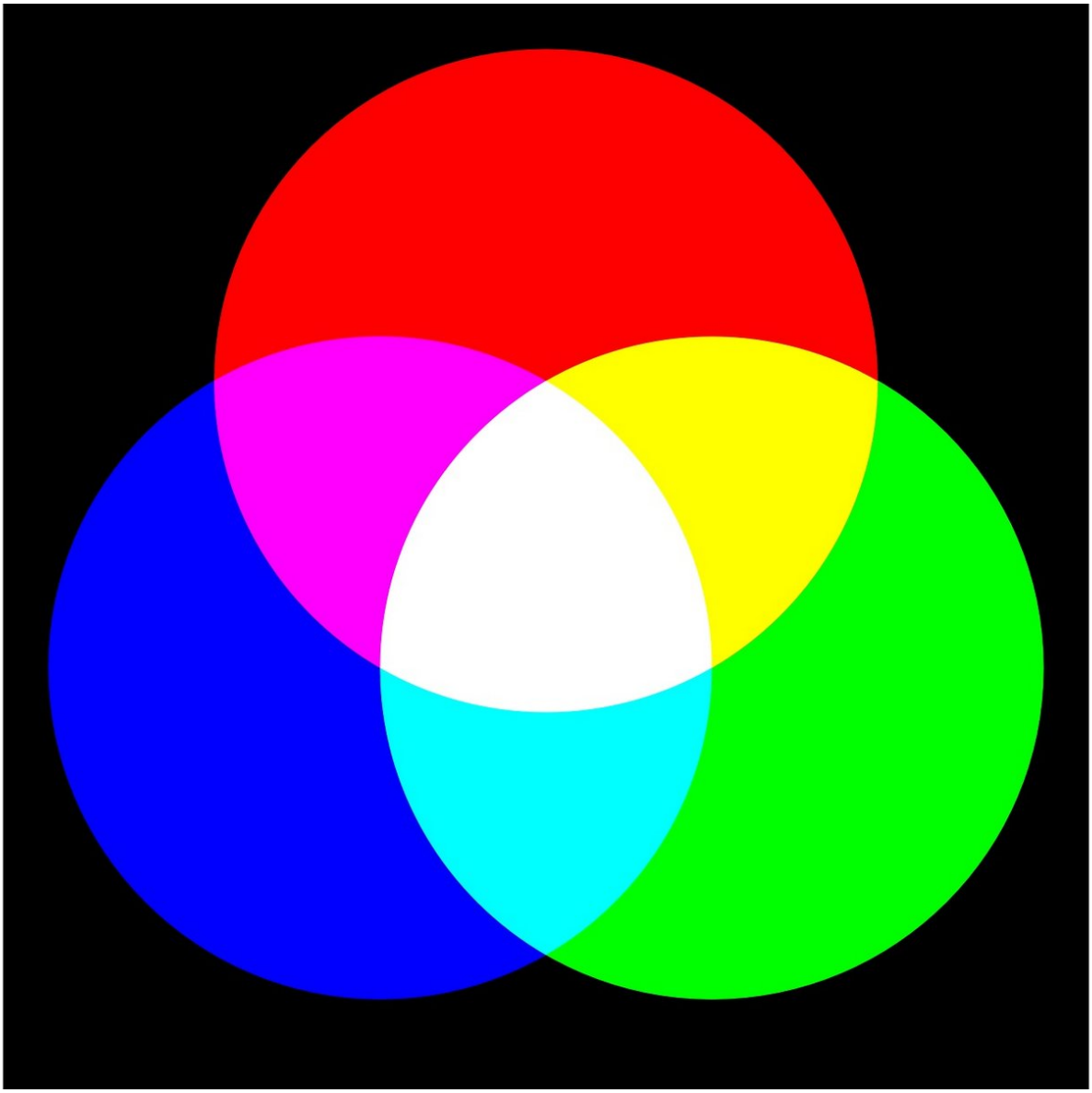
# C - Colour Mixing

Let's use the Sparkle module and some switches to make a colour mixer!



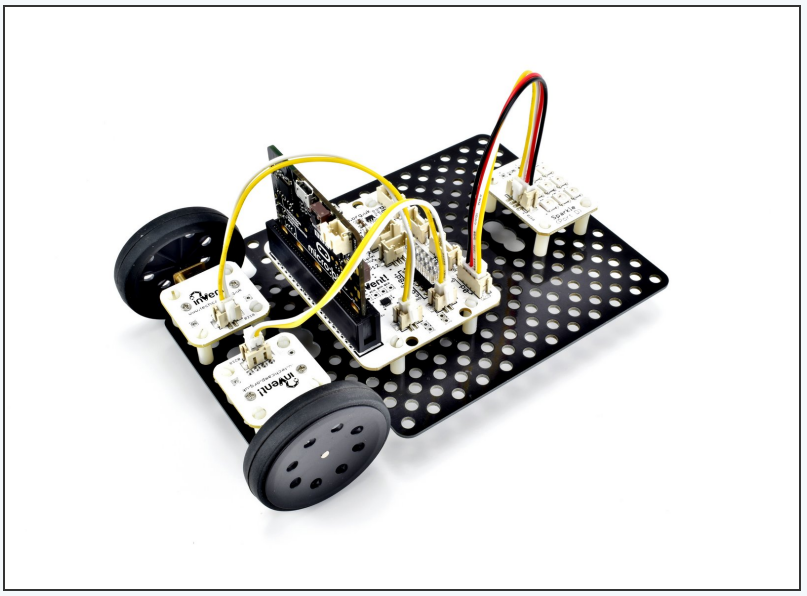
# INTRODUCTION

Let's use the Sparkle module and some switches to make a colour mixer!

Step 1

## Colour Changing Light

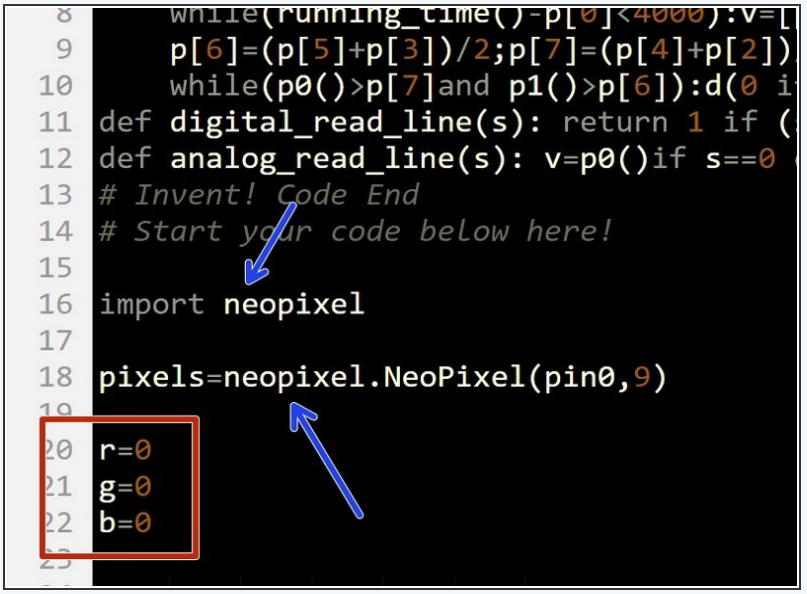
- Remember how each Sparkle contains **3** different colour **LEDs?**
- We're going to make a program using **variables,** that allows you to change the Sparkle colour using **2 switches.**
- **Assemble your robot** like the picture first!



Step 2

## Setup the Variables

- **Import** the neopixel code, and setup the neopixels at the start of the program as before.
- We are going to need **3 variables** this time - one for **red**, one for **green**, and one for **blue**
- **Add three new variables** at the top of the program.
- Call them **r**, **g** and **b** (for red, green and blue), an **initialise** them to 0.
- We're going to use each of these variables to **remember** the amount of red, green and blue light we want.



Step 3

## Start the Program

- Our program needs to always be **updating** the colour of the Sparkles, so when the variables change, the Sparkles change **too!**
- Add a **while True:** loop, and add some lines that continually set **all the sparkles** using the values in the r,g, and b variables.

```
14 # Start your code below here!
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 r=0
21 g=0
22 b=0
23
24 while True:
25     for i in range(0,9):
26         pixels[i]=(r,g,b)
27     pixels.show()
28
```

Step 4

## Test it out!

- Before we can test it, we need to set r,g, and b to something other than 0, or the sparkles **won't do anything!**
- **Change the values** to whatever you like, and test your program so far.

```
14 # Start your code below here!
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 r=0
21 g=0
22 b=0
23
24 while True:
25     for i in range(0,9):
26         pixels[i]=(r,g,b)
27     pixels.show()
28
29
```

This document was generated on 2022-01-01 09:47:46 PM (MST).

Step 5

## Using a Switch

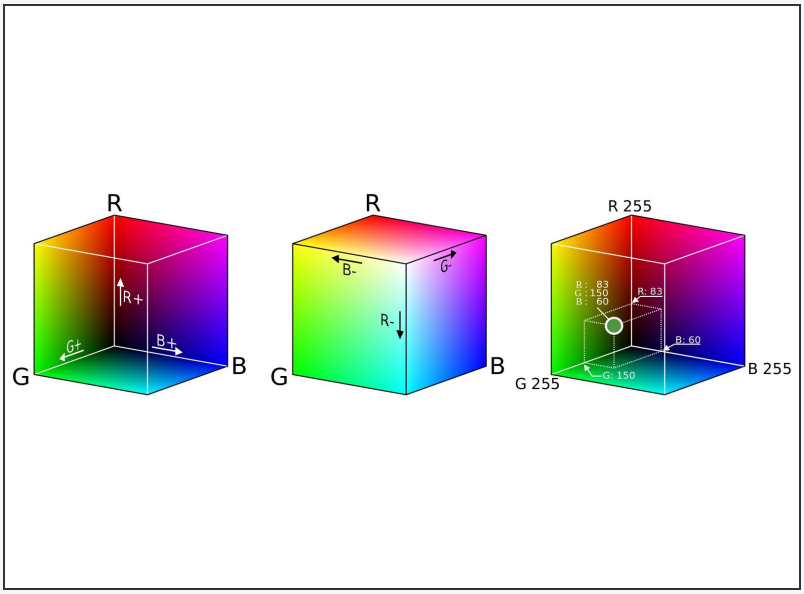
- Now let's add some blocks to let use **change the colour** of the Sparkles using a **switch.**
- You will need to **add the two switches** to your robot - plug them into **P1** and **P2**.
- Inside the loop, add an **if statement** that checks if the first switch **(P1)** is pressed.
- If the switch is pressed, **increase r by 10.**
- Add a **sleep** of **200** milliseconds after r is increased. Otherwise, holding down the switch will **change r very quickly!**
- At the start of the program, **set r, g and b to 0 again** so you can see what is happening more clearly when testing your program.
- There are some **hints** in the picture if you need them!

```
if pin1.read_digital()==1:
    r=r+10
    sleep(200)
```

Step 6

## Back to 0?

- You probably noticed when testing your program that if you press the button enough times, the amount of red seems to **start from zero again.**
- This is because the amount of red, green and blue in a Sparkle **cannot be more than 255!**
- When r is **more than 255** (when you have pressed the button more than 25 times), the Sparkle will show the **value of r minus 255.**
- For example - 260 is **more** than 255, so the Sparkle will actually show **260-255=5!**



Step 7

## Add Some Green

- We can now change the amount of red, but we want a colour **mixer!** Let's add **another colour** using the other **switch.**
- Add another **if statement**, that checks the **second switch (P2)** and increases **g** by **10** if it is pressed.
- This should be quite similar to the first one!

⚠ Make sure to **test** your program properly before moving on. What do you notice when you add equal amounts of red and green?

```
14 # Start your code below here!
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 r=0
21 g=0
22 b=0
23
24 while True:
25     for i in range(0,9):
26         pixels[i]=(r,g,b)
27     pixels.show()
28
29     if pin1.read_digital()==1:
30         r=r+10
31         sleep(200)
32
33     if pin2.read_digital()==1:
34         # Your code here
35
```

Step 8

## 3 colours, 2 Switches

- You may have noticed that we have **run out** of switches for the **third colour**, blue - but there is a solution!
- What if we made a program that could increase b by 10 if **both switches are pressed at the same time?**
- To do this, inside the **if statement** that checks the first switch, put an **if/else** statement that checks the **second** switch.
- **If** the second switch is pressed as well, **increase b** by 10 instead.
- **Else**, just increase **r by 10** as before.
- Don't forget to **wait 200 milliseconds!**
- There are some **layout hints** in the picture if you need them.

```
15
16 import neopixel
17
18 pixels=neopixel.NeoPixel(pin0,9)
19
20 r=0
21 g=0
22 b=0
23
24 while True:
25     for i in range(0,9):
26         pixels[i]=(r,g,b)
27     pixels.show()
28
29     if pin1.read_digital()==1:
30         if pin2.read_digital()==1:
31             b=b+10
32             sleep(200)
33         else:
34             # Increase r by 10 here
35
36     if pin2.read_digital()==1:
37         g=g+10
38         sleep(200)
39
```

Step 9

## Fixing the two-switch blue/green problem

- You've probably found the last program doesn't quite work correctly - if you press both switches, the **blue and green both increase!**
- This is because the **second if statement** is still **true** if we are holding down **both switches,** so g is also increased by 10.
- Add an **if/else statement** inside the **second** if statement **just like the one inside the first** if statement to fix this.
- **Congratulations** - you have made a switch controlled colour mixer!


Step 10

## White Light Buzzer

- When r, g and b are all the **same**, the Sparkles will be **white.**
- **Add code** to your program to:
  - Check if r, g and b are all the same
  - If they are, **sound the buzzer!**
  - You will need to **add the buzzer** module and plug it into **P8.**

Here's a hint - you can check if 1 condition **AND** another condition is true by simply writing **and** - for example, *if pin1.read_digital()==1 and pin2.read_digital()==1* is only true if both P1 **and** P2 are pressed.