



MISSION 5: Fence Patrol Lesson 3 (Objectives 6-8)

Time Frame: 40-45 minutes

Project Goal: Students will use data to count line detections and display the count in binary using the red user LEDs.

Learning Targets

- I can add a “wait” feature to a program.
- I can define a function that returns a Boolean value.
- I can make a contact counter to show each line-detect on the user LEDs.
- I can use a programming technique to avoid a runtime error.

Key Concepts

- Engineers build in safety features, like waiting for a button press before starting.
- Reusing code is fundamental. This lesson uses the same or similar code from two previous programs.
- The use of Boolean values (and variables) is also fundamental. They can be used as a toggle, like turning on/off LEDs, and also as a detection tool.
- Functions can return a value as well as receive one. To return a value, use the “return” statement. If a “return” statement isn’t explicitly given, the function will return None.

Assessment Opportunities

- Mission 5 Lesson 3 Log (digital)
- Submit completed program *LineSense*
- [Mission 5 Obj. 6-8 Review Kahoot!](#)

Success Criteria

- ☐ Use a while loop to wait for a button press
- ☐ Use a return statement in a function
- ☐ Call a function that has a return statement
- ☐ Use a function’s return Boolean value as a condition
- ☐ Define and increment a counter variable
- ☐ Display a number in binary using LEDs
- ☐ Avoid a runtime error by resetting a counter variable when it reaches its maximum number

Teacher Materials in Learning Portal

- Mission 5 Lesson 3 Slides
- Mission 5 Lesson 3 Log
- Mission 5 Lesson 3 Answer Key

Additional Resources

- [Mission 5 Obj. 6-8 Review Kahoot!](#)
- LineSense_obj5 sample code (learning portal)
- Testing Surfaces paper (learning portal)

Vocabulary

- **Return statement:** Exits the function and sends a value back to the code where the function was called.
- **Runtime error:** A coding error that happens when the program is actively running.

New Python Code

<pre>while True: if buttons.was_pressed(0): break</pre>	<p>(Review) Wait loop. A safety feature; the ‘bot waits until BTN-0 is pressed before continuing the program.</p>
<pre>return is_detected return got_line</pre>	<p>Function return The value of the variable is returned to the function call.</p>
<pre>hit = scan_lines() if detect_line()</pre>	<p>Function call The value of the return is used in the assignment or if statement.</p>
<pre>leds.user(line_count)</pre>	<p>Use a variable to turn on user LEDs. Line count needs to have a value from 0 to 255.</p>



```
line_count = line_count + 1
if line_count == 256:
    line_count = 0
```

Reset a counter variable when it reaches its maximum number.

Real World Applications

Many everyday items count things. Computers and sensors are used in many real world applications, especially to gather data for better decision making. Some examples are:

- In manufacturing, like counting parts on an assembly line
- In retail, such as customers entering stores or how many items of something are on the shelf
- In agriculture, with crop monitoring and livestock counting
- In healthcare, such as counting blood cells
- In smart cities, controlling traffic flow and monitoring pedestrians

Also, many of the real world applications don't have a computer screen to display information. LEDs and other methods of visualizing data are seen throughout everyday items.

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.
- This lesson uses the "wait" code segment from Mission 4 Lesson 3 RobotMoves. Students should have access to the program so they can copy and paste the code.
- The lesson introduces returning a value from a function. Some examples are given, and students have some sample code to try in the mission log.
- The code for resetting a count variable when it reaches its max value is reviewed from Mission 4 Lesson 1 SweepLEDs. Students need to access the code, but you can review it if you want.

Extensions / Cross-Curricular:

- Play a tone when the counter variable is incremented.
- **MATH:** This program uses a function that returns a value. Discuss what a function is in math. Four sample functions are given in the mission log. Students can also write their own and practice evaluating functions and returning the value.
- **MATH:** Review binary numbers and why 255 is the maximum number for 8 LEDs. Come up with other problems and discuss how many bits, or LEDs would be required to display the max number. For example, how many bits are required to keep track of all student numbers, or books in a collection?
- 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. There are a couple of tables to fill out. Prepare the assignment to give through your LMS.
- Have the Test Surfaces paper available for each student or programming pair.
- 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 reviews the parts of a function and the vocabulary associated with it. Students can share their answers, or compare with each other.

- Question: Fill in the chart by identifying the parts of a function (function definition, parameter, argument and function call)

Mission 5 Lesson 3 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.

Teaching tip: Mission Introduction -- slides 3-5

This mission is divided up into four lessons. The third lesson focuses only on the third goal.

Teaching tip: Objective #6 -- slides 6-11

This objective reviews the need for safety when operating a robot. The code from Mission 4 Lesson 3 is referenced. Students can copy and paste from the previous program into their new program. When they run the code, nothing should happen until BTN-0 is pressed.

Teaching tip: Objective #7 Part 1 -- slides 12-16

This is a long objective, so it is divided into two parts. Part 1 discusses adding a return in a function. A definition and examples are given. Students practice working with functions that return a value in their mission log.

Teaching tip: Objective #7 Part 1 Activity -- slides 17-20

Students add a return statement to both functions. After doing this part of the code, students should NOT run the program. It is continued in Part 2, and then ready for execution.

Teaching tip: Objective #7 Part 2 Activity -- slides 21-25

This part of the lesson completes Objective 7. In part 1, the functions were modified. In part 2, the main program is modified. After completing this activity, the program is ready for testing.

Teaching tip: Objective #8 -- slides 26-29


This objective is fairly short. It fixes the runtime error by resetting the count variable when it reaches its maximum value. This was also done in Mission 4 Lesson 1 and can be referenced.

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

Teaching tip: Extension

One suggestion is given for an extension. It is completely optional, but if you have students who always work more quickly than other students, these are a good way to challenge them. It adds a beep to the counter. This might drive you a little crazy!

If any extension is completed, students should do a File–Save As and give the extension program a new name. The LineSense program will be continued (and completed) in the next lesson.

Optional:  Mission 5 Obj 6-8 Kahoot! Review.

A review Kahoot! Is available for the three objectives.

Post-Mission Reflection:

The post-mission reflection asks students to think about functions and returns. They are asked to review the code for functions that have a return.

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



End by collecting the Mission 5 Lesson 3 Log.

SUCCESS CRITERIA:

- ☐ Use a while loop to wait for a button press
- ☐ Use a return statement in a function
- ☐ Call a function that has a return statement
- ☐ Use a function's return Boolean value as a condition
- ☐ Define and increment a counter variable
- ☐ Display a number in binary using LEDs
- ☐ Avoid a runtime error by resetting a counter variable when it reaches its maximum number