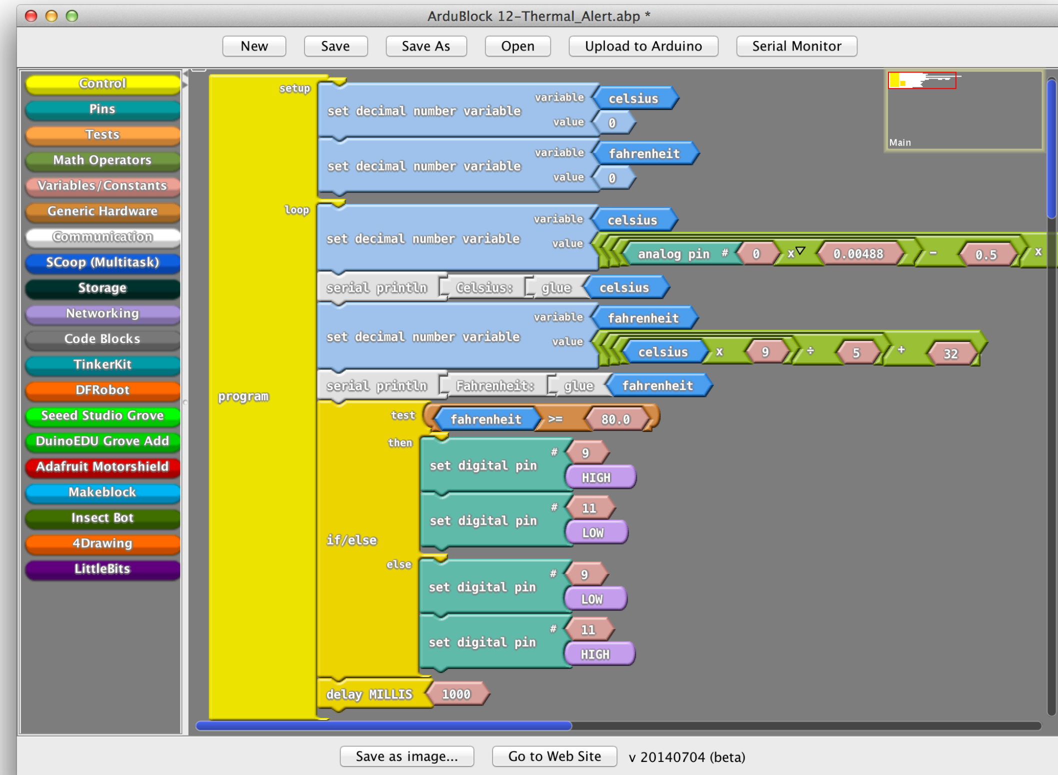

Figure 2: ArduBlock programming interface. From https://cdn.sparkfun.com/assets/learn_tutorials/2/6/1/ardublock_01.png

To assess the effectiveness of these efforts, we developed an online survey based on a combination of items from the Middle School Engineering Assessment developed by the Assessing Women and Men in Engineering Project[12], Likert-type questions related to the content included in the outreach program, and open response items where students indicated their favorite aspects of the project, the most challenging aspects, and any recommendations for changes. Following IRB approval and securing consent from the students and their parents, we administered the survey online after the outreach activities were done using Qualtrics survey software. These data were tabulated using Microsoft Excel, and we present the results in the following section.

In addition to the surveys, we also conducted a focus group with nine students selected by the teacher from multiple sections of the class. Questions for the focus group explored what the students learned, their favorite activities, their understanding of coding, and connections that they were able to make to technologies that they interact with on a daily basis like computers and smartphones. We recorded this conversation, and subsequently analyzed the recording over multiple times to identify themes in the students' responses.

Results

A total of 47 students responded to the survey, 32 in eighth grade, 14 in ninth grade, and 1 student did not provide this information. Table 1 shows the students' ratings of different aspects of the curriculum that they participated in. The results show that students were most interested in the activities that combined hardware and software, including blinking LEDs and working with the sensors and were least interested in the programming activities, with Hour of Code being the least popular activity.

Table 1: Participants' ratings of aspects of the outreach project

	Really didn't like it (1)	**Kind of didn't like it (2)**	**Kind of liked it (3)**	**Really liked it (4)**	**Mean**
Use Ardublock to Blink LEDs	1	1	13	32	**3.62**
Seeing electronic respond with code I wrote	1	3	11	32	**3.57**
Experimenting with different kinds of sensors	1	3	13	30	**3.53**
Able to see and touch hardware	1	5	10	31	**3.51**
Ardublock and Sandbox	3	3	11	30	**3.45**
Programming with Ardublock	2	2	20	23	**3.36**
Hour of Code	5	13	16	13	**2.79**

Table 2 shows the students' assessments of the overall impact of participation in the outreach activity. The most important impacts included helping students to understand how technology work and increased interest in engineering, while indicating that the activities had less of an impact on future course-taking plans or ability to be successful at school. These are consistent with the overall goals of our effort, which was focused more on developing specific coding and electronics skills and less specifically on developing interest in STEM careers or improving the participants' performance in school.

Table 2: Overall impact associated with participation in the outreach activity.

	Not at All (1)	Slightly (2)	Moderately (3)	A Great Deal (4)	**Mean**
Helped me to understand how technologies like phones and computers work	1	9	19	18	**3.15**
Helped me understand computers better.	1	6	29	11	**3.06**
Increased my interest in studying engineering in college.	1	15	15	16	**2.98**
Increased my confidence in my ability to participate in engineering projects or activities.	3	11	22	11	**2.87**
Made me think more about what I will do after graduating from high school.	8	7	19	12	**2.76**
Made me more confident in my ability to succeed in engineering.	5	9	25	7	**2.74**
Made me to decide to take different classes in school (including college) than I had planned	5	20	13	9	**2.55**

to.					
Made me decide to work harder in school.	7	19	15	6	**2.43**

Table 3 shows the participants' attitudes and skills related to science, math, engineering and their applications. The questions came directly from the AWE instrument and seem to mainly indicate a lack of alignment with the activities these items are being used to assess, but nonetheless provide some useful insights. These results indicate that this group of students has decent confidence in their ability to be successful at science and math, but do not see strong connections between the programming and electronics activities and their ability to be successful in these subjects.

Table 3: Participants' Attitudes and Skills

	Never (1)	Sometimes (2)	Very Often (3)	Always (4)	**Mean**
I can get good grades in science.	1	12	14	19	**3.11**
I can get good grades in math.	3	13	15	15	**2.91**
I know where I can find the information that I need to solve difficult problems.	1	15	22	8	**2.80**
I can explain math or science to my friends to help them understand.	5	15	11	15	**2.78**
In lab activities, I can use what I have learned to design a solution to a problem.	1	19	17	9	**2.74**
I can effectively lead a team to design and build a hands-on project.	3	20	13	10	**2.65**
I can use what I know to design and build something mechanical that works.	2	20	18	6	**2.61**
When I see a new math problem, I can use what I have learned to solve the problem.	4	21	14	8	**2.55**
I can use what I have learned to teach myself how to program a computer game.	11	17	13	5	**2.26**

Table 4 shows participants' interests related to STEM subjects and how they prefer to learn about these subjects. Overall, the participants indicated a strong preference for novel hands-on or project-based learning activities and less interest in math or science classes or activities.

Table 4: Participants' Interests

	Strongly Disagree (1)	Somewhat Disagree (2)	Somewhat Agree (3)	Strongly Agree (4)	**Mean**
More time should be spent on projects in science or technology activities at school.	2	0	21	23	**3.41**
I like learning how things work.	1	3	21	21	**3.35**
I like to learn to use new computer software.	3	6	15	22	**3.22**
Doing experiments in science class is frustrating.*	20	14	8	4	**3.09**
I would rather solve a problem by doing an	3	11	12	20	**3.07**

experiment than be told the answer.					
I am interested in learning more about how computers work.	3	9	21	13	**2.96**
Science is a difficult subject.*	14	15	9	6	**2.84**
I look forward to science class in school.	6	6	23	9	**2.80**
I get bored when I watch programs on channels like Discovery Channel, Animal Planet, Nova, Mythbusters, etc.	14	17	6	9	**2.78**
Science is too hard when it involves math.*	13	16	10	7	**2.76**
I feel comfortable with using a computer to make graphs and tables.	4	16	17	9	**2.67**
I look forward to math class in school.	7	14	13	12	**2.65**
I would like to (or already do) belong to a science or technology activities club.	11	16	11	8	**2.35**
I like to get science books or science experiments kits as presents.	17	12	12	5	**2.11**

* Indicates reverse-coded items

In addition to the previously described Likert-type items, the survey also included an open response section where we asked the students to describe their favorite portion of the outreach project, what they found most challenging, and any changes they would recommend. Similar to the results in **Table 1**, 18 of the 47 students who responded to the favorite question indicated they liked worked with the hardware. 14 of these students indicated programming as their favorite activity, and 8 indicated a combination of hardware and software. For the most challenging part of the activity, 20 of the students answered writing code, 10 mentioned specific exercises that they worked on, and 5 gave answers related to the logistics of completing the exercises such as not having enough time or difficulty getting enough attention from the instructor. Finally, when asked about what they would like to change, the majority of students responded indicated that they did not feel any change was necessary, while 8 students mentioned wanting more time, 6 wanted to work in different groups, five wanted more time with the hardware, and 4 indicated that the instructional materials were confusing or should be changed.

To further understand the students' experiences, we also conducted a focus group with 9 students to understand their experiences with the electronics and programming activities. Overall, the students who participated in the focus group described experiences and learning outcomes similar to what we have presented in the survey results. They indicated they also really liked working with the LEDs, because "you could create patterns, and got to choose from lot of options" and it "Feels good to understand and build something yourself."

The students also described how they were able to use what they learned to understand computers, phones, and tablets, such as "the blink indicator on your phone." They also said that "Arduino is a baby step for computers", and understood that "Technology works based on inputs and outputs and if/then statements. Complicated things are made up of simple commands." One student believed, "You could make a whole human artificial intelligence as just a combination of if this happens do this." They also understood coding and programming as "the language of computers", saying "It's like a sentence", or "A bunch of gibberish, like another language", but that they "Feel like they speak it a little more." Overall, the students in the focus groups indicated that it was a positive experience that helped them to understand both programming and electronics and begin to understand the complexity of modern technologies like smartphones and computer games.

Conclusions

Based on both the survey results and the focus group conversation, the integration of electronics and programming via a series of activities utilizing the Digital Sandbox microcontroller board represents a more engaging method of teaching programming skills to middle school students than purely computer-based programming activities. Participants consistently described activities that integrated electronics and programming as more engaging than those that focused exclusively on programming. The participants also were able to recognize the connections between the relatively simple programming and electronics design tasks they completed and more complicated technological artifacts incorporating hardware and programming elements such as smartphones and computers.

In the future, we hope to explore how learning embedded electronics and computer programming complement each other. In addition, this project involved both a significant time commitment to being in the classroom working directly with the students and relied on a low student to teacher ratio of 1 to 9. Even with this low ratio, students sometimes felt that it was difficult to get assistance in a timely manner. To address these issues of time commitment, we are interested in exploring how technologies like online tutorials and remote debugging sessions can be used to allow us to bring these activities to a wider audience while reducing the amount of time that university faculty need to be physically present in the classroom.

Acknowledgements

Thank you to Warren Hull at South Middle School for supporting our work, and thank you to the students in his technology classes for participating in this research.

References

1. Wilson, C., Sudol, L. A., Stephenson, C. & Stehlik, M. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age. ACM and CSTA*. (The Association for Computing Machinery and the Computer Science Teachers Association, 2010). at <http://runningonempty.acm.org/fullreport2.pdf>
2. Hour of Code. *Code.org* (2015). at <https://code.org/learn>
3. Girls Who Code. (2016). at <http://girlswhocode.com/>
4. National Academy of Engineering & National Research Council. *Engineering in K-12 education: Understanding the Status and Improving the Prospects*. (The National Academies Press, 2009).
5. National Academy of Engineering. *Standards for K-12 engineering education?* (National Academies Press, 2010).
6. National Academy of Engineering. *The Engineer of 2020: Visions of Engineering in the New Century*. (The National Academies Press, 2004).
7. Grover, S. & Pea, R. Computational Thinking in K–12 A Review of the State of the Field. *EDUCATIONAL RESEARCHER* **42,** 38–43 (2013).

8. Partovih, H. Transforming US education with computer science. in *45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014, March 5, 2014 - March 8, 2014* 5 (Association for Computing Machinery, 2014). doi:10.1145/2538862.2554793
9. Nikou, S. A. & Economides, A. A. Measuring student motivation during 'The Hour of Code' activities. in *14th IEEE International Conference on Advanced Learning Technologies, ICALT 2014, July 7, 2014 - July 9, 2014* 744–745 (Institute of Electrical and Electronics Engineers Inc., 2014). doi:10.1109/ICALT.2014.218
10. Garcia, D. D. *et al.* One-day activities for K-12 face-to-face outreach. in *46th SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2015, March 4, 2015 - March 7, 2015* 520–521 (Association for Computing Machinery, Inc, 2015). doi:10.1145/2676723.2677337
11. Huang, B. Digital Sandbox (Arduino Programmers Kit) – Curriculum Exchange. in 26.547.1–26.547.2 (2015). doi:10.18260/p.23885
12. Assessing Women and Men in Engineering (AWE) Project. Pre-College Implementer Guide. at <https://www.engr.psu.edu/awe/secured/director/precollege/MS_HS AWE Survey Impementer Guide.pdf>