Edison and the mystery line

Did you know that Edison robots have different sensors that can detect different things? One of these sensors is the line tracking sensor. Can you program Edison to use this sensor to follow a mysterious line?

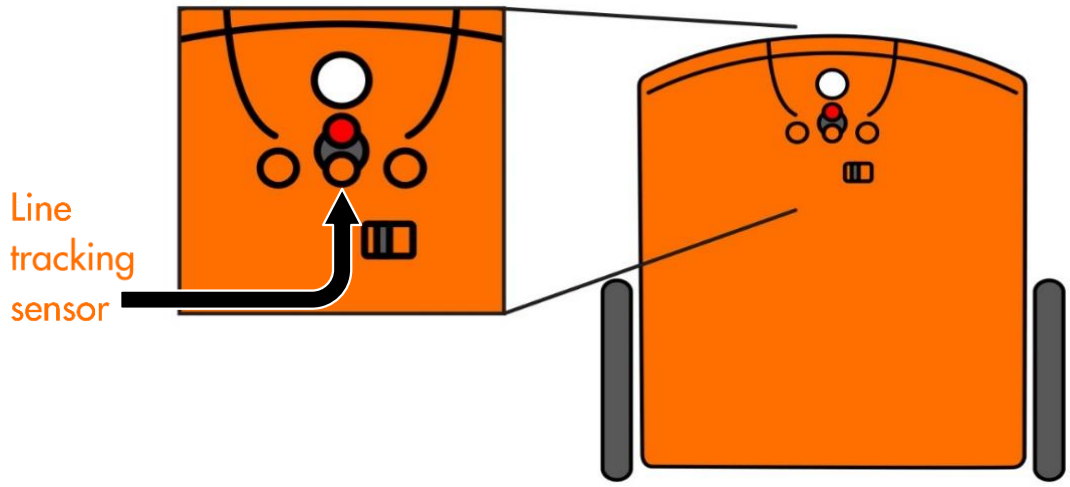
To program Edison to follow the mystery line, there are a few things we need to learn:

- ☐ Part 1: How does Edison's line tracking sensor work?
- ☐ Part 2: How do you master the mystery line?

Is this your first time using Edison robots or EdScratch? Start with the activity *Let's get started with Edison and EdScratch* first! Ask your teacher for a copy.

Part 1: How does Edison's line tracking sensor work?

The line tracking sensor is the sensor that lets Edison see the difference between dark and light surfaces. The sensor is located on the bottom of Edison, near the power switch.



The line tracking sensor is made up of two parts: a red LED and a light sensor. Look at your Edison robot's line tracker. Do you see the two parts of the sensor?

The line tracking sensor works by shining light from the red LED onto the surface below the robot. The light sensor then measures how much of that light bounces up from the surface. Edison stores the value of the reflected light as a light reading. The more light that is reflected back to Edison, the higher the light reading.

Will a white surface or a black surface reflect more light back to Edison?

Use the activity sheet on page 8 to test whether a white or black surface is more reflective to Edison. Turn Edison on and press the round button twice so that the red line tracking LED comes on. Lift Edison up from the paper slightly and have a close look at the round spot of light that the LED shines on the surface. Compare how bright the spot of light appears when placed on a black surface and then on a white surface.

1. Which surface reflects more light back to Edison, a white or a black surface? Why do you think that?

Hint!

The more light that is being reflected, the brighter the spot will appear on the surface below.

By measuring how much reflected light is coming from the surface below the robot, the line tracking sensor lets the robot 'see' the difference between dark and light surfaces. Edison doesn't see colours like a human does, however.

The robot can only tell if a surface is reflective or non-reflective. A reflective surface will shine back a lot of light from the red LED, and a non-reflective surface will shine back very little light.

Edison sees white surfaces as reflective and black surfaces and non-reflective. What about other colours?

2. Will Edison see a red surface as reflective or non-reflective? What about a blue surface? Or green? Use the activity sheet on page 8 to test all three colours using the red line tracking LED.
   Hint: if there is a bright spot similar to what you see on a white surface, a lot of light is being reflected, and the robot will see that colour as 'reflective'.

Colour	Reflective or non-reflective?
Red	
Blue	
Green	

We can use Edison's sensors to create inputs in EdScratch programs, telling Edison to look for different types of events and instructing the robot what to do when those events occur.

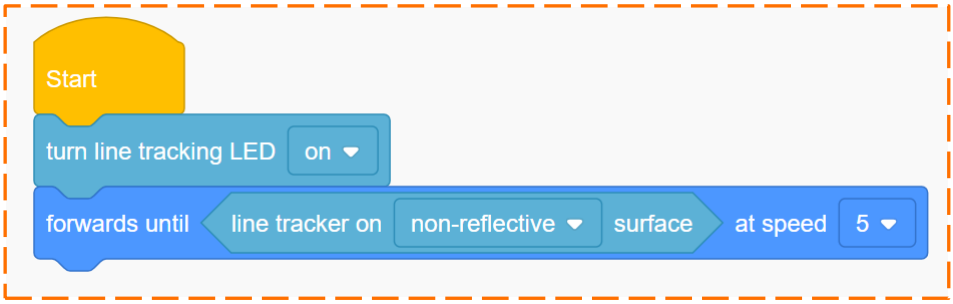
Jargon buster

Inputs are the information and instructions that you give a computer. When you write a program for your Edison robot, you are telling the robot what you want it to do by giving it inputs. Edison's microchip then processes the information to tell the robot what to output as part of the input-process-output cycle.

An event is something that happens outside of the program code that affects how the program runs. An event might be a button being pressed or information being relayed from a sensor.

Let's try using the line tracking sensor in a program which tells Edison to drive until it detects a black line. To write this program, you will need to use blocks from the Sensing category in EdScratch.

Look at this program:



The first code block in this program turns the line tracking LED on. Whenever you want to use the line tracking sensor in a program, you need to turn it on.

Why is that?

Some of Edison's sensors are always on and checking for events. The sound sensor that can detect claps is an example of this 'always on' type of sensor.

Other sensors, like Edison's line tracker, are off by default. You need to include code in your program to turn these sensors on. Just turning the line tracking LED on isn't enough, however. You also need code to tell the sensor what event to check for (reflective surface or non-reflective surface) and what to do if that event is detected.

Write the program in EdScratch and use the activity sheet on page 8 to test it with your Edison robot. Line your robot up on the outline facing the black line on the activity sheet and run your program. Does Edison stop at the black line?

What other things do you think you might be able to program Edison to do using the line tracking sensor?

Part 2: How do you master the mystery line?

Believe it or not, you can get Edison to follow a line neither you nor the robot has ever seen!

Edison's line tracking sensor can detect if the surface below the robot is reflective or non-reflective. You can use this sensor to get Edison to behave in different ways, such as driving until it detects a black line, then stopping. You can also use this sensor to program Edison to follow a black line, even if you don't know what that black line looks like. To do this, you first need to create an **algorithm**.

Jargon buster

An **algorithm** is a broad set of instructions to solve a set of problems. An algorithm lays out a process or a set of rules to be followed in order to solve any problem in the set.

Computer programs often use algorithms, but programs and algorithms are not the same thing.

A computer program is a collection of instructions that tell a computer to perform a specific task. An algorithm lays out the logic for how to solve a whole set of problems, not just one specific task. You can write a computer program that uses an algorithm, but not all computer programs are algorithms.

Algorithms are really helpful because using an algorithm lets a person, or a computer, solve a whole set of problems, even if you don't know every detail.

In computer programming, we often want to create instructions for a computer to follow in order to solve a whole set of problems. By using an algorithm, we can write a program that will let the computer solve any problem in the set. Without an algorithm, we would need to write a new program for every single problem individually.

Why is that?

Let's say you want to teach your friends how to make fruit pies. If you know that all of your friends have apples, you can just write down one recipe for apple pie.

Not all of your friends might have apples, however. What if one of your friends has blueberries, another has cherries, and a third has apples? They cannot all follow the apple pie recipe. You would need to write a separate recipe for each different fruit.

What if you don't know what fruit each of your friends has? How could you teach them to make fruit pies?

No matter what fruit they have, all of your friends need to follow the same basic instructions: make the dough, fill the pie with the fruit, then bake the pie.

This new set of instructions is an example of an algorithm.

Algorithms let you write computer code that will solve whole sets of problems rather than needing to write a new program for every single problem individually.

Task 1: Follow a black line

You know that you can use Edison's line tracking sensor to detect black (non-reflective) and white (reflective) surfaces. We can create an algorithm that uses this sensor to get Edison to follow any black line.

Why is that?

Let's say you draw a black line for Edison to follow. To get Edison to follow your line, you could write a program which makes Edison drive the exact path of the line. If you make a new line, however, you will need to write a whole new program for that new line.

Instead, you can create an algorithm.

This algorithm will solve a set of problems: 'follow any black line'. Any specific line you make for Edison to follow is a new problem inside this set.

Using the algorithm to guide the logic, you can then write a program which will work for all of the problems in the set. This way, a whole new program isn't needed for each new problem.

You can plan out an algorithm using pseudocode, just like you do when you plan out a program.

Jargon buster

Pseudocode is a way of writing out a program in a simple, easy-to-read format. Instead of worrying about syntax, pseudocode uses normal words to descrive what the program will do.

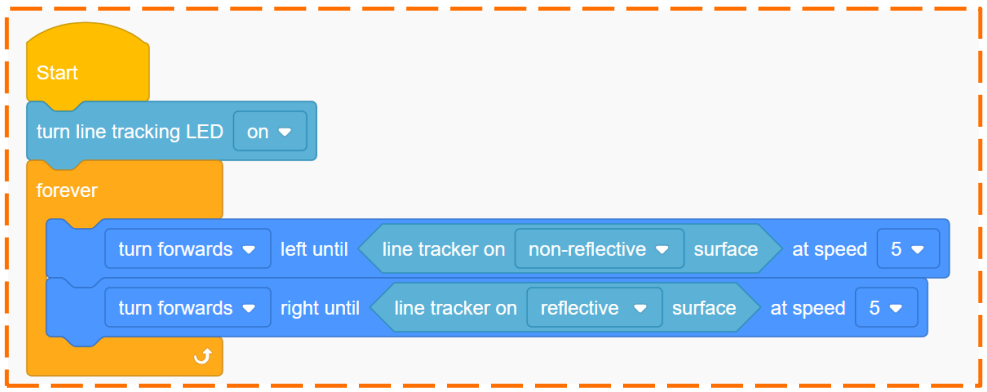
Pseudocode looks a bit like a simplified programming language, but it isn't based on any specific programming language. That's why pseudocode can be used to plan programs in any coding language.

Here is an algorithm in pseudocode that will allow Edison to follow any black line:

```
turn line tracker on
loop forever
        drive forwards left until the robot detects a black surface
        drive forwards right until the robot detects a white surface
```

This algorithm says that the robot should drive forwards to the left until the line tracking sensor is on a non-reflective (black) surface. Then the robot should drive forwards to the right until the line tracking sensor is on a reflective (white) surface. The robot should keep doing this behaviour forever.

This algorithm can be used to write a program in EdScratch:



Can you see how the algorithm's logic is inside the program?

Write the line-following program in EdScratch and download it to your Edison robot. Use the activity sheet on page 9 to test the program. You need to start Edison with the line tracker on a white surface, not directly on the black line.

1. How does the robot move when you run the program? Look at the program and think about the logic in the algorithm. Why does the robot move that way?

Task 2: Follow a mystery black line

The program you wrote uses an algorithm that is designed to solve any problem in the set of problems: 'follow any black line'. That means this same program should let Edison follow any black line you, or anyone else makes!

Make your own line to test. Use a black marker on white paper or make a line using black tape on the floor or a desk. Run the program in Edison to test out your line. Can Edison follow your line?

2. Was Edison able to follow your line? If you had any problems, describe them. What do you think caused the problems?

If you want, you can take the mystery further. Make new black lines and exchange them with a partner. Will your program be able to successfully complete the new mystery line?

Extra challenge! There's more than one way to follow a line

Just how many different ways could you write a program in EdScratch that will solve the 'follow any black line' set of problems? More than you might think! This challenge is to write at least two different programs that use the basic logic of the 'follow any black line' algorithm.

Hint!

Think about how you can apply the logic of the algorithm in EdScratch. Look at different conditionals, including until blocks, if blocks and if-else blocks. Don't forget about the Events category too!