

GPIO Inputs - Button LED Control

In the [GPIO LED Control](#) section, we looked at how to create an output pin on the GPIO and turn it on (3.3v) and off (0v).

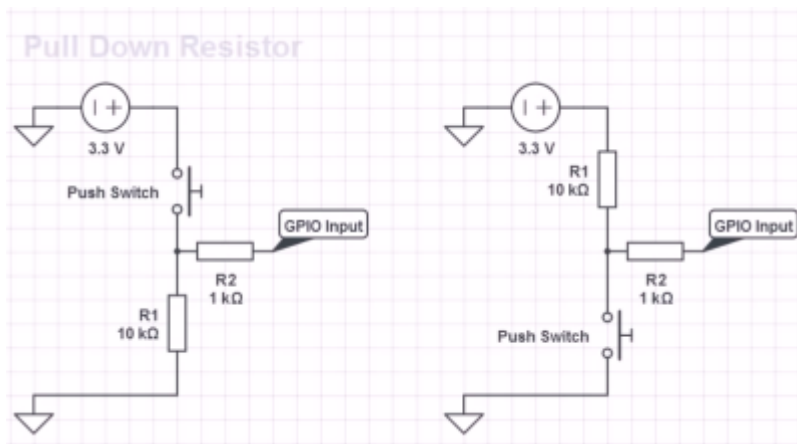
In this section we will look at how to create an input pin, that is a GPIO pin that looks for an input (which will be a voltage). When we talk about voltages, the Raspberry Pi only sees a voltage in terms of an on (3.3v) or off (0v). However, other voltages will fall in to either the on or off state, and roughly speaking, these are:

```
Off = 0v - 1.2v
On = 1.3v - 3.3v
```

Remember, the GPIO pins are 3.3v as they are logic devices, attach more than 3.3v and you might have to buy a new Raspberry Pi!

Pull Up, Pull Down

As we know, the value of a GPIO pin can be between 0v - 3.3v, this is a bit of a problem, because we need a pin to be at a known value so we don't get false positives, for this reason the Pi GPIO actually has built in Pull Up and Pull down resistors.



So for any input that we are waiting for a high signal (3.3v) we will use pull down so it stays a 0v, for any pin we are waiting for a low signal (0v) we will use a pull up so it stays at 3.3v (until a button press makes it low)

If you are using your Raspberry Pi via SSH or a Remote Desktop session, you need to enable remote access of the GPIO Pins, instructions for this are [here](#):

From the command line, create a new file for editing:

```
sudo nano LED-Button.py
```

```
#import modules
import RPi.GPIO as GPIO    # This imports the GPIO library that allows the use of the GPIO pins,
import time                # This imports the time library (for delays among other things)
                             # These libraries are built in to Raspbian.

GPIO.setmode (GPIO.BOARD)  # This sets the GPIO pin numbering. Our LED is connected to Pin 12,
                             # so we can reference it by using BOARD as pin 12. However there is
                             # another option (BCM) where we can reference a pin by it's name, pin
                             # 12 is called GPIO18 (a reference to its place on the chip).

GPIO.setup(12, GPIO.OUT)   # Sets the GPIO pin as output. (as opposed to input, which would
                             # read in a voltage (but only in terms of a 0 or a 1))

GPIO.output(12, 0)         # sets the GPIO Pin 12 to low (so 0v)
time.sleep(1)              # waits one second (settling time)
GPIO.output(12, 1)         # sets the GPIO Pin 12 to high (so 3.3v) LED will turn on
time.sleep(3)              # waits 3 seconds, so LED will be on for 3 seconds
GPIO.cleanup()             # Resets all the GPIO pins to their default state (LED will go off)
```

Last update:

2023/03/09 22:35 gpio_inputs_-_button_led_control https://wiki.alanwalker.uk/doku.php?id=gpio_inputs_-_button_led_control&rev=1470772116

From:

<https://wiki.alanwalker.uk/> - **WalkerWiki** - wiki.alanwalker.uk

Permanent link:

https://wiki.alanwalker.uk/doku.php?id=gpio_inputs_-_button_led_control&rev=1470772116

Last update: **2023/03/09 22:35**

