

**Focus:** Coding a Soil Moisture Sensor

**Grade Level:** 7

**Session Length:** 45-60 minutes

### Driving Questions

- How can we use block-based coding to write code for a soil moisture sensor?

### NGSS Links

- Planning and Carrying Out Investigations
- Designing Solutions

### Systems Thinking Characteristics

- Identifying Hidden Dimensions of the System

*In the seventh session of Project Crystal Code, students apply what they've learned about circuits and block-based coding to write a program for a soil moisture sensor.*

Students revisit the schematic for a soil moisture sensor that they designed in Session 5, then review the four coding concepts that they explored in Session 6. Afterwards, they match the coding concepts to elements of the soil moisture program that they'll need to write and then put together a program to turn a micro:bit computer into a soil moisture sensor.

Optionally, if students have access to real micro:bit computers, they can test and calibrate their soil moisture sensor.

### Learning Outcomes & Assessments

<i>By the end of this module, students will be able to...</i>	<i>You can assess this using...</i>
<b>1. Explain</b> how a soil moisture sensor works.	Field notebook entry; Student reflections
<b>2. Write</b> a block-based program that tells a micro:bit computer to measure the amount of moisture in the soil.	Field notebook entry; Class discussion
<b>3. Reflect</b> on how technology and coding can help us to answer scientific questions.	Student reflections

Session Overview

Section	Description	Length	Format
<b>Launch</b>	Students meet Erick again, who introduces them to their final task in this sequence: writing code for a soil moisture sensor.	5 minutes	Whole class
<b>Explore</b>	Students revisit their schematic for a soil moisture sensor, then review the four coding concepts that they practiced in Session 6. They then plan and construct a program to take soil moisture readings.  Optionally, if students have access to a micro:bit, they can test and calibrate their soil moisture sensor.	30-40 minutes  20-30 minutes	Individual or Research teams
<b>Share</b>	Students share their final results and reflect on the role that technology plays within science.	5-10 minutes	Whole class
<b>Reflect</b>	Students reflect on their experience during Sessions 5-7 and their next steps to test their hypotheses by collecting data.	5 minutes	Individual

## *Overview of the Soil Moisture Sensor Project*

---

In Sessions 5-7 of Project Crystal Code, students will be tasked with designing, coding, and building their own environmental sensors that can measure the amount of water in the soil in our mulch treatments. Although this part of the program is optional, it offers an opportunity for participants to practice computational thinking and coding skills while developing a deeper understanding of the environmental system.

### *Why We Do This*

One component of systems thinking is being able to identify and understand invisible parts of the system. By building and coding their own soil moisture sensor from scratch, students will develop a better understanding of what they're measuring -- and thus how invisible parts of the system work -- than if they simply used a provided soil moisture meter.

This project also offers the opportunity for students to be introduced to block-based coding in the context of solving a real-world environmental problem. Often, coding and computational thinking is taught in isolation from other scientific disciplines. It is our hope that by integrating coding into an environmental science program, we will be able to both help students see how computer science skills have become an important part of scientific work, while also inspiring interest in coding amongst students who might not normally choose to take a traditional computer science class.

*Where Today's Session Fits In*

During the culminating event of Session 7, students apply ideas from Session 5 and Session 6 to plan and write a program that will allow the micro:bit to measure the amount of moisture in the soil.

Throughout all three sessions, it is important to remind students of the ultimate goal so that the sessions don't feel disjointed from the rest of the Project Crystal Code program. By learning how to build and code a soil moisture sensor, we will be able to measure how much moisture is in the soil, allowing us to test our hypothesis about how mulch affects soil moisture.

Session 5		Session 6	Session 7	
<i>Exploring circuits, conductivity, and resistance</i>	<i>Designing a plan to physically build a soil moisture sensor</i>	<i>Practice block-based coding skills</i>	<i>Writing a block-based program to turn a micro:bit computer into a soil moisture sensor</i>	<i>Assembling the soil moisture sensor, downloading the code, and testing it (optional)</i>

***You are here!***

### *Writing a Program for a Soil Moisture Sensor*

This session revisits student understanding about how a soil moisture sensor works (which was introduced in Session 5), and then challenges students to match coding concepts (introduced in Session 6) to elements of a program in order to construct code that will tell a micro:bit computer to read soil moisture data.

This session is designed to position students to do more than simply follow step-by-step instructions. Instead of simply putting coding blocks in the correct order, they are challenged to think through elements of what they want the code to do, and then apply the concepts that they learned in Session 6 to write a program.

In addition to the coding concepts introduced in Session 6, one new concept is introduced in Session 7: using the micro:bit's Pin 1 as an input to measure soil moisture. Because this is a little more challenging to construct in blockcode, the session provides students with a link to a starting template on the MakeCode website that already has this block in place.

If students truly struggle to construct the program or if you are very short on time, this session also provides a link to a MakeCode template which has the necessary coding blocks already selected. You can share this link with students and give them the task of assembling the blocks in the correct order, rather than writing the program from scratch.

### *Testing & Calibrating Your Soil Moisture Sensor*

Obviously, writing a program for a soil moisture sensor does not need to be the final step in this process. If you have access to real micro:bits and your class has additional time, you can also challenge students to test their soil moisture sensors in the real world, comparing dry areas of soil with wet areas.

Additionally (and ambitiously), students can even figure out how to calibrate their soil moisture sensor so that the readings are accurate. This last step can be very challenging and will likely take an additional class period. If you'd like to take on this final challenge, we provide some tips at the end of this session guide on how you might approach this.

### Virtual Materials

---

- Session 7 Google Slides Presentation: <http://bit.ly/3jys9N3>
- Session 7 Coding Instructions: <http://bit.ly/3tlt5ml>
- Session 7 Field Notebook Template (optional): <http://bit.ly/3q6ZLEh>
- MakeCode Template: <http://bit.ly/3aOFIDZ>
- MakeCode Template with all Coding Blocks: <http://bit.ly/2OocNze>

### Each student will need...

---

- A device with internet access (a computer, smartphone, or tablet will all work!)
- Field notebook and pencil
- Access to their soil moisture sensor schematic from Session 5
- Session 7 Coding Instructions (optional)
- Access to a **micro:bit and a micro USB cable** to download the code (optional)
- Additional supplies to construct the soil moisture sensor: 2 **nails**, 2 **wires with alligator clips**

### Before You Start Teaching

---

- Decide how you want students to work on the Microsoft MakeCode website. Students can work independently or in research teams. Note that if students work virtually in teams, one student will need to share their screen with the others and be in charge of manipulating the platform.
- Although it is not necessary to have access to a real micro:bit to practice coding, decide whether you want to acquire micro:bits so that students can download and test their final program. If you do want to have them practice downloading their code and testing it in the real world, [Slides 15-17](#) give specific instructions on how to do this.
- Copy over the [Session 7 Slideshow](#) for your chosen platform to your own Google Drive account. Test to make sure that the videos work. (If not, you may have to check the permissions on the Crystal Cove Conservancy Youtube Account.)
- Decide how you will assess student progress during this lesson. You can ask students to take a screenshot of their final code and paste it into their field notebook (or another virtual document). Alternatively, you can monitor students as they work during the session, ask students to demonstrate their code for you, or have them reflect in their field notebook at the end of the process to check for understanding.

## Learning Sequence

---

### Launch

#### *Putting it All Together (5 minutes)*

1. Open the [Session 7 Slideshow](#) and play the video on [Slide 2](#) for your class. In this video, Khai introduces the task for Session , which is to apply what students have learned in Sessions 5 and 6 to write a program for a soil moisture sensor.
2. After watching the video, move on to [Slide 3](#), which gives an overview of what students will do and learn during Session 7.

### Explore

#### *Part 1: Reviewing What We've Learned (30-40 minutes)*

1. Continue to [Slide 4](#), where Erick gives students an overview of how a soil moisture sensor works. He then asks students to review the soil moisture sensor schematic that they designed in Session 5 to refresh their understanding of how soil moisture sensors work.
2. Advance to [Slide 5](#) and ask students to get into their research teams. Give them a few minutes to review their schematics and discuss how a soil moisture sensor works.
3. Then, move on to [Slide 6](#) and play the video of Erick. Here, he reviews the four coding concepts that students explored in Session 6: Using block-based code to write instructions, using inputs to produce outputs, creating conditional statements, and using variables. In addition, students also learned how to use the math blocks to make calculations.

After giving an overview, Erick asks students to use their field notebooks to review each concept.

4. Move on to [Slide 7](#). Here, students are asked to set up a table in their field notebook, or use the table in the field notebook template to review the six different coding concepts and terms. You can either ask students to break into their research teams to complete the table or go over it together and complete it as an entire class.

<b>LED Display</b>	On a micro:bit, this is a screen that can be programmed to show icons, text, numbers, or a graph.
<b>Input</b>	Something that tells a computer to create an output. For a micro:bit, this can take the form of a button being pressed or information coming from a sensor.
<b>Output</b>	You can tell a computer to create this when it receives a particular input. Often, this takes the form of a graphic shown on the LED display.
<b>Conditional Statement</b>	This tells a computer that <b>IF</b> a specific input is received, <b>THEN</b> it should do something specific.
<b>Variable</b>	Used as a short-cut to help a computer remember specific information.
<b>Math blocks</b>	A type of micro:bit coding block used to add, subtract, multiply, or divide.

**Part #2: Writing a Soil Moisture Sensor Program**

5. Continue to *Slide 8*. Here, Erick returns students to their culminating task: to write a program that will tell a micro:bit to act as a soil moisture sensor. He reviews the requirements for the program, describing the four individual tasks that it will need to perform. Finally, he asks students to match the different coding concepts that they've learned to the requirements for the program.

6. On *Slide 9*, students are challenged to review the requirements for the program and to match a coding concept to each one. Give them time to complete the table -- and then, if there is time, review it as a class.

<i>Soil Moisture Code Requirements</i>	<i>Which coding concept(s) will you use to write this part of the code?</i>
1. Define "Soil Moisture" as information coming in from Analog Pin 1.	Define a variable
2. Take in Soil Moisture data and plot a bar graph of it on the LED display.	Use inputs to produce outputs; Show something on the LED display
3. If button A is pressed, then show a number on the LED display representing the amount of moisture in the soil.	Create a conditional; Use inputs to produce outputs; Show something on the LED display
4. Convert the soil moisture number value (which is out of 1023) into a percentage (out of 100).	Use math blocks to add, subtract, multiply, or divide



7. Move on to [Slide 10](#) and play the video. Here, Erick introduces the final bit of information that students will need to use in order to code their soil moisture meter: using the micro:bit computer's analog pins. This is a little trickier to code, so he says that students will be provided with this bit of code to get them started.
8. When students are ready, move on to [Slide 11](#). Link them to the template on the MakeCode website and challenge them to put together what they've learned as they write a program for a soil moisture sensor!

As students work, circulate between the groups as you're able and observe their discussions. If any research teams are truly struggling to put the code together, you can share the link on [Slide 12](#) to a more structured template, which gives them all of the coding blocks they will need.

### Share

#### *Sharing Our Thoughts (5-10 minutes)*

1. Bring the group back together as a class. Open [Slide 13](#), and give one or two student teams a chance to share the program that they wrote and explain how it works.
2. Move on to [Slide 14](#), and ask students to reflect on the overall process of creating a soil moisture meter. Why did we go through this process? Did they enjoy coding? What did they find challenging? How did they overcome these challenges?

Ask students to reflect on the role that technology plays in science. How can we use things like coding and computer science to answer real-world scientific questions? Elevate ideas about how often during scientific investigations, we need to create tools to help us measure things we can't see (like soil moisture). We can also use technology to create models and analyze data. (You can also draw the connection that just as students used conditional statements in their code, they also used conditional statements to explain relationships between different factors when they were building their model.)

### Extend

#### *Optional Extensions: Testing and Calibrating a Soil Moisture Sensor (30-45 minutes)*

If you have access to physical micro:bits, ask students to download their program onto their micro:bit ([Slides 15-17](#)) so that they can test them in person. After they assemble the soil moisture sensor using the nails and wires, encourage them to use their sensor outdoors to compare wet soil to dry soil. (One way to do this is to measure a dry area, then pour water onto it to see how the reading changes.)

If students continue to experiment with their sensor, they may notice that the soil moisture percentages they receive aren't accurate. The micro:bit's sensors are very sensitive, which means that a very high reading (which reads as 100%) may be closer to 50% or 60% of actual soil moisture.

It is possible -- though challenging -- for students to calibrate the micro:bit sensor by setting upper and lower bounds for the readings. To do this, we recommend completely drying out some soil and then taking a measurement to set as a lower bound (0%). (An easy way to dry out the soil is by baking it at a low temperature in the oven.) After, add an equal weight of water to the soil and take a reading to set as 50% soil moisture. You can then use the math blocks to convert the original voltage reading into a more accurate soil moisture percentage.



Reflect

*Reflecting on Session 6 (5 minutes)*

1. At the end of the discussion, advance to **Slide 18** in the slideshow and play the video, where Khai will invite them to spend a few minutes reflecting on their experiences in Sessions 5-7.
2. Move on to **Slide 19**, which will share reflection questions. Ask students to spend five minutes reflecting on their experiences today in their field notebook.
3. Finally, thank the class for their time today. Tell them that when you gather again, they will help Crystal Cove Conservancy record data for our three research questions so that we can test our hypotheses!