



1. GETTING STARTED

So you want to make an LED light up, using your Raspberry Pi and Python?

First things first, GPIO stands for General Purpose Input Output, LED stands for Light Emitting Diode. A diode allows a current to flow, usually in one direction. Current is a flow of electrons or more commonly known as electricity!

2. WARNING

You **can't** just connect an LED and light it up it may blow up the light or even blow up your Pi. All normal electrical safety procedures should be followed. If unsure, don't do it!

```
import RPi.GPIO as GPIO
import time

choice = raw_input("Do you want to
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
GPIO.output(11, False)
```

3. PARTS / COMPONENTS

You require the following parts, **leads** (*male to female ends*), **LEDs**, **Resistors** suitable for the LED, **Solder-less Breadboard**, (*The clever little breadboard allows you to create a circuit without having to solder any components. It is important to make sure you know which way the connections run, usually horizontally*), and your **Raspberry Pi**. It is also worth getting the most up to date Disk Image for you Pi from the official site.

To calculate what resistor to use with which LED use this tool, <http://www.hebeiltd.com.cn/?p=zz.led.resistor.calculator>

To calculate the resistance value use this useful online tool: <http://www.csqnetwork.com/resistcolcalc.html>

Or, most electronics shops will calculate the LED and resistance to use before you buy.



6. PROGRAMMING CODE

Using **Python 2.6** create the program below.

```
Import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
GPIO.output(11, True)
```

- GPIO.setup(11, GPIO.OUT) tells the Pi which PIN the current is following out of,
- GPIO.output(11, True) tells the Pi that the output is true, therefore the LED is on
- To turn the LED off use **False**, GPIO.output(11, False)
- Use the **import Time**, and **time.sleep(4)** between the True and False, (On and Off) to make the LED turn on for 4 seconds then turn off.

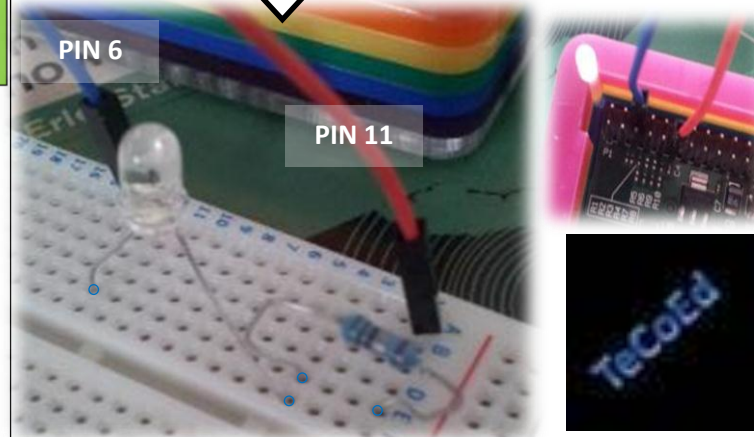
4. CONNECTING THE WIRES

The first part is to connect the wires, the first from **PIN11**, this is the 'output' wire which you will program to provide a current to the LED. The second wire, **PIN6** is for the current to return back to the Pi and complete the circuit. (This PIN is rated 0V). If you imagine that PINs11 and 6 are taps and are joined together with a pipe. If you turned the "PIN 11 tap" on the water would flow from PIN11 through the pipe to 6. This happens with the current, it flows from PIN 11 through the LED to PIN 6.

You may be required to install the GPIO module. Open the LX Terminal and type the code, `sudo apt-get install python-rpi.gpio`

5. CONNECT THE LED AND THE RESISTOR.

The long leg of the LED is the positive end **which must be** connected to the **PIN 11** wire, the current. If you get this around the wrong way you can blow the LED. **PIN 6** feeds the current back to the Pi. PIN6 should be in line with the LED short leg.



7. RUN THE CODE

The program will only run in the **Super User** mode open the LX Terminal and run using the **SUDO** command.

```
Sudo python the nameofyourprogram.py
```

```
AttributeError: 'module' object has no attribute 'raspberrypi' ~ $ sudo python GPIO.py
```



COMMON ISSUES

- Broken LED, wires in wrong PIN, coding error, components not pushed into breadboard, Resistor too high, broken wire