TechTrends (2018) 62:197–203
https://doi.org/10.1007/s11528-017-0248-3

ORIGINAL PAPER

Paper Circuits: A Tangible, Low Threshold, Low Cost Entry to Computational Thinking

Victor R. Lee[1] · Mimi Recker[1]

Published online: 9 January 2018

Abstract
In this paper, we propose that paper circuitry provides a productive space for exploring aspects of computational thinking, an increasingly critical 21st century skills for all students. We argue that the creation and operation of paper circuits involve learning about computational concepts such as rule-based constraints, operations, and defined states. Moreover, paper circuitry materials are low cost, provide a low threshold to entry, and draw upon the familiarity that already exists with respect to paper as a hands-on and interactive medium. Paper circuitry thus provides multiple points of entry for students who are unfamiliar with computational thinking ideas while also supporting creative, artistic and crafting activities. It also provides an important alternative to the typically steep learning curve associated with learning a programming language. We define paper circuitry and associated technologies, show how they afford key dimensions of computational thinking, and present examples of paper circuit projects created by students.

Keywords Computational thinking · Paper circuitry · Art and design · Tangible computing

Introduction

Computational thinking and programming skills have been increasingly identified as critical 21st century skills for all students (di Sessa 2000; Guzdial 2008). Wing (2006), one of the most cited advocates, defined computational thinking as "solving problems, designing systems, and understanding human behavior, by drawing on concepts fundamental to computer science" (p. 33). More recently, a consortium comprised of U.S. educational non-profit organizations and technology companies released the *K-12 Computer Science Framework Steering Committee* (2016). It defines computer science standards for students such as "developing and using abstractions, collaborating around computing, creating computational artifacts, testing and refining computational artifacts, and communicating about computing." Finally, in exploring how computational thinking specifically aligns with mathematics and science instruction, Weintrop et al. (2016) identified four areas: data practices, modeling and simulation practices, computational problem solving practices, and systems thinking practices. All of these researchers emphasize the idea that computational thinking is evident in many different places and spaces, and not just in computer programming languages.

The notion that computational thinking transcends coding is an important point for educational technologists and computing educators. Too often, the complexities associated with writing text-based computer code and the need for precise syntax, rules, and formalisms create the sense for many students that computational thinking is difficult and has a steep learning curve. For those who maintain the core constructivist commitment that new understandings build upon prior ones, it is important for us to identify where and how forms of low-threshold computational thinking already exist and can be encountered. For instance, Berland and Lee (2011) showed that collaborative strategy board games provide a rich space for engaging in computational thinking, by way of allowing for simulations, debugging, and algorithm development. This has since been extended to other tabletop games, such as Mancala, which exhibit similar properties (Phelps et al. 2017). An entire computing curriculum dedicated to teaching computing ideas without computers, *CS Unplugged*, has also

✉ Victor R. Lee
victor.lee@usu.edu

Mimi Recker
mimi.recker@usu.edu

1 Department of Instructional Technology and Learning Sciences, Utah State University, 2830 Old Main Hill, Logan, UT, USA

been developed (Bell et al. 2009). Eisenberg (2010) has also demonstrated how computation can also be embodied and expressed through craft objects, such as beads.

We propose that *paper circuitry* provides another productive space for exploring aspects of computational thinking. Like other *computer-less* computational thinking environments, the creation and operation of paper circuits involves rule-based constraints, operations, and defined states. Moreover, as explained below, paper circuitry materials are low cost, provide a low threshold to entry, and draw upon the familiarity that already exists with respect to paper as an interactive and manipulable medium (Shorter et al. 2014). This enables hands-on, creative, artistic, and crafting elements to be foregrounded in which people learn through physical and intellectual engagement in the world (Lee and Fields 2017; Peppler 2013).

In this way, paper circuitry provides multiple points of entry for students who are less familiar with computational thinking ideas. Moreover, as shown below, paper circuit activities can align to the emerging interest-driven emphasis of Connected Learning (Ito et al. 2013) as well as have a more constructionist orientation (Papert 1980). Finally, paper circuitry provides an important counterpoint to the typically steep learning curve associated with learning a programming language.

In this paper, we begin by defining paper circuitry and associated technologies, show how it affords key dimensions of computational thinking, and present examples of teenage students participating in after school programs who became deeply engaged in paper circuitry projects. We conclude with a discussion of future trends and technologies in this area.

What Is Paper Circuitry?

To build a simple paper circuit, all that is needed is a 2-dimensional surface (e.g., paper), conductive tape (e.g., copper), a low-voltage power source (e.g., 3-volt coin cell battery), and a component that takes power (e.g., an LED). This simple set-up can be purchased for a group of 20 participants for less than US$1.50 per person (Williams 2017).

The conductive tape is used to create an electronic **circuit** from the power source through the component. Figure 1 shows a simple circuit that lights up an LED once completing, thus providing immediate feedback on the intended goal state. In its simplest form, there is nothing hidden with respect to the circuit structure. All connections are plainly visible and manipulable, as are the power source and additional components. This is in contrast to what may be buried in the microscopic circuitry of a fabricated silicon logic board. A paper circuit construction can hide some of the circuitry by keeping the copper tape or battery on the underside of a paper or by layering sheets of paper, and doing so enables new aesthetic considerations.

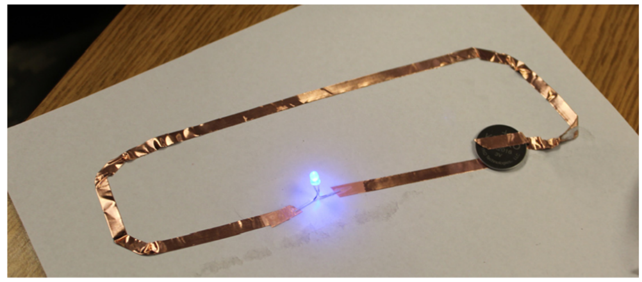

Fig. 1 A basic paper circuit

From that point, designs can quickly become more complex, by integrating more elaborate circuitry (in series or parallel fashion), with on/off switches and different components (e.g., sensors of sound, light, pressure). Designs can also draw upon artistic elements, as students add drawing, color, etc. (see Fig. 2; for more sophisticated examples, see the paper circuit-based crafts produced by Qi and Buechley 2010, 2014). With the immediate feedback provided by the components and the visibility of the circuit structure, students can quickly understand if their circuit is working correctly or not. If not, they need to engage in debugging activities to determine where their logic is faulty. The scale of materials and tangibility make systematic testing and identifying problems more transparent and approachable.

The precise materials suggested here are not the essential ones, in that other forms of metal tape, wire, aluminum foil, or conductive ink could be used as could other power sources and components. However, copper tape is already highly conductive and easily purchased online or at hardware stores, and the low amount of energy from a coin cell battery keeps materials safe and inexpensive.

What Aspects of Computational Thinking are Afforded by Paper Circuitry?

We do not wish to claim that paper circuitry integrates all aspects of computational thinking. Instead, we believe,

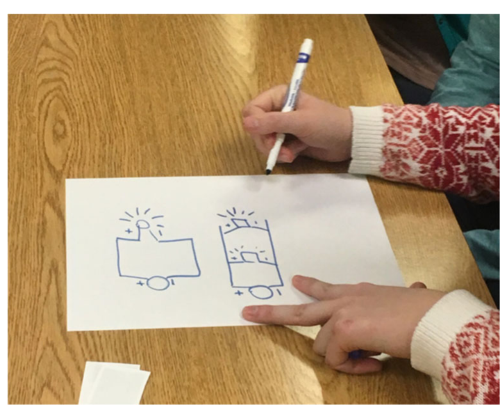

Fig. 2 Circuit diagrams that formalize spatial configurations for wire and components

that when well-implemented, paper circuitry and similar kinds of tangible circuitry activities such as e-textiles (Kafai et al. 2014; Peppler 2013) can expose students to productive thinking around several critical computational aspects (Grover and Pea 2013; Kafai and Burke 2014b; Weintrop et al. 2016). These include:

Algorithmic thinking involves creating a step-by-step plan that can be implemented by an artifact. For instance, creating a greeting card that lights up when opened or a succession of lights that will be lit as a complex set of switches involves creation of an algorithm for on and off states.

Conditional Logic involves *if-then* relationships in which binary states are critical in computation. With paper circuits, this often is expressed with switches that make a component turn on/off, but can also embed truth table structures such as XOR, for example when a pivot switch (a paper switch secured by a brad, for example) can be set to only allow for only one or another light to be operational.

Symbol Systems and Representations which typically involves defining variables, articulating data structures and abstractions, and writing code. In this context, it could involve sketching the planned paper circuit project as a simplified circuit diagram (Fig. 2) that is organized to manage given constraints (such as working only when the paper is folded or having circuits covered in a particular area).

Debugging involves systematic attempts to adjust a procedure to find the errors ("bugs") that are preventing a system from running properly. With paper circuits, this typically involves checking to see that the LEDs and battery are oriented the proper way, that all components are individually operational, and that connections are complete and without break or overlap.

Iterative and parallel thinking is a way of thinking about computational architectures but with paper circuits can be explored through the use of serial and parallel circuit designs and an examination of how electricity travels through these different configurations.

These computational aspects are present because paper circuits can be seen as modeling Boolean gates. Essentially, they turn off and on. An advantage of paper circuits over other forms of circuitry is that unlike the use of a breadboard or alligator clipped wires, mapping circuit structure onto a two-dimensional surface makes connections all simultaneously visible and easily traceable spatially.

In addition, as youth typically have past experience working with paper, the opportunities to embed circuits to create novel crafts are better foregrounded. This aesthetic exploration and expression is arguably a critical aspect of computational thinking that often gets missed when computer coding is made the dominant focus of instruction.

What Kinds of Computational Artifacts can Youth Create with Paper Circuits?

We recently offered several informal clubs and after-school activities in library settings for teenage youth around creating paper circuits, where our primary goal has been to identify and develop supports for librarians in these settings to enable them to enact new forms of educational programming that go beyond textual and information literacy emphases (Lee et al. 2017). As they were offered in an informal learning environment, the clubs were structured in such a way that minimal formal instruction was provided. Instead, examples of paper circuits, 1-page support handouts with pictures and diagrams showing some common tips, and the librarian were available to help guide youth. They also had access to the necessary materials (paper, some writing and drawing materials, scissors, LEDs, copper tape, and batteries). The clubs were offered on a drop-in and informal basis and lasted for one to two hours over the course of several weeks. While the goal of the overarching project and the current paper is not to provide an assessment of youth learning, we have been extensively documenting the development and implementation of these clubs through field notes and interviews (Lee et al. 2017). Below we describe the basic activity structure and some specific projects and activities.

Common Activity Structure

Within the library context, we have observed that youth often expect to participate on a drop-in basis. Furthermore, unless it is a school library doing planned instruction during designated class time, youth wanted as minimal formal instruction as possible. When lengthier presentations (exceeding more than a few minutes) were provided, we noticed students losing focus and talking with their friends or focusing on another activity. Conversations with students and with librarians reflected the expectation that at the library, the youth did not want to feel like they were in school.

While this was the temperament at the library, this did not fully preclude calling the group to attention or providing some initial direct instruction. In fact, centralized attention at the beginning and end were key, and the librarian or other program club facilitator could call everyone's attention to share an innovation discovered by one of the attending youth. As such, the base activity structure we have found most consistent with youth expectations was a very brief "learn" period where a challenge is established, materials are made public, and some basic tips are introduced by the facilitator. This would then be followed by a lengthier "make" period that took roughly 45 minutes or more if time allowed. During this time, youth worked on building circuits in response to the challenge and the librarian or other facilitator circulated to provide individual help or connect youth to various in-house resources available at the library. Intermixed in the "make" portion of the activity, the librarian would call out for attention and showcase new discoveries or inventions by youth. Finally, in the remaining portion of time, roughly 10-15 minutes, there would be a "share" time with different approaches taken to showcase the crafts that the youth made and explaining how they worked. Sometimes this involved individual students standing and talking about what they made. Other times, there was informal competition or awards given. Still others, exhibits were created with youth creations placed on display in select locations in the library.

Specific Projects

Figure 3 shows examples of paper circuit projects created by these youth participants. These are presented here to illustrate what youth can do in a single sitting and also serve to help describe and demonstrate some ways in which paper circuits can be used to explore computational thinking.

Conditional Logic For example, the magazine cutout of the flower picture with an LED in the middle includes a simple switch (left). The light comes on when the corner is pressed because the main circuit taped to the bottom sheet of paper is broken in one spot and has a small gap. On the underside of the magazine cutout, a small piece of copper tape is placed above the gap. Pressing that section down can close the circuit. In the context of an educational program, this is the time to make explicit that this will operate on the condition that contact is made between the two sheets. That contact can be sustained with some other removable object that can keep the circuit closed. For instance, securing a binder clip to apply pressure makes the circuit close for as long as the binder clip is present. Here, the conditions need to be carefully considered, although when established, become highly intuitive to youth. It is the same mechanism that was put into place with the Star Wars "light saber" that another youth made (Fig. 3, far right).

Advanced Conditional Logic A more advanced switch structure was integrated into a paper circuit project made by a student who was interested in creating something inspired by the *Harry Potter* universe. In the popular book series, one of the wizards (Dumbledore) possessed a "deluminator" that took the form of a lighter. When the lighter was used, it turned off a nearby lamp or lantern and the light from that lantern presumably became the light emanating from the deluminator. To implement this, the youth needed to devise a way to make a switch that worked such that only a lamp or the lighter would be on at a single time. In this specific project (Fig. 3, center left), the black cylinder was the lantern, and the printout was an image of Dumbledore found online on one of the library computers. Each had an LED and underlying copper tape circuit paths. However, the circuits were not closed, and required something else to close it. The solution this student devised involved a pivot switch that turned toward the lantern (and completed its circuit) or toward Dumbledore (and completed that circuit), but was not large enough to allow for both to be on at the same time. This was an instantiation of XOR logic, and was seen by all other program attendees as a very unique solution and a way to create a very specific set of conditions that would work.

Parallelism Finally, as an exercise in parallelism, many youth found organizing LEDs in serial fashion – with the electricity flow going throw one LED before it could go to another – could become tedious. When the circuit had a flaw, such as a weak connection, the youth would have to test (debug) all possible connections individually until the area needing repair was identified. Parallel circuits, like the one on the paper box (Fig. 3, center right), bypass that. When a single LED or LED connection fails

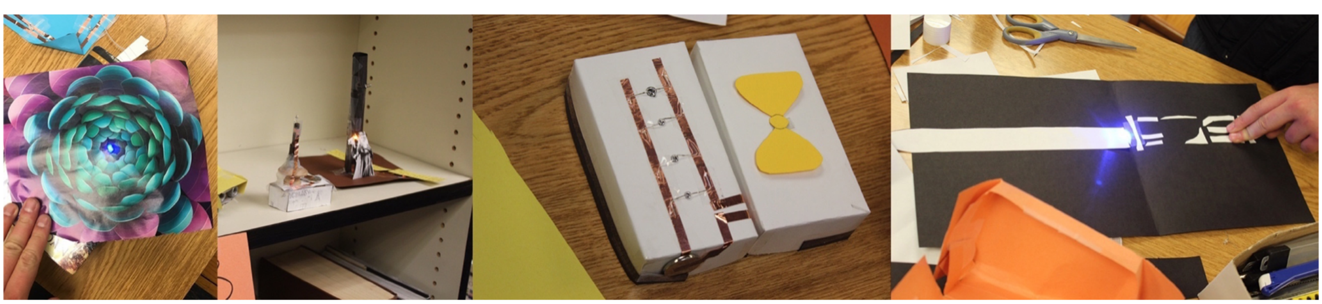

Fig. 3 Sample paper circuit projects conceived by and created by middle school students in a single day, including a light up flower picture, a re-enactment of a scene from the Harry Potter book series, a box that lights when closed, and a light saber from the Star Wars series that lights when a button is pressed

or poorly established, the other LEDs still operate. This is because the ability for electricity to flow through any LED is not dependent on the other LEDs since they are all connected to the circuit independently. The paper box, which also included a switch that was closed when the box was closed, was one instance of this although there were others. The trade-offs between serial and parallel designs could be considered in various paper circuit crafts, and maps onto considerations of various computational architectures.

Beyond showing how computation can be called out and enacted through paper circuits, these example projects also show several fun artistic and aesthetic dimensions where youth can express ideas that are personally meaningful for them. We have found that providing an overarching theme for youth and some easily repurposed materials (such as old magazines or packaging material) can provide some inspiration if they feel they have no ideas about what to create. Pre-made example projects of varying difficulty so students can see some of the possibilities with paper circuits are helpful as well, as they provide inspiration and also are worked examples that students can examine for ideas on circuit layout and become fodder for discussion.

More Advanced Paper Circuitry Experiences

We view paper circuit activities as an approachable entry point for novices. However, more advanced and computationally intensive approaches can be used as well. For example, Chibitronics has been a pioneer in this area through the development of circuit stickers that are interactive and can be placed on paper. They have also recently released a small microcontroller, the Chibi Chip (Fig. 4), that can be integrated with paper circuits. With the microcontroller, rather than having lights come on or off from manual control, a student can write code of their own to determine the sequence and duration of lights blinking. The code can be written as Arduino code or through a block-based programming interface modeled after Scratch. In the example shown in Fig. 5, different lights or light combinations can turn on or off depending on what number is produced by a random number generator.

The dominant form for computational thinking is associated with the production of computer code. However, with paper circuitry, core behaviors such as completing or breaking switches are akin to opening or closing gates – and can be performed without a microcontroller. The difference is that once one masters the creation of paper circuits, the Chibi Chip simply allows for more automation and more complex computations and operations to be managed, including sensing and looping. Debugging can

Fig. 4 The Chibi Chip and its accompanying data transfer and power cord

then expand beyond manipulations in the physical world to manipulations of computer code. The existence of this serves to further illustrate that paper circuits model fundamentals of computational processes and invite early forms of computational thinking. As individuals further pursue this interest, more familiar instantiations of computational thinking – involving depictions of computer code - are but a few steps away.

Implications

In recent years, the educational technology community has been broadening its focus to consider opportunities to support the development of computational thinking (Lee et al. 2017; Rich and Hodges 2017; Weintrop et al. 2016) across a range of ages and contexts. In line with a general goal of increasing access to opportunities to develop computational thinking, our assertion is that an approachable set of resources is merely a scribble, fold, and scissor-cut away. Paper circuits are a low-barrier resource for introducing learners to computational thinking, and it has a potential trajectory with new products that are appearing on the market (e.g., Chibi Chip). When considering that the current landscape of interest-driven learning, partially promoted by widespread interest in Connected Learning (Ito et al. 2013) and what can also be available in community settings such as libraries, we feel it is appropriate to articulate that paper circuitry is a viable starting point for computational thinking.

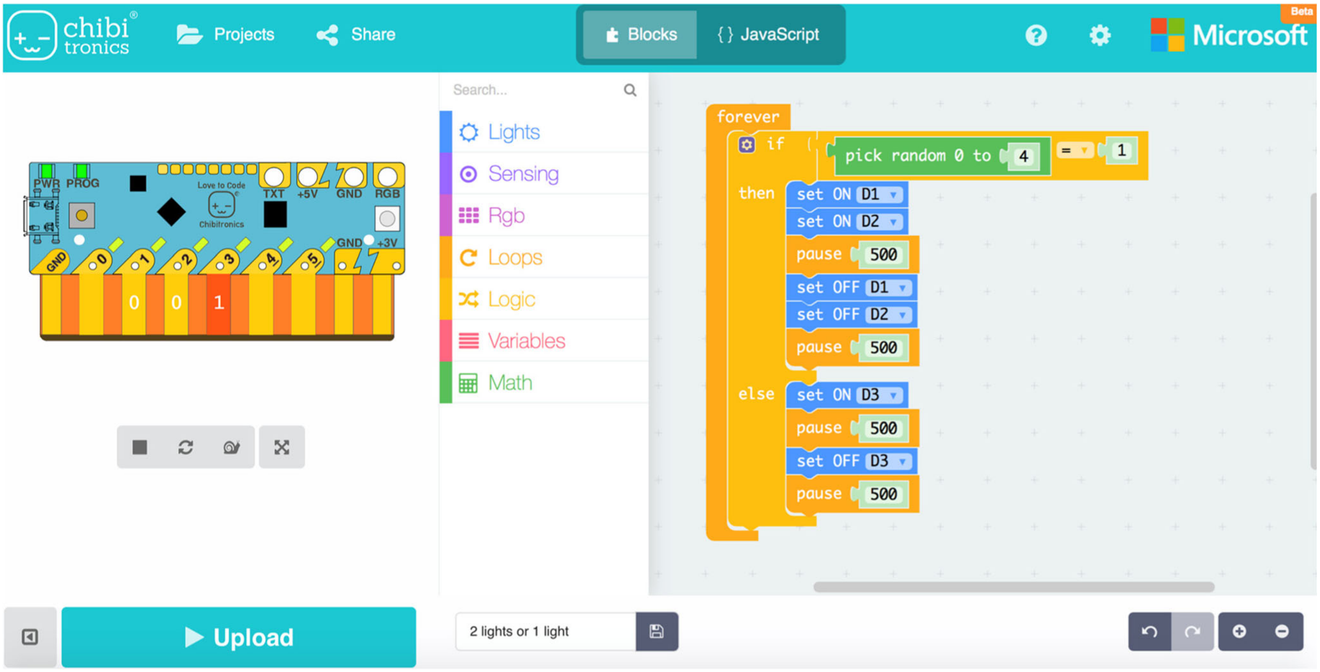

Fig. 5 A simple Arduino program for the Chibi Chip that will turn on a different light or combination of lights depending on what number is randomly generated

In terms of connection to theory about how to support learning, we are arguing that the paper, copper tape, LEDs, and batteries that many are using in their designs of educational settings are indeed low-threshold, interest-driven, craft-oriented computational thinking. We are making more visible the Constructionist orientation tied to current conversations around computational thinking (di Sessa 2000; Papert 1980) and the interest-driven push for Connected Learning intersect appropriately with materials that are familiar, low cost, and still open up a number of opportunities for complex and creative thinking.

The challenge that lies ahead is in how educational designers build upon this resource. While we have one possible trajectory that involves more complex circuit controls using proprietary chip interfaces, we still have much to understand about when and where paper circuits fit with respect to a trajectory of how computational thinking can be taught and learned. For instance, is it an appropriate introduction for more intensive electronic-textiles? Does it tend to be seen as dramatically different from working with text-based code? What kinds of analogies and connections must be made so that the fundamentally computational aspects of paper circuitry are mapped onto the logic gates and control flow processes that are inherently part of more traditional forms of computation? Through this paper, which articulates and extols the potential opportunities to be had with paper circuits, we hope that the community of designers and researchers who are reading this feel more empowered to explore this space further in the future and help to articulate how we can capitalize on these relatively inexpensive resources in the future.

Conclusion

We have argued that paper circuitry projects provide a powerful and complementary approach for providing a first exposure to students to computational thinking. The materials are familiar and low cost and can be implemented without requiring a computer. Their tangible nature enable youth to learn through physical as well as intellectual and artistic engagement in the world to quickly design creative projects. This is especially satisfying for novices who are interested in making something quickly but also want to go beyond rote introductory exercises that lack opportunities for self-expression. It is also easily approached by those who would like to facilitate or design such learning activities, as the overhead associated with these resources is low. As research has shown that computational thinking performance and interest towards programming depend on previous programming experiences (Kelleher and Pausch 2005), we believe that the generative computational capacity of paper provide a low barrier entry point for students before even writing their first line of code upon which subsequent coding experiences. With what is coming down the pipeline as far as digital expansions and supports, and what this community can conceive and iteratively refine, the opportunities for students to build upon and expand their understandings from paper circuits will only increase in the future.

Funding Funding for this work came from an Institute of Museum and Library Services grant, no. RE-31-16-0013-16 and from the Beverly Taylor Sorenson Arts Learning Program.

Compliance with Ethical Standards

Conflict of Interest Victor Lee declares that he has no conflict of interest. Mimi Recker declares that she has no conflict of interest.

Ethical Approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *Journal of Applied Computing and Information Technology, 13*(1), 20–29.

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning, 1*(2), 65–81. https://doi.org/10.4018/ijgbl.2011040105.

di Sessa, A. A. (2000). *Changing minds: Computers, learning, and literacy.* The MIT Press.

Eisenberg, M. (2010). Bead games, or, getting started in computational thinking without a computer. *International Journal of Computers for Mathematical Learning, 15*(2), 161–166. https://doi.org/10.1007/s10758-010-9167-5.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: a review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189x12463051.

Guzdial, M. (2008). Paving the way for computational thinking. *Communications of the ACM, 51*(8), 27.

Ito, M., Gutierrez, K., Livingstone, S., Penuel, B., Rhodes, J., Salen, K.,... Watkins, S. C. (2013). *Connected learning: an agenda for research and design*. Irvine: Digital Media and Learning Research Hub.

K-12 Computer Science Framework Steering Committee (2016). *K-12 Computer Science Framework* (p. 307). New York: ACM.

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

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR), 37*(2), 83–137.

Lee, V. R., & Fields, D. A. (2017). A rubric for describing competences in the areas of circuitry, computation, and crafting after a course using e-textiles. *International Journal of Information and Learning Technology, 34*(5), 372–384.

Lee, V. R., Lewis, W., Searle, K. A., Recker, M., Hansen, J., & Phillips, A. L. (2017). Supporting interactive youth maker programs in public and school libraries: Design hypotheses and first implementations. In P. Blikstein, D. Abrahamson (Eds.), *Proceedings of IDC 2017* (pp. 310–315). Stanford: ACM.

Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas*. New York, NY, Basic Books

Peppler, K. A. (2013). STEAM-powered computing education: Using e-textiles to integrate the arts and STEM. *IEEE Computer, 46*(9), 38–43.

Phelps, D., Benner, J., Munsell, J., & de los Angeles, G. (2017). *How K-5 students leverage computational practices and products to become expert Mancala players*. Paper presented at the 2017 Annual Meeting of the Jean Piaget Society, San Francisco.

Qi, J., & Buechley, L. (2010). Electronic popables: exploring paper-based computing through an interactive pop-up book. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, Cambridge.

Qi, J., & Buechley, L. (2014). Sketching in circuits: designing and building electronics on paper. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, Toronto, Ontario, Canada.

Rich, P. J., & Hodges, C. B. (Eds.). (2017). *Emerging research, practice, and policy on computational thinking*. Cham: Springer International Publishing.

Shorter, M., Rogers, J., & McGhee, J. (2014). Enhancing everyday paper interactions with paper circuits. In *Proceedings of the 2014 Conference on Designing interactive systems*.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.

Williams, P. (2017). *Paper Circuits*. Ann Arbor: Cherry Lake Publishing.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.