Understanding High School Students' Reading, Remixing, and Writing Codeable Circuits for Electronic Textiles

Breanne K. Litts
Utah State University
Logan, UT
001-435-797-0142
breanne.litts@usu.edu

Yasmin B. Kafai, Debora Lui, Justice Walker & Sari Widman
University of Pennsylvania
Philadelphia, PA
001-215-746-3468
kafai@upenn.edu, deblui@upenn.edu, justicew@gse.upenn.edu, sawidman@upenn.edu

ABSTRACT
In this paper, we examine students' learning about computing by designing, coding, and remixing electronic textiles with sensor inputs and light outputs. We conducted a workshop with 23 high school students ages 16-17 years who learned how to craft and code circuits with the LilyPad Arduino, an electronic textile construction kit. Our analyses not only confirm significant increases in students' understanding of functional circuits but also showcase students' ability in reading, remixing and writing program code for controlling circuits. In our discussion, we address opportunities and challenges of introducing codeable circuit design for integrating maker activities that include engineering and computing into K-12 classrooms.

CCS Concepts
• Applied computing → Education → Interactive learning environments

Keywords
Assessment; Coding; Remixing; Circuitry; Electronic Textiles; Arduino; Maker Movement

1. INTRODUCTION
The push to promote computational thinking for all [26] has recently been joined by efforts to promote STEM topics in K-12 education [13; 24]. Maker activities in which youth participate in hands-on experiences have been seen as a particularly promising vehicle to engage youth in interdisciplinary STEM activities [9; 20; 21]. One such maker activity which combines engineering, crafting, and programming are electronic textiles (e-textiles), which are codeable circuits constructed with conductive thread, sewable LED's and sensors, and sewable microcontrollers, like the LilyPad Arduino [2]. Using e-textiles activities in classrooms and afterschool workshops has been shown to raise girls' interest in computing [11], women's engagement in the larger DIY community [3], and students' overall interest in science [7].

So far most of the research examining what students learn through making e-textiles has focused on assessing either students' understanding of simple functional circuits [e.g., 8; 18] or their learning of programming concepts [12]. We know little about students' learning at the intersection of crafting, circuitry, and coding with electronic textiles. Learning how to make codeable circuits involves students in designing and crafting a functional circuit that also can be controlled via code. For instance, students design a circuit that can turn particular LED lights on or off with a switch or when a sensor reads data above or below a set value. Such activities not only provide a rich demonstration of how to integrate electronics and computing with crafting and creativity in computer science education but also present complex challenges for the design of instruction and assessment.

In this paper, we examine how students can learn about reading, writing, and remixing code in designing circuits with programmable features. We conducted a four-week-long workshop during which 23 high school students (ages 16-17) learned how to make e-textiles to address the following research questions: (1) Does students' understanding of a functional circuit improve after completing a complex e-textiles project? and (2) Can students read, design, and remix codeable circuits after a complex e-textile project? Our analyses not only show significant improvements between pre/post tasks of students' understanding functional circuits but also showcase students' ability to read, design, and remix code for controlling circuits. In our discussion, we address how this type of circuit design can lead to a better understanding of functionality and provides a promising context for integrating maker activities in K-12 computing activities.

2. BACKGROUND
The research on examining learning with e-textiles has been most closely connected with prior work in science education that examined students' understanding and misconceptions of circuit design. Osborne's [17] seminal work found that elementary school students tend to struggle with understanding circuitry and typically generate linear representations rather than loop-based representations of circuits. Moreover, these misconceptions persist with high school or college-age students, even with instruction [e.g., 3; 13]. Issues like current flow and polarity are two of the most prevalent misconceptions in students' learning of simple circuitry and scholars often point to the abstract nature of traditional teaching models and learning materials as prime contributors.

More recently, circuitry teaching and learning has expanded to include other conductive materials such as play dough [10], circuit tape [21] or stickers [22], or conductive thread, sensors and microcontrollers such as the LilyPad Arduino [8] or Adafruit's Flora [25]. In one study, Peppler and Glosson [18] measured students' ability to create a functional circuit before and after students participated in an out-of-school e-textiles

SIGCSE '17, March 8–11, 2017, Seattle, WA, USA.

DOI: http://dx.doi.org/10.1145/3017680.3017740

workshop using stickers of LilyPad-specific components (an LED, a battery holder with battery, and a switch). Findings revealed significant increases in students' ability to create a working circuit and their understanding of current flow, connections, and polarity. Likewise, Halverson and colleagues [8] found that students learning of circuitry improved in a music-based maker activity in museum and classroom settings.

While these studies of electronic textiles designs have demonstrated promising findings in helping students to learn about simple and parallel circuits and furthered their understanding of key concepts, they did not address student learning of programming circuits controlled by microcontrollers. It is at this intersection between engineering and computing that the design and coding of circuits takes students into STEM application areas that integrate computational thinking [6; 26]. The design of circuits becomes foundational for the design of software (and vice versa) that controls the input/output interaction of sensors, lights, and motors. One study that has examined this intersectional dimension of electronics and computing found that college students engaged successfully in creative prototyping with modular electronics [23]. Others have made efforts to assess students' abilities to problem solve, troubleshoot, and integrate scientific and design principles with fabrication technologies (including microcontrollers), but do not go so far as to specifically study the learning of codeable circuitry [2]. No studies exist, however, that have examined K-12 students' understandings of codeable circuit design with modular electronics or electronic textiles.

In this paper, we investigate high school students' understanding of functional circuit and software design through a combination of pre/post tasks that examine two key aspects: (a) reading codeable circuit designs and (2) writing and remixing functional code for controlling circuits. What is both interesting and challenging about codeable circuits is that they integrate two different modalities—the visual architecture of the circuit and the written text of programs—and how these are connected via a microcontroller. In each instance, students have to be able to interpret and produce not only the blueprints of circuit designs but also the code for microcontrollers (in our case the LilyPad Arduino board), which collectively work to perform certain actions such as turning on or off LED lights in patterns or reading data from light or touch sensors and reacting accordingly. Furthermore, in designing or remixing code for circuit diagrams, the intersecting features become critical for how inputs and outputs can be controlled. For instance, LEDs in a parallel circuit design can only be turned on or off concurrently giving the designer no control over individual lights' functionality. We have chosen to examine different competencies such as reading, writing, and debugging both code and circuit designs to provide students with different contexts in which to showcase their understanding, fully realizing that these contexts vary in difficulty for novice student designers.

3. METHODS

3.1 Participants and Workshop Design

We conducted this study with a class of 23 students (4 boys, 19 girls, 16-17 years old), who were STEM majors at a public high school with a diverse demographic: 44% African American, 35% Caucasian, 13% Hispanic, 3% Asian, and 3% Multiracial students. All but six students had completed an introductory e-textiles project at the end of the previous academic year [4]. The participating teacher, a trained biologist, was also the STEM coordinator of the school. The teacher put students in pairs balancing skills and expertise, personality traits, and existing friendships. All students (N=24) consented to participate in the study, but one student transferred and another one was not able to complete all tasks.

The teacher worked together with our team to prepare and guide 15 workshop sessions each lasting about 90 minutes during which student pairs collaboratively constructed an interactive sign that would be exhibited in a high-traffic area of the school. Each student-pair received a letter printed onto canvas, which was designed by an art major student at the same school; collectively the letters constructed a sign of the school's name. Each pair also received a LilyPad Arduino, LEDs, sensors, switches, and other e-textiles materials to make the sign interactive.

3.2 Data Collection and Analyses

Individual student debriefing interviews were conducted by researchers and lasted about 30 minutes. In all of these students participated in a series of circuit and coding tasks. Two of these tasks were administered in pre- and post-interviews, while a third was included only in post-interviews. All interviews were videotaped and transcribed, and all artifacts such as student-generated diagrams and handwritten comments were captured.

The structure of Task 1, "Designing a Functional Circuit", was modeled after Peppler and Glosson's [18] and Halverson and colleagues [8] e-textile-based circuitry task that used stickers with lights, switches, and batteries and asked students to "draw a working circuit with a light and a battery." In our analysis, we focused on three aspects: polarity, connection, and current flow and coded each specific feature as 1 (present/correct) or 0 (not present/incorrect). We also coded (1-yes, 0-no) whether or not students were able to draw a functioning circuit (i.e., a circuit that could actually power a light). Due to the small sample size and binary coding scheme, we conducted McNemar's tests to determine significance of changes in students' understanding for each of the circuitry features.

In Task 2 "Reading a Codeable Circuit", we asked students to read a piece of Arduino code, which made an LED attached to a LilyPad blink continuously, a basic code pattern most students learned in their previous e-textiles project. We provided students with a piece of paper presenting this code and a circuit diagram that featured an LED attached to a LilyPad. We asked students two questions: (1) "Do you have a sense of what the function of this code might be?" and (2) "Do you think this code would work with the following LilyPad circuit, why or why not?" Answers were recorded on video and then transcribed. The coding scheme for this task focused on students' understanding of specific parts of the program by explaining: (1) 'HIGH' means on and/or 'LOW' means off; (2) 'delay' as a command that controls time; (3) 'void loop' indicating a continuous or repeated action; (4) inputs/outputs (e.g., LED as an output); (5) how variables were named in the code; and (6) whether the program as a whole would work with the circuit shown (for more detail, see section 4.2). After coding students' answers, each student received a score ranging between 0 and 6, with a higher score reflecting a better understanding. Due to the small sample size and ordinal coding scheme, we conducted a Wilcoxon two-sample paired signed ranks test on these data to determine significance in changes in students' understanding.

At the end of the workshop, we gave students Task 3, a codeable circuit task that required reading, writing, and remixing code with a circuit diagram for the LilyPad Arduino. For Task 3a "Reading a Complex Codeable Circuit", we relied on the coding scheme for the Task 2 as our basic model but changed the last feature. We coded for students' ability to read the given switch and sensor-based program based on their ability to explain different program components such as variables, data input and output, and loops. We also scored students on their ability to understand the conditional if/then commands within the program—a more complex programming feature that was not present in the more basic program presented in Task 2 (for more detail, see section 4.3.2). This coding scheme resulted in each student receiving a number ranging between 0 and 6, with a higher score reflecting a better understanding of the program.

The coding scheme for Task 3b "Remixing a Complex Codeable Circuit" was based on students' ability to remix given lines of the code to shift the program output from asynchronously blinking LEDs to synchronously blinking LEDs. Here students needed to make the following changes in the program code: (1) move together the 'digitalWrite HIGH' lines, (2) move together the 'digitalWrite LOW' lines, and (3) move or eliminate the 'delay' lines from the program (for more detail, see section 4.3.3). Each of these changes was coded as either 1 (present/correct) or 0 (not present/incorrect), with the maximum score for this task being 3 points.

Finally, in Task 3c "Designing a Complex Codeable Circuit," students needed to design a circuit with four components (a LilyPad, two LEDs, and a switch) to match the given switch-based program, which included a conditional statement (if-then-else) and custom functions. In Task 3b students needed to read and explain this given program while in Task 3c students needed to remix parts of this code to change the behavior of the circuit from blinking asynchronously to blinking synchronously (for more detail, see section 4.3.1). Students responded to questions either verbally or by writing down their answers (including drawings, comments, and code), and these data were video-recorded, transcribed, and documented accordingly. The analytic coding scheme focused on students' ability to design a codeable circuit based on an existing program, and included three parts each with specific features such as connections, grounding, and current flow that we coded as 1 (present/correct) or 0 (not present/incorrect. We also coded (1-yes, 0-no) if the circuit as drawn would indeed work with the existing program.

4. FINDINGS

We first present results of students' ability in designing a functional circuit, highlighting circuitry concepts in which students demonstrated their greatest improvements. Second, we share results regarding students' ability in reading a codeable circuit. Finally, we describe results of students' ability to read, design, and remix a codeable circuit after participating in the e-textiles workshop during which they designed similar codeable circuits.

4.1 Designing Functional Circuits

We first sought to determine whether students significantly improved their ability to draw, label, and explain a working circuit diagram. Like in previous studies [8; 22], we found that students' ability to draw a working circuit significantly increased ($p<.05$, $p=.000$) from pre- to post-task. Specifically, after the e-textiles workshop, 78% (18) of students were able to draw a working circuit diagram whereas only 26% (6) were able to do so before the workshop. Furthermore, we examined what elements of a circuit demonstrated the biggest areas of improvement and found that students significantly improved their understanding of matching polarity ($p<.05$, $p=.002$). Students also significantly increased their understanding of circuitry as a loop ($p<.05$, $p=.004$) by significantly reducing the number of missing connections they had in their circuit ($p<.05$, $p=.021$). These results confirm findings from prior studies [8; 22] demonstrating that students can learn about circuits in e-textile activities.

4.2 Reading Codeable Circuits

We also investigated students' ability to understand the relationship between circuit design and program code by explaining a given code and diagram (see Figure 1). In this task, the Arduino code generates a basic blinking pattern, turning on and off the LED attached to the LilyPad with a repeated delay of 1000 milliseconds.

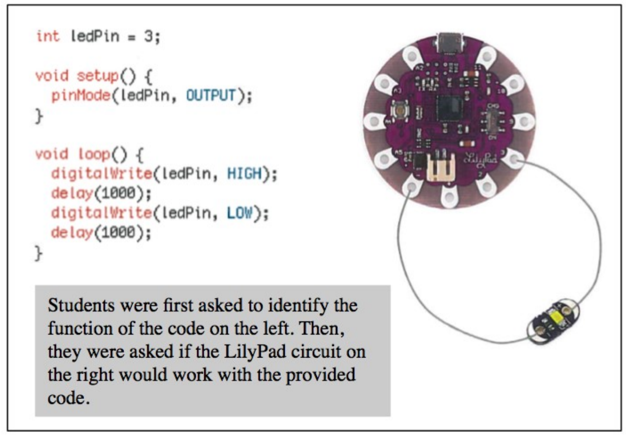

Figure 1. Reading Codeable Circuit.

We found that students' ability to read code to control a circuit improved after the workshop: while only one student in the pre-task got a score between 4 and 6, indicating high understanding, over 56.5% (13) of students reached this level in the post-task. The average score on the pre-task was 1.13 ($N=23$) and on the post-task it jumped to 3.35 ($N=23$) indicating that students significantly improved in their ability to decipher a program and circuit design. A Wilcoxon two-sample paired signed ranks revealed that the median post-assessment scores were significantly higher than median pre-assessment scores ($Z=-3.7364$, $p<0.01$).

4.3 Developing Codeable Circuits

The last task focused on students' ability to read, write, and remix a codeable circuit, thus giving us better insights into students' understanding of the relationship between software design and the circuit design for the LilyPad Arduino. As compared to the basic blink program provided in the last task, this program code was more complex since it contained a conditional statement (if-then-else) with a switch, control of multiple LEDs, and custom functions.

4.3.1. Reading a Complex Codeable Circuit

In this task, we asked students to read and explain program code that controlled a LilyPad Arduino and components. Students' overall average score for this task was 3.45 ($N=22$) out of 6 with a distribution as follows: no students scored a 0, 4.5% (1) of students scored a 1, 31.8% (7) scored a 2, 18.2% (4) scored a 3, 13.6% (3) scored a 4, 22.7% (5) scored a 5, and 9.1% (2) scored a 6 indicating the highest understanding. Students demonstrated the greatest understanding of HIGH/LOW as on/off (95.7% or 22 of 23), and the lowest understanding of both loops and conditional (if-then-else) statements (39.1% or 9 of 23 students). This finding illustrates that students' grasp of the specific features present in the code was still not very well developed after the workshop. One possible explanation is that because of the pair working arrangement, some students focused more intensely on coding and learned about the more complex aspects of the program (i.e., the conditional statement, inputs/outputs), while others focused on crafting or circuit design. Additionally, as seen in the task that involved reading a codeable circuit program (see Section 4.2), it was difficult to develop a consistent coding scheme because students' answers varied in specificity. For instance, many students may have delivered succinct answers without a step-by- step breakdown of the commands. In this way, the scores may have not accurately reflected students' understanding, but instead the comprehensiveness of their explanations.

4.3.2. Remixing a Complex Codeable Circuit

We also investigated students' ability to remix existing program code to change the behavior from synchronous to asynchronous blinking for a codeable circuit. Students' average score was 2.09 (N=22) distributed as follows: 40.9% (9) scored 3, the highest score, followed by 31.8% (7) scored 2, and then 22.7% (5) scored 1. We rated students on whether or not they made the following changes in the program code: (1) move together the 'digitalWrite HIGH' lines, (2) move together the 'digitalWrite LOW' lines, and (3) move or eliminate the 'delay' lines from the program. While more students indicated that they would move the 'digitalWrite HIGH' lines or 'digitalWrite LOW' lines together (19 or 86.4%, and 16 or 72.7%, respectively), only half indicated an interest in moving the delays (11 or 50%). These findings indicate that students had developed a basic understanding of program code to change functionality.

4.3.3. Designing a Complex Codeable Circuit

In this final task, we examined students' ability to design a working circuit that matched components such as two LED lights and switch with a pre-existing program, which includes a conditional statement (if-then-else) using the switch and custom functions. This task required students to make connections between the different components and the LilyPad, to ground the circuit, and facilitate current flow (see Figure 2). In terms of connections, students were able to create looped connections for each component (one connection each from the positive and negative ends): 91.3% (21) of students did this for the LEDs and 78.2% (18) did this for the switch. Students were also able to connect the positive end of the components to the correct LilyPad pin based on their understanding of the program: 78.2% (18) of students were able to properly connect the LED to pin A5, 82.6% (19) were able to properly connect the LED to pin 11, and 87.0% (20) were able to properly connect the switch to pin 9. However, students had more difficulty connecting the negative pole of each component appropriately in order to properly ground the circuit: 73.9% (17) of students were able to ground the left LED, 65.2% (15) of students were able to ground the right LED, and only 52.2% (12) of students were able to ground the bottom switch.

We also categorized students' different techniques for grounding the overall circuit. Here we found that grounding the components was accomplished by one or more techniques including: 73.9% (17) of students by connecting them directly to the negative pin of the LilyPad, 34.8% (8) of students by connecting to another component (which was eventually connected to the negative pin), and 0.04% (1) student by connecting it to negative connecting line. It should be noted here that the latter two techniques are considered more advanced, since they are an efficient use of thread and sewing. Interestingly, 17.4% (4) of students attempted to ground components by connecting them to undesignated pins on the LilyPad. While this would not have worked with the pre-existing program, it possibly reflected conversations that we had in the workshop about this particular technique of writing extra code in order to program a pin to be negative.

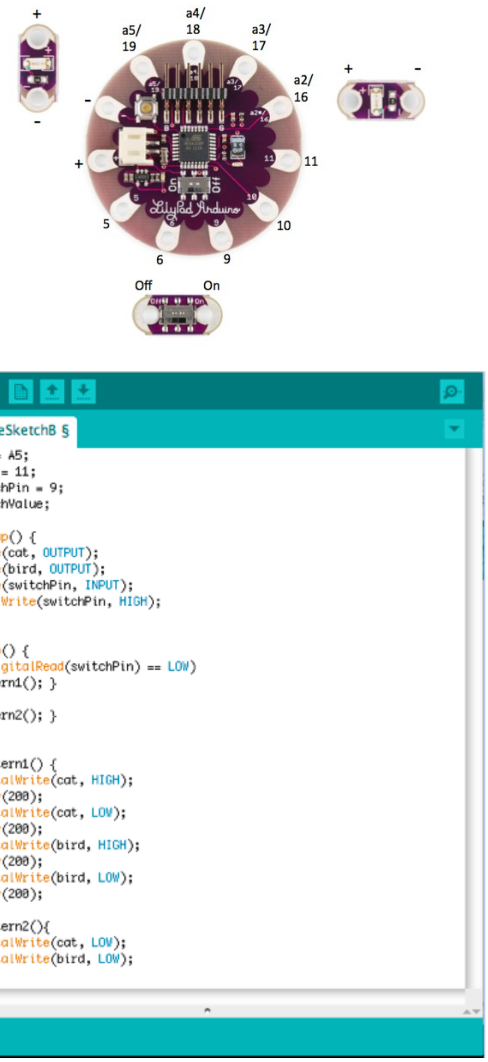

Figure 2. Designing a Codeable Circuit.

Finally, we examined students' ability to design a circuit that would ultimately work with the pre- existing program by looking at specific wiring characteristics of the circuit. Overall, 39.1% (9) of students created a circuit that would fully work with the existing code, meaning that all the intended actions written into the program (for the two lights and the switch) would function. In some cases (6 students, or 26.1%), students' circuit diagrams would work for some but not all components, so we did not rate these as working. Regarding the mistakes, 39.1% (9) of students were missing connections within the entire circuit and 8.7% (2) of students had redundant lines. However, none had crossed lines or short circuits.

5. DISCUSSION

The findings illustrate that after completing the e-textiles workshop, students were able to design functional circuits and understand more complex codeable circuits. Codeable circuits are a unique type of design in which both the blueprint of the circuit design and the control structure of the code must align in order for the LEDs, sensors, and switches to perform desired behaviors. Thus far, only college students have demonstrated learning this intersectional knowledge required for codeable circuits [23] and our study sheds light on high school students' learning in this area. In the following sections we discuss the opportunities and challenges in understanding and assessing this type of intersectional learning in computing activities.

5.1 Understanding Computing

Successfully working with codeable circuits requires a grasp of both circuit and software design, which makes these types of design activities particularly challenging for students working with e-textiles. In our study, nearly all students were able to read codeable circuits while some students were also able to successfully design and remix codeable circuits. This means that most students understood the interconnectedness of circuit and software design enough to interpret it, but only some were able to apply this knowledge by producing new code. It is possible that the pairing of students in teams to work collaboratively on their e-textile projects contributed to this result. We noticed that about half the students took on the role of 'coder' in their teams and this imbalance could have contributed to one group of students developing a better understanding in remixing and writing code.

Furthermore, we suspect that disciplinary knowledge is interdependent in e-textile activities. This dimension of computing learning can be especially challenging when students need to connect concepts from different disciplines, such as engineering and computing, with which they are often unfamiliar and which are rarely introduced together. Understanding one disciplinary area (e.g., circuitry) could be supportive of learning other areas (e.g., coding or design) and vice versa. We have some evidence supporting this, specifically from the interviews, when students made references to circuit diagrams when discussing their program code, thus providing an answer that integrated both visual and textual elements. This intersectional nature of e-textiles poses significant challenges to computing teachers in K-12. To support these interdisciplinary activities in the classroom, teachers will likely need to gain some familiarity with both disciplines. We need to investigate how teachers can provide differentiated support for students and teams often at various completion points in their projects.

5.2 Developing Assessments

Bringing e-textiles into the classroom also presents challenges for developing assessments to understand what students have learned. The assessment tasks developed by Peppler and Glosson [18] and Halverson and colleagues [8]—whose findings we replicated in this study—captured a particular aspect of circuit design knowledge in e-textile activities. Their assessments tools, though, primarily look at students' ability to understand and design simple functional circuits. In this workshop we pursued more complex projects involving codeable circuits, which required students to understand of multiple domains, namely the visual design of a circuit along with writing code. We assessed students' understanding of codeable circuits through activities that differed in complexity: from the ability to read a codeable circuit design to remix a codeable circuit with given components, and eventually to write code to capture this interdisciplinary knowledge and skill.

Within the design of our assessments, we incorporated elementary circuit design with various computational practices such as reading, writing, and remixing code. These tasks were designed to highlight the integrated nature of the activity in the following two ways. First, the visual presentation of the tasks was multimodal; that is, students were simultaneously presented with the text of a program alongside a LilyPad circuit design, they did not see one without the other. In this way, the presentation reinforced their integration. Second, while the tasks themselves focused on one mode over another (e.g., asking students to design a circuit, or asking them to explain or remix the code), the most correct answers depended upon their understanding and ability to manipulate both modes. Thus, our design of these tasks expands the growing body of assessment tools in the computational making literature with an eye toward its interdisciplinary, multimodal nature.

5.3 Moving Forward

The present study contributes to the growing body of research exploring learning of electronics and computing in maker activities and designing tasks to assess student learning. We could further the development of assessments by providing debugging activities which present students with faulty features of codeable circuit designs. This assessment approach has the benefit that we can control the difficulty of the bugs and present different scenarios ranging from simple syntax problems to more complex control structure issues in addition to varying circuit designs and functionality. We have already experimented with debugging activities in prior work where we crafted faulty e-textiles and then asked student teams to fix the problems [5]. We found that such e-textile debugging activities were more true-to-life, moving away from paper-and-pencil tasks we used in this study, and thus provided a more authentic context. We also found that students enjoyed the collaborative problem solving of a constrained task within the context of a larger self-directed e-textile project. They considered it a learning opportunity to put to test their newly developed understanding of circuit design and coding skills. We think that these are fruitful avenues to pursue for the development of authentic learning assessments.

In previous work, we also investigated the collaborative dimensions of computational maker activities by examining how student teams leveraged distributed expertise to mitigate the challenges of these complex design tasks [15]. In this study, we had students work in smaller teams, pairing two students. Our preliminary findings suggest that student pairs were better able to rely on each other's strengths to troubleshoot and debug problems in e-textile projects. Building on this work, we should continue to explore how to best leverage collaborative learning arrangements as possible solutions to support learning with maker activities in computer science classrooms. Another way to support student learning is through designing technologies that support interdisciplinary and multimodal learning. For instance, Modkit [16] connects visual diagrams with code by providing visual programming software for the LilyPad Arduino. The tradeoff with this specific tool, however, is that it eliminates the more authentic computer science practice of text-based coding. We need additional research in this area focused on designing and developing tools and resources to support the interdisciplinary, multimodal learning that such computing activities require.

6. ACKNOWLEDGMENTS
This work was supported by a grant (#1509245) from the National Science Foundation to Yasmin Kafai, Jane Margolis and Joanna Goode. Any opinions, findings, or conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation, Utah State University or the University of Pennsylvania.

7. REFERENCES

[1] Blikstein, P., Kabayadondo, Z., Martin, A., & Fields, D. (in press). The Exploration and Fabrication Technology Index (EFTi): Assessment of unexplored technology and engineering literacies in makerspaces and educational FabLabs. *Journal of Engineering Education*.

[2] Buechley, L. (2006). A construction kit for electronic textiles. In: *The proceedings of the 10th IEEE international symposium on wearable computers* (pp. 83-90). New York, NY: IEEE.

[3] Buechley, L., & Hill, B.M. (2010). LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. *Proceedings of the 8th ACM conference on designing interactive systems* (pp. 199-207). New York, NY: ACM.

[4] Duit, R., & von Rhöneck, C. (1998). Learning and understanding key concepts of electricity. In: Tiberghien A, Jossem EL, Barojas J (Eds), Connecting research in physics education with teacher education. *International Commission on Physics Education.* Available at https://www.univie.ac.at/pluslucis/Archiv/ICPE/TOC.html

[5] Fields, D. A. Searle, K., & Kafai, Y. B. (2016). Deconstruction kits for learning: Students' collaborative debugging of electronic textile designs. In *Proceedings of the 2016 FabLearn Conference*. New York, NY: ACM.

[6] Grover, S., & Pea, R. (2013). Computational thinking in K-12: a review of the state of the field. *Educational Researcher,* 42(1), 38-43.

[7] Gu, J., Tofel-Grehl, C., Fields, D. A., Sun, C., & Maahs-Fladung, C. A. (2016, April). Sewing electronics in class: Improving middle school students' science learning. Paper presented at the annual meeting of the American Education Research Association Annual Meeting. Washington, DC: AERA.

[8] Halverson, E., Lawson, C., Fleuchaus, E., Bakker, M., & Saplan, K. (2016). Learning circuitry through making. Paper presented at the annual meeting of the American Educational Research Association. Washington, DC: AERA.

[9] Honey, M., & Kanter, D.E. (Eds) (2013). *Design, make, play: Growing the next generation of STEM innovators.* New York, NY: Routledge.

[10] Johnson, S., & Thomas, A.P. (2010). Squishy circuits: a tangible medium for electronics education. *Proceedings of CHI 2010* (pp. 4099–4104). New York, NY: ACM.

[11] Kafai, Y. B., Fields, D., & Searle, K. (2014). Electronic textiles as disruptive designs: supporting and challenging maker activities in schools. *Harvard Educational Review 84(*4), 532-556.

[12] Kafai, Y. B., Searle, K. A., Fields, D. A., Lee, E., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: introducing computational concepts, practices and perspectives with E-textiles. *Transactions on Computing Education 14(*1), 1-20.

[13] Katehi, L., Pearson, G., & Feder, M. (Eds) (2009). *Engineering in K-12 Education: Understanding the status and improving the prospects.* Washington, DC: National Academies Press.

[14] Kock, Z.J., Taconis, R., Bolhuis, S., & Gravemeijer, K. (2013). Some key issues in creating inquiry-based instructional practices that aim at the understanding of simple electric circuits. *Research in Science Education*, 43(2), 579-597.

[15] Litts, B. K., Kafai, Y. B., & Dieckmeyer, E. (2015). Collaborative electronic textile designs by high school youth: Challenges and opportunities in connecting crafts, circuits, and code. In: The proceedings of the 2014 ACM SIGCHI FabLearn conference on creativity and fabrication in education.

[16] Millner, A., & Baafi, E. (2011). Modkit: Blending and extending approachable platforms for creating computer programs and interactive objects. *Proceedings of the 10th annual ACM SIGCHI conference on Interaction Design and Children* (pp. 250-251). New York, NY: ACM.

[17] Osborne, R. (1983). Towards modifying children's ideas about electric current. *Research in Science & Technological Education,* 1(1), 73-82.

[18] Peppler, K., & Glosson, D. (2013). Stitching circuits: Learning about circuitry through e-textile materials. *Journal of Science Education and Technology,* 22(5), 751-763.

[19] Peppler, K., Halverson, E., & Kafai, Y. B. (Eds) (2016a). *Makeology: Makerspaces as learning environments (Volume 1)*. New York, NY: Routledge.

[20] Peppler, K., Halverson, E., & Kafai, Y. B. (Eds) (2016b) *Makeology: Makers as learners (Volume 2)*. New York, NY: Routledge.

[21] Qi, J., & Buechley, L. (2014). Sketching in circuits: designing and building electronics on paper. *Proceedings of the 2014 ACM SIGCHI conference on human factors in computing systems* (pp. 1713-1722). New York, NY: ACM.

[22] Qi, J., Huang, A.B., & Paradiso, J. (2015). Crafting technology with circuit stickers. *Proceedings of the 13th annual ACM SIGCHI international conference on interaction design and children* (pp. 438-441). New York, NY: ACM.

[23] Sadler, J., Shluzas, L., & Blikstein, P. (2016). Building blocks in creative computing: modularity increases the probability of prototyping novel ideas. *International Journal of Design Creativity and Innovation*, *3*(1), 1-17.

[24] Smith, M. (2016, January 30). Computer Science for All. Retrieved from https://www.whitehouse.gov/blog/2016/01/30/computer-science-all

[25] Stern, B., & Cooper, T. (2015). *Getting started with Adafruit FLORA: Making wearables with an Arduino-compatible electronics platform.* San Francisco, CA: Maker Media.

[26] Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.