

hack
@CEWIT

Raspberry Pi 103

Basic Circuits

Matthew A. Cordaro

Part List

Resistors

One 220Ω , $1k\Omega$, & $10k\Omega$

LED

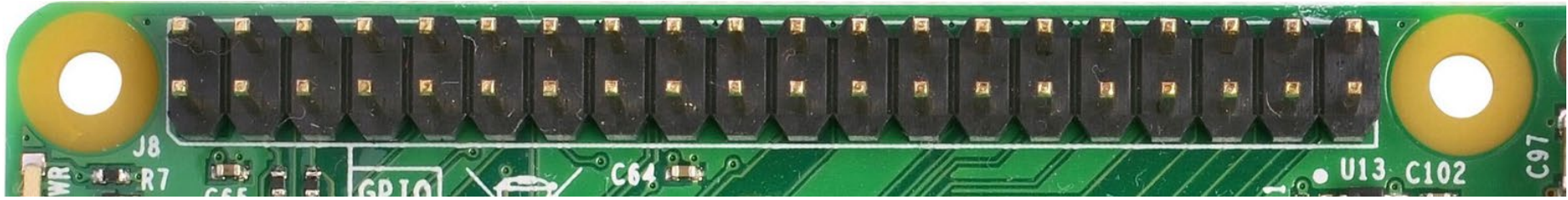
Red is Nice

Button

Make is a small one

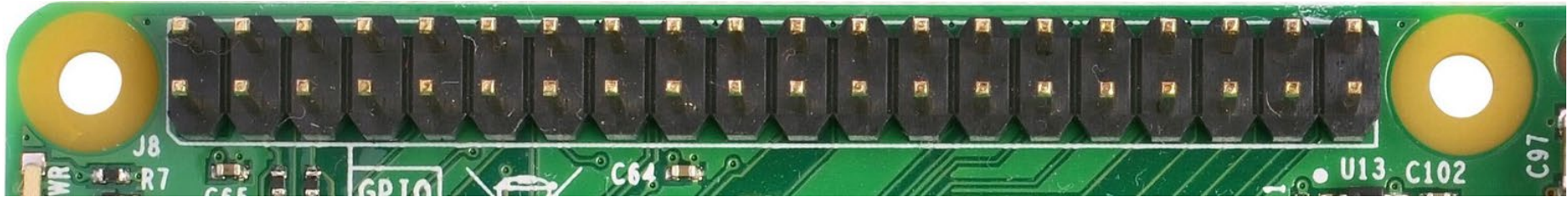
Attack the Hardware Lab!

GPIO



A pin whose behavior, including whether it is an input or output pin, is controllable by the user at run time

GPIO



pinout.xyz

The comprehensive GPIO Pinout guide for the Raspberry Pi

Get Ready for GPIO Types

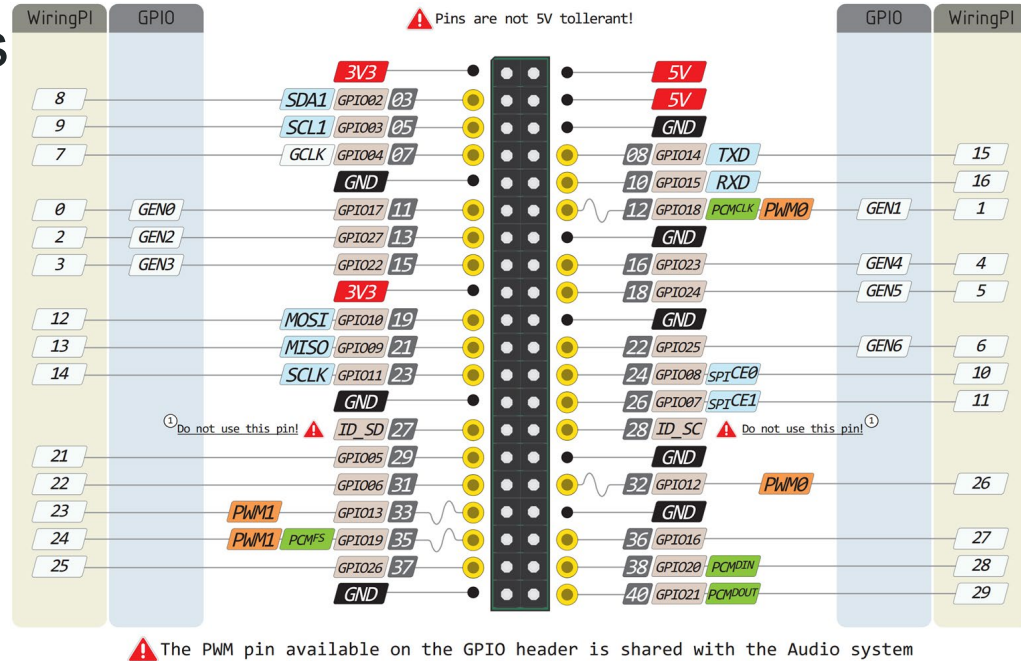
GPIO are your standard pins that simply be used to turn devices on and off. For example, a LED.

I2C (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This will typically take up 2 pins.

SPI (Serial Peripheral Interface) pins can be used to connect and talk to SPI devices. Pretty much the same as I2C but makes use of a different protocol.

UART (Universal asynchronous receiver/transmitter) are the serial pins used to communicate with other devices.

GND are the pins you use to ground your devices. It doesn't matter which pin you use as they are all connected to the same line.



Checkout pinout.xyz for an interactive pinout with greater details!

LED Blink

The Hello World of GPIO

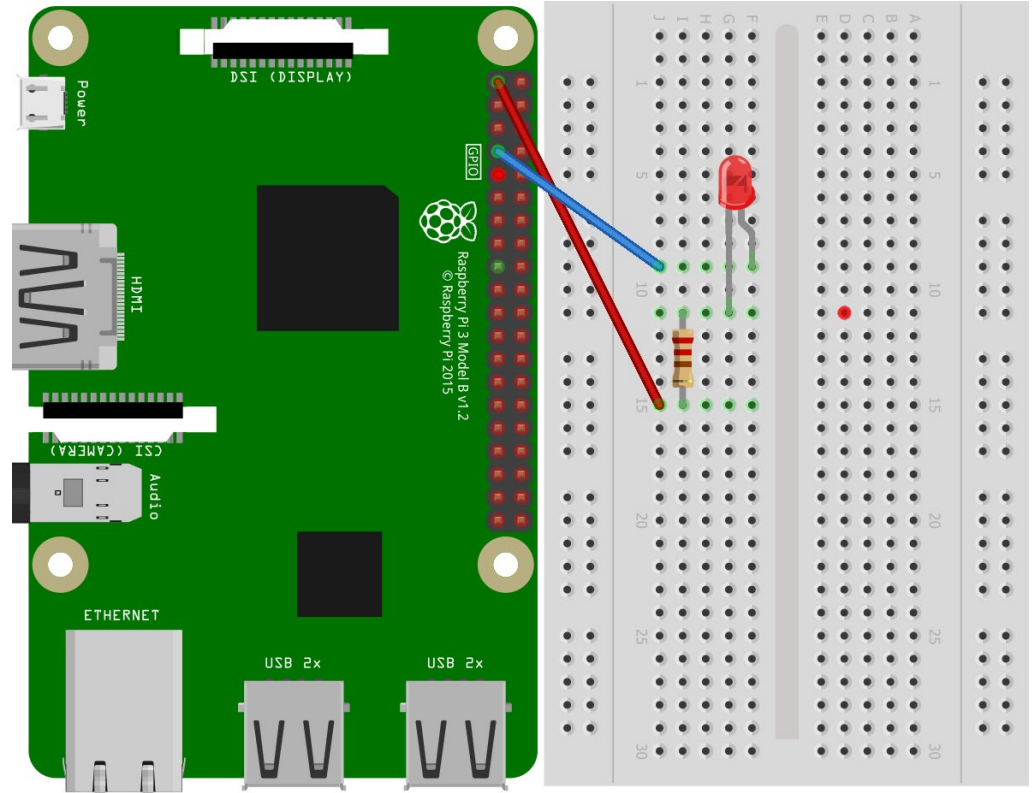
```
import RPi.GPIO as GPIO
import time

# To use Raspberry Pi board pin
# numbers
GPIO.setmode(GPIO.BOARD)

# Setup GPIO output channel
GPIO.setup(7, GPIO.OUT)

# Blink GPIO17 ad infinitum
while True:
    GPIO.output(7, GPIO.HIGH) # Off
    time.sleep(1)
    GPIO.output(7, GPIO.LOW) # On
    time.sleep(1)

GPIO.cleanup()
```



Hello Button

No Buttons about it!

```
# Start with the usual...
```

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
```

```
# PUD_UP because GND
GPIO.setup(7, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

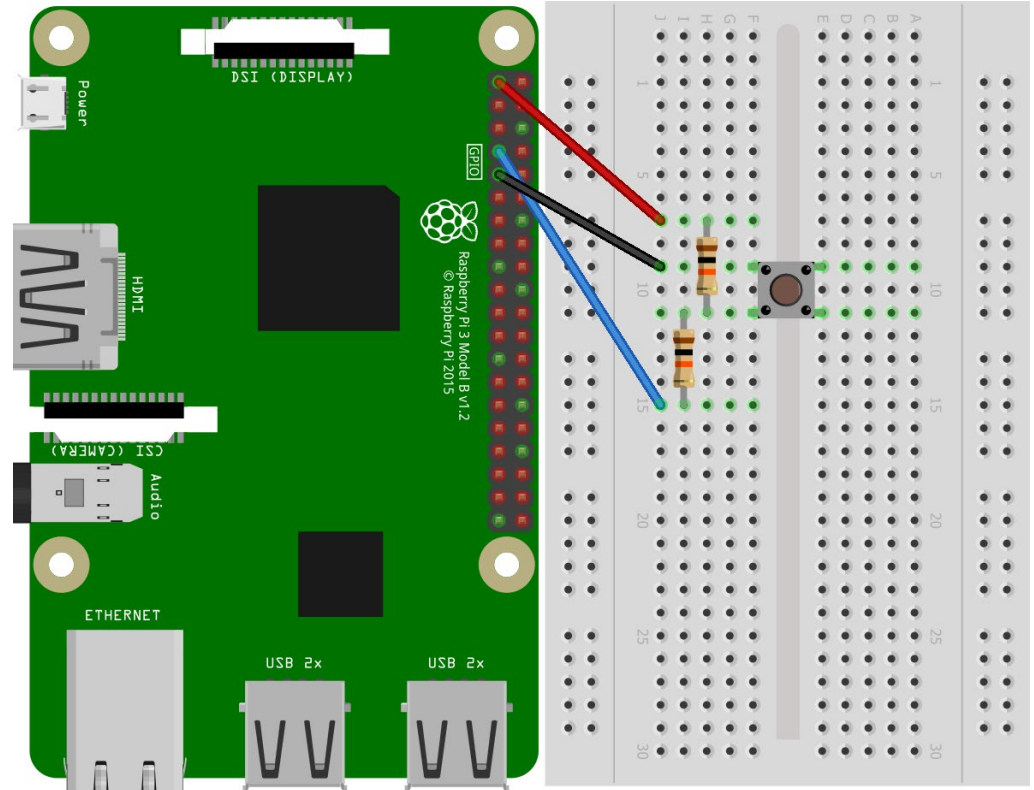
```
# One Example
```

```
def my_callback(channel):
    print "Hello World!"
```

```
GPIO.add_event_detect(7, GPIO.FALLING,
callback=my_callback)
```

```
while True: # Loop forever
    True
```

```
GPIO.cleanup() # clean up GPIO on normal exit
```



Hello Button v2

Buttons about it!

```
# Start with the usual...
```

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
```

```
# PUD_UP because GND
```

```
GPIO.setup(8, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
# Another Example
```

```
while True:
```

```
    try:
```

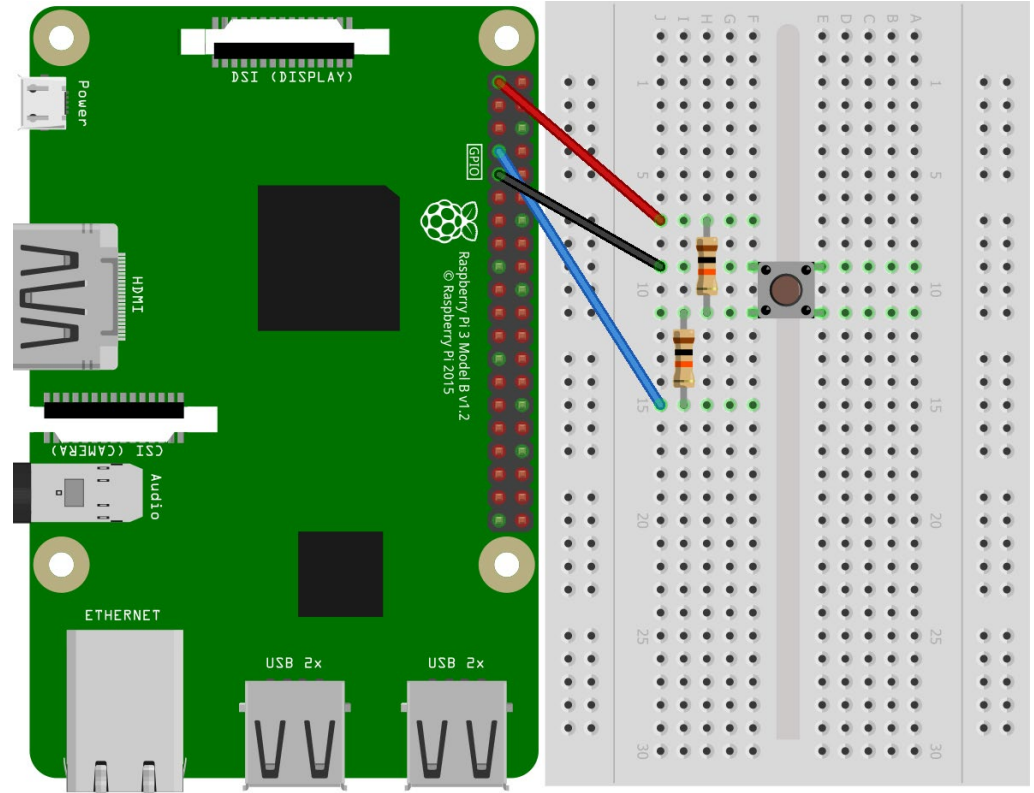
```
        GPIO.wait_for_edge(8, GPIO.FALLING) # Waiting
```

```
        print "Hello World!"
```

```
    except KeyboardInterrupt:
```

```
        GPIO.cleanup() # clean up GPIO on CTRL+C exit
```

```
GPIO.cleanup() # clean up GPIO on normal exit
```



Analog

The Raspberry Pi's GPIO pins are digital pins, so you can only set outputs to high or low, or read inputs as high or low

However, using an ADC chip (Analogue-to-Digital converter), you can read the value of analogue input devices such as potentiometers

MCP3008

PCF8591

Check the Hardware Lab

pinout.xyz

The comprehensive GPIO Pinout guide for the Raspberry Pi

