Robots and sequences

BY MERCEDES ANDERSON, BLANCA MILLER, JAMIE POSTON, ADAM KIRN, MELISSA JURKIEWICZ, AND DAVID FEIL-SEIFER

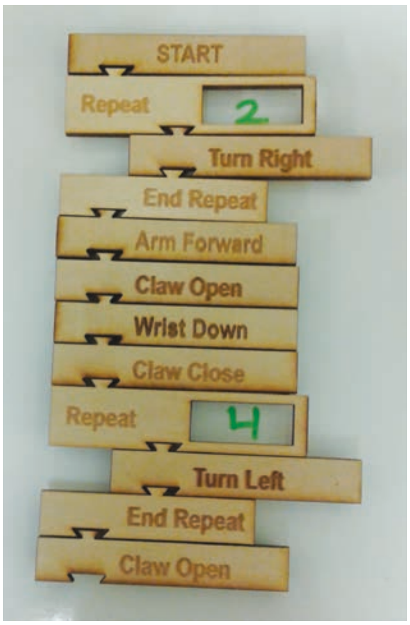
Robotics education often focuses on the design, construction, or programming of robots. However, a critical yet often overlooked component of understanding how coding allows robots to combine a sequence of smaller actions to create a more complex robotic action.

This lesson has been piloted in several middle school classrooms, in addition to several informal education programs on a University campus. Our lesson was designed as a means of introducing K–12 middle school students to engineering. This lesson can be paired with similar lessons on technology or robotics as a way of providing students with a tactile experience. Alternatively, this lesson can be paired with discussions on planning for a future profession by allowing students to investigate the area of robotics, which is increasingly involved in a broad range of careers. We recommend using this lesson as an introduction and jumping-off point for students' future learning of computational thinking concepts, as well as in lessons for other fields of engineering.

This lesson allows students to engage with robotics as they work through three modules exploring sequences. A *sequence* is a series of steps followed to complete a task. This lesson is structured as follows:

FIGURE 1: The magnetic code pieces used for this lesson assembled into a sequence

The sequence begins at the "START" block and proceeds from top to bottom in order.



1. Sequences are presented to students that make a robot accomplish a simple action (Figure 1).
2. In groups, students develop sequences to make a robot perform a specified task.
3. Students analyze code to find and fix errors that prevent a task from being accomplished.

Setup

Students should be arranged into groups of two or three to promote collaboration through discussion and sharing of diverse ideas. Group arrangements should use multiple criteria to scaffold and support diverse learners. Example considerations include students with different learning habits or processing speeds, English language learners, or students with special needs, or groups arranged for gender balance. Each group should allow students access to, and encourage participation in, the engineering design process. Each group should have one robot, one stack of 37 wooden, three-inch coding blocks for the robot arm to manipulate, and at least one set of code pieces. Stations should be set up similarly to Figure 2.

Safety

This lesson involves electronic components. Instruct students to not eat food or drink during the lesson. Students should be trained on safe handling of electronics and only interact with the robot's controls. Students should be supervised and follow the safety guidelines. Students do not need goggles during this lesson, as they are not using wires.

Engage (5–10 minutes)

Student engagement for this lesson begins with asking students to retrieve background knowledge about robots and identify what students already know about them through a paired peer discussion. Then, as a group, ask students to share what they discussed. Point out to students that they may have daily interactions with robots they may not recognize, such as self-checkout machines, automatic doors, and vehicles. Then, based on the class discussion about students' prior knowledge, have the class construct a working definition of *robot* before providing the technical definition: "A machine capable of automatically carrying out a complex series of actions."

Next, as a means of further demonstrating the definition of *robot*, students watch a video of a robot sorting lemons and limes into different containers. Ask students to discuss in pairs how the definition of *robot* applies to the video; that is, what complex action did the robot implement during the video and how did it implement that action automatically? Through a class discussion, lead students to the conclusion that the robot has been designed to perform a sorting action. Then, explain that the robot's programming is a series of sequential steps to complete its tasks.

Explore/Explain (25–45 minutes)

Conceptual exercise

Introduce to the class *sequence* as a vocabulary word and describe the three rules that all sequences follow: They go step-by-step, the order of these steps matter, and some steps can be switched with other steps without changing the overall result.

Begin a conceptual discussion of sequences by asking students to write down the steps they take when brushing their teeth. Record a student example on the board (see Online Supplemental Materials). When completing a sequence that arrives at a finished product (e.g., clean teeth), sometimes the order of certain steps matters and sometimes it does not. To help focus students' attention on this idea, ask them whether the order of the steps could be switched and if any steps need to be completed in a certain order. For example, they could have brushed their bottom teeth before brushing their top teeth and the result would have been the same; however, they could not "put brush away," prior to "grab brush." Highlight that there are many ways of accomplishing the goal of brushing one's teeth and none of the ways

CONTENT AREA

Problem solving and design process

GRADE LEVEL

5 and 6

BIG IDEA/UNIT

Sequences, problem solving

ESSENTIAL PRE-EXISTING KNOWLEDGE

Spatial orientation (knowledge of left from right)

TIME REQUIRED

60 minutes, with an additional 15 minutes for evaluation

COST

$1,200 for the entire lesson, including 10 robots and magnetic sets. However, this lesson can be done completely without them.

SAFETY

When role playing, students need to monitor their surroundings. Students should be instructed to avoid getting their hand or fingers in the way of the robotic arm.

demonstrated were incorrect. To encourage students during their sequence writing activity, we recommend reminding the class that different answers between groups do not indicate an error; instead, they are different avenues to the same end goal.

Activity 1: Sequences to actions

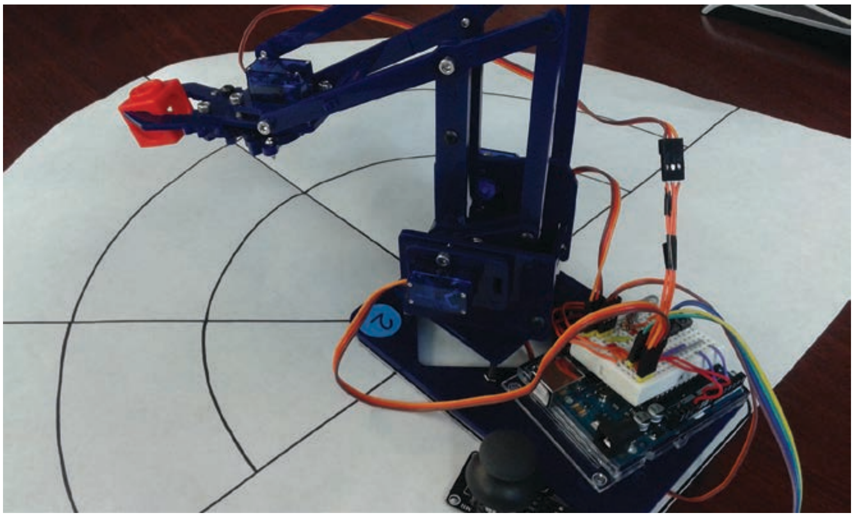
Prior to initiating this activity, remind the class that the robots will always start by facing forward (represented by line 3 on their mats, with the claw above 3-B). Take a moment to describe the remote's controls: button (open claw and close claw) and joystick (turn right, turn left, forward, and backward). Lastly, every sequence that students write must be tested with their own arms first before they move on to working with the robot.

Allow students about two minutes to explore all the actions the robot can take. Be sure that students notice that the robot starts with its claw above the first semicircle, moves in 45-degree increments, and is able to reach blocks that are located on the intersections drawn on their mats.

Give students a sequence (see Figure 3) and an initial block position on the mat. The block position shows where the block should be placed in relation to the robot (see Figure 4). Then, assign team roles of "programmers" and one or two "robot(s)." Programmers will read aloud each action, while robots(s) simulate the robot arm's actions with their own arm, being careful to not strike another student. Roles should be rotated for each activity. Role-play allows students to simulate what the given sequence will accomplish. Prior to acting out the sequence with the robot, each group predicts the block's final position. Students then use the robot to execute the provided sequence again. Once students manipulate the robot according to the given sequence, display the block's end position on the mat and invite groups with different final positions to share with the class. Invite groups to describe their thought processes and engage the class in a discussion to evaluate each other's ideas.

FIGURE 2: The MeArm robot arm placed on the mat used for this lesson

The robot holds a block in its claw. The sequence lessons require a group of students to create a code sequence that commands the robot to pick up a block from one location and place it at another.



Activity 2: Designing sequences

Show students a start and end block position. Students then design sequences by organizing the magnetic code pieces into a sequence that, when executed, will make the robot move the block to the end position (Figure 5). An example solution is shown in Figure 6. Have students test their proposed sequence by simulating with their own arm. Encourage students acting as robots to close their eyes while acting out the steps to ensure they are acting out only what is being read to them. After acting, groups should execute their sequence with the robot.

After the groups have created and tested a solution, have them trade their sequence with another group and test it. Seeing a different sequence allows students to consider alternative perspectives and assess whether the shared example achieves the objective. With this new perspective, students can redesign their own sequences and re-evaluate their solutions. Invite groups to share their solutions and discuss what is done well in each group's sequence. Things that are worth pointing out in student sequences are when steps are taken to plan ahead, such as not extending the arm forward until it has been turned toward the block to avoid knocking it over. Have students identify where their sequences have been written differently. For example, the robot could reach forward before turning or lifting the wrist up. Point out to students that neither solution is more correct as long as the goal is met. Remind them that this is the third property of sequences, that they can be flexible.

Activity 3: Redesigning or "debugging" code

Students set up their mat to look like the initial mat in Figure 7. During this module, students are provided with a sequence (Figure 8) that has an error, or *bug*, in it, along with a picture of the desired result of sequence implementation. The error in the sequence will have the robot drop the block on intersection 3-B when it should be placed on intersection

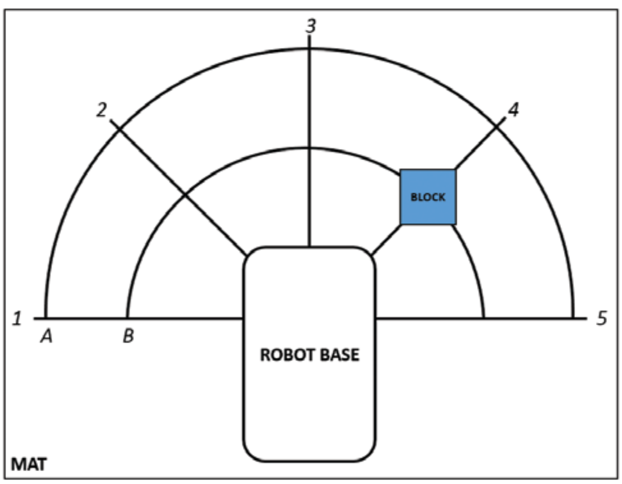
FIGURE 3: A sequence that, when followed, will result in the ending mat for activity 1

Note: There exist many alternative solutions that will also result in the same ending behavior, even if the path to get there is different. We present the simplest sequences, whenever possible.

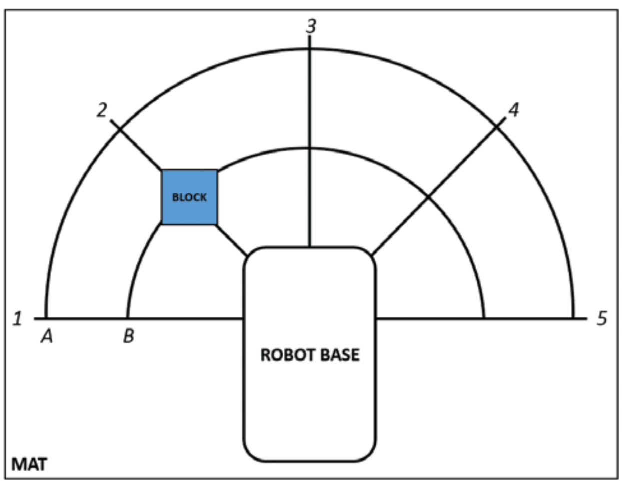
Step	Action
1	Start
2	Turn right
3	Claw open
4	Wrist down
5	Claw close
6	Turn left
7	Turn left
8	Claw open

FIGURE 4: The start and ending locations for activity 1

At the start of the activity, students should place a block on the same intersection as depicted by the blue square on the starting mat. At the end of the activity, if students follow the given sequence, the block will be at the same intersection as depicted by the blue square on the ending mat.



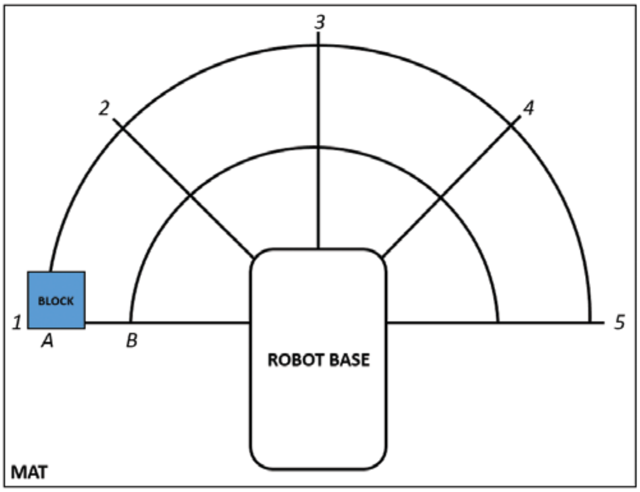
Start location. The block should be at position 4-B and the arm should be facing line 3, with the claw floating above position 3-B.



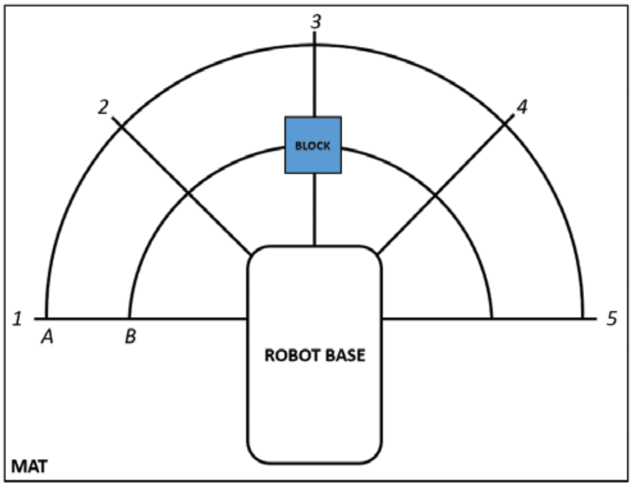
End location. The block should be at position 2-B.

FIGURE 5: The start and ending locations for activity 2

At the start of the activity, students should place a block at the same intersection as depicted by the blue square on the starting mat. During activity 2, students should write a sequence that will result in the block being moved to the same intersection as shown in the ending mat.



Start location. The block should be at position 1-A and the arm should be facing line 3, with the claw floating above position 3-B.



End location. The block should be at position 3-B.

FIGURE 6: An example of a sequence to create the ending mat for activity 2

Step	Action
1	Start
2	Turn left
3	Turn left
4	Claw open
5	Arm forward
6	Wrist down
7	Claw close
8	Turn right
9	Turn right
10	Arm backward
11	Claw open

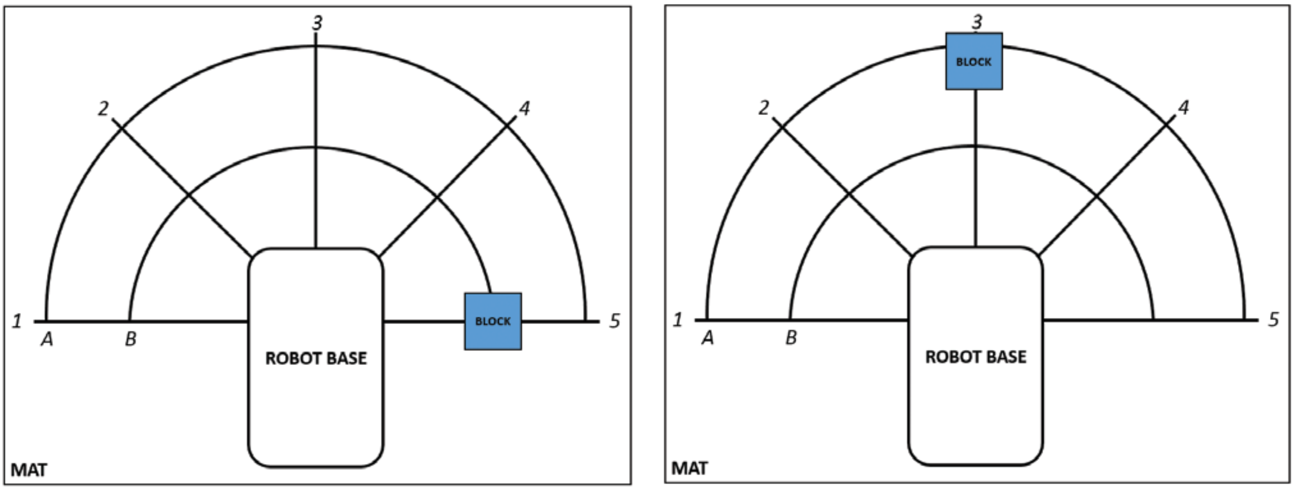
3-A. Figure 9 demonstrates how a fixed or *debugged* sequence might look. The additional code step, or *fix*, in this example is that the arm is extended forward before dropping the block at the expected intersection. Students may also determine that the robot should lift its wrist before moving the block so it does not drag on the ground. During this activity, students should rewrite the sequence and add steps as necessary in order to fix the bug.

Students identify the error in the sequence and determine how to fix it so the end position of the block is at the desired location. Redesigning or debugging sequences provides an opportunity for students to reflect on and brainstorm possible solutions to an error and identify the best solution based on their discussion, a critical aspect of the engineering design process. Allow students time to determine that missing just one step in a sequence can cause the robot to deviate significantly the end goal. This concept can be used to connect sequential logic to students' lives; for example, you cannot put on your shoes before you put on your socks. Students struggling to locate the error should be prompted to set up their mats and solve the problem without looking at the provided code.

After students have an idea of what their own sequence looks like, ask them to identify differ-

FIGURE 7: The beginning and ending locations for activity 3

At the start of the activity, students should place a block at the same intersection as depicted by the blue square on the starting mat. During activity 3, students will need to fix and rewrite the provided sequence for the block to reach the same intersection as shown in the ending mat.



Start location. The block should be at position 5-B and the arm should be facing line 3, with the claw floating above position 3-B.

End location. The block should be at position 3-A.

FIGURE 8: A sequence that contains an error, causing the actual ending mat the robot would create to be different than the one provided

Step	Action
1	Start
2	Turn right
3	Turn right
4	Claw open
5	Wrist down
6	Claw close
7	Turn left
8	Turn left
9	Claw open

FIGURE 9: An example of a sequence that corrects the error and has the robot create the appropriate ending mat for activity 3

Step	Action
1	Start
2	Turn right
3	Turn right
4	Claw open
5	Wrist down
6	Claw close
7	Turn left
8	Turn left
9	Arm forward
10	Claw open

ences between their sequence and the one provided. Then they should test the sequence. This process allows students to evaluate causes and effects for working and nonworking sequences. After redesigning their sequences, students should once again trade sequences with another group and evaluate the solution. Once all groups have altered their sequences and retested, discuss any errors that were present in the original sequence, as well as solutions that were created.

Extend (5–10 minutes)

Give each group a worksheet that has two blank mats, one mat for an initial block position and one for a resulting block position (download "Robot Arm Sequencing Worksheet" with this article's Online Supplemental Materials). Each group will draw an initial and final block position on the corresponding worksheet mats, then trade with another group to produce a sequence to move a block from the initial to the final block positions. Students design a sequence in the space provided for the original group to grade. This exercise simulates a sandbox environment where students are afforded time to create problems, followed by an opportunity to work through the code design process at a personal pace.

Evaluate (15 minutes)

An evaluation for this lesson and a corresponding answer key are provided online (see Resources for lesson support materials). The assessment features problem-solving questions based on the lesson's learning objectives, including questions that ask students to apply their skills in debugging and sequence generation. Additionally, conceptual understanding questions are posed regarding how sequences are applied to areas not explicitly related to computer science. To complete the assessment, we recommend providing students with 15 minutes at the end of the lesson or using the assessment as a homework assignment. To review the material and provide students the opportunity to discuss what they learned and justify their thought processes with a peer, we also recommend pairing students to discuss their assessment responses when they return to the classroom.

About the robots

The robots used for this lesson are constructed from the Mime Industries' MeArm schematic. Resources on how to construct the robot, remote (which allows students to move the robot as they desire), and magnetic code pieces are available online (see Resources for lesson support materials).

The initial construction of materials for this lesson was done by Ms. Anderson (see lesson support materials in Resources for instructions). In addition, Ms. Anderson can provide construction assistance to Science Scope readers via e-mail. The authors would not recommend young students assist in the building of these robots, as they require builders to be dexterous and have patience working with parts and the manual. However, older students who are able to follow instructions may be encouraged to help construct the breadboard for the remote and work with wires. During the construction of the breadboard and the rest of the remote, the components should not be connected to electricity so there is no risk of electrocution; however, as they will be working with small wires, goggles may be necessary. If students are old enough to use hot glue, they can be encouraged to help put together the magnetic code pieces by gluing magnets to the back of the wooden pieces.

One obstacle for incorporating robotics lessons into K–12 classrooms is the involvement of resources that many schools and teachers cannot afford. To address this challenge, we designed a robotics lesson that does not require computers. This lesson costs roughly $150 per robot (usable by a two- or three-person group), as shown in Figure 10. If only one robot per class can be afforded, the robot should sit on an independent table, and groups can come up one at a time to test their sequence with it after they first test their sequence using their own arm as a robot.

This lesson can also be delivered without the use of the robots and/or the wooden magnetic code pieces, which are wooden pieces with steps engraved on the front and magnets glued to the back, which teachers would construct along with the robots. To replace the robot arms, have students take turns acting as the robot and using their own arm during activities. The magnetic code pieces can easily and cheaply be replaced by paper strips with the same directions. The lesson objectives and activities should remain the same, with the exception of the steps that ask students to implement their code with their robot.

FIGURE 10: **Materials and budget for a class of 30 students**

Reported prices are our best estimate for listed items, and will vary based on location and sales. For a more detailed list, including links to all the items, please visit see Resources for lesson support materials. This website also features detailed guides on how to construct the robot and remote, and provides the program for the robots.

Item	Amount	Price
Supplies to build robot		
Acrylic sheets (A2 sized)	10	$50
Motors	50	$200
Rubber feet	16	$10
Nuts	100	$5
Screws	100 each of sizes: 6 mm, 8 mm, 10 mm, 12 mm, 20 mm, 30 mm	$55
		Subtotal: $320
Supplies to build remote		
Arduino Unos	10	$300
Tiny breadboards	10	$10
Resistors	50	$10
Pins	400	$10
Wires	350-piece kit	$15
Dupont female connectors	40	$5
Long Dupont female connectors	40	$45
Buttons	50	$10
Joysticks	10	$50
		Subtotal: $455
Supplies to build magnetic code pieces		
*Wood (1/8” × 4” × 24”)	10	$45
*Magnets (1/2”)	100	$65
*Magnetic whiteboard	10	$95
		Subtotal: $205
Miscellaneous supplies		
*Blocks	10	$15
USB wall plug	10	$50
USB cable	10	$90
*Paper mats	10	$45
		Subtotal: $200
		Total without robot (items indicated by *): $250
		Total with robot: $1,180

Differentiated instruction

Educators should encourage student participation and frequent check-in by walking around the classroom and prompting students with questions that promote critical thinking, emphasize the learning objectives, clarify misconceptions, and guide them to higher levels of conceptual understanding (e.g., “Why did you choose that approach?”; “How is your idea different from that of your partners?”; “Have you considered a different approach?”; “Where is sequence flexibility applied?”). Teachers should model a level of comfort with the content and materials, as students are less likely to have previously worked with robotics and semiexposed electrical components. Additionally, this lesson should be integrated within existing classroom structures to maintain consistency for students who face challenges with new contexts.

Teachers should provide explicit statements with visual reinforcement of step-by-step instructions to effectively manage the lesson’s progress. Walk through each of the block actions to demonstrate the movements and alleviate any reading barriers that students might have. In addition, it may be helpful for the class to draw pictures representing what each block action is and create a guide to use as they complete the activities. This lesson asks students to implement their code using their own arms prior to using the robotic arm. This is a key characteristic of this lesson, as it involves students in testing their code using an interactive and kinesthetic learning approach. Students with limited motor capabilities should be paired with students who are willing to provide support to access this aspect of the lesson. Overall, the strategies previously mentioned should be used together to differentiate and facilitate the learning of an introductory computing concept for all students. ●

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the National Science Foundation (CNS-1542465) the Nevada Space Grant Consortium (NNX15AI02H).

REFERENCES

National Governors Association Center for Best Practices and Council of Chief State School Officers (NGAC and CCSSO). 2010. *Common core state standards*. Washington, DC: NGAC and CCSSO.

NGSS Lead States. 2013. *Next Generation Science Standards: For states, by states*. Washington, DC: National Academies Press. www.nextgenscience.org/next-generation-science-standards.

RESOURCES

7Bot Desktop Robot Arm working with assembly line—*https://youtu.be/sLKXQvUkWAU?t=24*

Building the MeArm Deluxe Kit—*http://learn.mime.co.uk/docs/building-the-mearm-deluxe*

Materials and support for building the lesson—*https://unpluggedrobotics.blogs.unr.edu*

ONLINE SUPPLEMENTAL MATERIALS

Student worksheets, example sequences, vocabulary, additional materials list, assessment, and assessment answer key—*www.nsta.org/scope1901*

Mercedes Anderson (*mercedesanderson@nevada.unr.edu*) is an undergraduate student in the Department of Computer Science and Engineering at the University of Nevada, Reno, in Reno, Nevada. **Blanca Miller** (*blanca.miller@vectorinstitute.ai*) is researcher and professional development associate at the Vector Institute in Toronto, Ontario. **Jamie Poston** (*jposton@nevada.unr.edu*) is a graduate student in the Department of Computer Science and Engineering at the University of Nevada, Reno, in Reno, Nevada. **Adam Kirn** (*akirn@unr.edu*) is assistant professor of engineering education in the Colleges of Engineering and Education at the University of Nevada, Reno, in Reno, Nevada. **Melissa Jurkiewicz** (*jurkiewicz_ma@mercer.edu*) is assistant professor of science/STEM education in the Tift College of Education at Mercer University in Atlanta, Georgia. **David Feil-Seifer** (*dave@cse.unr.edu*) is assistant professor in the Department of Computer Science and Engineering at the University of Nevada, Reno, in Reno, Nevada.