

SNAP CIRCUIT – BASIC LED CONTROL

WARNING

Make sure you pay close attention to the details, as failure to do so could result in your Raspberry Pi suffering damage and invalidating the warranty. We cannot be held responsible for any damage resulted with incorrectly following the guide.



DESCRIPTION

The Raspberry Pi is very good at talking to many different electrical components with the help of digital input and outputs, serial and with Bluetooth and Wi-Fi. LEDs are great indicators to notify if a specific criteria is met – Such as a new email has been received, code has finished compiling or just to simply let you know if it will rain today.

PHYSICAL COMPUTING?

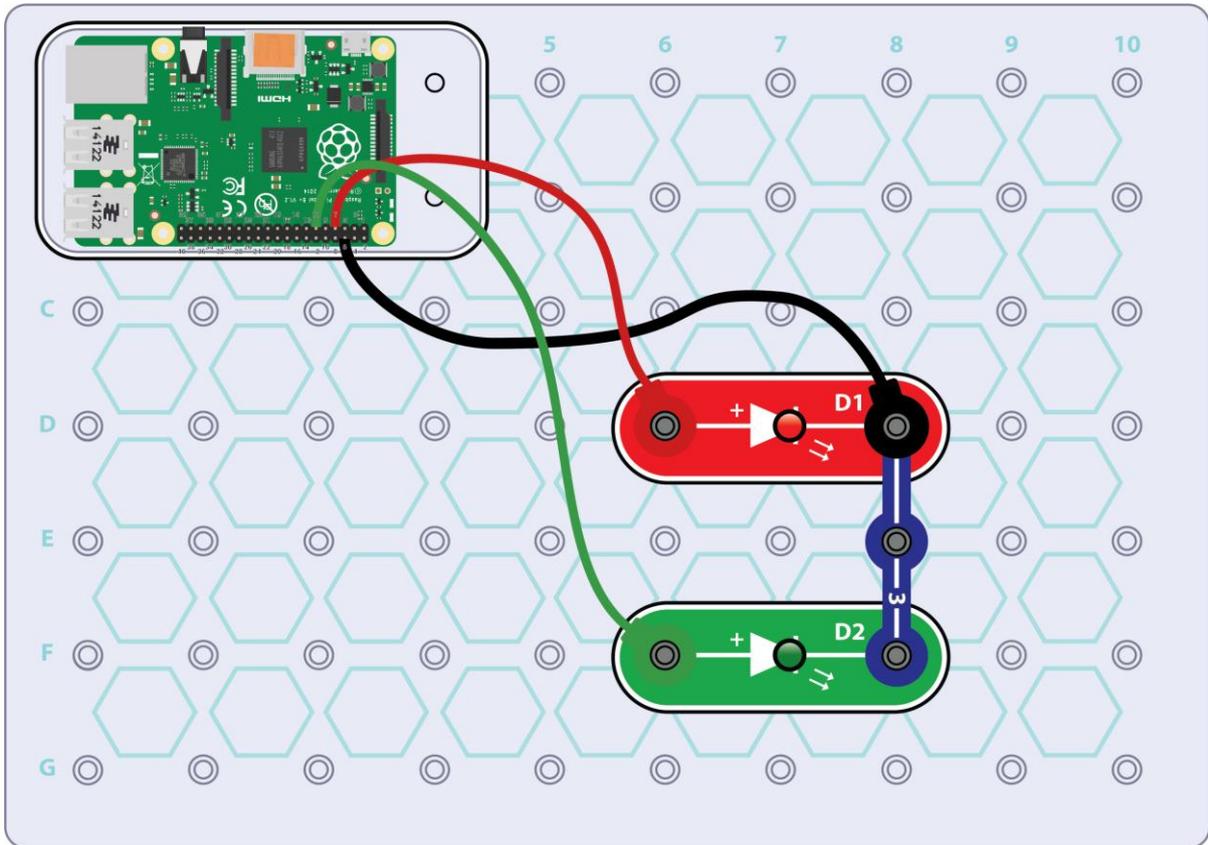
One of the easiest first steps in to physical computing, which really is a fancy name for making stuff you can see and touch, such as LEDs work in a way that you program them to like flash when something happens on screen. This is easily shown with the names; Hardware and Software.

Hardware is the physical things you can see and touch, such as the screen, mouse and keyboard - Including in this case - LEDs.

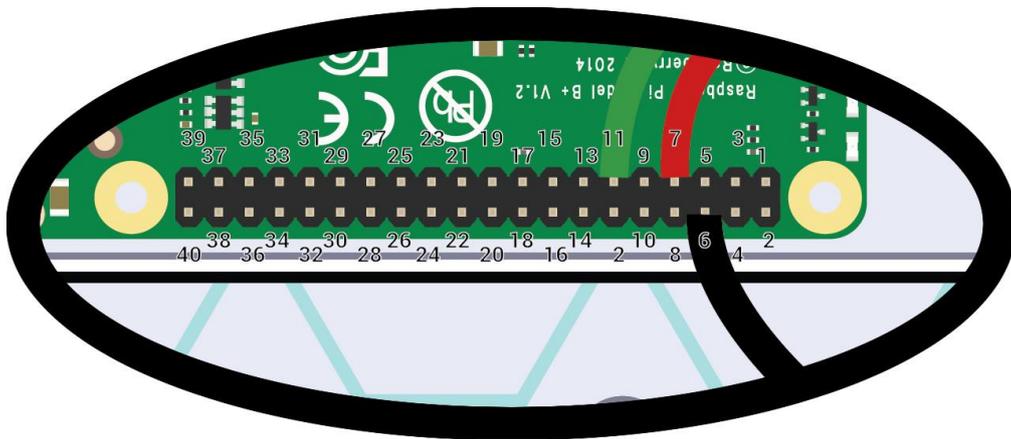
Software is what runs on the computer, biggest being the operating system like Windows, or Raspbian for the Raspberry Pi, then Word Processors, Python or games.

By using Python to mix these 2 worlds together, you will be able to add even more than normal, such as flashing LEDs when a new email or message arrives.

THE CIRCUIT



Firstly place the D1 and D2 LEDs on the base plate. These will go first as the 3 snap bridge sits on top.

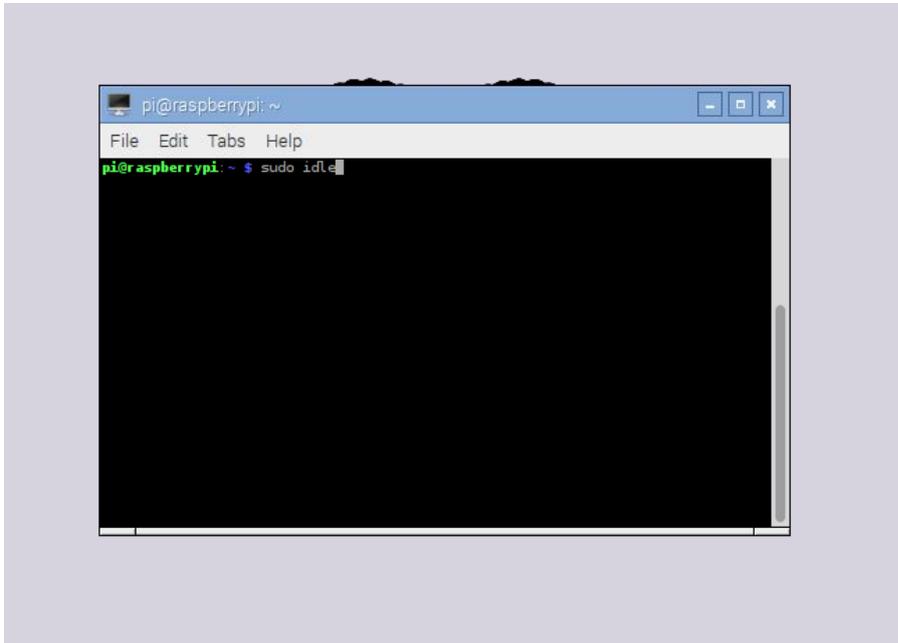


Now to connect the Snap to Pi wires between the Raspberry Pi and circuit.

Carefully count the number of pins on the Raspberry Pi before attaching any wires, be sure to double and triple check that the wire matches the correct pin!

THE PYTHON CODE

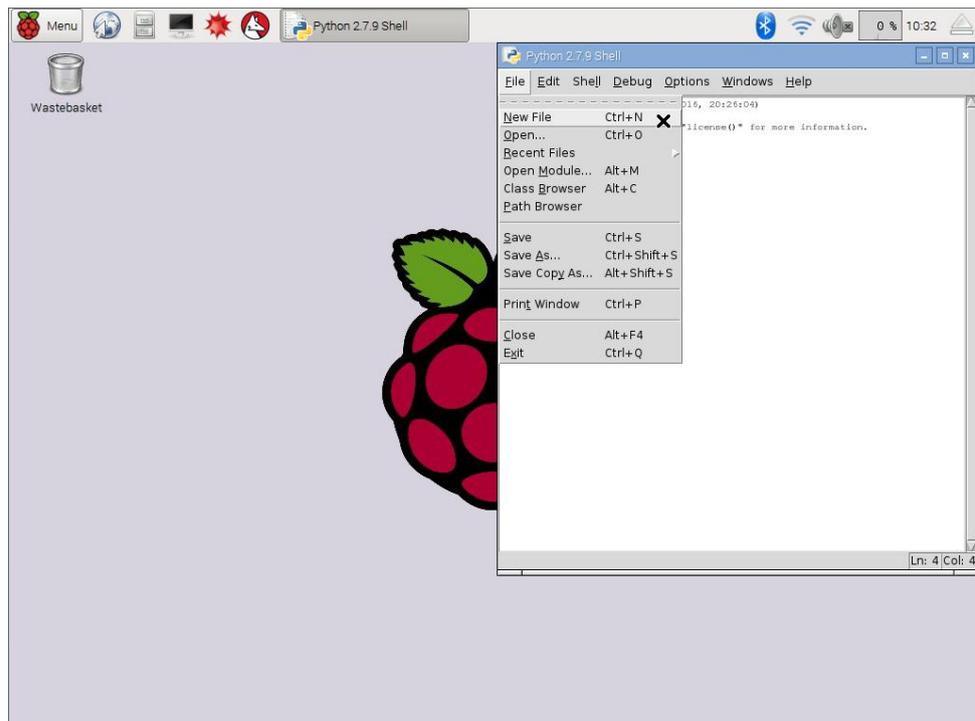
With the Raspberry Pi booted and showing the desktop, open Terminal by either clicking the Black computer screen icon on the task bar, or click Pi Menu (start button), followed by Accessories and then Terminal.

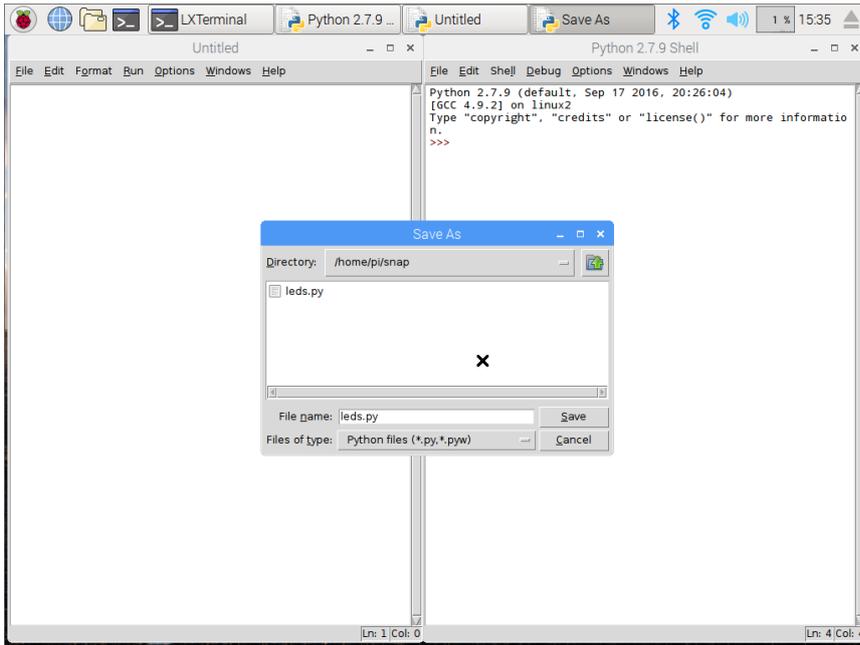


In this box type in `sudo idle`. This will open IDLE for typing our Python code with permission to control the GPIO pins.

Click File and then New File;

In the window that pops up, click *File, Save As* and give it a name to remember





Type in the following code in to this window, pay attention to capitals and spacing;

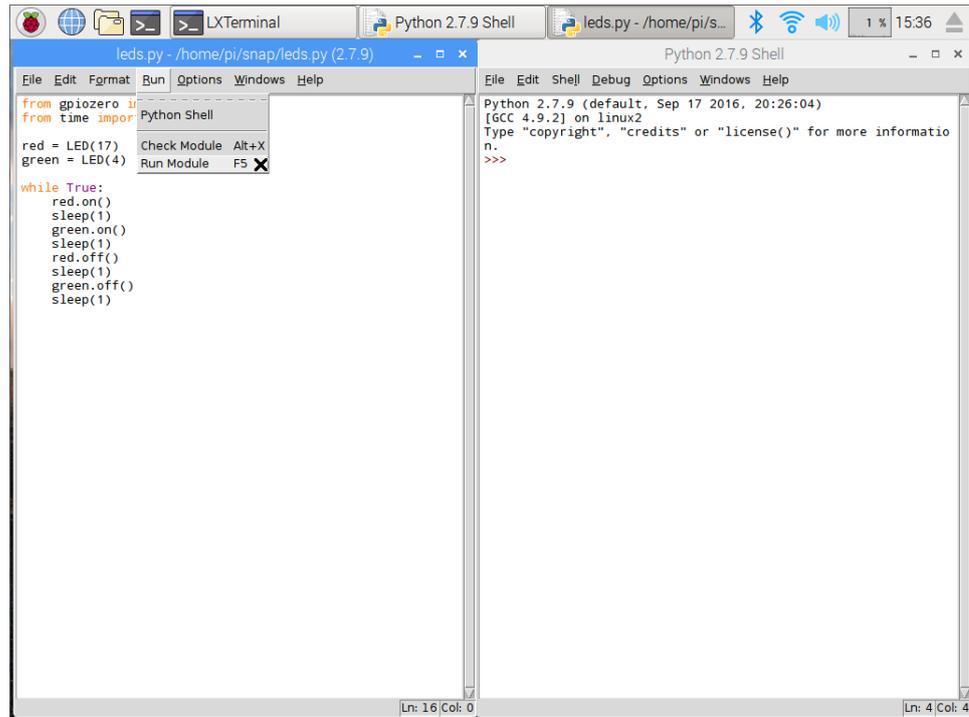
```
from gpiozero import LED
from time import sleep

red = LED(4)
green = LED(17)

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

Now double check the code to make sure no mistakes were made, otherwise the code might run and not give the expected results, or present you with an error.

When you are happy, click Run, and then *Run Module*.



The screenshot shows a Python IDE window titled 'leds.py - /home/pi/snap/leds.py (2.7.9)'. The code in the editor is as follows:

```
from gpiozero import LED
from time import sleep

red = LED(17)
green = LED(4)

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

The IDE also shows a Python Shell window on the right with the following output:

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
```

Immediately the LEDs will start to turn on and off in sequence. As this was created with a `while True:` loop, this will run forever until told to stop.

To actually stop the script, just hold CTRL and press C.

WHAT IS HAPPENING

The Python code will run in a forever loop, the first line after the imports and setting up is `red.on()`. Self explanatory – It turns the red LED on, and keep it on for 1 second with the help of `sleep(1)`. Reading through the rest of the code, next it turns the green LED on, for another second before turning the LEDs off again. Once it reaches the end, the loop begins again.

The LEDs were set up with the GPIO BCM numbering for GPIO 4 (pin number 7) and green with GPIO 17 (pin number 11). More LEDs can be added by choosing a spare GPIO pin – Which there are plenty of.