

# Blink

1

Get to know the Raspberry Pi, its GPIO pins, and Python IDLE  
Make an LED flash on and off

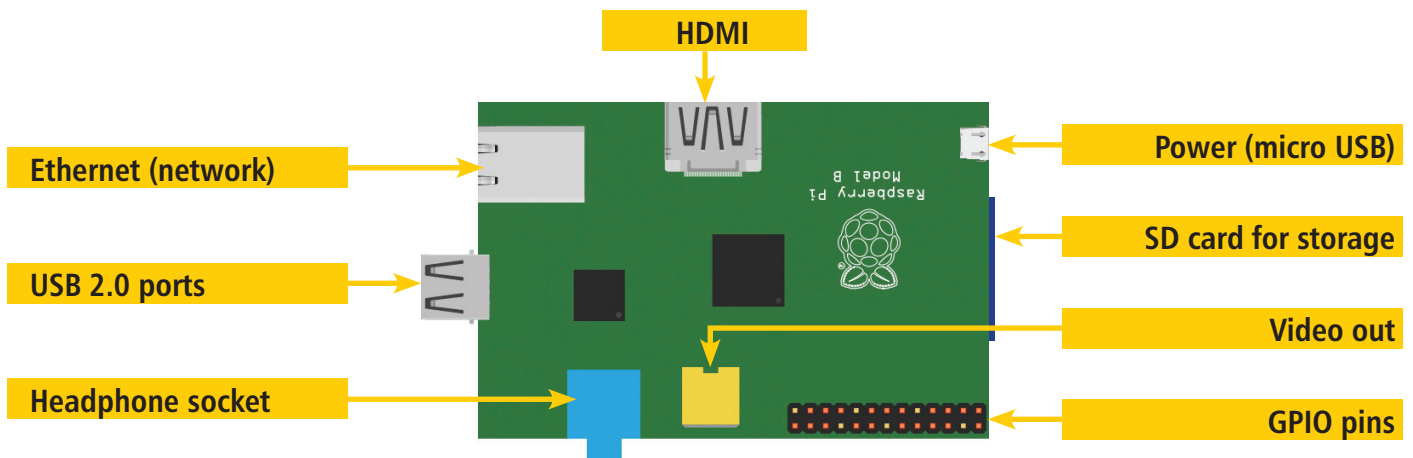


30 mins

## 1. Introduction to components

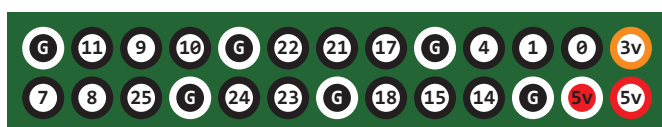
### The Raspberry Pi 1A

This tutorial is for the Raspberry Pi 1A, which has a different layout of GPIO pins to later versions. Please go to [mr.langford.co](http://mr.langford.co) for updated worksheets as they become available.



### GPIO pins

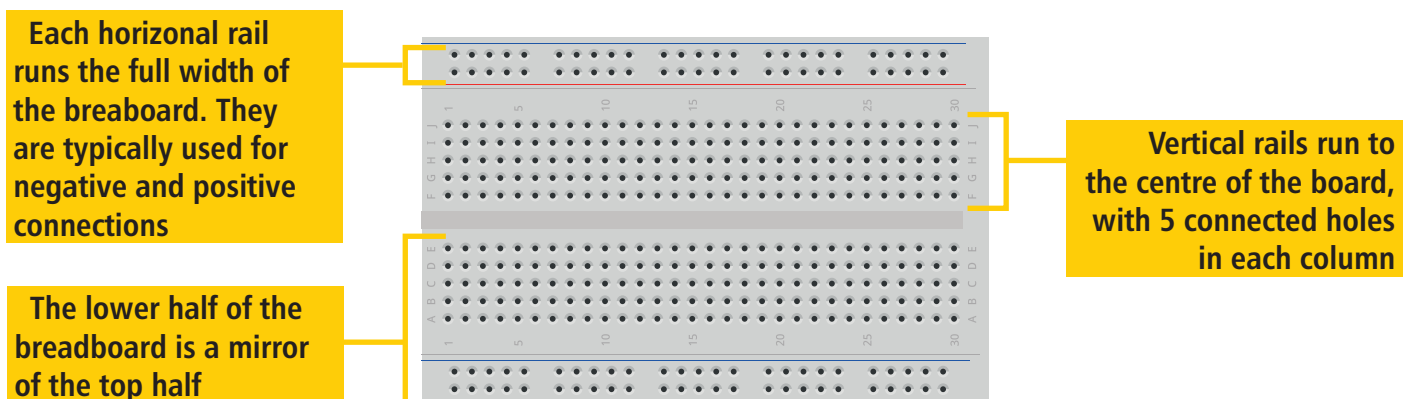
We will be connecting electronic components to the Pi's GPIO (general purpose input output) pins, and controlling them using Python 3. We will refer to the Broadcom pin connection numbers. The pin layout is shown below:



- +3.3 Volt pin
- +5 volt pin
- GPIO pins
- Ground pins

### Breadboard

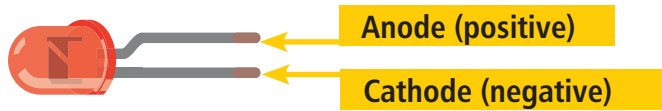
The pins of electronic components are inserted into holes in the breadboard. This allows you to quickly and easily build and change electronic circuits without the need for soldering.



## 2. Components list

### Light emitting diode (LED)

LEDs must **always** be used with a resistor, or they may be damaged. LEDs must be used the correct way around. The short pin (straight leg in diagrams) is the cathode (negative).



The components may have sharp points or edges. Be careful when using them.

### Resistor

Reduces current flow in a circuit. Resistance is measured in ohms, and is shown using a code of different coloured stripes. For this project, you need a 220 ohm resistor (red, red, brown).

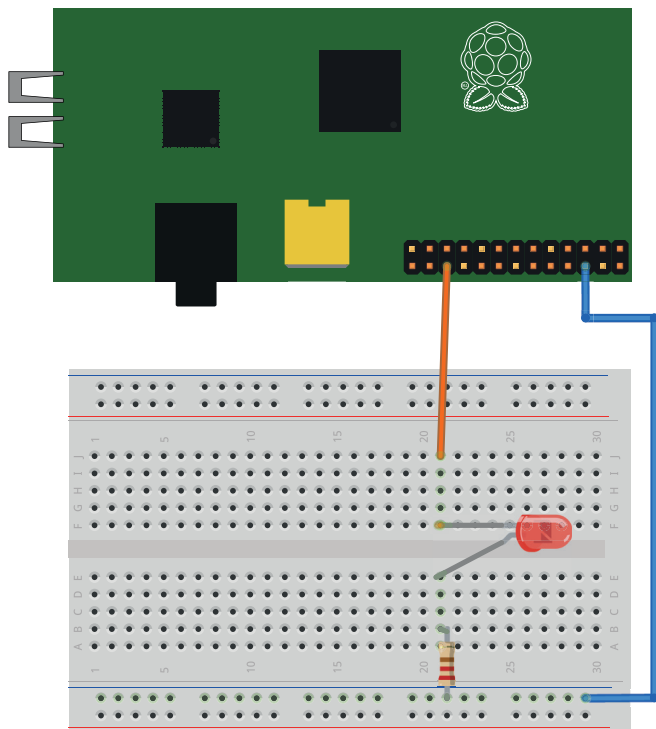


## 3. Connect components

Connect your components as shown.

The colour of the wires (called jumpers) is not important.

Remember to check that the LED is inserted the correct way around. Inserting it the wrong way around will not damage it, but it just won't do anything.



## 4. Python program

Enter the program below. Save it and run it.

To quit it, press CTRL and C, or close the Shell.

```
1. # Import Raspberry Pi GPIO library
2. import RPi.GPIO as GPIO
3.
4. # Import sleep function from the time module
5. from time import sleep
6.
7. # Ignore warnings
8. GPIO.setwarnings(False)
9.
10. # Use BCM pin numbering
11. GPIO.setmode(GPIO.BCM)
12.
13. # Set pin 8 to be an output pin
14. GPIO.setup(8, GPIO.OUT)
15.
16. # Print a message to the shell so that we
17. # know the program is running
18. print("Program running")
19.
20. # Run forever
21. while True:
22.     GPIO.output(8, GPIO.HIGH) # Turn on
23.     sleep(1) # Sleep for 1 second
24.     GPIO.output(8, GPIO.LOW) # Turn off
25.     sleep(1) # Sleep for 1 second
```

## 5. Challenge activity

Congratulations on creating your first electronics project! Can you add another LED which is on when the first one is off, and vice versa?

## Credits

Based on "Blink" by Soren - <https://raspberrypi.com/making-a-led-blink-using-the-raspberry-pi-and-python>