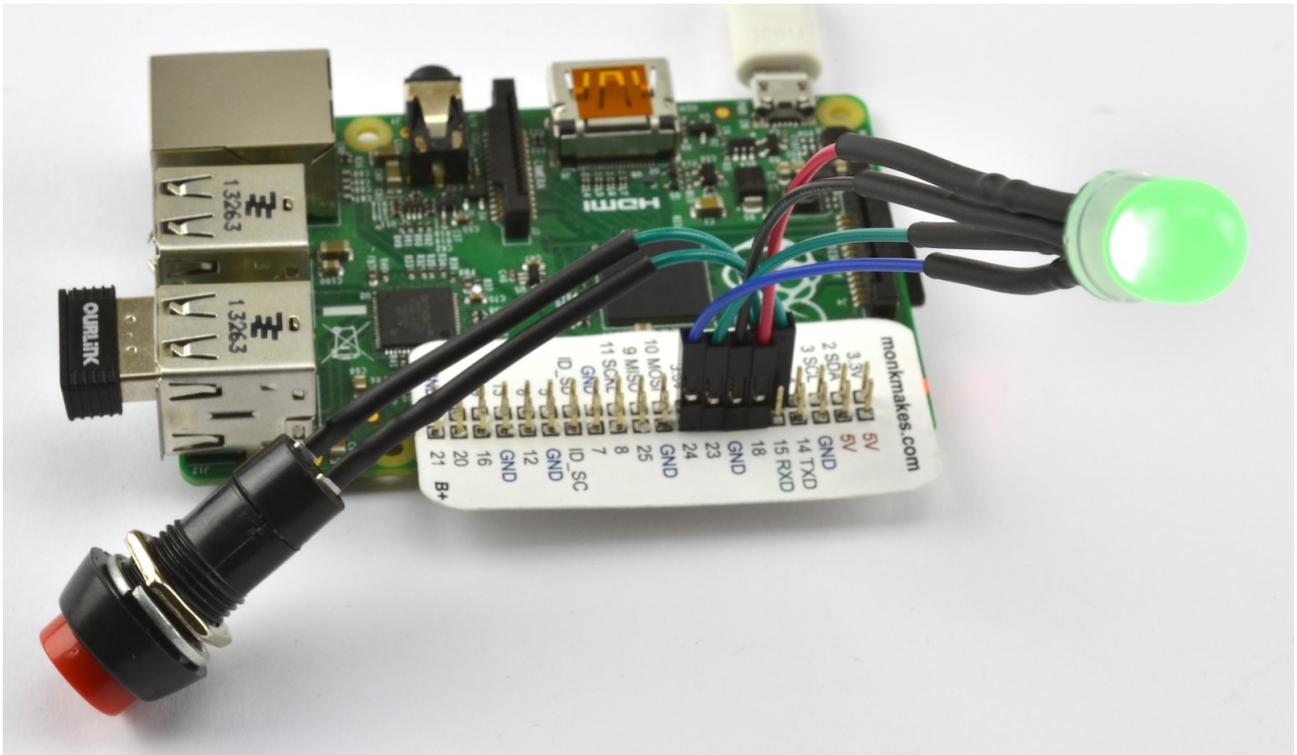


MonkMakes Raspberry Squid and Squid Button

The Raspberry Squid RGB LED and the Squid Button are useful little add-ons that can be directly connected to a Raspberry Pi, by simply pushing their header leads onto the GPIO header of the Raspberry Pi.



They are available in a number of product configurations:

- Individual Raspberry Squid RGB LED
- Twin pack of Raspberry Squid RGB LEDs (with Raspberry Leaf GPIO template)
- Squid Button Individual (momentary action)
- Squid Button Individual (latching)
- Squid Button Twin Pack (momentary action) (with Raspberry Leaf GPIO template)
- Squid Combo Kit (1x LED, 2x Button, 1x Raspberry Leaf)

Raspberry Squid

The Raspberry Squid is an RGB LED with built-in resistors and header lead sockets that can fit directly onto GPIO pins of a Raspberry Pi.

There are three connections (red, green and blue leads) that control the red, green and blue channels of the LED's colour respectively. The fourth (black) lead should be connected to GND on the Raspberry Pi.

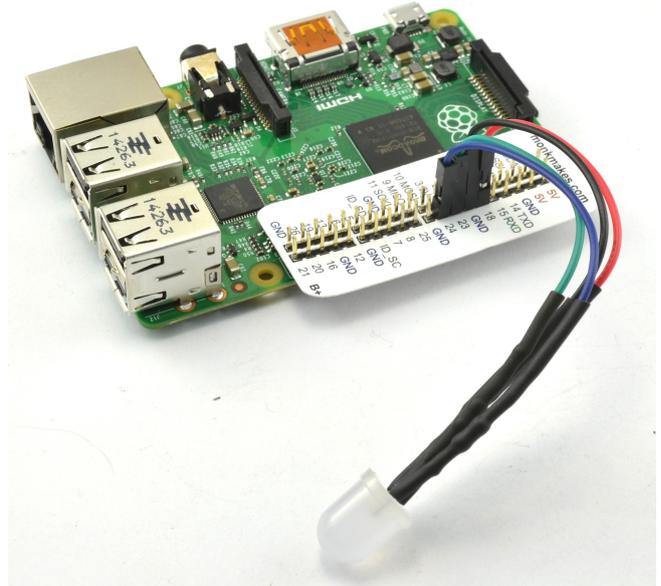


Connecting

Plug in a Squid or connect up an RGB LED as follows:

- Black, Common cathode of the LED to GND (the one between GPIO 18 and 23 is most convenient)
- Red squid lead to GPIO18
- Green squid lead to GPIO23
- Blue squid lead to GPIO24

You can use other pins, but these are the most convenient as the GPIO pins are all in a row (including the GND connection) in the same order as on the Squid itself.



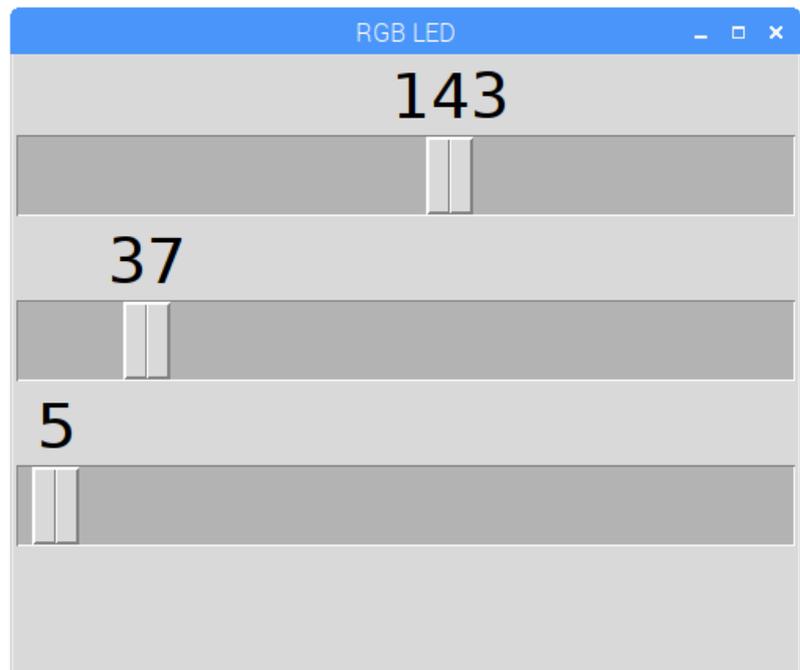
Software

There is a Python library on GitHub here: <https://github.com/simonmonk/squid> for all the squid products. You will find full instructions for using this, as well as for making your own Raspberry Squid there.

However, now that GPIO Zero, the standard Raspberry Pi GPIO library, directly supports RGB LEDs like the Raspberry Pi, you can just use that.

You will find full documentation here: https://gpiozero.readthedocs.io/en/stable/api_output.html#rgbled

Here is an example using the GUI Zero and GPIO Zero libraries that provides three sliders to mix RGB colours.



```
from gpiozero import RGBLED
from guizero import App, Slider
from colorzero import Color

rgb_led = RGBLED(18, 23, 24)

red = 0
green = 0
blue = 0

def red_changed(value):
    global red
    red = int(value)
    rgb_led.color = Color(red, green, blue)

def green_changed(value):
    global green
    green = int(value)
    rgb_led.color = Color(red, green, blue)

def blue_changed(value):
    global blue
    blue = int(value)
    rgb_led.color = Color(red, green, blue)

app = App(title='RGB LED', width=500, height=400)

Slider(app, command=red_changed, end=255, width='fill',
height=50).text_size = 30
Slider(app, command=green_changed, end=255, width='fill',
height=50).text_size = 30
Slider(app, command=blue_changed, end=255, width='fill',
height=50).text_size = 30

app.display()
```

Squid Button

The Squid Button is in fact a momentary action push switch with trailing leads that end in female header sockets that will fit over GPIO pins on a Raspberry Pi, Node MCU or ESP32 board.

There is also a latching version of the Squid Button that when pressed stays in the on position until you press it again to 'unclick' it.

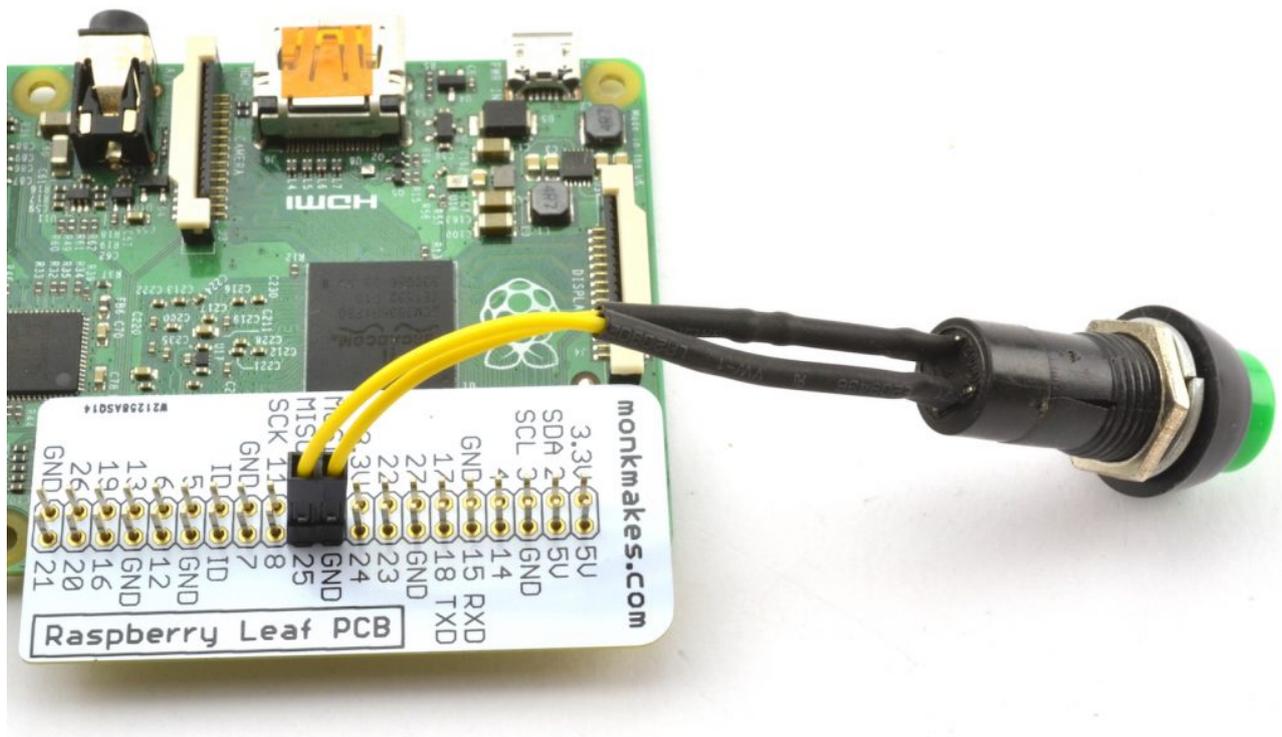
To reduce the chance of accidental short-circuits when using the Button, each Button has a built-in 470Ω resistor that does not affect the switching action when connected to a GPIO input, but does mean that if the switch is accidentally connected across power connections your Raspberry Pi will not come to any harm.

It also gets used to make a Power-on switch for your Raspberry Pi as described here: <http://www.doctormonk.com/2017/01/on-switch-for-raspberry-pi.html>



Connecting

Attach the button as shown below, with one lead connected to GND and the other to GPIO25. If you don't have a Leaf, count along the pins to find the right position.



It does not matter which lead of the button connects to which of the two GPIO pins.

Software

The Squid Button acts just like any other button connected between GND and a Raspberry Pi GPIO pin acting as a digital input with its internal pull-up resistor enabled.

The easiest way to use this is to use the standard GPIO Zero Python library as illustrated in the example below, that assumes a switch between pin 25 and GND as shown in the figure above:

```
from gpiozero import Button

button = Button(25)

while True:
    if button.is_pressed:
        print("Button Pressed")
```