



MISSION 5: Fence Patrol Lesson 1 (Objectives 1-2)

Time Frame: 35-40 minutes

Project Goal: Students will read and display data from CodeBot's line sensors

Learning Targets

- I can use `ls.read()` to get real-time line sensor values.
- I can display values in the console panel's debug window to get real-time sensor values.
- I can print values to the console panel while running a program to display real-time values.
- I can assign a Boolean value to a variable by using an if statement as part of the assignment statement.

Key Concepts

- Analog sensors are non-contact sensors used in many industrial and commercial applications.
- The console panel can be used to print real-time data using the `print()` statement.
- A dark line on a light background will have a larger value, toward the high end of the range. A light line on a dark background will have a smaller value, toward the low end of the range.
- A variable can be assigned a Boolean value by using an if statement as part of the assignment statement.

Assessment Opportunities

- Mission 5 Lesson 1 Log (digital)
 - Includes a chart for recording reflective values of different surfaces
- Submit completed program **LineSense**
- [Mission 5 Obj. 1-2 Review Kahoot!](#)

Success Criteria

- ☐ Read values from a single line sensor
- ☐ Use the debugger to see values in real time
- ☐ Print a value to the console panel
- ☐ Assign a Boolean value to a variable using an if statement

Teacher Materials in Learning Portal

- Mission 5 Lesson 1 Slides
- Mission 5 Lesson 1 Log
- Mission 5 Lesson 1 Answer Key

Additional Resources

- [Mission 5 Obj. 1-2 Review Kahoot!](#)
- LineSense_obj2 sample code (learning portal)
- Testing Surfaces paper (learning portal)
- More surfaces for testing reflectivity, such as a student desk, a folder, a piece of carpet, etc.

Vocabulary

- **Line sensors:** Photo reflective sensors that detect lines and boundaries beneath your 'bot.
- **API:** Application Programming Interface – the details of how your program interacts with different services it needs.
- **Analog:** Infinite variation, like from dark to light or cold to hot.
- **Digital:** Incremental variation, within a specific range.
- **ADC:** Analog to Digital Converter.

New Python Code

| | |
|---|--|
| <code>val = ls.read(0)</code> | Read a line sensor. The number of the line sensor is the (argument). It returns an integer between 0 and 4095. |
| <code>print(val)</code> | Print the value of a variable to the console panel. |
| <code>print("Line sensor value = ", val)</code> | Print the value of a variable with a text message. |

Real World Applications

Many everyday items use sensing and control techniques. Two examples are robots and self-driving cars.



Teacher Notes:

- The slides should replace the instructions in CodeSpace. Code will be similar to CodeTrek, but a little different. All goals will be met.
- Objective 1 requires students to use the debugger. It has been awhile since students used the debugger. You may want to do this part together, or model it first.
- In objective 2, there is a slide on APIs. This isn't necessary for the lesson, and you can skip it if you want to.
- Extra slides are added that go into a little detail about analog and digital. You can use them or skip over them. This lesson is on the shorter side, so you have time to discuss the topic.

Extensions / Cross-Curricular:

- Expand the chart on reflectivity by adding more surfaces.
- **MATH:** Add an additional chart for distance from the sensor. Use the same surfaces, but increase the distance from the sensor. Compare the values. Create a line graph for the data.
- **SCIENCE:** Expand on the information about analog and digital. Look at examples of each and compare the data that each produces.
- Supports **language arts** through reading instructions, guided notes, and reflection writing.

Preparing for the lesson:

- Look through the slides and workbook. 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. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS.
- Print the Test Surfaces paper for each student or programming pair. You will also want additional surfaces available for students to use for testing. They can be things like a folder or notebook, a piece of carpet, colored construction paper, shiny wrapping paper, etc.
- 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 question asks students to think about sensors and how they might be used. Students can share their answers, or compare with each other.

- Question: What are some things you would like CodeBot to detect?

💻 Mission 5 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. Part of the mission log is to fill out a chart, so working in pairs is a good suggestion for this lesson.

💡 Teaching tip: Mission Introduction -- slides 3-8

This mission is divided up into four lessons. The first lesson focuses only on the first part of the first goal. Students answer three questions in their mission log.

💡 Teaching tip: Objective #1 -- slides 9-10

These slides introduce the line sensors and their parts.



 **Teaching tip: Objective #1** -- slide 11

This slide gives the definition of an API and a few examples. It isn't necessary for the lesson and can be skipped. It is included here because it is in the CodeSpace instructions.

 **Teaching tip: Objective #1** -- slide 12

This slide introduces analog data and that it is used by line sensors. This information is from CodeSpace.

 **Teaching tip: Objective #1** -- slides 13-17

Additional information about analog and digital data. These slides are optional, but recommended.

 **Teaching tip: Objective #1** -- slide 18

This slide talks about analog to digital conversion (ADC).

 **Teaching tip: Objective #1** -- slide 19

This slide completes the instructions from CodeSpace by showing the function for reading a line sensor.

 **Teaching tip: Objective #1 Activity** -- slides 20-22

Students start a new file and copy the code from the slide. Then they must use the debugger to meet the goals. You may need to do this objective together, or model it first for the students. Make sure the console panel is open, and students have clicked on the "Debug" window. They will not stay on the "Console" window. Then they have to click on "Globals" to see the val variable.

 **Teaching tip: Objective #2** -- slides 23-25

This objective introduces the print() statement. Students do not need to use the debugger, but they will open the console panel. If the "Debug" window is still active, they need to click on "Console" to make that the active window. The prints will be displayed there.

The concept of REPL is not addressed here. It is introduced and used in Mission 6.

 **Teaching tip: Objective #2 Activity** -- slides 26-28

Students modify their code with the lines on slide 28. They open the console panel and observe the line sensor readings. This code adds a sleep() to slow down the display. It is slightly different from CodeTrek.


 **Teaching tip: Objective #2 Mission Log** -- slide 29

Students complete the mission log by using different testing surfaces underneath CodeBot and recording the line sensor reading. Have a test surface paper printed for each student or programming pair. Also have other surfaces available for testing. The chart uses the three test surfaces, plus five more of the students' choice. You can shorten or lengthen the chart, depending on the pacing of the lesson.

At the end of this objective, students should have a working program to turn in.

 **Teaching tip: Extension**

There isn't an extension planned for this lesson. However, if you have time remaining in your class, you can have students test more surfaces. Or you can add a chart to the lesson and change the distance of the testing surface. The first chart places all testing surfaces directly under CodeBot. Students can experiment with the same surfaces at different distances from the line sensor. They can even create a graph.

Optional:  Mission 5 Obj 1-2 Kahoot! Review.

A review Kahoot! is available for these two objectives.

 **Post-Mission Reflection:**

The post-mission reflection asks students to review the values of a line sensor reading, and think about future uses of line sensors.

You can use an extension or cross-curricular activity as post-mission activity.

End by collecting the Mission 5 Lesson 1 Log.



SUCCESS CRITERIA:

- ☐ Read values from a single line sensor
- ☐ Use the debugger to see values in real time
- ☐ Print a value to the console panel
- ☐ Assign a Boolean value to a variable using an if statement