



MISSION 9: All Systems Go! Lesson 1 (Objectives 1-4)		Time Frame: 45-50 minutes									
<p>Project Goal: Students will use system sensors to monitor battery power and display battery health with a UI.</p> <p>Learning Targets</p> <ul style="list-style-type: none"> I can use system sensors to read battery voltage and power source. I can print the capacity percentage using a table. I can calculate the capacity percentage using the equation of a line. I can create a user interface to display battery health using the power LED. 		<p>Key Concepts</p> <ul style="list-style-type: none"> The 'bot can measure its own battery voltage, and also if the 'bot is powered by USB or batteries. The data returned by the system sensors can be used to show alerts to avoid problems. Even though CodeBot doesn't have a screen, its LEDs can be used to display sensor data. A graph of battery discharge can be approximated with a straight line. Another LED on CodeBot is just above the power switch. It can be turned on/off with code. 									
<p>Assessment Opportunities</p> <ul style="list-style-type: none"> Mission 9 Lesson 1 Mission Log Submit completed program BatteryTest Mission 9 Obj. 1-4 Review Kahoot! 		<p>Success Criteria</p> <ul style="list-style-type: none"> <input type="checkbox"/> Read battery voltage using system sensors <input type="checkbox"/> Indicate battery or USB power using system sensors and the power LED <input type="checkbox"/> Display capacity percent using a table <input type="checkbox"/> Calculate and display capacity percent using the equation of a line <input type="checkbox"/> Use the power LED as a UI to communicate battery health 									
<p>Teacher Materials in Learning Portal</p> <ul style="list-style-type: none"> Mission 9 Lesson 1 Slides Mission 9 Lesson 1 Mission Log Mission 9 Lesson 1 Mission Log Answer Key 		<p>Additional Resources</p> <ul style="list-style-type: none"> Mission 9 Obj.1-4 Review Kahoot! BatteryTest_obj2 sample code (learning portal) BatteryTest_obj3 sample code (learning portal) BatteryTest_final sample code (learning portal) BatteryTest_ext sample code (learning portal) 									
<p>Vocabulary</p> <ul style="list-style-type: none"> System sensors: Sensors that read internal settings, like power and temperature. Under load: Batteries are being used to power something, like turning on LEDs or running motors. Float (review): A decimal number, either positive or negative. y = mx + b: The equation of a straight line. User Interface (UI): A way for a person and a machine to communicate, which includes screen, keyboards and LEDs. 											
<p>New Python Code</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #FFF9C4; padding: 5px;"><code>system.pwr_volts()</code></td> <td style="padding: 5px;">Returns a float (decimal number) for the current voltage, either from USB or batteries.</td> </tr> <tr> <td style="background-color: #FFF9C4; padding: 5px;"><code>system.pwr_is_usb()</code></td> <td style="padding: 5px;">Returns an integer 0 if the power switch is set to batteries and 1 if USB.</td> </tr> <tr> <td style="background-color: #FFF9C4; padding: 5px;"><code>leds.user(15)</code></td> <td style="padding: 5px;">Turn on the first four user LEDs so the battery is under load.</td> </tr> <tr> <td style="background-color: #FFF9C4; padding: 5px;"><code>leds.user(0)</code></td> <td style="padding: 5px;">Turn off all user LEDs.</td> </tr> </table>				<code>system.pwr_volts()</code>	Returns a float (decimal number) for the current voltage, either from USB or batteries.	<code>system.pwr_is_usb()</code>	Returns an integer 0 if the power switch is set to batteries and 1 if USB.	<code>leds.user(15)</code>	Turn on the first four user LEDs so the battery is under load.	<code>leds.user(0)</code>	Turn off all user LEDs.
<code>system.pwr_volts()</code>	Returns a float (decimal number) for the current voltage, either from USB or batteries.										
<code>system.pwr_is_usb()</code>	Returns an integer 0 if the power switch is set to batteries and 1 if USB.										
<code>leds.user(15)</code>	Turn on the first four user LEDs so the battery is under load.										
<code>leds.user(0)</code>	Turn off all user LEDs.										



<code>pct = (v / 2) - 2</code>	Use the equation of a line to calculate the percentage, given the volts.
<code>leds.pwr(True)</code> <code>leds.pwr(False)</code>	Turn on the LED indicator for power. Turn off the LED indicator for power.

Real World Applications

You already use this kind of code daily! Many electronic devices check for battery power to ensure device reliability, extend battery life, and prevent unexpected failures. Some examples are:

- Cell phones and laptops
- Electric vehicles
- Industrial robots
- Hospital and emergency services equipment

Teacher Notes:

- This lesson follows the instructions in CodeSpace fairly closely. It is chunked into smaller bits of information and it occasionally gives extra examples or definitions.
- The code in this lesson is similar to CodeTrek. It is simplified a little for ease of typing, and some non-required code is left out. All goals will be met.
- The activities include a lot of testing, which uses the console panel for printing results. The console panel can be resized so show more text. Just drag up the gray line above the tab headings for a longer window.
- All CodeBots should have batteries in their battery pack for testing. The batteries don't need to be fresh. In fact, having batteries at a variety of power levels can be more interesting.
- The lesson uses the equation of a line to calculate the percentage. If your students haven't learned the equation of a line yet, you can just give them the code, or have a short lesson on the concept.

Extensions / Cross-Curricular:

- Create an additional user interface to display battery health. Turn on all user LEDs for full power, and decrease the number of turned on LEDs as the battery power decreases. When all user LEDs are off, the battery is close to dead.
- **SCIENCE:** Have a lesson on electricity, voltage and/or battery power. Compare battery power to USB power from the computer.
- **MATH:** This lesson uses percentages. Have a lesson on how to calculate a percent.
- **MATH:** This lesson uses the equation of a line to calculate percentages. Have a lesson on finding the equation of a line. Or have students graph line equations.
- **STEM:** Suppose CodeBot had a display screen. Design a UI that would display battery health and any other system information you are interested in.
- Supports **language arts** through reading instructions, guided notes, and reflection writing.

Preparing for the lesson:

- If your students skipped Mission 8 as recommended, make sure you have entered the classroom dashboard and unlocked Mission 8 so students can continue to Mission 9.
- Look through the slides. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS.
- All CodeBots need four AA batteries. They do not need to be fresh.
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.



Lesson Tips and Tricks:

💡 Teaching tip:

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods.

👥 Pre-Mission Warm-up: -- slide 2

Students can write in their log first and then share, or discuss first and then write in their log. The warm-up questions review CodeBot movement. Students can share their answers, or compare with each other.

- Question: What are some sensors on CodeBot you know how to use?
- Question: What other information from a CodeBot sensor would you like to have access to?

📄 Mission 9 Lesson 1 Activities:

The Chrome browser works best, but other browsers also support CodeSpace. Each student will complete a Mission Log. Students could work in pairs through the lesson, or they can work individually. This lesson requires testing during each objective, so working in pairs could be advantageous.

💡 Teaching tip: Mission Introduction -- slides 3-4

This mission is divided up into three lessons. This lesson completes the first goal. Students answer one question in their mission log.

💡 Teaching tip: Objective #1 -- slides 5-6

Battery voltage is introduced. Students learn about CodeBot battery power and testing under load.

💡 Teaching tip: Objective #1 -- slides 7-9

Students learn about the functions used to read system sensors to track battery voltage and the power source.

💡 Teaching tip: Objective #1 Activity -- slides 11-15

Students use the Console Panel to practice using the battery functions. Students record testing results in the mission log.

💡 Teaching tip: Objective #2 -- slides 16-18

Students look at a chart that plots the discharge of a battery over time. The chart is simplified to a straight line, and a chart of voltage points is created. This data is used to convert voltage into capacity percentage.

💡 Teaching tip: Objective #2 -- slide 19

Algorithm for the battery tester program.

💡 Teaching tip: Objective #2 Activity -- slides 20-25

Students start a new file for the battery tester program. They will create two functions and a main program. All the code is given in the slides. Students follow all the instructions and record two results in the mission log.

💡 Teaching tip: Objective #3 -- slides 26-27

Students learn a more accurate way to calculate the capacity percentage is to use the equation of a line. A short description of the parts of the equation is given.

💡 Teaching tip: Objective #3 Activity -- slides 28-32

Students modify one of the functions to use the equation of a line to calculate the capacity percentage. All the steps are given in the slides. Students follow all the instructions and record results in the mission log. They will run the program several times to test all the data.

💡 Teaching tip: Objective #4 -- slides 33-34

The user interface is introduced, and students are given a way to communicate battery health using the power LED.



💡 Teaching tip: Objective #4 -- slides 35-39

The algorithm for the battery health UI. The code is given step-by-step with explanations. Students can refer back to the algorithm if needed when completing the activity.

💡 Teaching tip: Objective #4 Activity -- slides 40-45

Students add a function for the battery health UI. The code is given, and then step-by-step instructions for testing. Again, students record the results in the mission log. For the last test data case, students select their own battery voltage.

💡 Teaching tip: Extension – slide 46

One extension suggestion is given for the lesson. Students can use the user LEDs as a battery health UI. There are many ways to accomplish this task. Sample code with one solution available in the learning portal.

Optional:  Mission 9 Obj 1-4 Kahoot! Review. A review Kahoot! Is available for these four objectives.

 Post-Mission Reflection:

The post-mission reflection asks students to think about how a battery tester can be used in other CodeBot programs. Answers can vary widely.

You can use a cross-curricular activity for a post-mission activity.

End by collecting the Mission 9 Lesson 1 Log.

SUCCESS CRITERIA:

- Read battery voltage using system sensors
- Indicate battery or USB power using system sensors and the power LED
- Display capacity percent using a table
- Calculate and display capacity percent using the equation of a line
- Use the power LED as a UI to communicate battery health