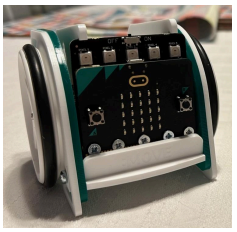
Selby 0 1 CoderDojo

Robot Buggies
Light them up!



You're going to programme the lights on the buggy, they take their instructions from the micro:bit. You can still use the LED's on the micro:bit itself at the same time. Use this guide, and the guide on how to drive the robot buggy

Adding the Neopixel blocks

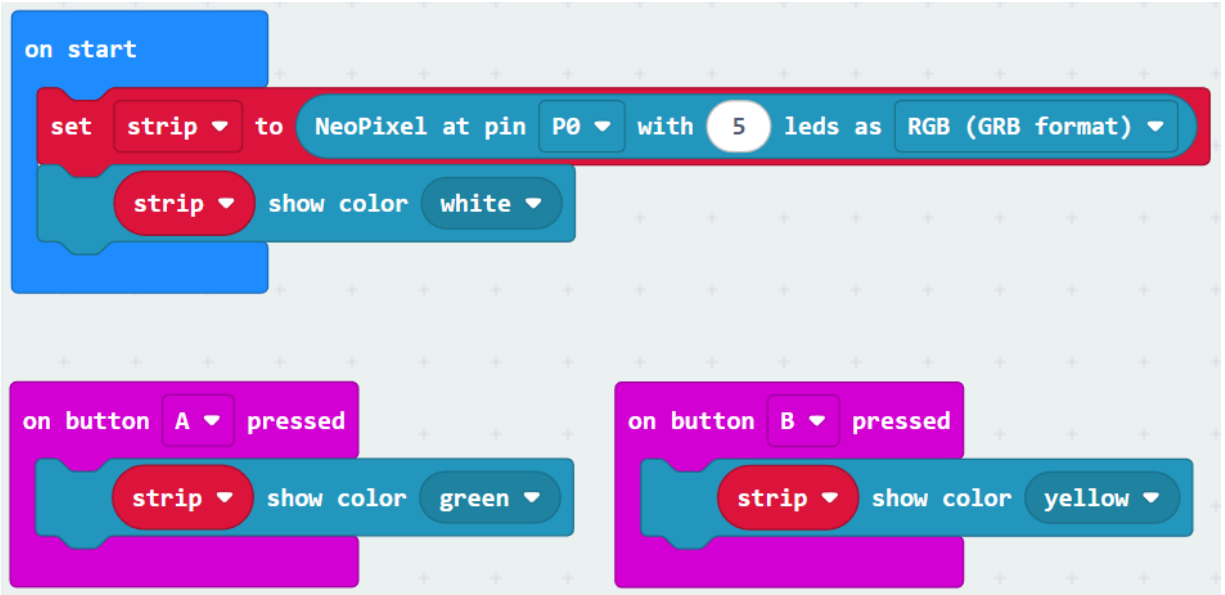
- Before we start building the code, we need to install some special blocks for our robot buggy.
- In a web browser, go to makecode.micobit.org and start a new project. If you have just finished the driving project and it's still open, you can continue working on that project.
- Look for the grey extensions button.
- In the search bar, type "neopixel", click on the card titled neopixel.
- You should now see a new turquoise group called Neopixel. Take a look at these blocks.

Creating a variable to store the LEDs

- Neopixel stores a record of the LEDs in a variable and we give the variable a name.
- Think of the variable name like an address or a phone number. We're going to write some instructions to send to that variable.
- Click on the 'Neopixel' group, and look for the very top block, it should start 'set strip to'.
- Drag this block into the 'on start' block.
- This block has automatically created the variable for us, its name is 'strip'. You can change this name if you want, but you need to use the same name later in your code or it won't work.
- Lastly, we only have 5 LED's on the board, so change the number 24 to 5. Your code should look like this:

Setting all the LED's to 1 colour

- The easiest way to get started is to set all the LEDs to one colour.
- In the 'Neopixel' group find 'strip show color red'. Drag this under the last block.
- *Remember - if you changed the variable name from 'strip' you'll also need to change it here.*
- Click on the word 'red' and change it to 'white'.
- Now go into the 'input' group, and drag an 'on button A pressed' block into your code.
- Go back to 'Neopixel' and drag another 'strip show color red' block into your input block.
- Pick another colour, whichever you like!
- Grab another button input block, but this time lets use button B.
- Drag another 'strip show color red' block into your button B, and pick another colour.



- Make sure the small switch on the top of the buggy is switched to off.
- Plug the micro:bit into the computer, don't remove the micro:bit from the buggy.
- Download your code to the micro:bit.
- Leave the robot plugged in as you're going to be testing lots of different bits of code.
- Flick the switch on top of the micro:bit to on and see what happens.
- Now press the A and B buttons on the micro:bit, did the LEDs change colour?
- Well done on getting this far!! 🎉🥳🎉🥳
    - What other micro:bit blocks could you use to let the computer control the lights?
    - Try playing with other inputs to change the LED colours.

Setting all the LED's to a rainbow of colours

If you like a variety of colours, you can get your code to automatically show a rainbow on the LED's!

- Computer code runs very fast, if we keep adding LED blocks one after the other, they will change faster than we can see.
- Lets fix that by adding a pause after our first 'strip show color' block.
- After that, add a 'strip show rainbow 1 to 360' block.

- Download the code to your micro:bit and see what happens.
- Remember to leave the micro:bit plugged in!
- Now try changing the numbers 1 and 360 to different numbers, what happens?

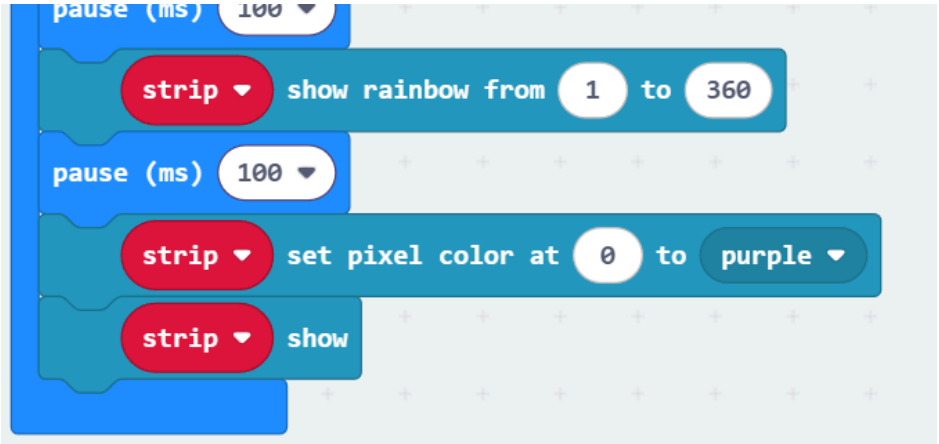
Setting 1 LED to 1 colour

We might want to only light up 1 LED, this takes a bit more thinking, than you might think! Before we get into the code it's important to think about how a computer counts numbers.

If I asked you to count to 5, you'd probably say 1, 2, 3, 4 then 5 right? Computers do it differently, computers always start at 0, not 1. So if you tell a computer to count to 5, it would say 0, 1, 2, 3, then 4.

Why is this important? Well, the LED's are in a row of 5. If we want to switch on the first LED, we need to send that instruction to LED 0, not LED 1. Let's get into the code!

- Add another pause under your rainbow block.
- Go into the 'Neopixel' group, and this time click the more button.
- Find the 'strip set pixel color at 0 to red' block and add this to your code.
- Change the colour, then download your code.
- Did the LED change to the colour you wanted? If not, don't worry, let me explain!
- Look at the blocks that turn on the white lights and the rainbow, they say **show.**
- The block you just added says **set**. What this means is that we've told the LED what colour to be, but we've not told it to show us yet.
- Now find the 'strip show' block, and add it to the end.

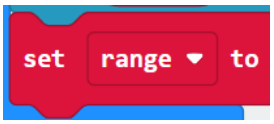


- Now try downloading your code, now it should work!
- Try adding other 'strip set pixel' blocks, but this time change 0 to 1, 2, 3 or 4.
- If you're not sure if it's working correctly, try deleting the rainbow block.

Setting some LED's to 1 colour

Setting 1 LED at a time can be very slow, what if we want to just change a couple of LEDs? We're going to use another variable, and this time we're only going to store 2 LEDs inside it. We're going to work on the 3rd and 4th LED, but remember, the computer calls these LED 2 and 3!

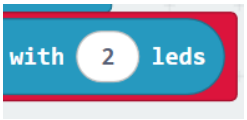
- Go into the 'Neopixel' group again, and look for the second block, it will probably be called 'set range to strip range from 0 with 4 LEDs'.
- Drag this to the bottom of your code. Let's look at this code in more depth:
    - The first part is choosing which variable to set:



    - The second part is telling the code to look at the first variable called 'strip':

    - Range from means, from 'strip' start at LED 2, what humans call the third LED:

    - With 2 LEDs means, from the start point, include 2 LEDs in this variable.



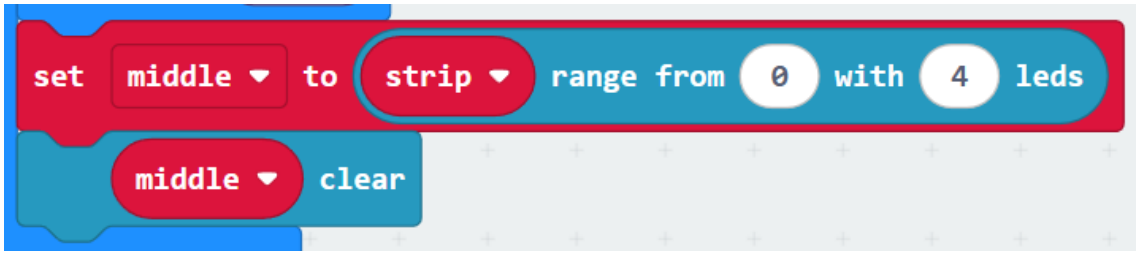
- Your block of code should look like this, change the numbers in your code to match.

- Great job!! Now all we need to do is tell the new variable what colour we want it to be.
- Go into the 'Neopixel' group, and grab another 'strip show color' block.
- Add this to the bottom of your code, and click the white arrow next to the word strip.
- In the menu, click the word 'range'.
- Now download the code to your micro:bit and give it a try! Did it work like you wanted?
- Why don't you try with different numbers, does it make sense?

Turning off the LED's

We need to know how to turn LEDs off. Strangely, this needs 2 blocks to work, let's take a look!

- With this code, we're going to turn off LEDs 2, 3 and 4. Remember that the computer knows these as LEDs 1, 2 and 3.
- First let's make a new variable to store the LEDs we want to turn off.
- Add a 1 second pause block at the bottom of your code.
- Now go into the Neopixel group and get the second block again, it will probably be called 'set range2 to strip range from 0 with 4 leds'
- Drag this block to the bottom of your code.
- Let's rename this group to 'middle'.
- Click on the word 'range2' and from the menu click 'Rename variable…'.
    - In the pop up box, type the word middle, and click ok.
- We need to start at the LED we count as 2, but that the computer counts as 1.
    - Change the first number 'range from' to 1.
- We want to turn off the middle 3 LEDs, so change the second number to 3.
- Now go back into the Neopixel group and find the block 'strip clear'.
- Add this to the bottom of your code, and change 'strip' to 'middle'.



- Download the code to your micro:bit, did it work?
- A bit like when we set the colour earlier, we've told the LED we want it to clear, we just haven't told it to do it yet.
- This will sound odd, but we need to add a show block to the bottom, then it will work.
- Remember to change 'strip' to 'middle' or it won't work!
- Try downloading your code now, it should all be working correctly, well done!
- You can also use the clear and show blocks with the full set of LEDs, just change the variable you've selected first!

Well done, you've got to the end of the LED instructions. Now, can you put these together with the driving instructions???