CHAPTER 11

Why Pair Block-Based Programming with Physical Computing Devices?

In order to provide equity in the area of computer science, we *must* provide all students with the engaging activities that allow for creating, modifying, and/or using devices that interact with the word around them. While they may not look like a traditional computer/robot, these physical devices are able to sense their environment (inputs), process that information, and then perform some type of action (output)—recall the Input-Process-Output feedback cycle from chapter 6. Therefore, throughout the remainder of this chapter, we will use the words "physical computing device" synonymously with the word "robot."

Coding to Learn and Learning to Code

When educators allow students to learn through the interaction between the virtual world and physical world, they create opportunities for students to code to learn (Resnick, 2012). Imagine creating a Scratch project (in the virtual world) that would ask a variety of questions about space and astrology. Then balloons would float upward with statements that represented truths and misconceptions. Players would use a stick (in the physical world) to pop the misconceptions out of the air. In this example they would be coding to learn about space and astronomy.

While research in the 1990s on how children read led to the catchy phrase “In K–3 children are learning to read, and in 4–12 children are reading to learn,” (Chall et al., 1990), elementary teachers of today understand that reading is much more complex. Learning to read and reading to learn should be happening simultaneously within *all* K–5 classrooms. The same concept applies when it comes to learning to code and coding to learn.

Just as *reading* to learn will open doors and opportunities to learn other things, so will *coding* to learn. Too often, there is an inequity of resources within our schools of poverty and color, and we can only actualize CSforALL (Computer Science for ALL Students; see chapter 12 for more information on this organization) if we have the physical computing devices for students to engage with. But . . . having the devices is only one of the hurdles. Teachers must also be willing to code to learn and learn to code, and that is why you have picked up this book!

Design Thinking

Design thinking is a powerful process of problem solving that begins with understanding what the end product or goal might be. When working as a team, design thinking requires the skills of creativity, reasoning systematically, and working collaboratively. When working with robots, it is helpful to look at the design-thinking process (figure 11.1). This process is cyclical, and there is a lot of back and forth between the stages, especially for our youngest coders. The stages of the design-thinking process are empathy, definition, ideation, prototype, and testing.

1. Empathy involves creating a “user centered” solution.
2. Definition requires interpreting what you learned in order to identify what the user needs.
3. Ideation is the process of generating, advancing, and communicating ideas.
4. Prototyping allows for manipulation and testing of ideas using a prototype or preliminary copy.
5. Testing involves refining the prototype based on user feedback.

When using the design-thinking process, there are many opportunities for social-emotional learning through working with robots like Dash & Dot, including lessons in anger and frustration when having to share. One such curriculum that has earned the ISTE seal of alignment is FUNecole. It has been endorsed by

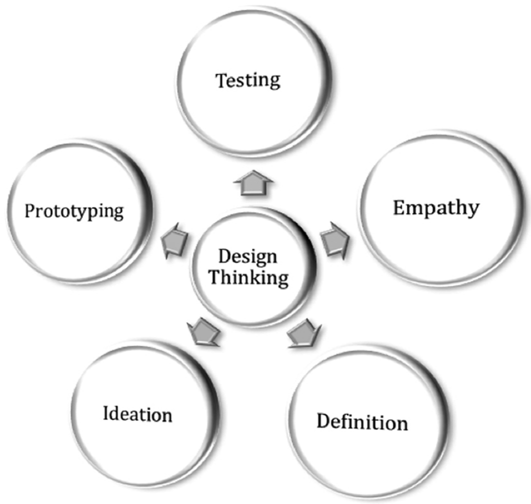

Figure 11.1. Stages in the design thinking process.

Cambridge International for Cambridge ICT Starters Qualification and includes alignment to the ISTE student standards, CSTA computer science standards, and is working toward the CASEL seal of alignment for social-emotional learning. Within the curriculum, there are many opportunities to help students feel compassion and empathy for the problems others face. These are characteristics that will help build ethical and strong future leaders.

Mississippi Teacher Alicia Verweij discusses how teachers have used FUNecole both in person and virtually in this video: **youtu.be/FxZFEsUUf0A**

Learning First, Physical Computing Devices Second

The Triple E Framework, developed by Professor Liz Kolb at the University of Michigan, School of Education, was created to ensure that teaching was put before technology tools. Her framework, as seen in figure 11.2, begins with ensuring students have authentic *engagement* with the learning. Robotics (the technology tool) checks the boxes of focusing on the task at hand, motivating students to start the learning process, and creating the shift of moving from passive to active social learners.

Robotics also meets her targets on *enhancing learning* as well. When students become creators of content, they naturally demonstrate a more sophisticated

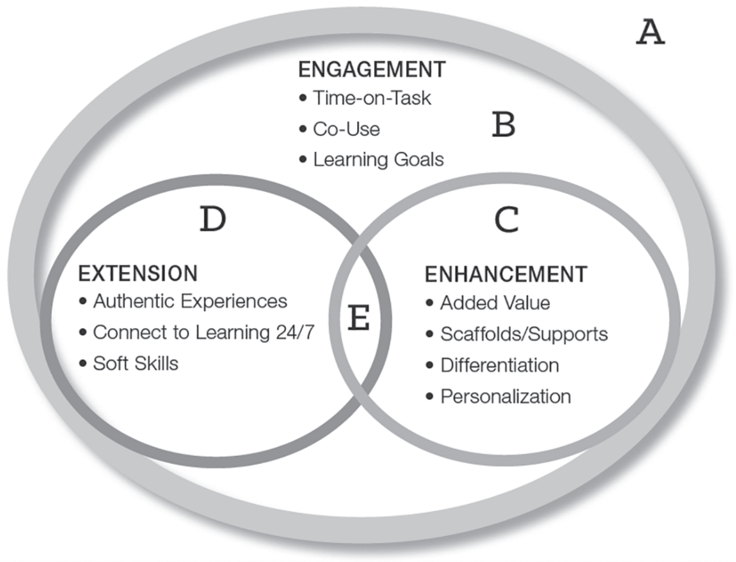

Figure 11.2. The Triple E Framework. Source: Liz Kolb, 2017.

understanding of not only core content, but also computer science standards. When it comes to the technology of block-based coding, the nature of the language itself supports the scaffolding of concepts. Students with a more limited coding experience will sequence repetitive steps in their algorithms, while more experienced coders will begin to use loops for repetitive steps. Most importantly, robots demonstrate learning goals using a performance-based task. Learning and assessment goes well beyond traditionally used tools.

Finally, physical computing devices *extend learning* by building on skills that students can use in their everyday lives to solve everyday problems they may encounter (Kolb, 2017). For example, phones are physical computing devices that run apps, and MIT's APP Inventor uses a block-based coding environment for students to create their own apps. MIT's most recent App Inventor Challenge revolved around the coronavirus. The June 2020 winner was an eight-year-old from India who created an app to generate positive vibes with India's ancient Vedic chants to help with the pandemic effort.

Try it!

EXPLORE MIT'S APP INVENTOR

The Getting Started guide and other resources on the website will help you get started creating apps:
appinventor.mit.edu/explore/ai2

Most Importantly: PLAY

The American Academy of Pediatrics is well versed in research on the power of play! According to the 2007 article “The Importance of Play in Promoting Healthy Child Development and Maintaining Strong Parent-Child Bonds,” “Play allows children to use their creativity while developing their imagination, dexterity, and physical, cognitive, and emotional strength” (Ginsburg, 2007). The key for learning is that these things occur when play is child-driven; the key for educators is that you don’t need to be the content expert when it comes to technology and computer science.

It is through play that children learn to create and explore their world. As they begin to master their world, play helps them build confidence and build the resiliency they will need to face future challenges. It is also through play that children learn group-work skills such as sharing, negotiation, conflict resolution, and self-advocacy skills. Play even helps adults by adding joy to life, helping relieve stress, and making the learning of new skills *fun*!

While the next chapter outlines commonly used physical computing devices that can be found in schools across the world, it is not meant to promote any specific devices or be an all-encompassing list. It does, however, uncover a common thread behind all the stories of each product: *play*.

CHAPTER 12

Physical Computing Resources

When working to achieve CS education for all, many have turned to the importance of CSforALL's mission statement:

> CSforALL's mission is to make high quality computer science an integral part of the educational experience of all K–12 students and teachers and to support student pathways to college and career success. (CSforAll, n.d.)

CSforALL is a central resource for individuals and organizations interested in K–12 computer science education. They connect providers, schools, districts, funders, and researchers working toward the goal of providing quality CS education to every child in the U.S. Their organization supports school districts through their SCRIPT: Strategic CSforALL Resource & Implementation Planning Tool. It is a planning framework that guides teams of district administrators, school leaders, and educators through a series of collaborative visioning, self-assessment, and goal-setting exercises. Teams walk away with solid plans and action steps to create or expand upon a computer science education plan for their students.

EXPLORE CSFORALL EDUCATION RESOURCES

Join the movement to bring computer science to all students! Learn more about CSforALL at **csforall.org**.

CSforALL SCRIPT framework:
bit.ly/2JCeiaG

Read the report “CS for What? Diverse Visions of Computer Science Education in Practice” at **bit.ly/3mXInjn**.

While schools all across the world are focusing on computer science and computational-thinking professional development plans, there are some essential steps that all educators can take to help actualize high-quality coding and computational thinking across the curriculum. In achieving equity, we must realize that every community faces a different learning context. Each have their own strengths and resources, as well as their own challenges and funding priorities. Values should drive what computer science education looks like.

One Size Does Not Fit All

Since not every district has the same physical computing devices, the following sections will outline some of the commonly used resources and how current, practicing educators are using them. Some have written district grants for funding, others have gone to Donorschoose.org, and some have even written state-level grants in order to secure the physical resources they need. Oftentimes, professional development is done through individual initiative or in groups banned together with a growth mindset and the mantra of “We will figure this out together.”

Ideally, a district will have a strategic plan in how they are implementing computer science standards and how the resources they select will fit into that structure. There will also be thought put into how those standards will be assessed. If you are one of the pioneers in bringing computer science and/or computational thinking to your district, hopefully you will find the words of wisdom, contained in the following chapters, from fellow educators useful. If you have already begun your CS journey, hopefully you will find some new take-aways that will enhance your journey.

And if you have already established your CS pathway, hopefully you will feel a sense of reassurance and a gold nugget or two.

Dot, Dash, and Cue

After Wonder Workshop's CEO and cofounder Vikas Gupta's first child was born, he soon realized that in order to prepare his daughter for the world of tomorrow, she would need to be armed with robust technical skills—and the ability to code. He also witnessed firsthand how quickly his daughter learned when she was having fun. So Vikas was inspired to invent a way to engage children to learn to code at a young age.

The Wonder Workshop robots (Dot, Dash, and Cue) put the power of play into the hands of the children, while at the same time allowing for creativity and a ton of flexibility for classroom teachers. Programming languages for these adorable robots include icon-based apps, blocked-based Blockly, and text-based JavaScript. Due to their versatility, you will find these products in more than forty-three countries and in over 20,000 schools worldwide.

For educators just starting out, these robots provide a ton of resources. They offer "Blockly Print Outs," which are block-based coding templates than can be printed out for unplugged activities or even sent home with students. Their Global Educator Community has over 1,000 members and is a great place to connect for support, as well as new ideas. They also have "Robotics Team Role Cards" you can download, a blog to follow, tips and tricks, and a STEAM Activity Pack.

Try it!

Check out the Wonder Workshop educator toolkit to see all their amazing resources: **makewonder.com/toolkit-educators.**

Try out these six STEAM activities for Dot, Dash, and Cue: **bit.ly/363NXKe.**

To grow your PLN (professional learning network), I encourage you to connect with Jasmine Saab and read her blog post about cross-curricular ways to use Dot, Dash, and Cue in the classroom. She shares her passion of coding and combines as many subjects as possible when utilizing robotics with her fourth grade students. Table 12.1 shows how her projects line up against the standards.

TABLE 12.1. Jasmine Saab's Lessons for Dot/Dash/Cue with Subject Area and Standards Connections

LESSON TITLE	SUBJECT AREA	STANDARDS	CSTA STANDARDS
2 Digit by 2 Digit Multiplication	MATH	**4.NBT.B.5:** Multiply a whole number of up to four digits by a one-digit whole number, and multiply two two-digit numbers, using strategies based on place value and the properties of operations. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.	**1B-AP-11:** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **1B-AP-10:** Create programs that include sequences, events, and loops and conditionals. **1B-AP-13:** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences (the audience). **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. **1B-AP-16:** Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. **1B-AP-18:** Describe choices made during program development using code comments, presentations, and demonstrations.
Order of the Planets	SCIENCE	How will you make your Bee-Bot go from the letter H to the letter Q? Why didn't the Bee-Bot do what you wanted it to do?	
Battle of Bunker Hill	SOCIAL STUDIES	**D2.His.1.3-5:** Create and use a chronological sequence of related events to compare developments that happened at the same time.	

Content area standards are from NGSS, CCSS, C3 framework, and CSTA standards.

Try it!

JASMINE SAAB'S CODING & ROBOTICS LESSONS
The full lesson plans and rubrics can be downloaded from Jasmine's blog post **(bit.ly/32ecGdM)**.

Makey Makey

Underneath the guidance of Mitch Resnick, creator of Scratch, two students from the MIT Media Lab initiated an academic and artistic project. What resulted was the creation of a small invention kit that comes with a Makey Makey board, wires, alligator clips, and a USB cable. Cocreators Jay Silver and Eric Rosenbaum believe that everyone is creative, inventive, and imaginative.

Students and teachers alike can let their imaginations run wild when pairing this physical computing device with their coding skills. For example, when paired with Scratch's block-based programming language, a budding eight-year old inventor created a physical "knife-and-log" interface for cutting virtual wood in an online game.

For educators just starting out, their website supports online classes that will walk you through how the Makey Makey works so you can start inventing. "Lesson Two: Hands on a Makey Makey" (bit.ly/34ZKdtC) could be used as the foundation for a hands-on workshop for administrators and/or educators to learn through play! Their philosophy of invention literacy fits well into any STEM or STEAM program.

To grow your PLN, I encourage you to connect with Julie Smith, the TECHIE teacher, in her blog post about academic ways to use a Makey Makey in the classroom. In her post she has links to a variety of projects she has implemented within core content areas. Table 12.2 shows how her projects line up against the standards.

Try it!

JULIE SMITH'S MAKEY MAKEY PROJECTS
The projects and instructions can be downloaded from Julie's blog post **(bit.ly/2TUo7CB).**

TABLE 12.2. Julie Smith's Makey Makey Projects with Subject Area and Standards Connections

LESSON TITLE	SUBJECT AREA	STANDARDS	CSTA STANDARDS
Affixes with Pencil Lead: Prefixes, Root Words, and Suffixes	ELA	**L.4.4.B:** Use common, grade-appropriate Greek and Latin affixes and roots as clues to the meaning of a word.	**1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
Money: Counting Coins	MATH	**2.MD.C.8:** Solve word problems involving dollar bills, quarters, dimes, nickels, and pennies using $ and ¢ symbols appropriately. Example: if you have two dimes and three pennies, how many cents do you have?	**1A-CS-02:** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).
Water Piano: Sound/Pitch	SCIENCE	**1-PS4-1:** Plan and conduct investigations to provide evidence that vibrating materials can make sound and that sound can make materials vibrate.	**1B-AP-11:** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **1B-AP-10:** Create programs that include sequences, events, and conditionals.
Interactive Compass Rose: Cardinal Directions	SOCIAL STUDIES	**D2.Geo.1.K-2:** Construct maps, graphs, and other representations of familiar places.	**1A-CS-02:** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).

Content area standards are from NGSS, CCSS, C3 framework, and CSTA standards.

Ozobot: Bit and Evo

Nader Hamda, the CEO and founder of Ozobot, recognizes that these are exponentially changing times and believes that many of the jobs today's students will have do not even exist yet. To help educators prepare students for potential jobs, Hamda's team at Ozobot had the vision of making it easy for educators to teach all subjects with their physical computing devices—and inspire all students to learn and create their own future.

It is no accident that the Ozobot team is made up of a diverse group of artists and engineers. They collaborate with children every day, as children are their inspiration. By watching students play, share, and create with their robots, Ozobot uses student feedback to make iterative improvements on their products. Like many of the physical computing devices, students can code both Bit and Evo using Google's Blockly. However, Ozobot has a unique screen-free color-code system by which non-readers can engage in coding their robot. By using markers or stickers on white paper, students can create color-code patterns along a solid black line.

The Bit and Evo robots use sensors to follow the solid black line, and when it reaches one of the color patterns, it will follow the command associated with the colors (for example, spin or go left/right).

To support educators, Ozobot Classroom has a free sign-up that includes interactive teacher training, single-click class setup, standards-aligned lessons for all grades and all subjects, and it integrates with Google Classroom. The Ozobot blog is another great resource; they recognize an educator of the month and share how that educator is using Evo and/or Bit to connect with content standards. For example, Kendra Jordan was their educator of the month for July 2020, and she shared how she engaged one of her students who was having a difficult time learning about coordinate planes in math. Since this scholar was interested in video games, she met his social-emotional needs by recognizing that he was a "gamer." She then showed him how video games used coordinate planes by the people who code them, thus turning a gamer into a game-changer! See the full blog post at bit.ly/362QvIo.

To grow your PLN, I encourage you to connect with Deanna Brewer and her cross-curricular lessons when using Ozobot in the classroom. She shares her passion for STEM by supporting students and teachers in integrating these bots into core content areas. Table 12.3 shares how her projects line up against the standards.

TABLE 12.3. Deanna Brewer's Ozobot Lessons with Subject Area and Standards Connections

LESSON TITLE	SUBJECT AREA	STANDARDS	CSTA STANDARDS
Mapping Out Geometry	MATH	**4.G.1.1:** Draw points, lines, line segments, rays, angles (right, acute, obtuse), and perpendicular and parallel lines. Identify these in two-dimensional figures.	**1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
Ozobots Life Cycles	SCIENCE	**3-LS1-1:** Develop models to describe that organisms have unique and diverse life cycles, but all have in common birth, growth, reproduction, and death.	**1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. **1B-AP-16:** Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.

Content area standards are from NGSS, CCSS, C3 framework, and CSTA standards.

Try it!

OZOBOT LESSON PLANS

Check out Deanna's full lesson plans at **nofearcoding.org/ozobot**.

Robo Wunderkind

Rustem Akishbekov, the founder of Robo Wunderkind, came up with the idea while building robots on the Arduino platform. The robots were too techie—not an easy entry into the world of technology—so he set out to make learning coding and robotics as fun and simple as playing with building blocks. These engaging robots beg for the use of creative thinking with Robo Wunderkind's unique LEGO-like compatible blocks, which include a variety of sounds, lights, and sensors.

The coding options for these physical computing devices include remote-control options, two visual icon-based coding apps, RoboBlockly, as well as using a free open-source Python API and an Arduino Library. What is most impressive is their philosophy of "Imagine – Build – Code – Reflect," as this helps to support children's cognitive development and offers a new way to learn coding, robotics, and STEAM skills through open-ended play. Their curriculum is built on a constructivist learning theory and serves children from kindergarten through middle school.

To grow your PLN, I encourage you to connect with the Howard B. Owens Science Center in Lanham, Maryland. It is a 27,500-square-foot facility that is owned and operated by Prince George County public schools. Its mission is to provide excellence in STEM education through student-centered programs and services. Their staff of teachers provide students with hands-on exploratory learning. Each year, first grade students come to the center to draw, build, code, and test robots to complete a science mission task.

Sallie Smith, the computer science program instructor, has the following to say about her program: "The blocks are color coded by function, [and] the blocks are easy to put together and can make many different kinds of robots, [and] the electrical components are inside the block modules and kids love to build, code and test these robots."

Table 12.4 shows how her program lines up against the standards.

Try it!

EXPLORE ROBO WUNDERKIND

Check out the program description and materials for the Design, Build and Code Robots program at **nofearcoding.org/wunderkind**.

TABLE 12.4. Sallie Smith's Design, Build and Code Robots program with Subject Area and Standards Connections

PROGRAM	SUBJECT AREA	STANDARDS	CSTA STANDARDS
Robo Wunderkind: Design, Build, and Code Robots	ELA	SL.2.5: Create drawings to clarify ideas, thoughts, and feelings.	1A-CS-01: Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.
	MATH	2.MD.A.3: Estimate lengths using units of inches, feet, centimeters, and meters.	1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
	SCIENCE	K-2 EST1-3. Analyze data from tests of two objects designed to solve the same problem to compare the strengths and weaknesses of how each performs.	

Content area standards are from NGSS, CCSS, C3 framework, and CSTA standards.

CoDrone Mini

Lifelong entrepreneur, baseball fan, and foodie, Hansol Hong, established Robolink in 2012. Hailing from Seoul, Korea, this robotics company has taken root in San Diego, California with its sole purpose of encouraging students to engage in STEM. According to this CEO, "We make fun and approachable robotics kits that bring computer science to life and teach real industry competency."

This company's mission is to make STEM education accessible, engaging, and fun for all. They offer a variety of teacher resources including quickstart guides, FAQ, as well as access to free lesson plans, video tutorials, and learning guides. The CoDrone Mini is an excellent resource for middle school math and science teachers. With their LED and gyroscope sensors, students can explore how the physical world works by plotting a variety of data points.

Given the recent demand for drone pilots, skilled technicians and programmers will also be needed. “Not only do students need to learn the skills of flying, they also need to master the skills of programming and computational thinking to solve complex problems” (Bartholomew, 2018). Utah State University has developed one such curriculum for 4th-8th grade students. While not specifically for the CoDrone Mini, educators could easily adapt the following basic unit outline, adapted from their full lesson plan:

Unit 1: Following safety procedures

- 1.1 Follow mini drone safety practices.
- 1.2 Sync control device to specific minidrone.

Unit 2: Flying the mini drone remotely

- 2.1 Hover the minidrone at specified altitudes.
- 2.2 Fly the minidrone in a square pattern without using pitch and roll controls.
- 2.3 Fly the minidrone in a square pattern using yaw controls.
- 2.4 Fly the minidrone through an obstacle course.

Unit 3: Flying the Parrot mini drone autonomously

- 3.1 Program the minidrone to fly a simple pattern.
- 3.2 Program the minidrone to fly through an obstacle course.

Try it!

CODRONE LESSON PLANS

To see the full lesson plan and resources, check out **bit.ly/2NeJwpX**.