

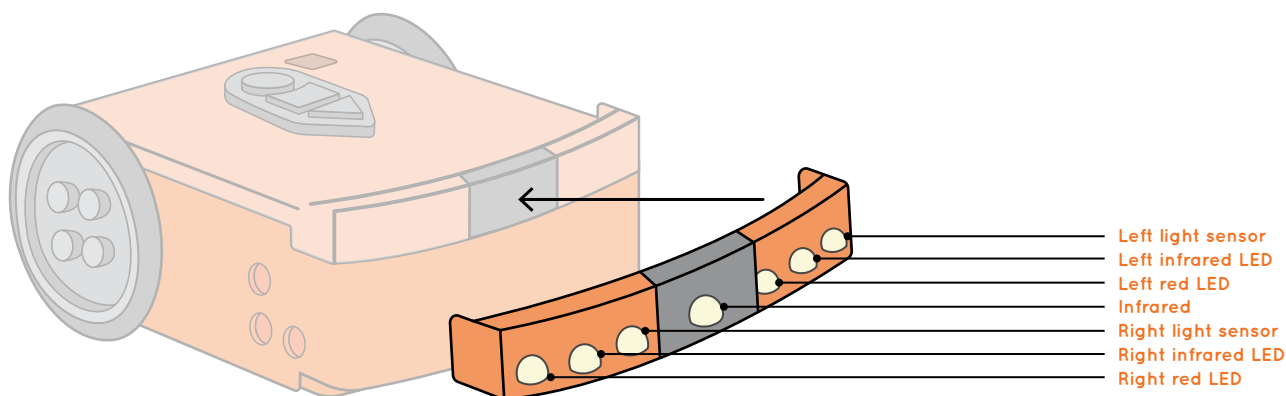
## Let's explore Edison's obstacle detection

Edison robots have different sensors that can detect different things. One of these sensors is the infrared light sensor which we can use to detect obstacles.

### Meet Edison's infrared light sensor

Edison's infrared light sensor is the sensor that lets Edison emit and detect infrared light. The sensor is made up of three parts all located across the front of Edison: two infrared LEDs (one on the right and one on the left) plus an infrared receiver in the middle. Look at your Edison robot. Do you see the different parts of the sensor?

Just like Edison's two red LEDs, the two infrared LEDs can emit light. However, because it's infrared, you won't be able to see it.



### Why is that?

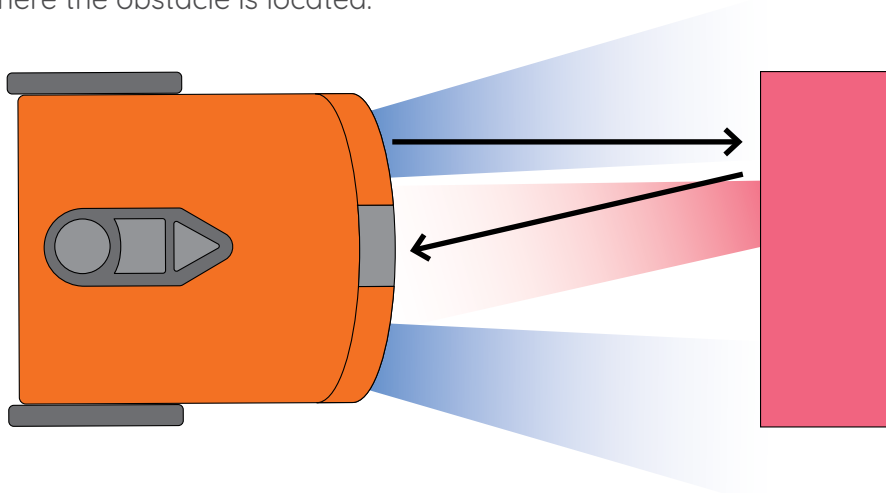
There is a wide range of light. People can see some of this range, but not all of it. Infrared light (which is also called IR light) is not visible to humans.

Even though we cannot see it, people use infrared a lot. For example, infrared light is used in TV remote controls. It's how the remote tells the TV to change the channel or turn up the volume!

Unlike us, Edison can detect infrared light using the infrared receiver on the front of the robot. One way we can use Edison's infrared light sensor is to detect obstacles.

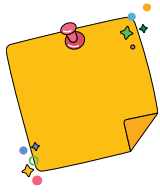
Edison can emit infrared light from the two infrared LEDs. If that infrared encounters an obstacle, like a wall, the light is reflected back towards Edison. Edison's infrared receiver detects the reflected light, telling Edison that there is an obstacle.

Depending on where the obstacle is, infrared light will bounce back from the left LED, the right LED or both. The reflected light tells the robot where the obstacle is located.



## Forward until obstacle

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.



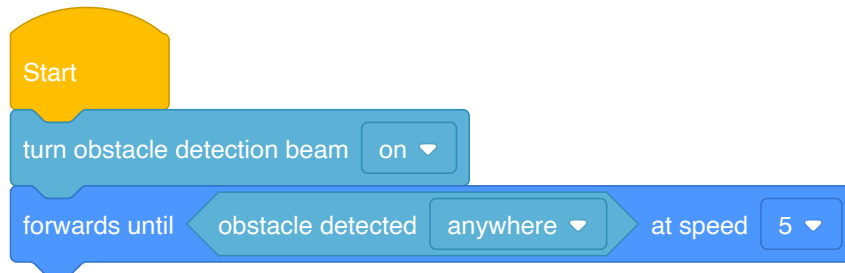
### Don't forget

**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 infrared sensor in a program which tells Edison to drive until it detects an obstacle. 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 on the obstacle detection beam.

Write the program in EdScratch. Download the program and test it using your Edison robot. You will need an obstacle to use in your test as well.

Some obstacles will work better with Edison than others. If an obstacle is too small or doesn't reflect enough infrared light, Edison cannot detect it. Select an object that is opaque but not too dark (don't choose black objects) and at least as tall as Edison. For this program, the wall of the room would be a good obstacle.



### Why is that?

Some of Edison's sensors are always on and checking for events. The infrared receiver is actually always on, but the infrared LEDs are not.

For Edison to detect obstacles, you need to include code in your program to turn on the obstacle detection beam. This turns on the infrared LEDs and tells the infrared receiver to look for reflected infrared light bouncing back to the robot. Just turning the obstacle detection beam on isn't enough, however. You also need code to tell the sensor what event to check for (in other words, where it should be looking for an obstacle) and what to do if that event is detected.



### Hint

If your robot isn't detecting your obstacle, try a different object. If it still isn't detecting the obstacle, you might need to calibrate the obstacle detection. Ask your teacher for the special barcode to calibrate obstacle detection.

Test the program to see how it works.

## Detect and avoid

Instead of just stopping when it detects an obstacle, you can get Edison to avoid any obstacle it detects and keep driving. To do this, you need to create an algorithm to solve the set of problems: 'detect and avoid any obstacle'.

Use pseudocode to write a detect-and-avoid algorithm.

Once you have your algorithm, write an EdScratch program for Edison that uses the logic from that algorithm. Download your program and test it out using Edison and some obstacles.

Even when professional computer programmers write programs, they run into problems. Describe one problem you had creating your detect-and-avoid algorithm or program. What did you do to solve the problem?

## Challenge up: If line, go right. If obstacle, go left

Edison robots have different sensors that can detect different things. You can use multiple sensors in a single program, getting the robot to react to different types of events.

### What to do

Write a program so that your Edison robot moves through a grid, checking each new section of the grid for any non-reflective (black) surfaces or obstacles.

Your program should tell Edison that if the robot detects a black surface, it needs to turn right, but if it detects an obstacle, the robot should turn left instead. Test your program out using the activity sheet on page 134.

See if you can write a program so that your Edison robot starts on the outline, then uses its sensors to get to the parking zone goal.



### Hint

Sensors like the obstacle detection beam and line tracker need to be turned on to work. Your program also needs to tell Edison what event the sensors should look for and how to react to that event.

Some of Edison's sensors, including obstacle detection, generate and store data when they detect specific events. It is good coding practice to clear the sensor data, especially when you use sensor events in conditionals nested inside loops. You don't want the data from a previous loop to affect the next loop!

It is also wise to clear the sensor data at the start of a program, just in case the robot has old data stored from a previous program.

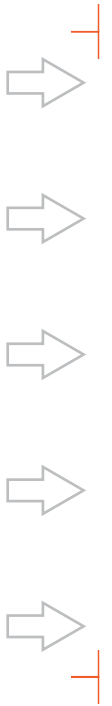
Pseudocode helps us plan, comments help us debug. Use them to help you plan and test your program!

Write your program in the text box on the next page. Alternatively you can add a photo to the image box on the next page.

Name \_\_\_\_\_

# Activity sheet: If line, go right

Place obstacle over here.



Place obstacle over here.

