

Pico 1.5 LED strips using Fade & Random

If you have not done so already look at Pico 1.4 LED strips guide. That guide shows you how to import the neopixel library and set up your Pico and breadboard.

If you have completed the 1.4 guide, let's do more with LED strips!

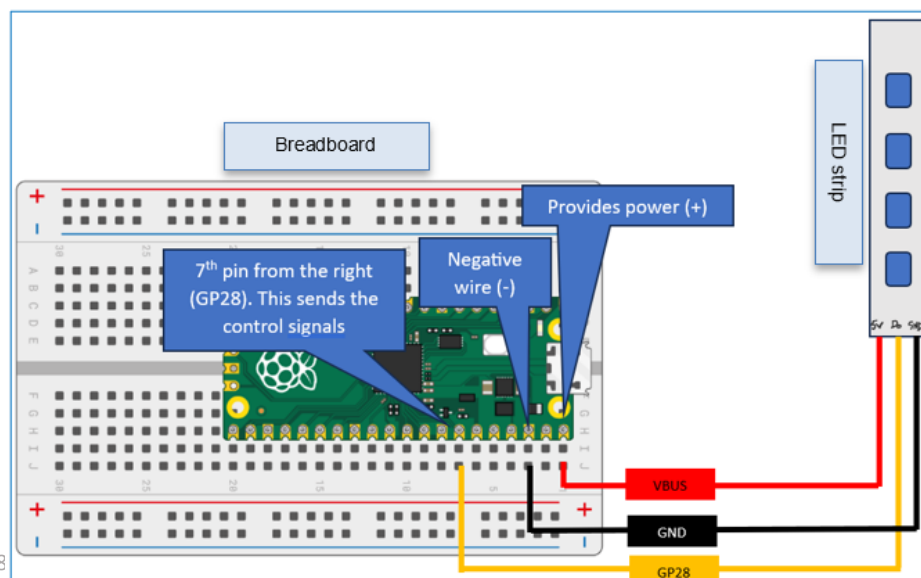
In this guide we will look more closely at the code so we can control the LEDs.

- ✓ We will make the LED strip fade in and get brighter.
- ✓ We will use the **random** function to select colours.
- ✓ Use **random** to select which LEDs light up



Contents

Test Code:	2
Challenge 1 – more LEDs and different colours:.....	2
Changing the 'Strip Brightness' variable.	3
Challenge 2 – fading LED strip:.....	4
Using Random with LEDs	4
Challenge 3 – Random:	5
Challenge 4 – Random on each loop:.....	6
Using Random with individual LEDs:.....	6
Challenge 5 – Random colour with while loop:.....	6
Random colour and random position:	7
Challenge 6 – Twinkling light strip:.....	8
Challenge 7 – Random strip brightness:.....	8



Test Code:

Copy/paste this test code:

```
1  ###Example 1 - Control individual LEDs###
2  #Libraries
3  from machine import Pin
4  from neopixel import Neopixel
5  import time
6
7  #Variables for LED strip
8  numpix = 14
9  strip = Neopixel(numpix, 0, 28, "RGB")
10 strip.brightness(42)
11
12 #colour variables
13 red = (0, 255, 0)
14 green = (255, 0, 0)
15 blue = (0, 0, 255)
16 blank = (0,0,0)
17
18 #commands
19 while True:
20     strip.set_pixel(0, blue)
21     strip.set_pixel(1, red)
22     strip.set_pixel(2, green)
23     strip.show()
24
```

Test Code:

```
###Example 1 - Control individual LEDs###
#Libraries
from machine import Pin
from neopixel import Neopixel
import time

#Variables for LED strip
numpix = 14
strip = Neopixel(numpix, 0, 28, "RGB")
strip.brightness(42)

#colour variables
red = (0, 255, 0)
green = (255, 0, 0)
blue = (0, 0, 255)
blank = (0,0,0)

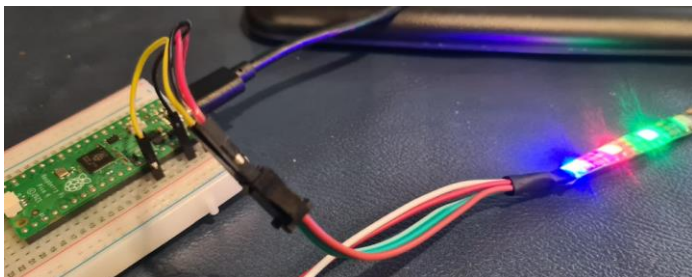
#commands
while True:
    strip.set_pixel(0, blue)
    strip.set_pixel(1, red)
    strip.set_pixel(2, green)
    strip.show()
```



Click the green run button on Thonny to test this code.

Did your strip light up the first three LEDs?

Was one LED blue, the next red and the third one green?



Challenge 1 – more LEDs and different colours:

- ✓ Can you make more LEDs light up?
- ✓ Can you add more colour variables?
- ✓ Can you change the brightness of the LEDs?
- ✓ What is the maximum brightness value?

You can use the [W3Schools colour picker website](https://www.w3schools.com/colors/colors_picker.asp) to find out the RGB values for different colours.

Changing the 'Strip Brightness' variable.

You have a line of code that sets the brightness of your LED strip.

```
10 strip.brightness(42)
```

This is a variable, so it can be changed.

Try changing this variable value to see what happens.

```
10 strip.brightness(5)
```

Try out different values.

What are the lowest and highest settings that work?

I do not believe that Neopixel can set the brightness of individual LEDs on the strip.

If you find one, let me know!

But we can still use the strip.brightness() variable in fun to make the whole strip fade or brighten.

In the example below I have moved the variable for strip brightness into a For Loop.

```
1  #####Fading Strip Brightness###
2  #Libraries
3  from machine import Pin
4  from neopixel import Neopixel
5  import time
6
7  #Variables for LED strip
8  var1 = 0
9  numpix = 14
10 strip = Neopixel(numpix, 0, 28, "RGB")
11
12
13 #colour variables
14 red = (0, 255, 0)
15 green = (255, 0, 0)
16 blue = (0, 0, 255)
17 blank = (0,0,0)
18
19 #commands
20 for x in range(5):
21     strip.brightness(var1)
22     strip.fill(red)
23     strip.show()
24     time.sleep(1)
25     var1 = var1 + 50
26
27 #command after escaped the loop
28 strip.fill(blue)
29 strip.show()
```

This sets the starting value for strip brightness.

Every time this loops, it checks the value for strip brightness.

Every time this loops, the value assigned to var1 goes up by 50.

Challenge 2 – fading LED strip:

Use this test code to solve this challenge. 

- ✓ Can you make it start faded and get brighter but take longer and do it smoothly.
- ✓ Can you make it get to full brightness, and then fade back to nothing.
- ✓ Can you make it fade from nothing to full brightness, fade back to nothing and then repeat the process with a different colour?

Test Code:

```
###Fading Strip Brightness###
#Libraries
from machine import Pin
from neopixel import Neopixel
import time

#Variables for LED strip
var1 = 0
numpix = 14
strip = Neopixel(numpix, 0, 28, "RGB")

#colour variables
red = (0, 255, 0)
green = (255, 0, 0)
blue = (0, 0, 255)
blank = (0,0,0)

#commands
for x in range(5):
    strip.brightness(var1)
    strip.fill(red)
    strip.show()
    time.sleep(1)
    var1 = var1 + 50

#command after escaped the loop
strip.fill(blue)
strip.show()
```

Using Random with LEDs

In previous activities you may have used the **random function** like this:

```
1  import random
2
3  shoplister=["tea","crackers", ]
4
5  #print the whole list
6  print(shoplister)
7
8  #print one element from the list, chosen at random
9  print(random.choice(shoplister))
10
```

Line 3 creates a list called **shoplister**.

This list contains two **elements** :["tea", "crackers",]

Line 9 randomly prints an element from **shoplister**.

We can use random in the code for our LED strip.

Line 22 creates a list called **colourlist**.

This list stores 3 **elements**.

```
21 #using random
22 colourlist = [red,green,blue]
23 colour = random.choice(colourlist)
```

Line 23 randomly an element from the list at random.

This then gets saved into the **colour** variable.

The **colour** variable is then used in a For Loop.

```
25 #commands
26 for x in range(5):
27     strip.fill(colour)
28     strip.show()
29     time.sleep(1)
30     strip.fill(blank)
31     strip.show()
32     time.sleep(0.1)
```

Challenge 3 – Random:

Use this test code as a starting point.

- ✓ Can you add more elements to the list?
- ✓ Can you change the lighting sequence?

Test Code:

```
###Use RANDOM to select colour###
#Libraries
from machine import Pin
from neopixel import Neopixel
import time
import random

#Variables for LED strip

numpix = 14
strip = Neopixel(numpix, 0, 28, "GRB")
strip.brightness(50)

#colour variables
red = (0, 255, 0)
green = (255, 0, 0)
blue = (0, 0, 255)
blank = (0,0,0)
purple = (204, 0, 255)

#using random
colourlist = [red,green,blue]
colour = random.choice(colourlist)

#commands
for x in range(5):
    strip.fill(colour)
    strip.show()
    time.sleep(1)
    strip.fill(blank)
    strip.show()
    time.sleep(0.1)

#command after escaped the loop
strip.fill(purple)
strip.show()
```

Challenge 4 – Random on each loop:

The problem with the test code above is that the colour selected is the same for all 5 loops.

- ✓ Can you develop the program, so it assigns a new value to **colour** every time it loops?

Using Random with individual LEDs:

Let's see if we can use the random function with some of the LED programming we have learned recently.

In the 1.4 guide we used a command such as this:

```
12 strip.set_pixel(1, red)
13 strip.show()
```

Line 12 selects the colour and the LED to light up.

We can set the **colour** using a variable instead of hard coding the colour **red**.

Update your code, removing the for loop and using these commands.

```
21 #using random
22 colourlist = [red,green,blue]
23
24 #commands
25 strip.set_pixel(1, colour)
26 strip.show()
27 time.sleep(0.2)
```

When you run the program, does it display a different colour **each** time you click run?

Challenge 5 – Random colour with while loop:

- ✓ Can you develop the above code, so it loops endlessly?
- ✓ It should select a different colour each time it loops.

Random colour and random position:

We can use the random function for more than just colour choice.

Use the random function to select which LED on the strip we want to light up.

We used this code in a previous example.

The LED to light up is hard coded as 1.

```
27 strip.set_pixel(1, colour)
```

We can use **random** to select a number between 1 and 14 randomly.

```
24 #commands
25 ledno = random.randint(0,13)
26 colour = random.choice(colourlist)
27 strip.set_pixel(ledno, colour)
28 strip.show()
29 time.sleep(0.2)
```

We do *not* need to create a new list (like we did for colour).

We can use **random.randint()** to select a number between 0 and 13.

The range is 0 to 13 because I am using a 14 LED strip.

Adjust your code accordingly.

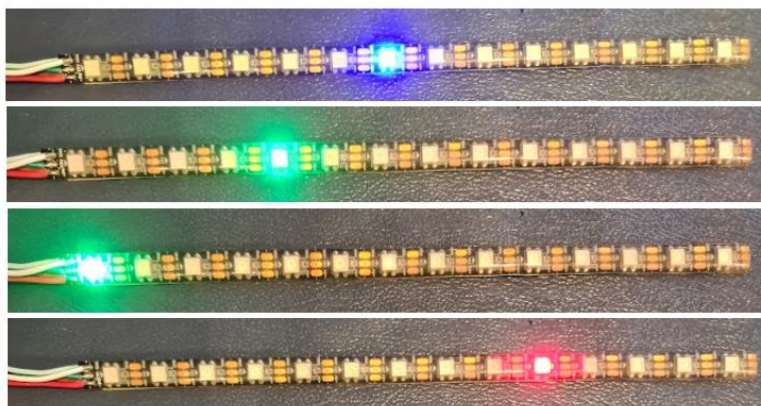
On line 25 an integer value gets assigned to the variable **ledno**.

This variable is then used on line 27.

It selects which LED should light up.

Update your code as shown above.

When you run it, does it select a different colour and position each time you run?



Challenge 6 – Twinkling light strip:

Place the code shown above into a For Loop.

- ✓ Can you make it so the colour and position are randomly selected?
- ✓ Make it loop 10 times.

You will probably get an output that looks something like this.



- ✓ Develop your code so one LED goes blank before the next one lights up.
- ✓ Can you make it loop ten times as a fast sequence then times as a slow sequence?
- ✓ Once you have got this working can you then put it all into a While Loop so it loops endlessly?

Challenge 7 – Twinkle two lights:

- ✓ Can you develop your program so two LEDs light up on each loop?

Here is a clue:

```
ledno1 = random.randint(0,13)
ledno2 = random.randint(0,13)
```

Challenge 7 – Random strip brightness:

- ✓ Can you create a variable to be used for strip brightness?
- ✓ Can you randomly assign a value to that variable?
- ✓ Can you develop your twinkling light program so it fades in and out as well as randomly selecting lights and colours?

