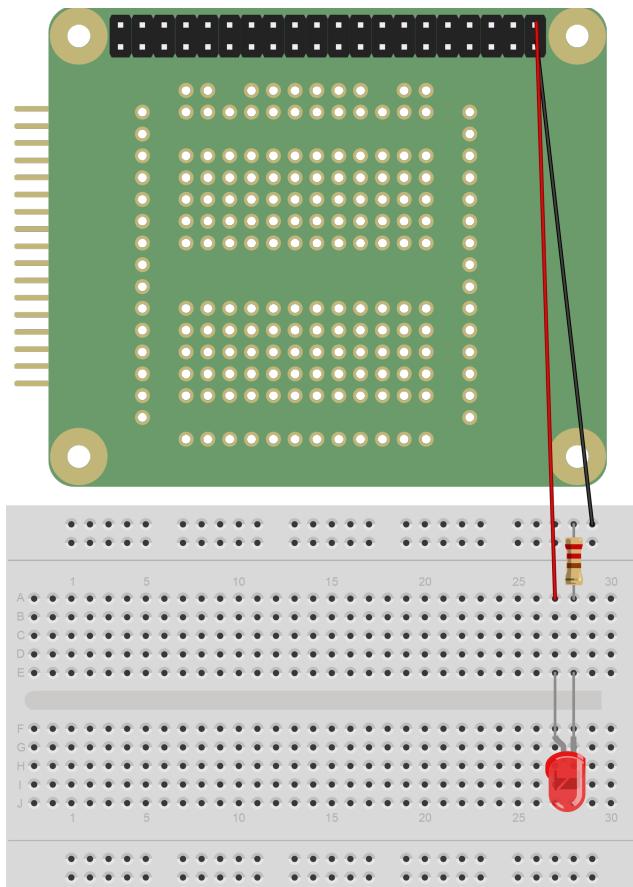


Your First Circuit



Take a look at this circuit.

You will need an LED, a resistor and two male/female jumper cables.

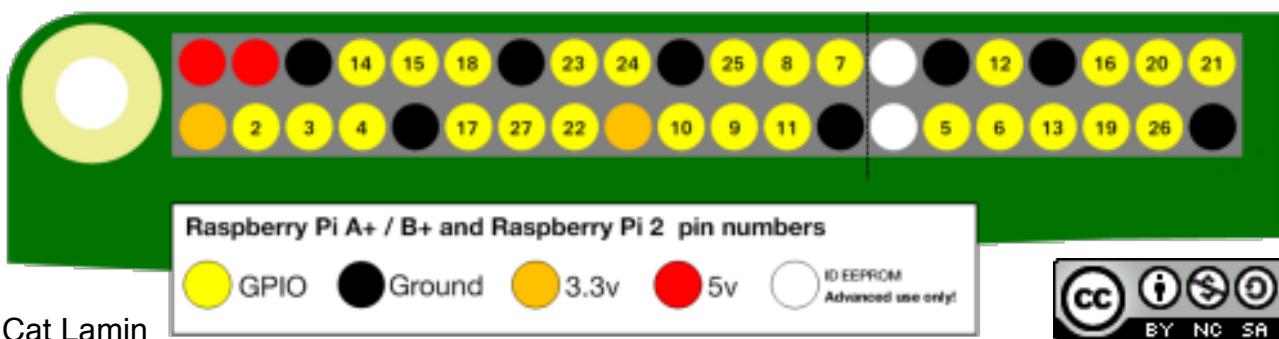
One of the jumper cables needs to run from the bottom right hand pin on either the Raspberry Pi or the proto board (this is a GROUND pin) into a ground rail on your breadboard.

You will need to place your LED so that the two legs are in different rails of the breadboard as shown in the diagram.

The shorter, negative leg, needs to be connected to ground by attaching a resistor from the rail it's on, into the ground rail.

On the rail with the positive leg, you will need to use a jumper cable to connect it to the top right hand pin, which is called pin 21.

The diagram below shows the numbers of the pins - you may need it later!





Scratch

To use Scratch with the Raspberry Pi you need to use 'broadcast' blocks.

First we need to tell Scratch that we're using the GPIO pins and that we have put an LED on pin 21.

We will need to broadcast:

gpioserveron

and

config21out

Hit the green flag to run this code.



To get turn your LED to turn on and then off, try broadcasting

gpio21on

wait 1 sec

gpio21off

Can you make your LED blink (not flicker)?

Python

Scratch is great, but it has its limits. Now it's time to look at a text-based programming language: Python

Let's look at the code we just wrote in Scratch and re-write it for Python.

First, we need to import some libraries - this tells Python we're using the GPIO pins with LEDs. We also can't use the 'sleep' command

```
from gpiozero import LED
from time import sleep
```

without importing the function from the time library.

Next, we need to create a variable - we can call our variable anything we want, I've chosen to call mine 'red' because it's the red LED. I've then told Python that 'red' refers to the LED on pin 21.

```
from gpiozero import LED
from time import sleep

red = LED(21)
```

We're ready to run some code - we're going to use the statement 'while True' to create a forever loop.

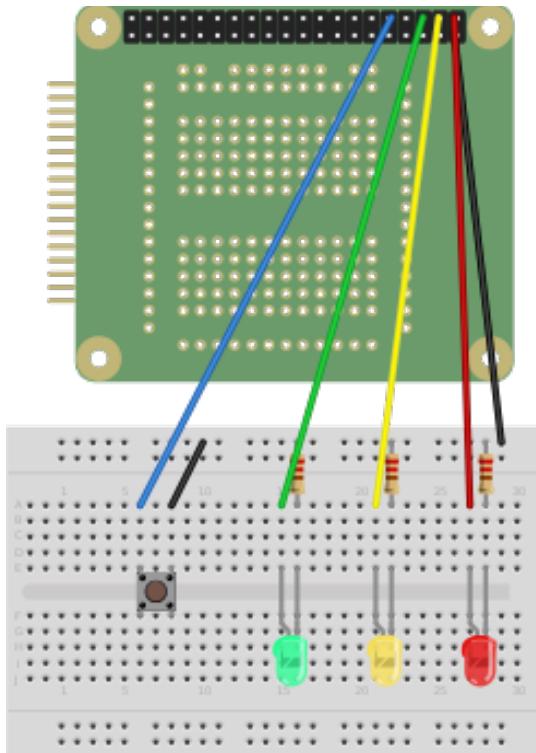
```
from gpiozero import LED
from time import sleep

red = LED(21)

while True:
    red.on()
    sleep(1)
    red.off()
    sleep(1)
```

Try adjusting the number in brackets after the word 'sleep'. Try decimals and single digit numbers to see how it affects the LED.

Next Steps



Here is an example of how you would wire up your circuit to include three LEDs and a button.

For each LED, you'll need to create a variable e.g.

```
amber = LED(20)
```

If you want to use a button, you will need to import the 'button' library and create a button variable e.g.

```
from gpiozero import Button
button = Button(12)
```

The button is connected to pin 12, which is why we've used 12 in brackets.

Try some of the codes below to see what they make your LED and buttons do! Can you modify your code to alter different LEDs?

```
from gpiozero import LED, Button
from signal import pause

red = LED(21)
button = Button(12)

button.when_pressed = red.on
button.when_released = red.off
```

```
from gpiozero import LED, Button
from time import sleep

red = LED(21)
button = Button(12)

button.wait_for_press()
red.on()
sleep(3)
```

```
from gpiozero import LED, Button
from time import sleep

red = LED(21)
button = Button(12)

while True:
    button.wait_for_press()
    red.toggle()
```