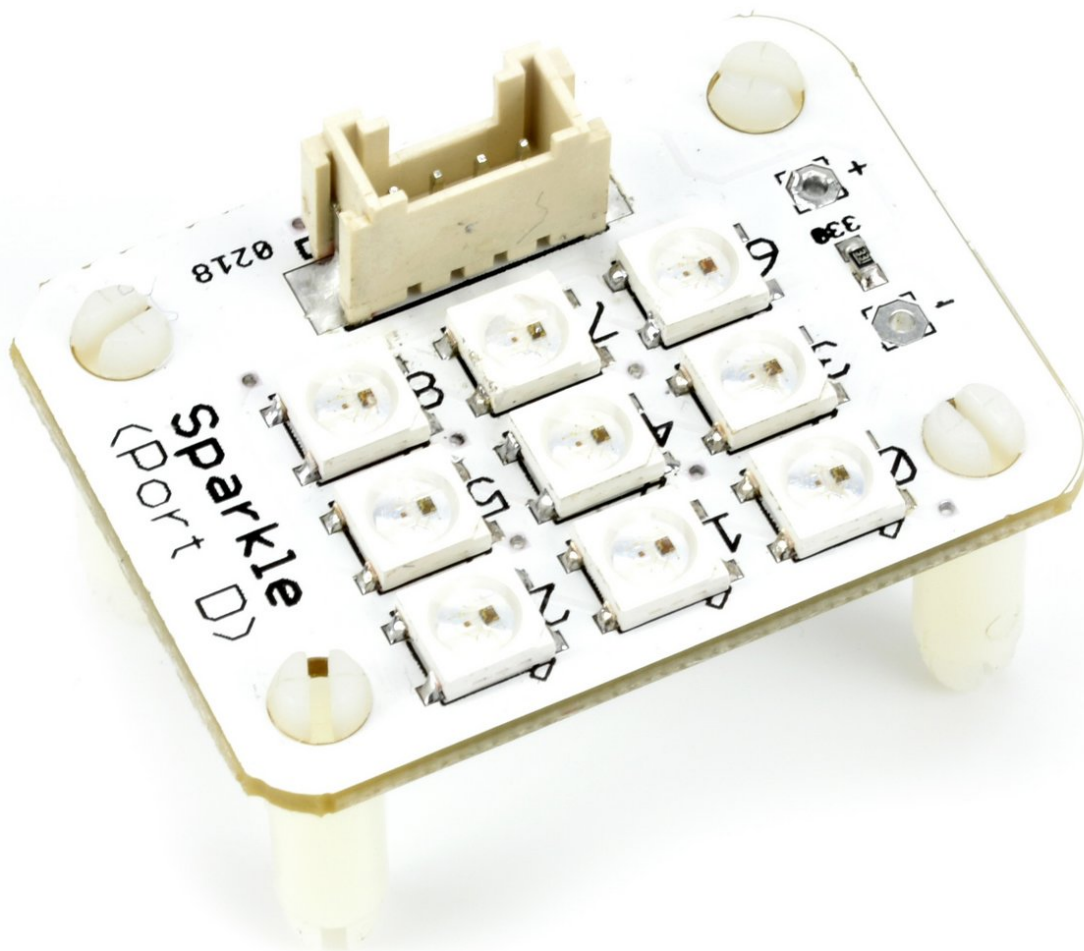


C - Sparkle Feedback

You may have found that when testing your robot, it was sometimes difficult to know what was going wrong as there was no feedback about the sensors!



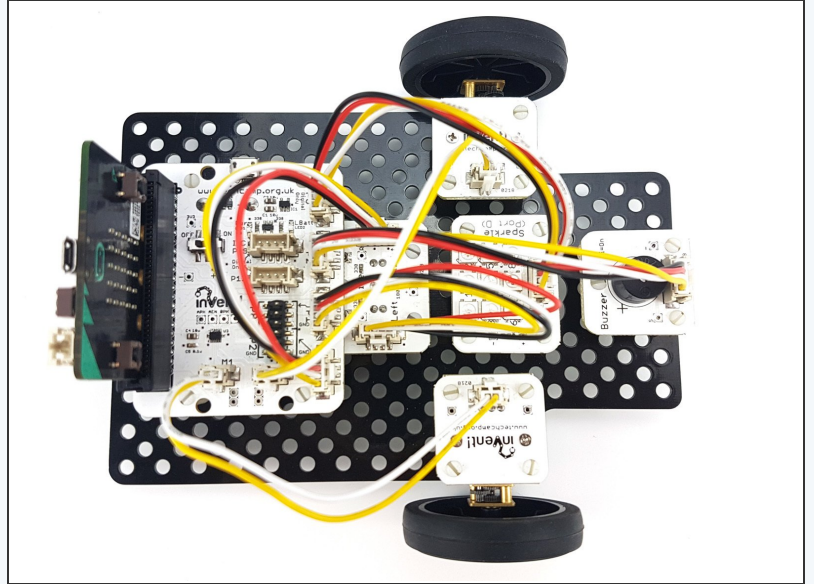
INTRODUCTION

You may have found that when testing your robot, it was sometimes difficult to know what was going wrong as there was no feedback about the sensors!

Step 1

Assemble Your Robot

- Let's add some more modules to allow our robot to give **feedback** about the line it is following.
- Add the Line sensor underneath as before then add **Sparkle** and **buzzer** modules to your robot like in the picture.
- It can be quite hard to see the wiring with so many things plugged in! The **motors** and **line sensor** should be the same as before (M1, M2 and P0, P1)
- Plug the buzzer into **P2**, and the Sparkles into **P8**



Step 2

Two Sensor Program

- Load up your **two sensor line follower program** from the last lesson.
- It should look something like our **example** in the picture.

```
13 # Invent! Code End
14 # Start your code below here!
15
16 calibrate_line_sensors()
17
18 drive_motor(0,100)
19
20 while True:
21     if digital_read_line(1)==1 and digital_read_line(0)==1:
22         drive_motor(0,0)
23
24     if digital_read_line(1)==1 and digital_read_line(0)==0:
25         drive_motor(1,50)
26         drive_motor(2,-50)
27
28     if digital_read_line(1)==0 and digital_read_line(0)==1:
29         drive_motor(1,-50)
30         drive_motor(2,50)
31
32     if digital_read_line(1)==0 and digital_read_line(0)==0:
33         drive_motor(0,50)
34
35
```

Step 3

Left Sensor

- Let's use the **outer** two Sparkles (**0** and **2**) to show the state of the two **infrared sensors**.
- If the **left** sensor is on the line (**0**), turn the **left** Sparkle **green**.
- If the **left** sensor is off the line (**1**), turn the **left** sparkle **red**.
- Let's change the program so the line follower code isn't used for the moment - we can do this by making it into a **function**.
- Move the line follower code out of the **while True:** loop and into a function called **line_follower** that takes no inputs.
- Write the code in the picture to test the left sparkle - **test it by moving it across the line!**

```
15
16 import neopixel
17 pixels=neopixel.NeoPixel(pin8,9)
18
19 calibrate_line_sensors()
20
21 def line_follower():
22     if digital_read_line(1)==1 and digital_read_line(0)==1:
23         drive_motor(0,0)
24
25     if digital_read_line(1)==1 and digital_read_line(0)==0:
26         drive_motor(1,50)
27         drive_motor(2,-50)
28
29     if digital_read_line(1)==0 and digital_read_line(0)==1:
30         drive_motor(1,-50)
31         drive_motor(2,50)
32
33     if digital_read_line(1)==0 and digital_read_line(0)==0:
34         drive_motor(0,50)
35
36 while True:
37     if digital_read_line(1)==1:
38         pixels[0]=(255,0,0)
39     else:
40         pixels[0]=(0,255,0)
41     pixels.show()
42
```

Step 4

Right Sensor

- Let's use the right Sparkle (**Sparkle 2**) to do the same thing for the **right hand sensor**.
- **Add some more code** to set the **right** Sparkle in the same way as the left one, but using the **right sensor**.
- There is a **hint layout** if you need it!
- Don't forget to **test it properly**.

```
31         drive_motor(2,50)
32
33     if digital_read_line(1)==0 and digital_read_line(0)==1:
34         drive_motor(0,50)
35
36 while True:
37     if digital_read_line(1)==1:
38         pixels[0]=(255,0,0)
39     else:
40         pixels[0]=(0,255,0)
41     if # Right sensor off line
42         # Set right sparkle red
43     else: # Right sensor on line
44         # Set right sparkle green
45     pixels.show()
46
```

Step 5

Centre

- Let's add some more blocks so the **middle Sparkle** (Sparkle 1) is **green** when **both** sensors are on the line.
- This is when both left **and** right are **0**!
- Add a third **IF/ELSE** block that checks if both left **and** right are **0**, and sets the middle Sparkle **green** if they are.
- There is a **hint layout** if you need it.
- Don't forget to use **and**.

```
35
36 while True:
37     if digital_read_line(1)==1:
38         pixels[0]=(255,0,0)
39     else:
40         pixels[0]=(0,255,0)
41     if digital_read_line(0)==1:
42         pixels[2]=(255,0,0)
43     else:
44         pixels[2]=(0,255,0)
45     if # Both sensors read 0
46         # Middle sparkle green
47     else:
48         # Middle sparkle red
49     pixels.show()
50
```

Step 6

Make it Better

- You may have thought the last program **didn't look quite right** - as the robot moved over the line, **two** Sparkles changed at once!
- It would be more sensible for the program to set **just the middle** Sparkle green when both sensors are on the line, instead of **all three**.
- We need to change the code to do this, to set the colours of **all three Sparkles** for each **combination** of sensor states.
- In the picture is the structure you will need - **a 3 part if, else if, else statement** for checking if the robot is to the **left, right** or on the **centre** of the line.
- **Add lines of code** to set the sparkles properly!

```
1     drive_motor(2,50)
2
3     if digital_read_line(1)==0 and digital_read_line(0)==0:
4         drive_motor(0,50)
5
6 while True:
7     if # Left off line, right on line
8         # Right sparkle green, others red
9     elif # Right off line, left on line
10        # Left sparkle green, others red
11    else: # Both on line
12        # Middle sparkle green, others red
13    pixels.show()
14
15
16
17
```

Step 7

Combine the Code

- You may have noticed that our Sparkle code and line following code have some **very similar IF blocks**.
- **Move** the Sparkle lines into the **matching IF statements** in the line follower function, to **merge** your line follower and Sparkle programs together.
- **Test** it properly on the track!



Step 8

Red Lights for Stop

- What about when the robot comes **off the line completely**?
- Add some more Sparkle lines to the correct **IF statement** to turn all the Sparkles **red** if the robot is **off the line**.



Step 9

SOS - I'm Lost!

- Our robot should probably make more of a fuss if it **gets lost completely** and doesn't know where the line is.
- **Add** some code to make the buzzer **beep SOS** if the robot goes off the line.
- You can look back at the **Morse Code lesson** if you can't remember how to do this!

