B - Robot Police

Make a program using your Sparkle and buzzer modules, that turns your robot into a police robot to keep the planet safe.

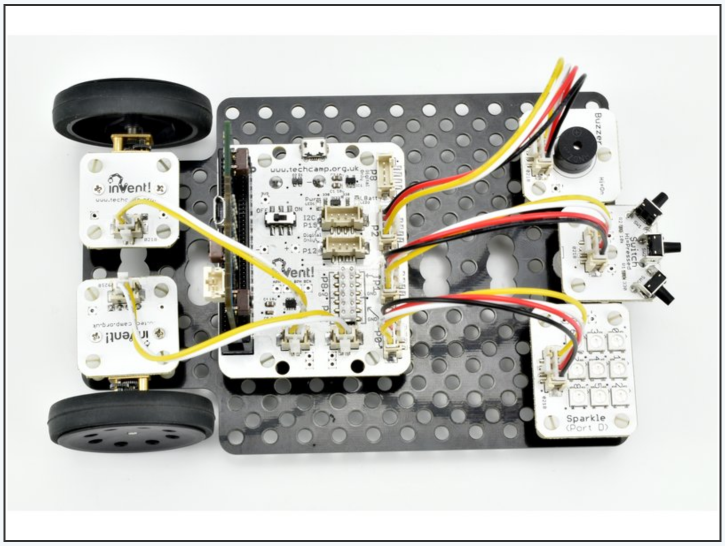
This document was generated on 2022-01-01 09:47:36 PM (MST).

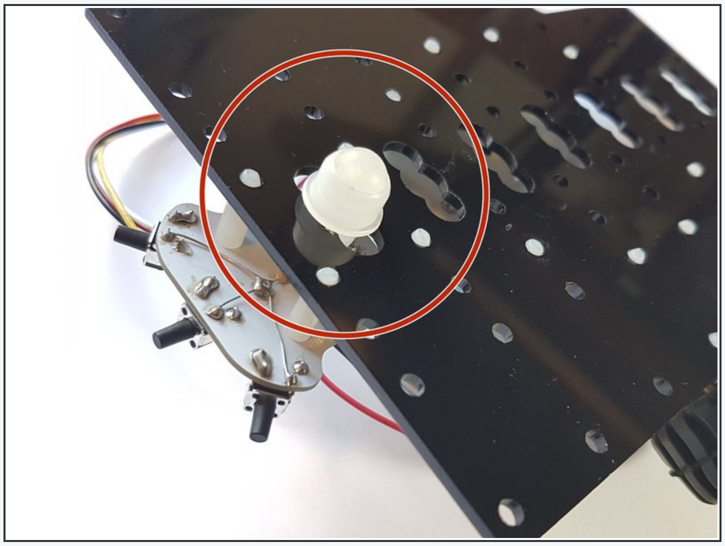
INTRODUCTION

Make a program using your Sparkle and buzzer modules, that turns your robot into a police robot to keep the planet safe.

Step 1

Assemble the Robot





- We're going to be using a **lot of modules** to turn our robot into a police car!
- **Carefully** assemble your robot like the picture. The connections should be:
  - Buzzer > **P2**
  - Switch > **P1**
  - Sparkles > **P0**
  - Left Motor> **M1**
  - Right Motor > **M2**

Step 2

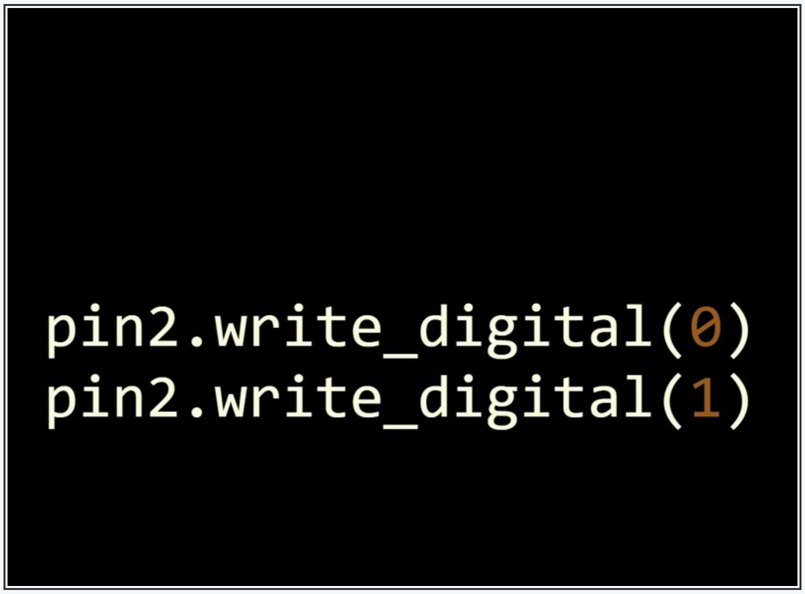
Reds and Blues

- Now we know how to use the **sparkle module,** let's start by programming all the sparkles to **flash red and blue** like a police car.
- Your program should:
  - Turn all sparkles **red**
  - Wait **half a second**
  - Turn all sparkles **blue**
  - Wait **half a second**
  - **Repeat** this forever!
- Make sure to **test your program** works.

Step 3

Add the Siren

- Let's add the **buzzer** to the flashing lights to make a **siren!**
- **Add some lines** to your program so the buzzer is:
  - **Buzzing** when the sparkles are **red**
  - **Off** when the sparkles are **blue**
- There are some **hints** in the picture if you can't remember which functions to use!



```
pin2.write_digital(0)
pin2.write_digital(1)
```

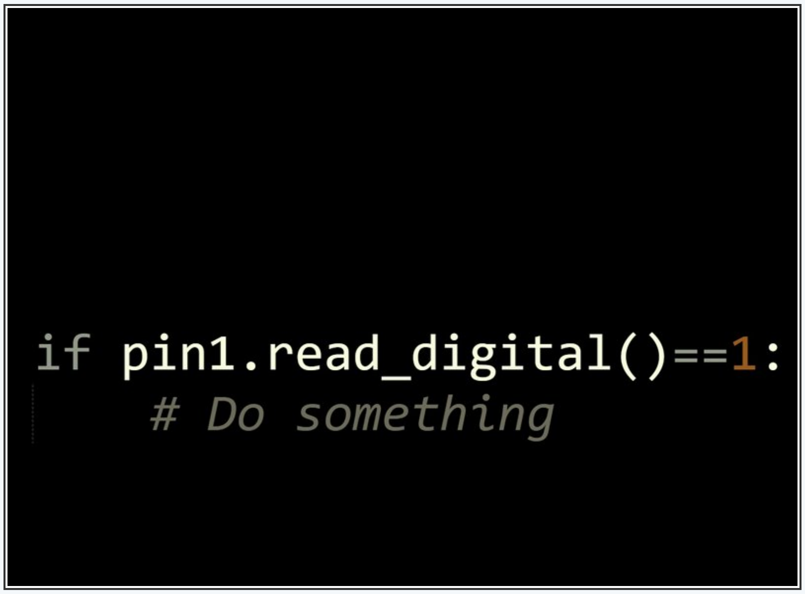
Step 4

Switch Activation

- Police cars don't have their lights and sirens on **all the time,** that would be very irritating!
- Let's **add a switch** so the police robot only flashes and buzzes when we **press** it.
- Add an **IF statement** that checks if the **switch** is pressed, and **move** your sparkle code inside the IF statement so it is run **only when the switch is pressed.**

⚠ Don't forget, make sure all your code is **inside a while True: loop** so the switch isn't just checked once!

- Test out your program - **does it work how you expected?**

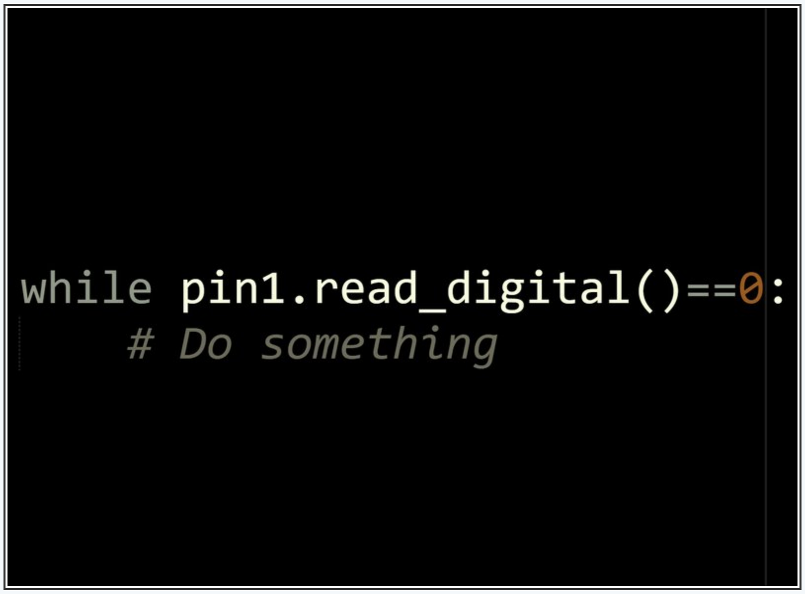


```
if pin1.read_digital()==1:
    # Do something
```

Step 5

While Loops

- Currently, the siren and lights only run **once** when we press the switch. We really want them to run **forever** when the switch is pressed!
- We need to use another loop for this - luckily we can use a **while loop!**
- Don't forget - a while loop will run all the time the condition is **True**
- For example, the while loop in the picture will run **until the switch is pressed!**
- Inside the if statement, put your sparkle code **inside a while loop that will run forever**.

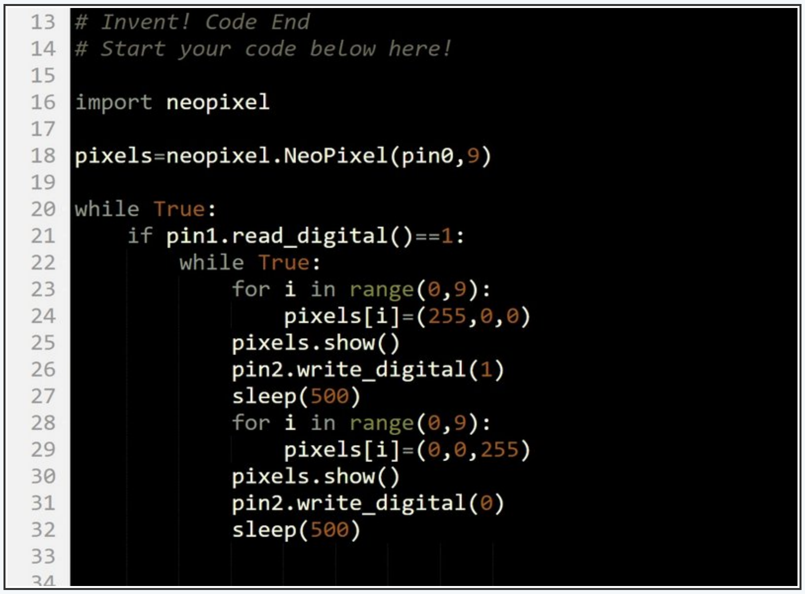


```
while pin1.read_digital()==0:
    # Do something
```

Step 6

Switch De-activation

- Now we can **activate** the police robot with the switch, but we can't **turn it off** again!
- We need to change the while loop, so it only runs **until the switch is pressed again** .
- **Change the condition** in the while loop so it uses a **read_digital** function to only run until the switch is pressed.
- **Test it out** - does it work as expected?
- The picture has an example of what the code should look like **before this step!**

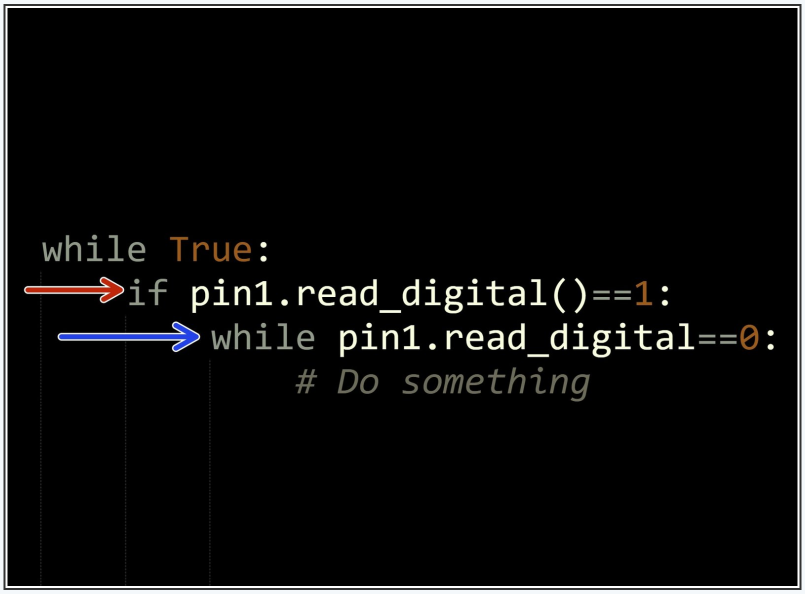


```
# Invent! Code End
# Start your code below here!

import neopixel

pixels=neopixel.NeoPixel(pin0,9)

while True:
    if pin1.read_digital()==1:
        while True:
            for i in range(0,9):
                pixels[i]=(255,0,0)
            pixels.show()
            pin2.write_digital(1)
            sleep(500)
            for i in range(0,9):
                pixels[i]=(0,0,255)
            pixels.show()
            pin2.write_digital(0)
            sleep(500)
```

Step 7

Waiting for Switches

- You might find it **doesn't work very well**, and it sometimes **takes a few attempts** to turn the lights on or off.
- Our problem is that the robot runs the program **very, very fast!**
- Even if we just tap the switch, the robot thinks so quickly it thinks we are **holding it down.**
- Look at the simplified version of the code in the picture:
    - As soon as we press the switch, the **conclusion** of the if statement is run
    - The robot **checks** the condition of the while loop **before we can possibly have time to take our finger off the switch,** so the loop is **never run** (the condition is false as P1 is still 1), and the program goes back to the **red arrow**
    - This can happen **many times** before we have released the switch, so we have **no idea** whether the lights will be running or not!

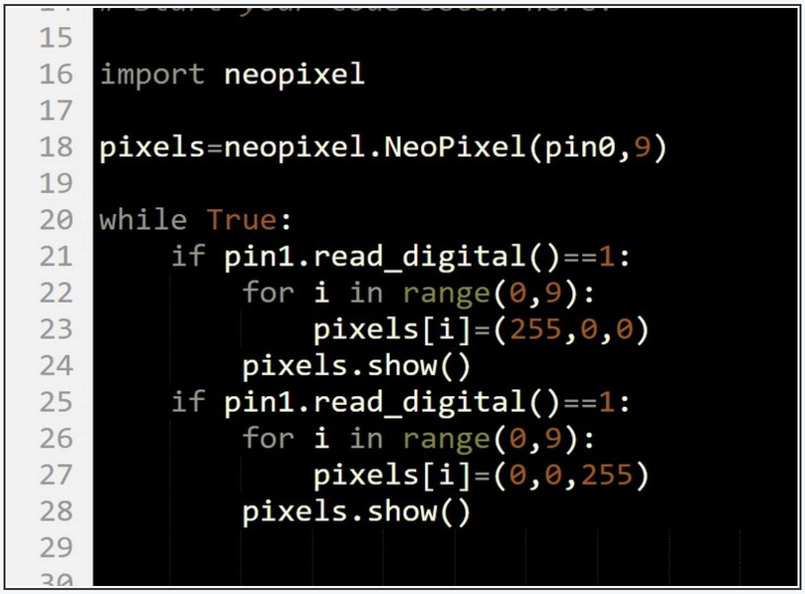


```
while True:
    if pin1.read_digital()==1:
        while pin1.read_digital==0:
            # Do something
```

Step 8

Waiting for Switches

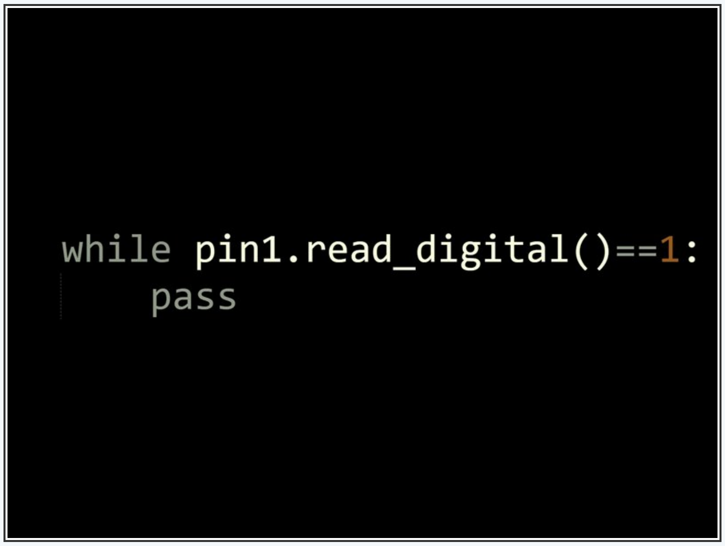
- This can be **hard to understand!** Let's make another short program to understand this problem with **an example.**
- **Save you program as a separate file** and build the program in the picture.
- This program turns the sparkles red if we press the switch, then blue if we press it again - **test it out!**
- It is **impossible** to accurately make the sparkles red or blue, as the robot is too fast - we need to make it **wait for us to let go of the switch** to fix this.



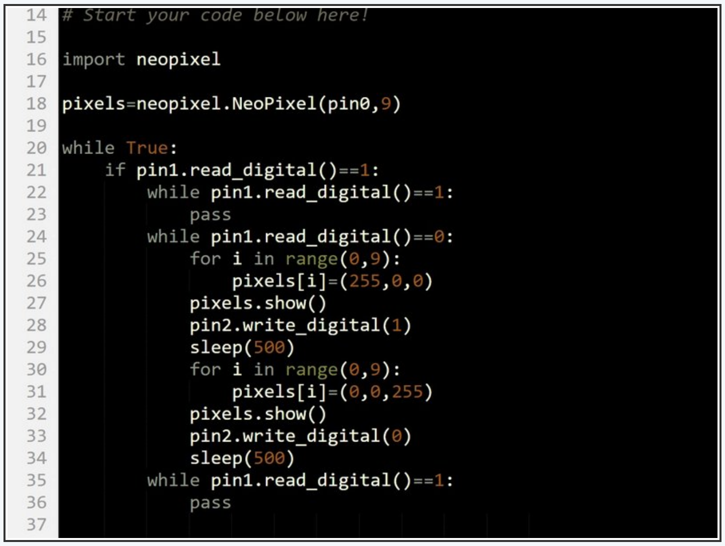
```
import neopixel

pixels=neopixel.NeoPixel(pin0,9)

while True:
    if pin1.read_digital()==1:
        for i in range(0,9):
            pixels[i]=(255,0,0)
        pixels.show()
    if pin1.read_digital()==1:
        for i in range(0,9):
            pixels[i]=(0,0,255)
        pixels.show()
```

Step 9

Waiting for Switches



```
while pin1.read_digital()==1:
    pass
```



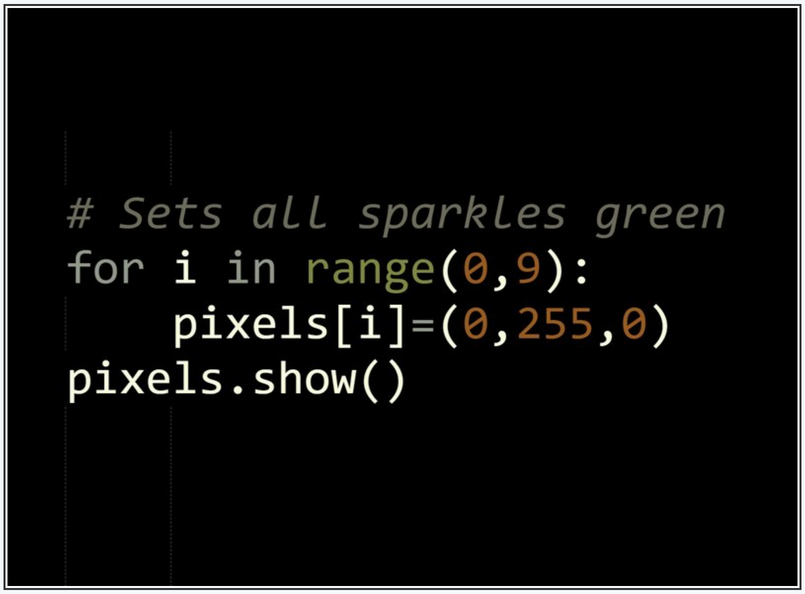
```
# Start your code below here!

import neopixel

pixels=neopixel.NeoPixel(pin0,9)

while True:
    if pin1.read_digital()==1:
        while pin1.read_digital()==1:
            pass
        while pin1.read_digital()==0:
            for i in range(0,9):
                pixels[i]=(255,0,0)
            pixels.show()
            pin2.write_digital(1)
            sleep(500)
            for i in range(0,9):
                pixels[i]=(0,0,255)
            pixels.show()
            pin2.write_digital(0)
            sleep(500)
        while pin1.read_digital()==1:
            pass

```

- Luckily, we can use a **empty while loop** to wait until we let go of the switch.
- A while loop with no code in it will simply **stall the program (wait)** until the condition is **false**.
- We need to add **pass** inside the loop as it can't be completely empty - the pass line doesn't actually do anything though.
- We want to wait until the switch is released (reads **0**), so our condition should be **pin1.read_digital()==1**
- Go back to your police car program, and **add** this empty while loop into your program, **right at the start of the if statement.**
- We need to do the same thing **after the while loop that flashes the sparkles,** so when we press the switch to turn them off they don't get **started again immediately!**
- Add another empty while loop like the first one at the **end of the if statement.**

Step 10

Business as Usual

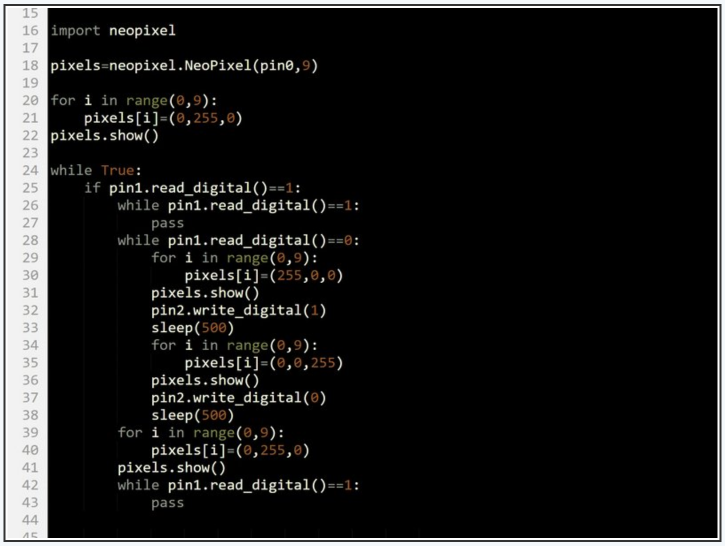
- You may have noticed that the sparkles **stay blue** after we turn off the police lights and siren - let's change them so they are **green!**
- Add some more sparkle lines so that:
  - The sparkles are set to **green** at the start
  - They are **reset** to green **after the end of the main while loop.**



```
# Sets all sparkles green
for i in range(0,9):
    pixels[i]=(0,255,0)
pixels.show()
```

Step 11

Chase the Criminals



```

import neopixel

pixels=neopixel.NeoPixel(pin0,9)

for i in range(0,9):
    pixels[i]=(0,255,0)
pixels.show()

while True:
    if pin1.read_digital()==1:
        while pin1.read_digital()==1:
            pass
        while pin1.read_digital()==0:
            for i in range(0,9):
                pixels[i]=(255,0,0)
            pixels.show()
            pin2.write_digital(1)
            sleep(500)
            for i in range(0,9):
                pixels[i]=(0,0,255)
            pixels.show()
            pin2.write_digital(0)
            sleep(500)
        for i in range(0,9):
            pixels[i]=(0,255,0)
        pixels.show()
        while pin1.read_digital()==1:
            pass

```

- The final part of our police program is to make the robot **chase** after our criminal!
- **Check the picture** for what your code should look like so far.
- **Add some code** to make your robot **drive forwards** at **full speed** when the switch is pressed and the sparkles are flashing, then **stop** when it is pressed again.

Step 12

Improve your Police Program

- For this extension challenge, you need to complete a few tasks:
- **Change the buzzer code** so that your buzzer beeps **faster than the sparkles change** to make it sound more realistic.
- When your robot is chasing the criminal, make it **turn left and right** in a **weaving motion** instead of just moving forwards.