techcamp.org.uk

A - Starting Lights

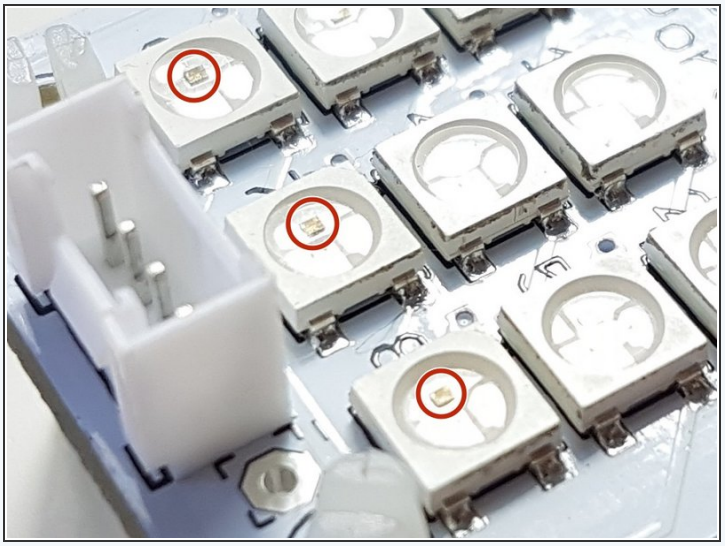
Learn how to use the Sparkle module by creating some starting lights for a race around the planet.

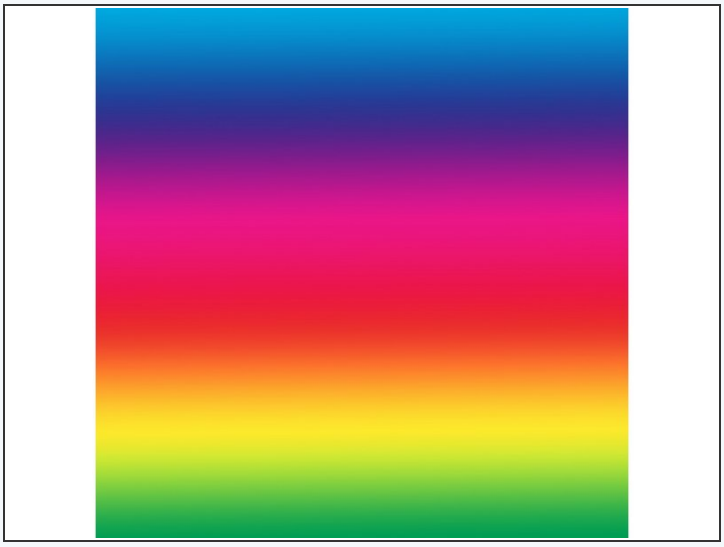
INTRODUCTION

Learn how to use the Sparkle module by creating some starting lights for a race around the planet.

Step 1

What are Sparkles?



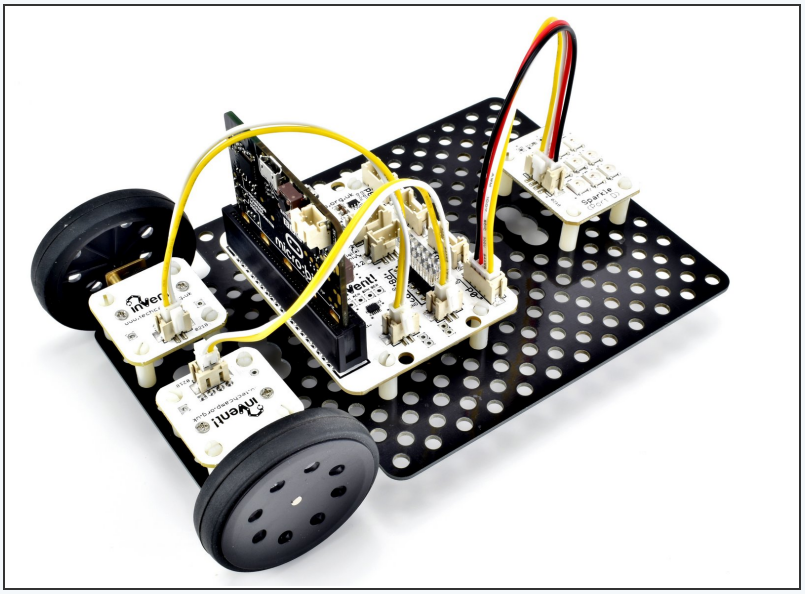


- Sparkles are very useful - they are **LEDs,** just like the red/green LED from before, but much cleverer!
- They have small chips inside them, which allow you to control many LEDs using **just one output.** If you look really closely you might be able to see them.
- They are also **three LEDs in one** - there is a **red**, **green** and **blue** LED in every sparkle.
- We can control these three internal LEDs **separately,** and mix them together to create **any colour!**
- This is the same way pixels in your computer screen work - **have a look at the chart** to see all the possible colours we can make.

Step 2

Connect your Sparkle Module

- Build up your robot like the picture.
- Plug the sparkle module into **P0**.



Step 3

Test Your Sparkles

- For now, let's **test the sparkles** by building the simple program in the picture - hopefully they **all turn red** when you **program your robot!**

⚠ Don't stare at the sparkle board for too long - it's very bright!

```
12 def analog_read_line(s): v=p0()if s==0
13 # Invent! Code End
14 # Start your code below here!
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 for i in range(0,9):
21     pixels[i]=(255,0,0)
22
23 pixels.show()
24
25
26
27
```

Step 4

How does it work?

- Let's go through this program and see how it works.
- **import neopixel** - this loads some extra code that allows us to use the sparkles. You need this in any program you want to use them in!
- **pixels=neopixel.NeoPixel(pin0,9)** - this sets up **pixels** to refer to a set of sparkles, connected to **pin0**, with **9** sparkles in total.
- **pixels[i]=(255,0,0)** - with the for loop changing i from 0 to 8, this line sets each sparkle to **red**. *pixels[0]* sets the first sparkle, *pixels[1]* sets the second one and so on.
  - The three numbers in this line set the amount of red, green and blue respectively - so 255,0,0 sets red at full, and green and blue completely off.
- **pixels.show()** - this **updates** the sparkles, and you need this line every time you change any of the sparkle colours.
- **Try changing the amounts of red, green and blue**, and see what colours you can make.

```
12 def analog_read_line(s): v=p0()if s==0
13 # Invent! Code End
14 # Start your code below here!
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 for i in range(0,9):
21     pixels[i]=(255,0,0)
22
23 pixels.show()
24
25
26
27
```

Step 5

Different Sparkles, Different Colours

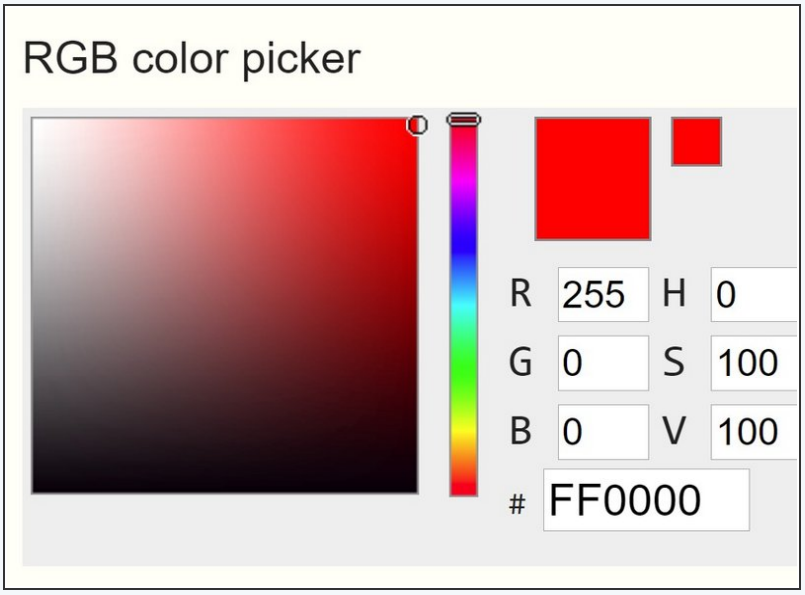
- Remember, we can also control each sparkle **individually** if we take out the for loop.
- We can use **pixels[pixelNumber]=(red,green,blue)** to set the colours of individual sparkles - have a look at the sparkle board to see which sparkle is which number.
- **In programming numbers start from 0,** not 1 - so for nine sparkles, the first is 0, the second is 1 and the last is sparkle 8.
- Use **three lines of code** to **set three sparkles to a different colour.**

⚠ Don't forget to put a **pixels.show()** line at the end, or you won't see anything happen.

Step 6

Choosing Colours

- It can be difficult to know what to set the red, green and blue to to get a **specific colour!**
- Try this online colour picking tool (http://www.rapidtables.com/web/color/RGB_Color.htm) - you can pick any colour you like, and it will give you the **red, green and blue values** you need.



Step 7

Extension Challenge - Starting Lights

- Let's make a set of **starting lights** for a race across the planet surface.
- Check out the F1 starting lights in the video - can you put together a program using **sparkle** and **wait** blocks to **make your own?**
- The lights should **turn red 3 at a time,** then **all go green** at the same time.