

PubNub®

Internet of Things 101: Building IoT
Prototype with Raspberry Pi

Feb 9 and 11, 2016 at Forward 4 Conf



Tomomi
(@girlie_mac)



Bhavana
(@bhavanallo)





#ForwardJS

While you are waiting...



Connect Wi-Fi

Wi-Fi Network: NewCircle Student

Password: opensource99

* This requires you to sign in from a captive portal



[https://github.com/pubnub/
workshop-raspberrypi](https://github.com/pubnub/workshop-raspberrypi)

You should have:

- Raspberry Pi 2
- Micro SD card (preloaded w/ Raspbian)
- Mini Wi-Fi adapter
- Micro USB power supply
- HDMI Cable
- Wires
- Breadboard
- LED
- Resistors
- PIR sensor
- DHT22 sensor



What You Will Learn Today



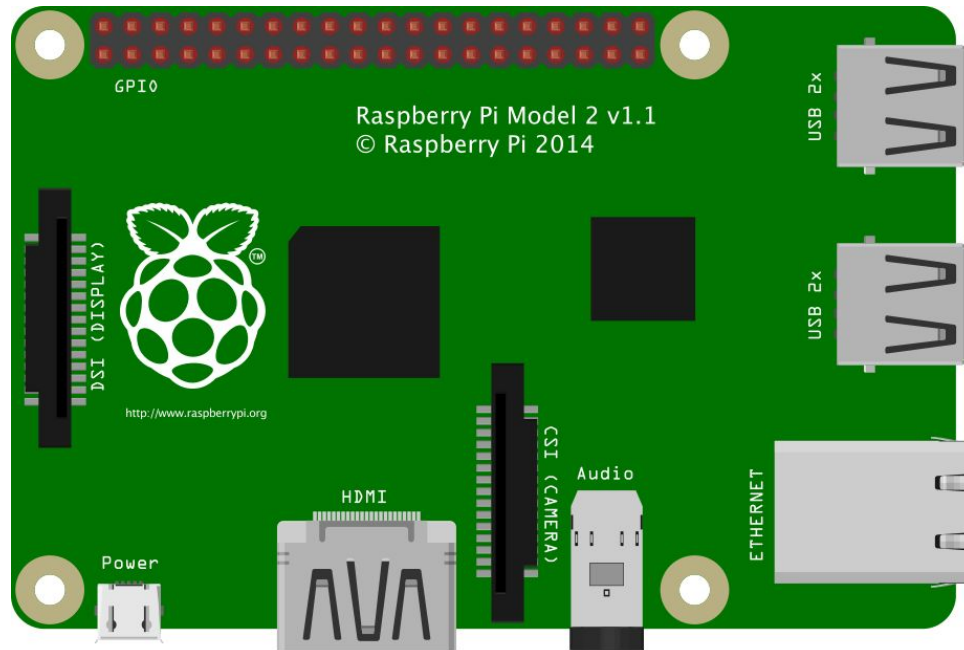
1. How to start up Raspberry Pi & start Raspbian OS
2. How to connect the Pi remotely from your laptop
3. How to send & receive data with PubNub using Python
4. How to wire a LED & resistor to Pi using breadboard
5. How to program Pi to blink the LED
6. The First IoT : Remote-controlled LED from web interface
7. Projects: Using sensors (Work on your own)

1. Powering the Raspberry Pi

Setting up Your Pi



1. SD CARD



2. WI-FI ADAPTER

3. TO KEYBOARD

4. TO MOUSE

6. USB TO POWER SOURCE

5. TO MONITOR

Starting Raspbian



Username: pi

Password: raspberry



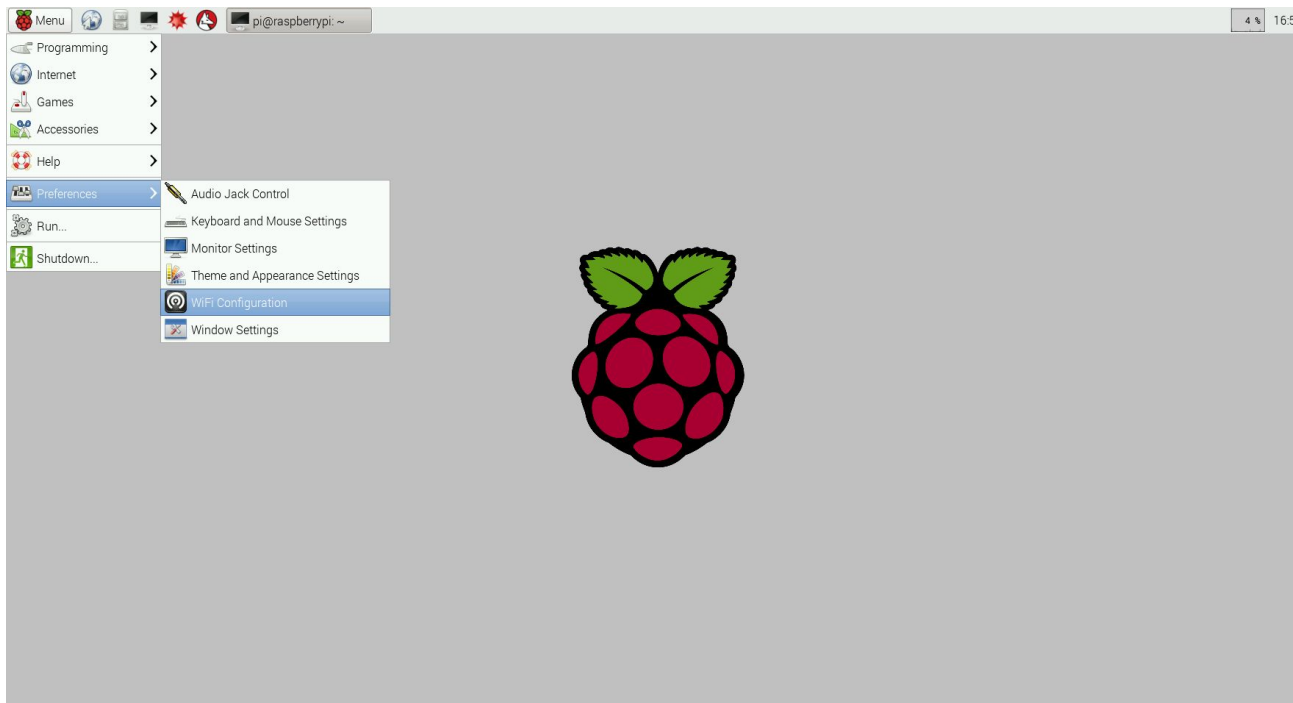
```
pi@raspberrypi ~$ startx
```

(Don't worry about the error dialog. Just dismiss it!)

Wi-Fi Configuration



Menu > Preference > WiFi Configuration



Wi-Fi Network: **NewCircle Student** Password (PSK): **opensource99**

2. Remote Connect the Pi



<https://github.com/pubnub/workshop-raspberrypi/blob/master/remote-vnc.md>

Remote Connect Pi



Getting your Pi's IP address

```
pi@raspberrypi ~$ hostname -I
```

★ You'll need the IP address when you connect the Pi from your computer!!!

Remote Connect Pi



You can choose from the followings:

- SSH to your Pi from terminal, and keep working on the terminal
- SSH with a client, e.g. [Cyberduck](#), and use your usual IDE
- Use VNC (Virtual Network Computing), and work on the virtual GUI

SSH into your Rasp Pi



SSH to Pi from your laptop

(Terminal on Mac/Linux, PuTTY on Windows):

me@MyMac ~\$ **ssh** *pi*@10 96.70.1

Your Pi's username

A purple arrow points from the text "Your Pi's username" to the word "pi" in the command. An orange arrow points from the text "Use your Pi's IP!" to the IP address "10 96.70.1" in the command.

Use your Pi's IP!

If SSH-ing fails, try:
\$ *sudo raspi-config*
on your Pi

Remote Access w/ VNC



Remote-access to Raspberry Pi's graphical interface.

1 . On your laptop:

- Install VNC Viewer (Client)

<http://www.realvnc.com/download/viewer/>

2. On your Pi (Either by SSH, or directly):

- Install VNC server

Remote Access w/ VNC



Install **Tight VNC Server**

```
pi@raspberrypi
```

```
~$ sudo apt-get install tightvncserver
```

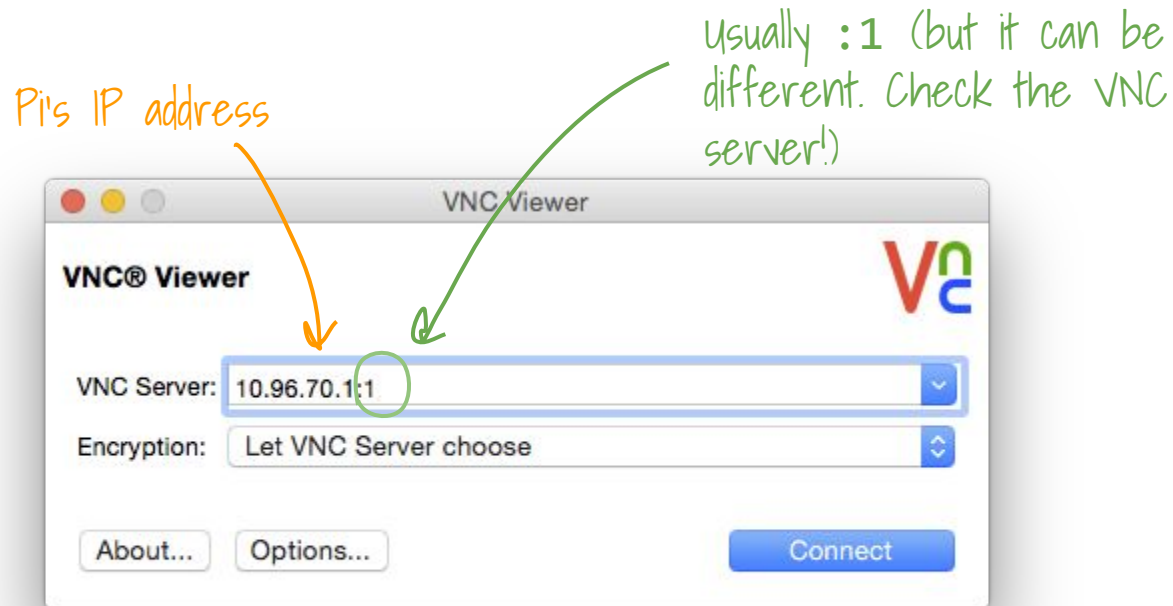
Run the server

```
pi@raspberrypi ~$ tightvncserver
```

Remote Access w/ VNC



Run the client



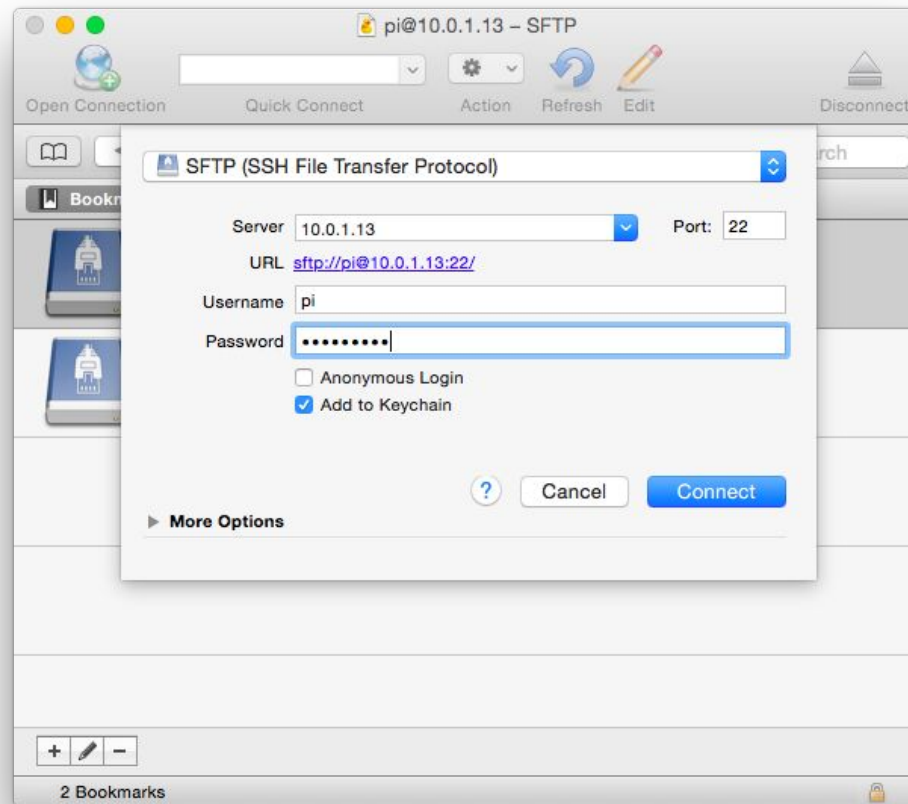
SSH w/ Cyberduck



- No virtual GUI, but faster than using VNC
- You can use your fave IDE to edit files
- Mac users can keep using Mac keyboard shortcuts

Download the client at: <https://cyberduck.io/>

SSH w/ Cyberduck





3. Get Started w/ PubNub Python SDK

Update System First



Update the System's package list

```
~$ sudo apt-get update
```

Upgrade the installed packages to the latest versions

```
~$ sudo apt-get upgrade
```

Get Started w/ Python



Install python and pip

```
~$ sudo apt-get install python-dev
```

```
~$ sudo apt-get install python-pip
```


Get Started w/ PubNub



Install pubnub libs

```
~$ sudo pip install pubnub
```

Hello World w/ PubNub



<https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/helloworld>

Hello World w/ PubNub



Import & init (hello.py)

```
import sys
```

```
from pubnub import Pubnub
```

```
pubnub = Pubnub(publish_key='pub-c-123...',  
subscribe_key='sub-c-456...')
```

Hello World w/ PubNub



Publish (Sending data)

```
channel = 'hello-pi'
data = { 'username': 'SpongeBob',
        'message': 'Hello world from Pi!' }
```

Use your own name & message!

```
def callback(m):
    print(m)
```

```
pubnub.publish(channel, data, callback=callback,
               error=callback)
```

Hello World w/ PubNub



Run your program

```
~$ sudo python hello.py
```

Hello World w/ PubNub



Subscribing data you are publishing



<http://pubnub.github.io/workshop-raspberrypi/web/hello.html>

3. Using the Debug Console

Debug Console

A screenshot of the PubNub Developer Console interface. The browser address bar shows 'www.pubnub.com/console/'. The page has a navigation bar with links: COMPANY, CUSTOMERS, PRESS, CONTACT, BLOG, and a search bar. Below the navigation bar are links: TUTORIALS, PRICING, DEVELOPERS (highlighted), HOW IT WORKS, and USE CASES. The main heading is 'View and Debug all PubNub Events'. The interface is divided into several sections. At the top left, there's a 'SUBSCRIBE' button with a green checkmark and a label '4. subscribe'. Below it, there's a 'channel' input field with 'hello-pi' and a green checkmark, labeled '1. channel'. To the right, there's a 'publish key' input field with 'demo' and a green checkmark, labeled '2. pub key'. Below that, there's a 'subscribe key' input field with 'demo' and a green checkmark, labeled '3. sub key'. At the bottom left, there's a 'message' input field with '{"text": "hey"}' and a green checkmark, labeled '5. monitor data'. The bottom right section shows a 'presence' and 'history' log. The 'presence' log shows a message: 'Tue Apr 14 2015 16:23:51:523 : <hello> {"action": "join", "timestamp": "1429053831", "uid": "fbaf3508-2429-4aaa-a8ff-6c09c15c19c", "occupancy": 2}'. The 'history' log shows a message: 'Tue Apr 14 2015 16:29:52:678 : <hello> {"username": "Your name", "message": "Hello World from Pi!!!"}'.

<http://pubnub.com/console/>

1. channel: **hello-pi**
2. pub key: **demo**
3. sub key: **demo**

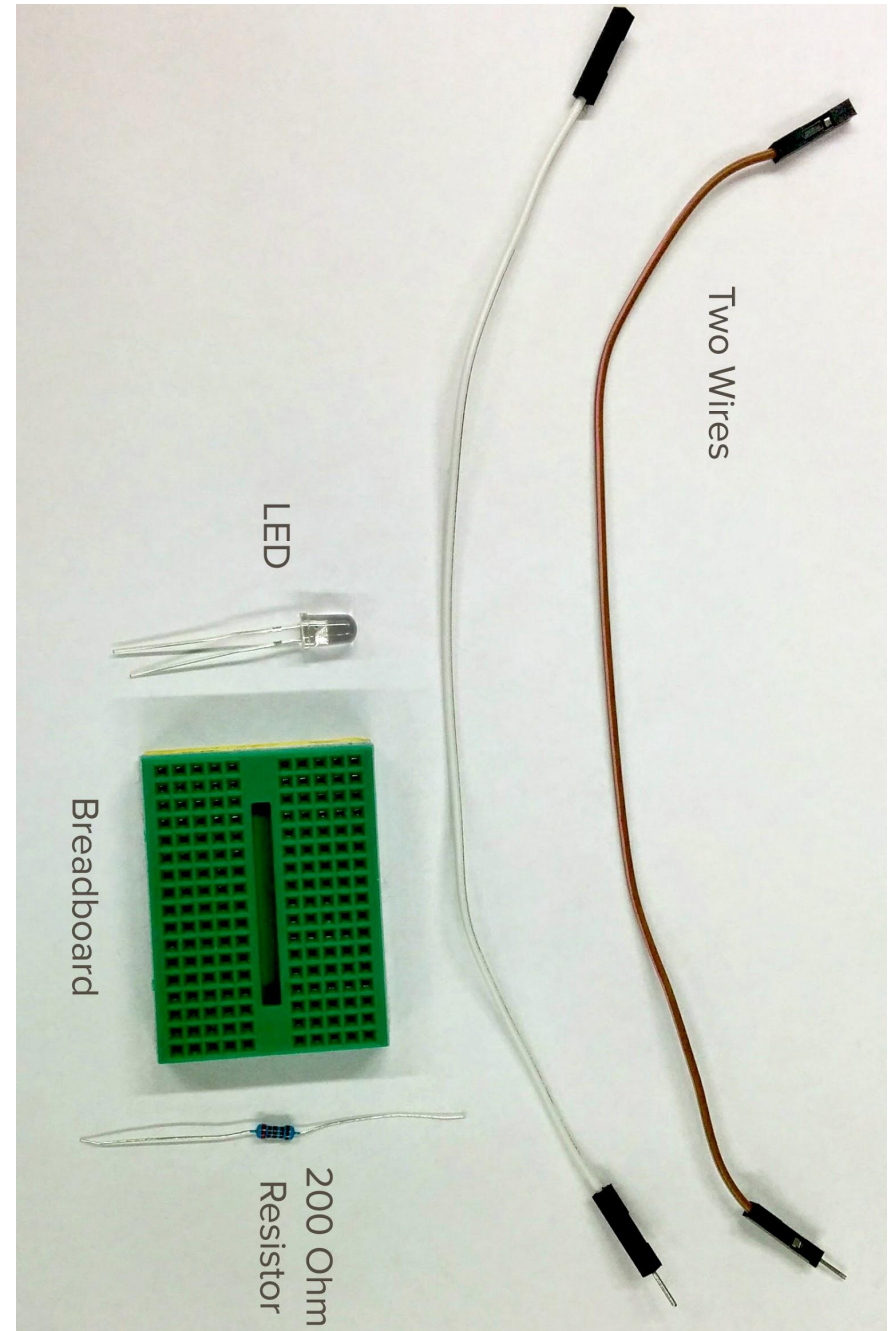
4. Blinking LED



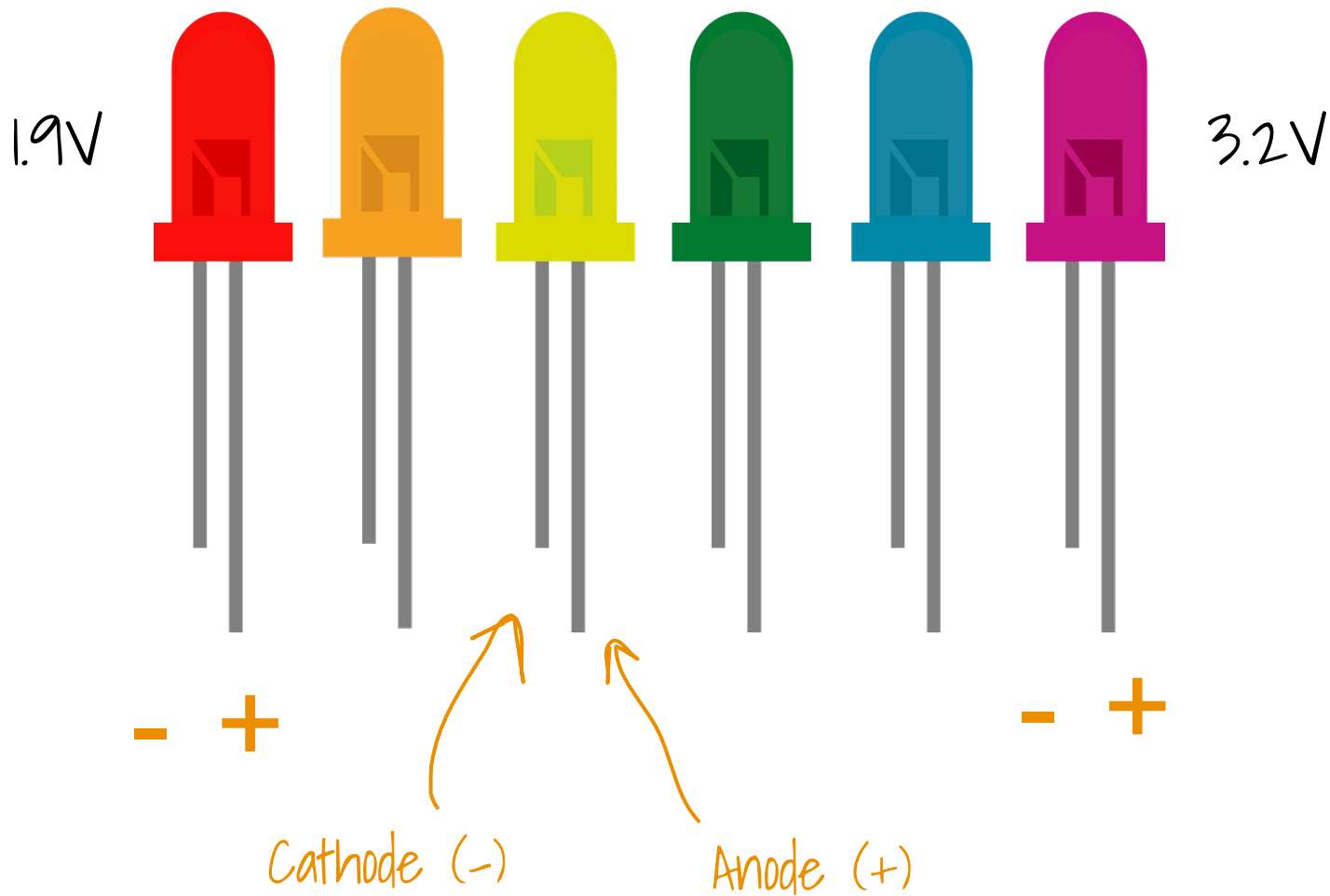
<https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/led>

Blinking LED

- Raspberry Pi 2
- 1 LED (1.9 - 3.2V)
- 1 Resistor (200Ω)
- 1 Breadboard
- 2 M-to-F jumper wires, 2 colors



OMG Physics!



OMG Physics!

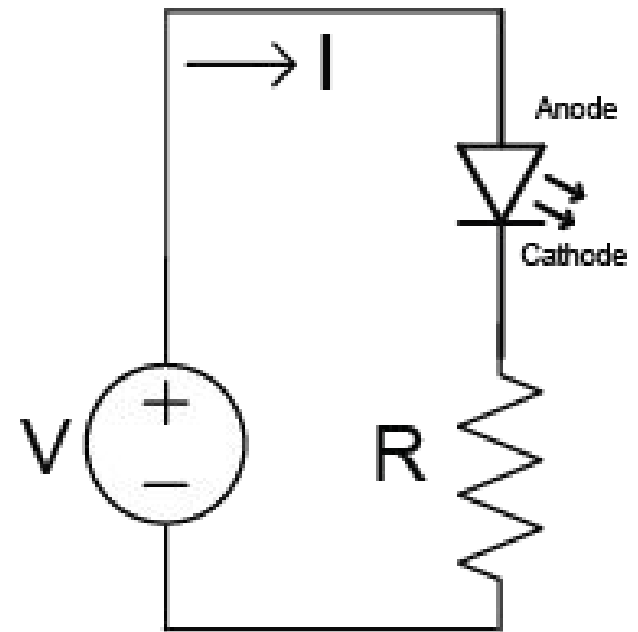
source voltage (V)

forward voltage (V)
(LED voltage drop)

$$R = \frac{V_s - V_f}{I}$$

resistance (Ω)

current thru
the LED (A)



OMG Physics!

A handwritten calculation for the resistor value R . The formula is $R = \frac{3.3\text{v} - 1.9\text{v}}{0.02\text{ A}} = 70\ \Omega$. Annotations include: 'source voltage (V)' pointing to 3.3v, 'forward voltage (V) (Red LED voltage drop)' pointing to 1.9v, 'resistance (Ω)' pointing to R , and 'current thru the LED (A)' pointing to 0.02 A. The result 70 Ω is highlighted in green.

source voltage (V)

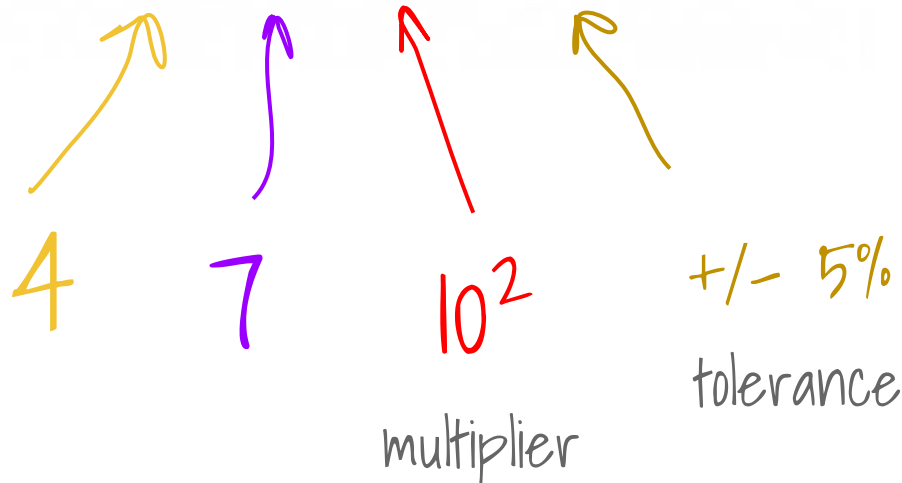
forward voltage (V)
(Red LED voltage drop)

$$R = \frac{3.3\text{v} - 1.9\text{v}}{0.02\text{ A}} = 70\ \Omega$$

resistance (Ω)

current thru
the LED (A)

4-band Resistor Color Code



$$47 \times 100 = 4.7 \text{ k Ohms}$$

5-band Resistor Color Code

PubNub®



2 0 0 10^0 $\pm 1\%$
multiplier tolerance
 $200 \times 1 =$
200 Ohms

Learn more at: <https://learn.adafruit.com/multimeters/resistance>

Breadboard

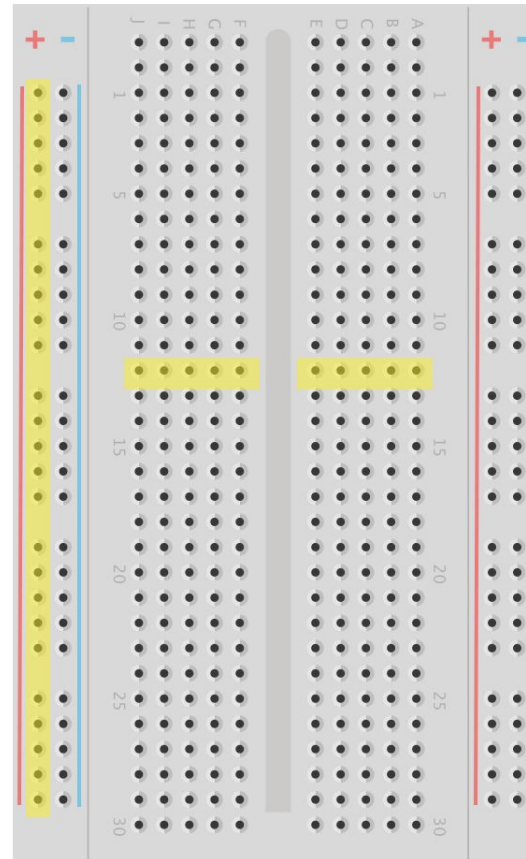
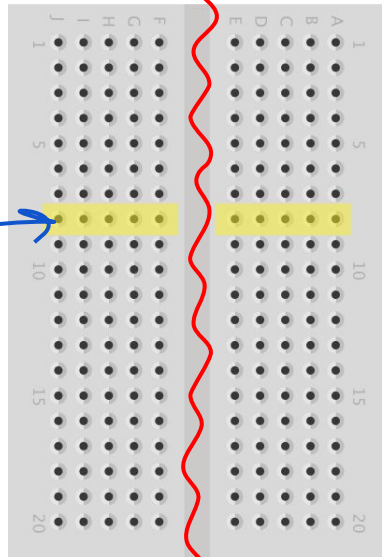
An electronics breadboard is a fundamental tool to build circuits. It is solderless, and great tool for prototyping.

We are using this kind today!

Connected!
conductive metal
strips goes
horizontally

Mini

not connected !

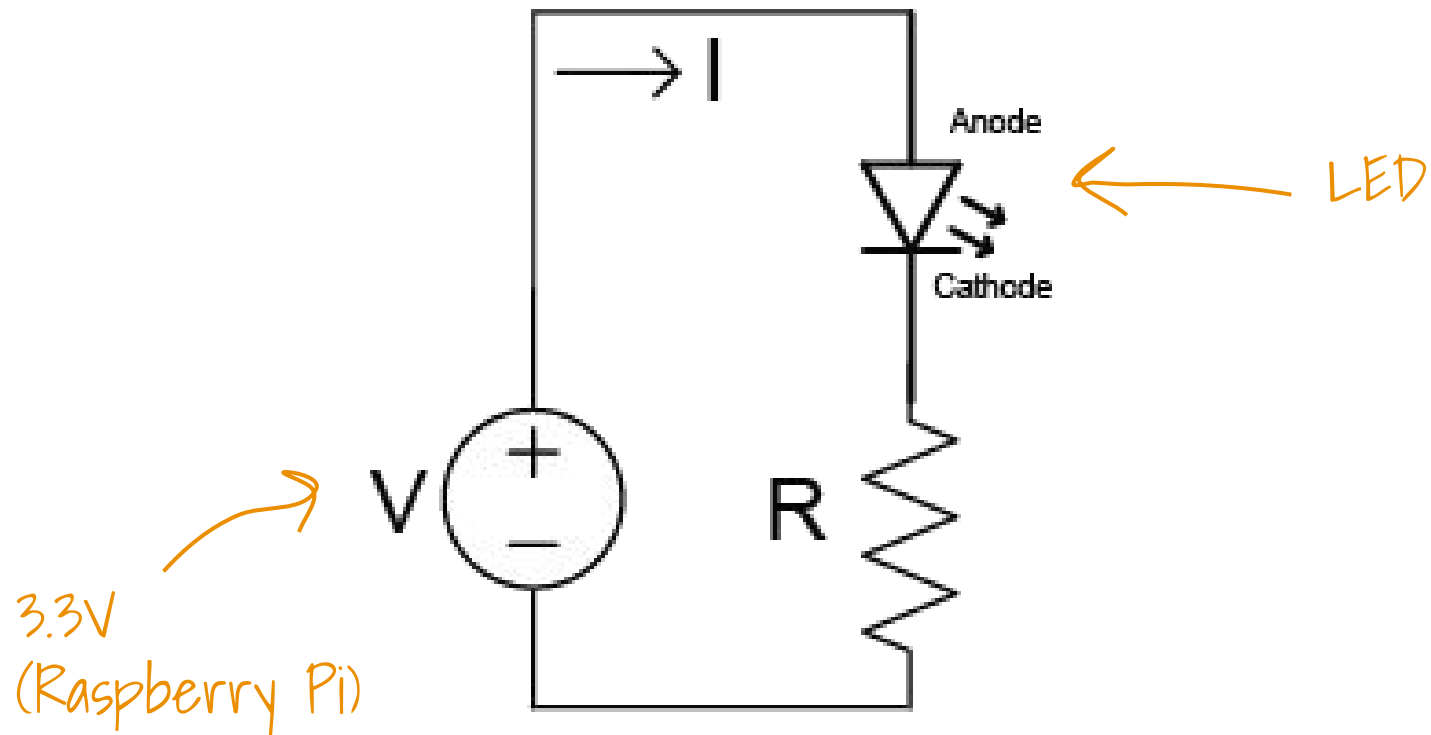


You may find this
type of breadboard
when googling
circuits. They have
power rails that
goes vertical!

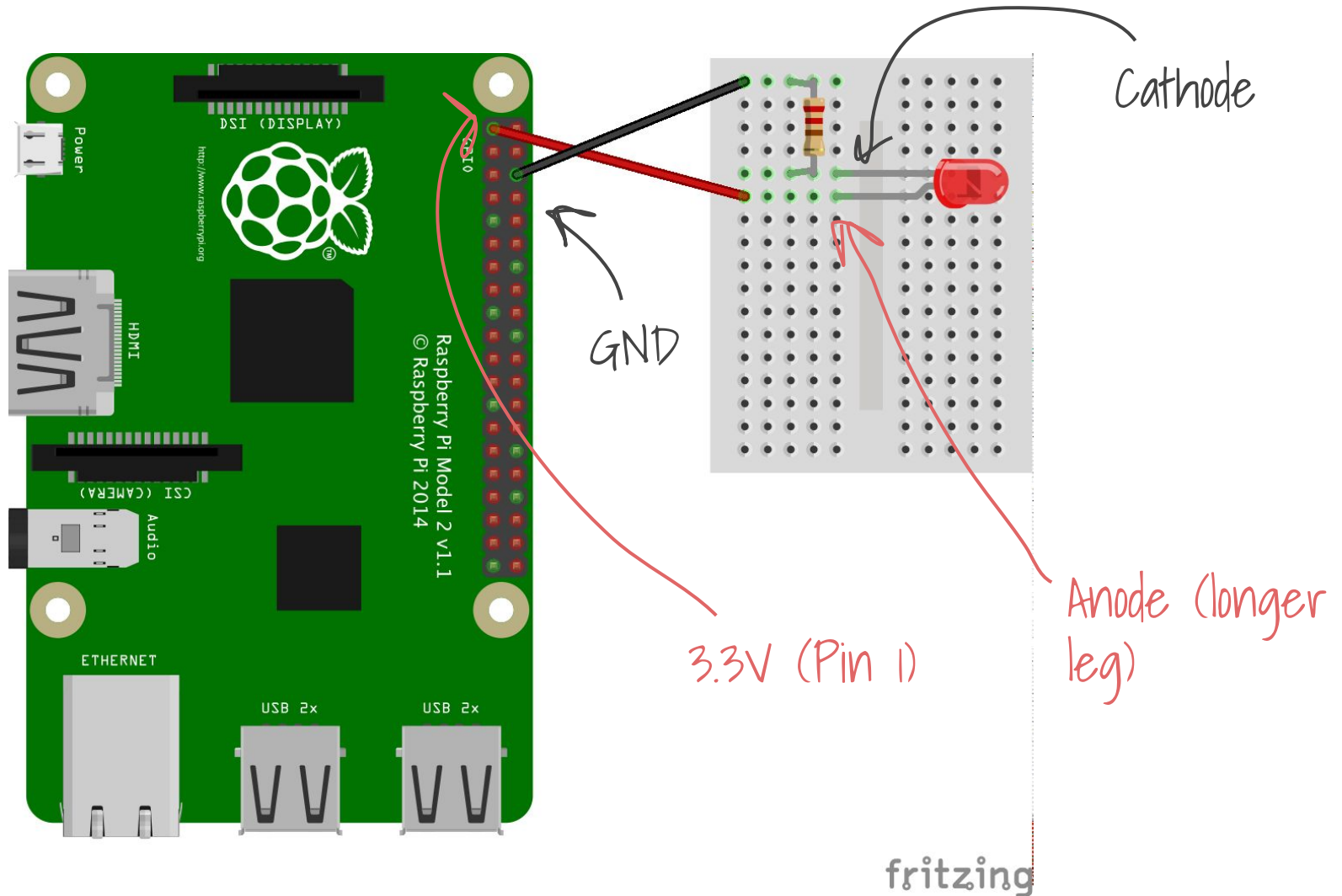


400-pin

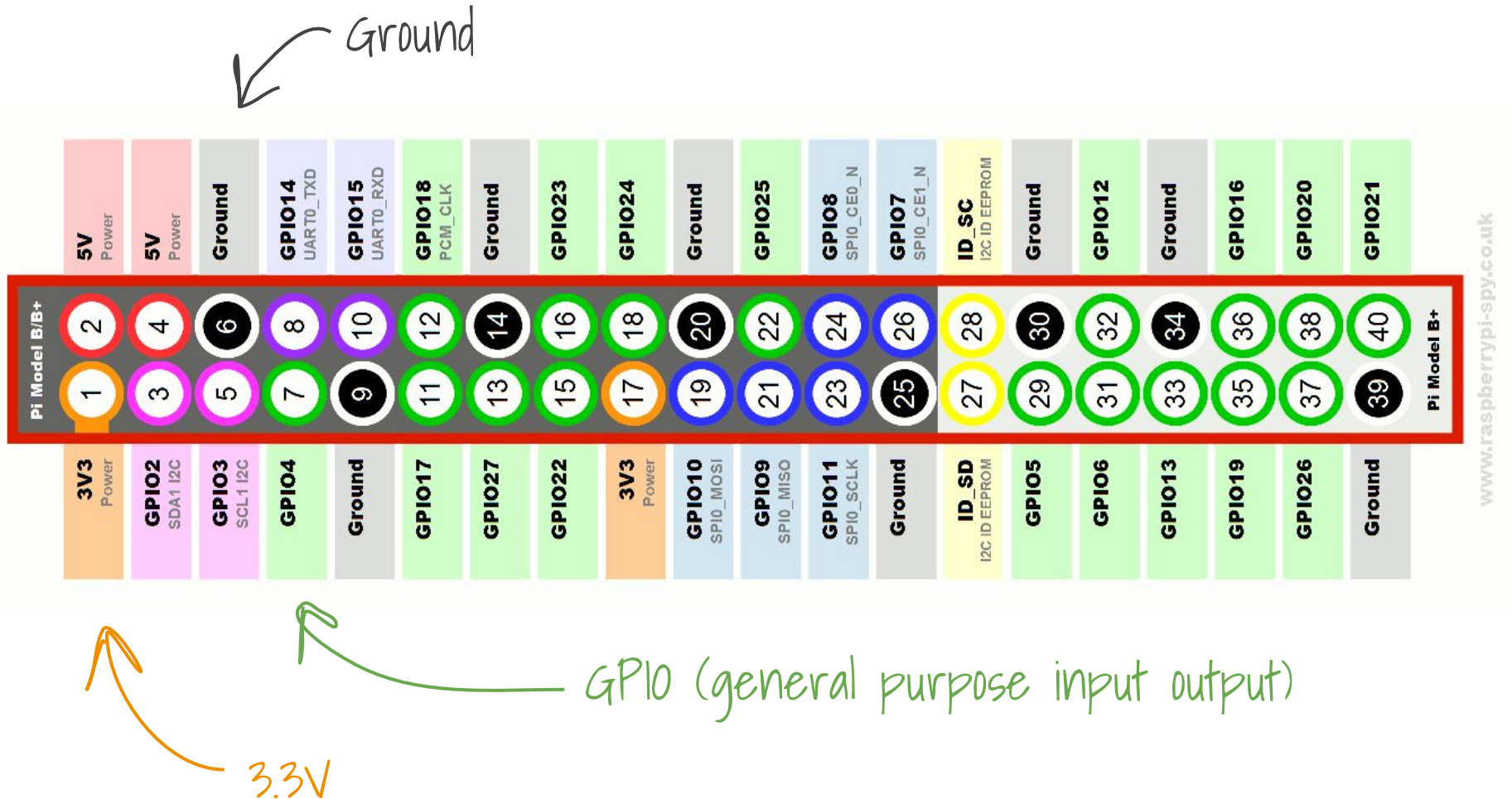
Turning LED on



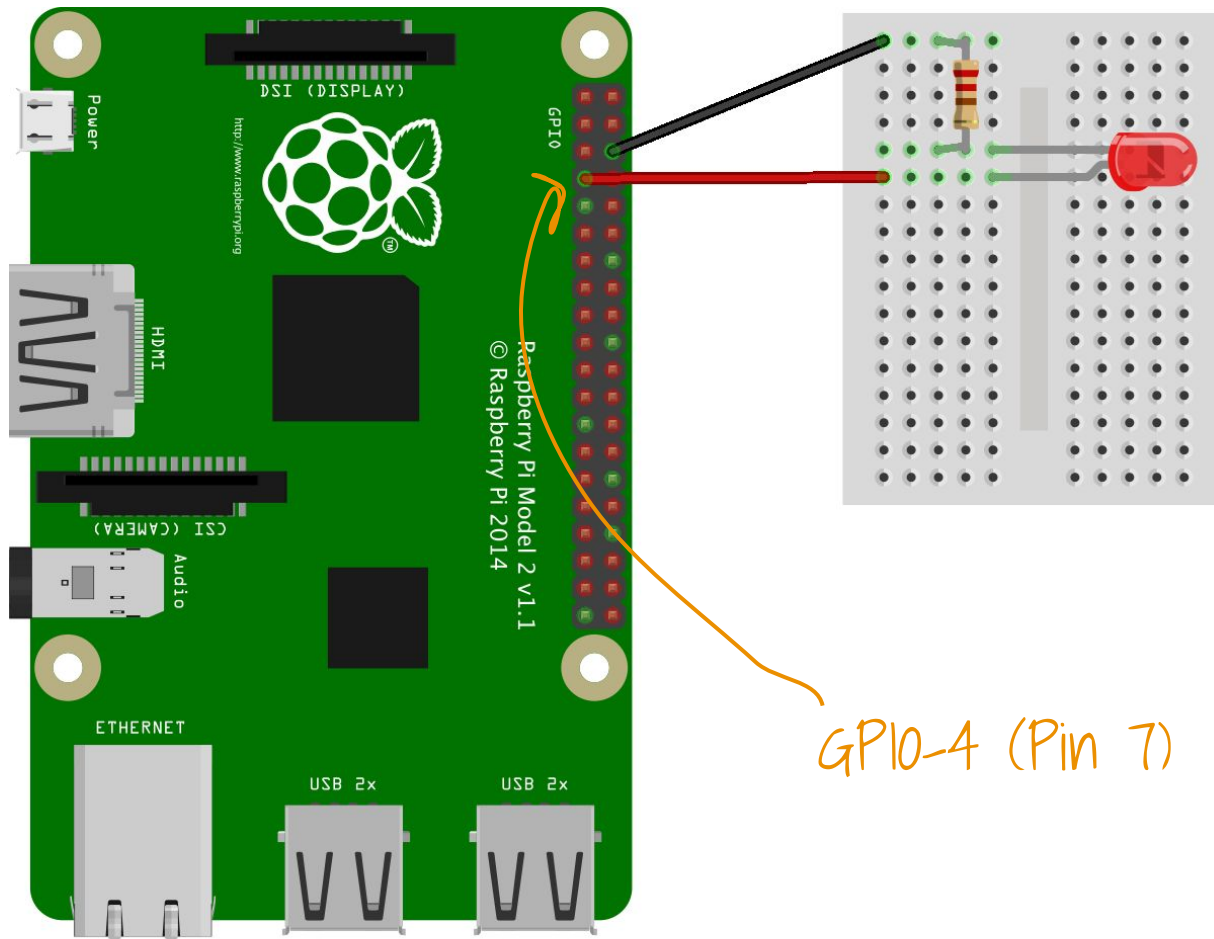
Turning LED on



Raspberry Pi 2 Pins



Programming LED



GPIO-4 (Pin 7)

Programming LED



```
import RPi.GPIO as GPIO
import time
```

import RPi.GPIO libs

```
GPIO.setmode(GPIO.BCM)
```

set pin type. use BCM, not pin number

```
LED = 4
```

GPIO 4 pin (Pin 7)

```
GPIO.setup(LED, GPIO.OUT)
```

set LED pin as output

```
while True:
```

```
    GPIO.output(LED, True)
```

```
    time.sleep(0.5)
```

```
    GPIO.output(LED, False)
```

```
    time.sleep(0.5)
```

toggle light pin signal to low/high to make it blink.

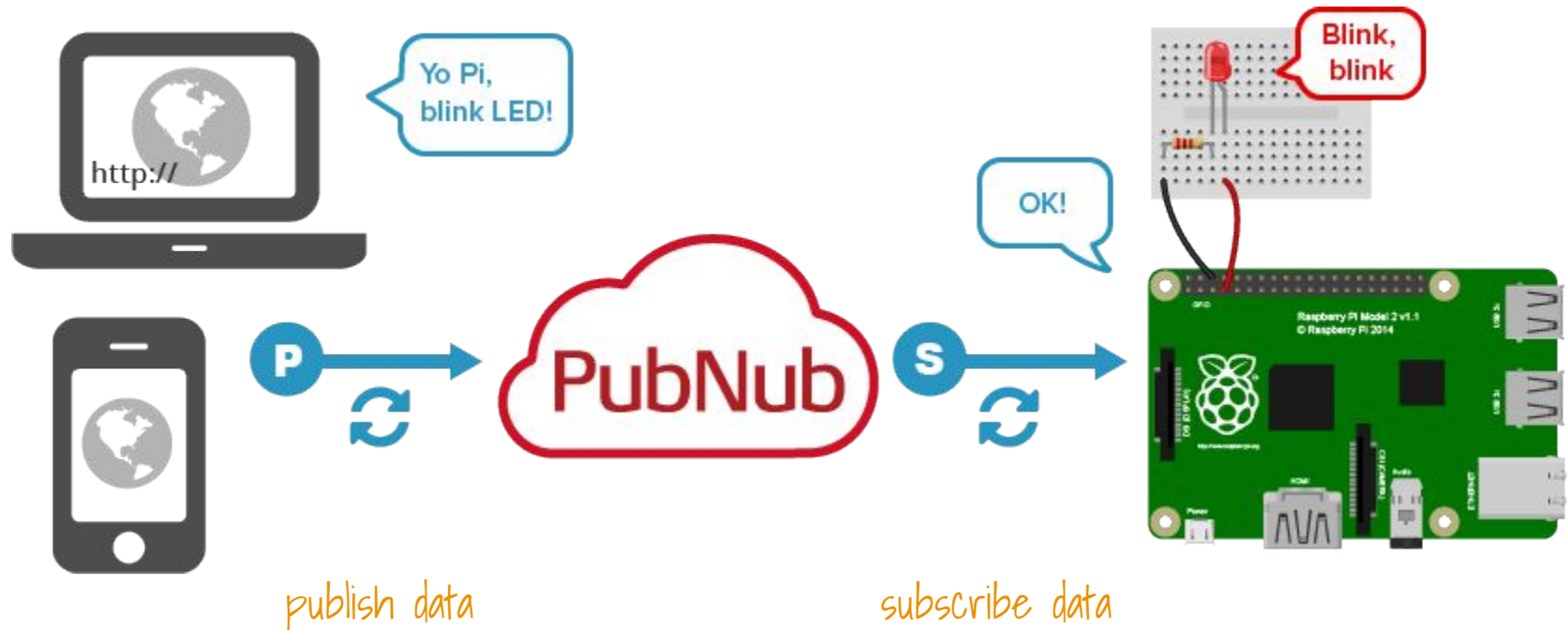
5. Introduction to IoT

Making it IoT: Remote-Controlled LED



[https://github.com/pubnub/
workshop-raspberrypi/tree/master/
projects-python/remote-led](https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/remote-led)

Making it IoT: Remote-Controlled LED



Making it IoT: Remote-Controlled LED



Subscribing data from a web client

```
pubnub = Pubnub(publish_key='demo', subscribe_key='demo')  
channel = 'disco'
```

```
def _callback(m, channel):  
    if m['led'] == 1:  
        for i in range(6):  
            GPIO.output(LED_PIN, True)  
            time.sleep(0.5)  
            GPIO.output(LED_PIN, False)  
            time.sleep(0.5)
```

When the button is
clicked on browser, it
publishes data, {'led': 1}



```
button.addEventListener  
('click', publish);
```

```
pubnub.subscribe(channels=channel, callback=_callback, error=_error)
```

Making it IoT: Remote-Controlled LED



Disco!

<http://pubnub.github.io/workshop-raspberrypi/web/disco.html>

IoT & PubNub Case Study: Insteon



<http://www.insteon.com>

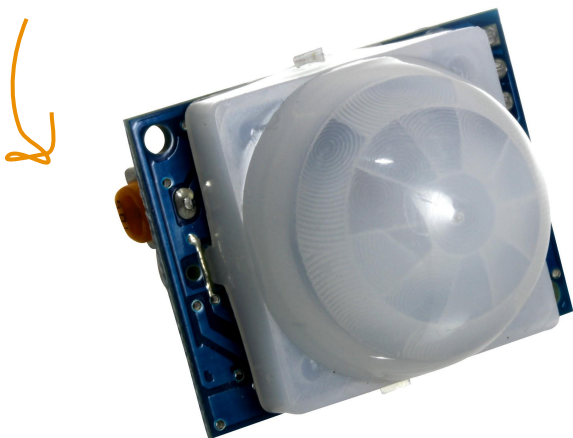


6. Go conquer IoT

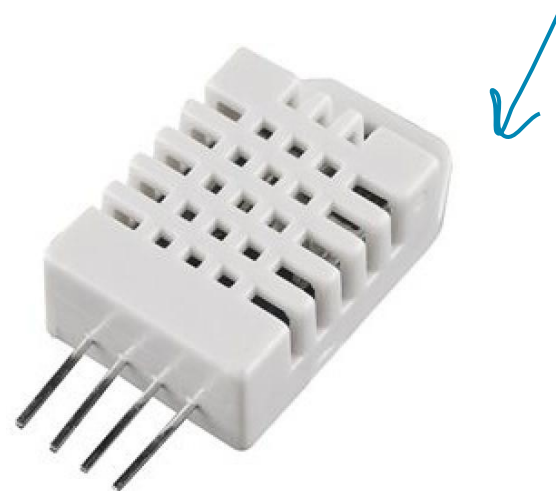
Projects

- Pyroelectric IR Motion sensor
- Combinations of sensors and LED
- DHT22 Temperature & Humidity sensor

PIR sensor



DHT22 sensor



PIR Motion Sensor



It detects motions by measuring changes in IR radiation when an object moves around it.

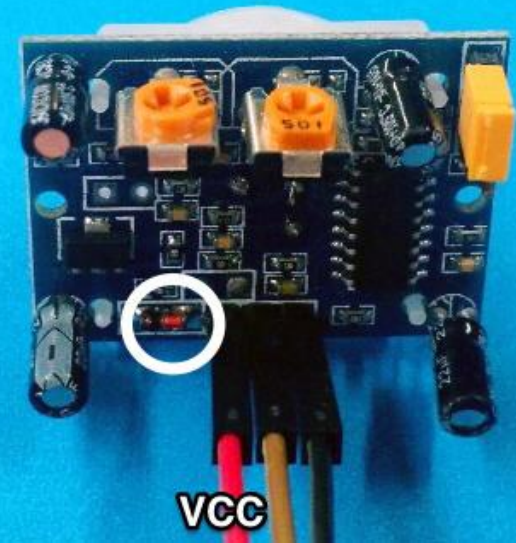
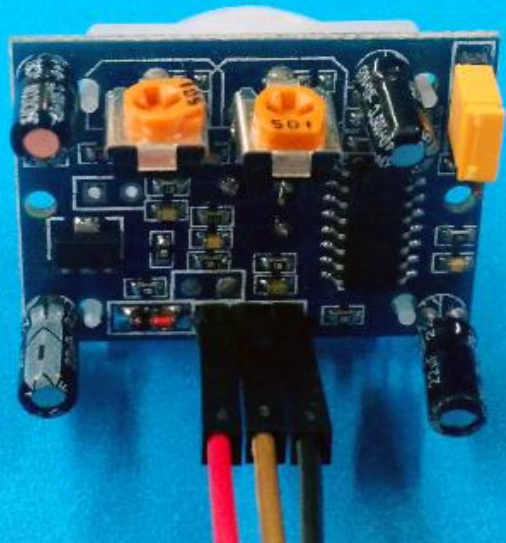


<https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/motion-sensor>

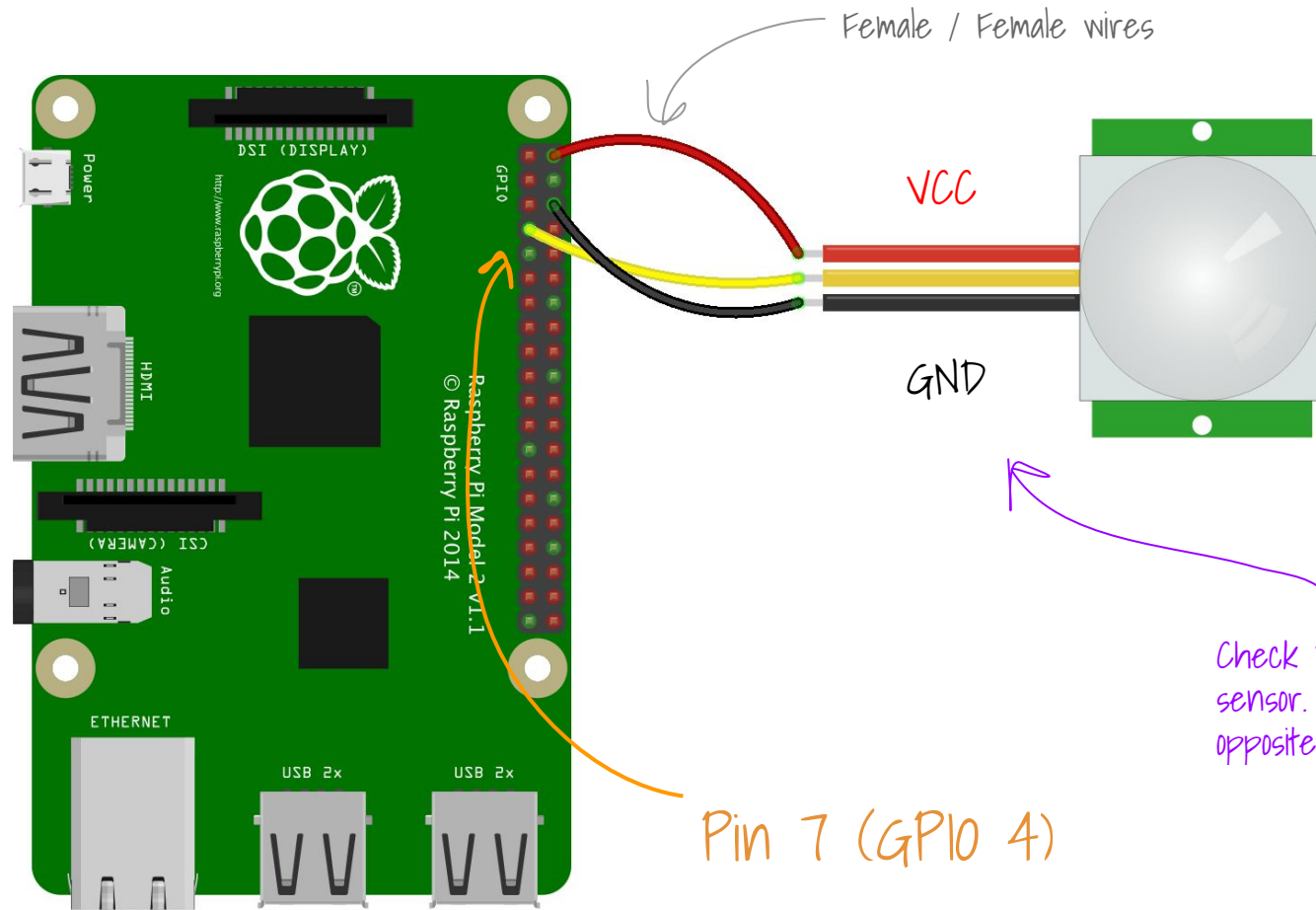


<http://pubnub.github.io/workshop-raspberrypi/web/motion.html>

PIR Motion Sensor



PIR Motion Sensor



PIR Motion Sensor w/ LED



Combination of the PIR motion sensor with a LED as a visual indicator

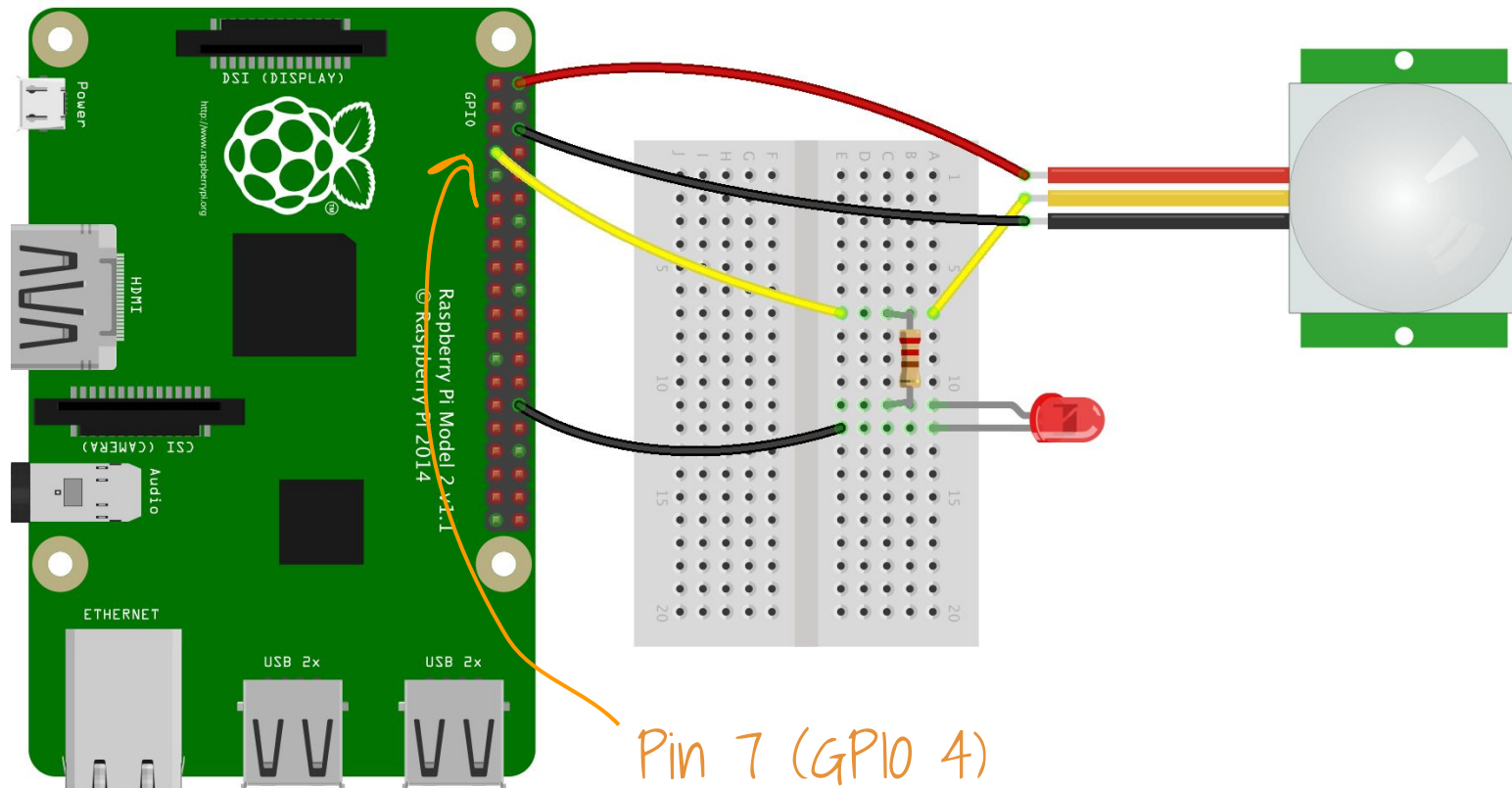


<https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/motion-led>



<http://pubnub.github.io/workshop-raspberrypi/web/motion.html>

PIR Motion Sensor w/ LED



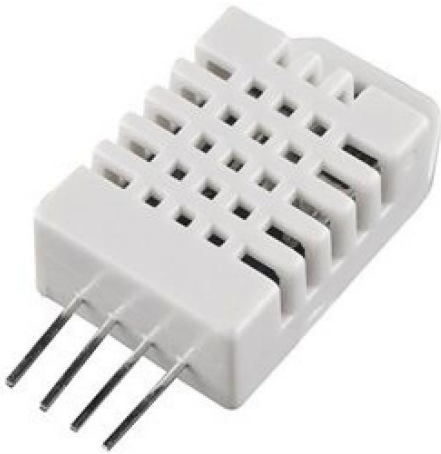
fritzing

Note: The circuit change only. The code remains the same.

Data Visualization with Temperature Sensor



It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin.

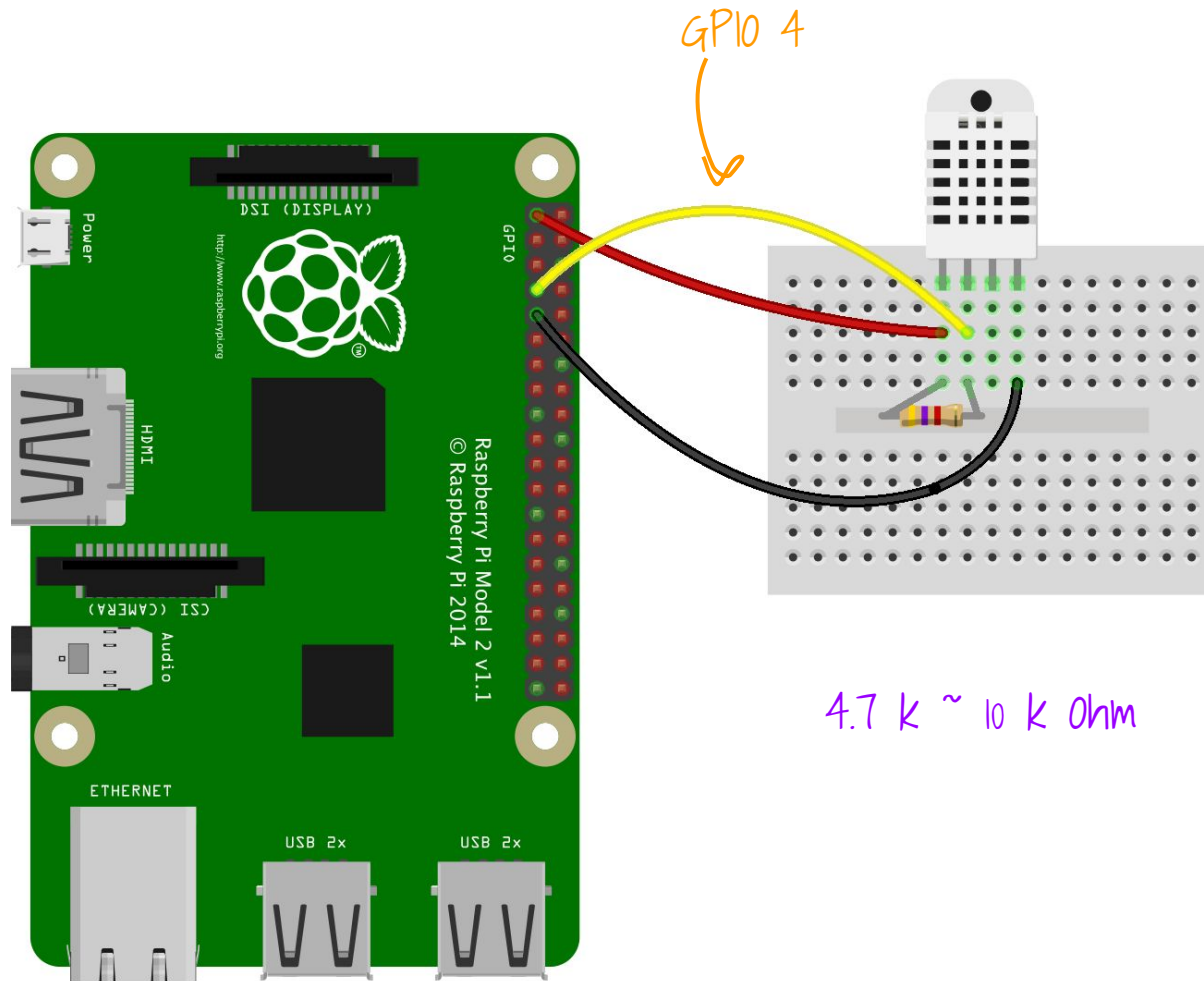


<https://github.com/pubnub/workshop-raspberrypi/tree/master/projects-python/dht22>



<http://pubnub.github.io/workshop-raspberrypi/web/temperature.html>

DHT22 Sensor



DHT22 Sensor



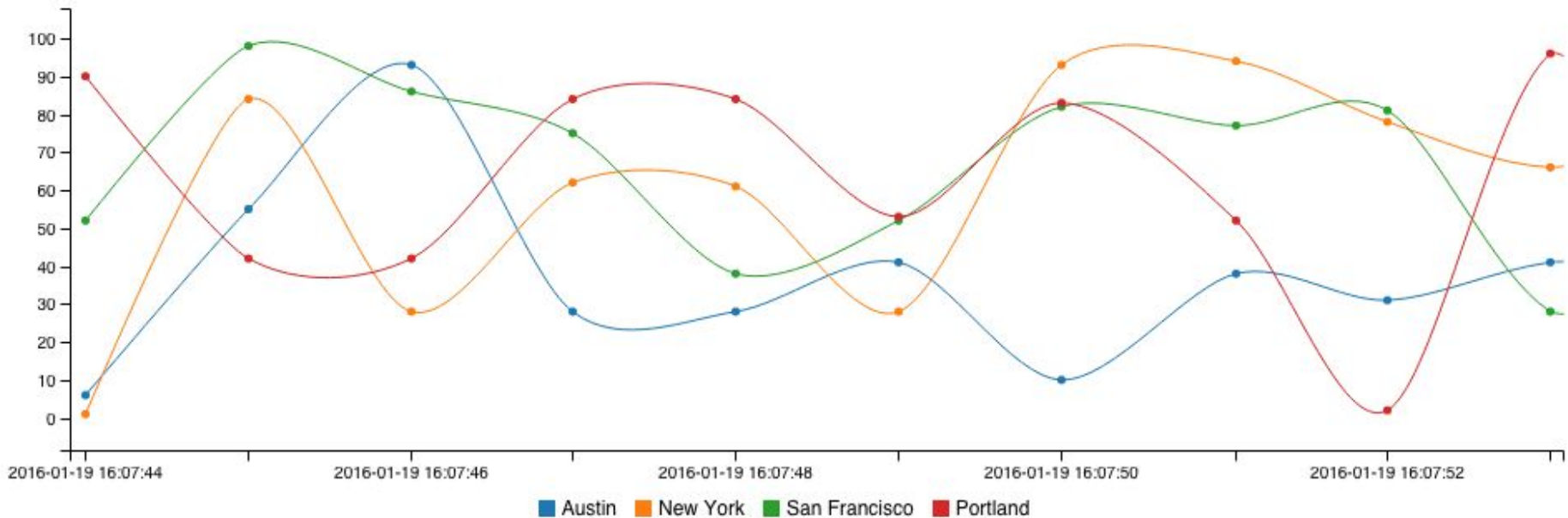
Download & Install Adafruit DHT library:

```
~$ git clone  
https://github.com/adafruit/Adafruit_Python  
_DHT.git
```

```
~$ cd Adafruit_Python_DHT
```

```
~$ sudo python setup.py install
```

Realtime Data Graphs & Charts



<https://github.com/pubnub/eon-chart>

You've got some extra time for one more project?

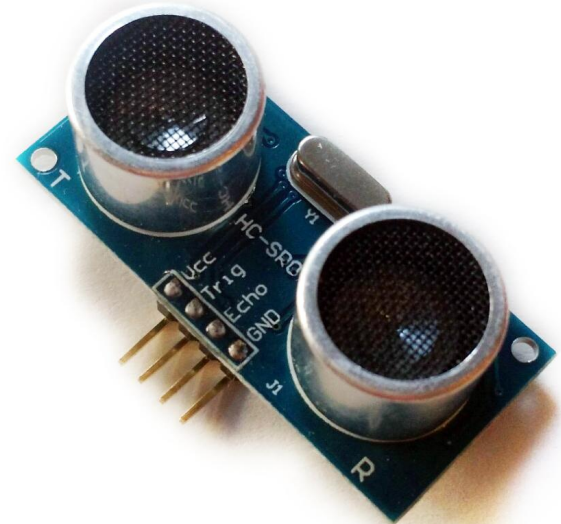
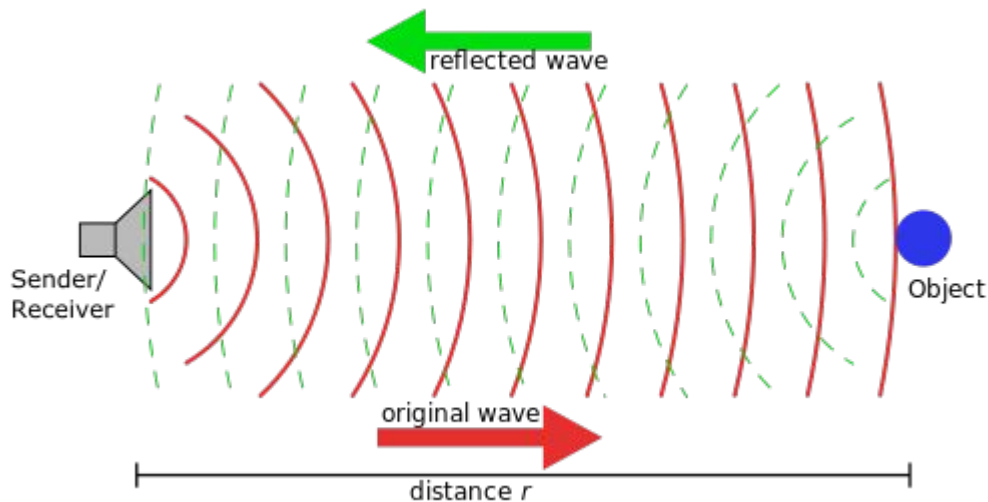
Come to front to pick up:

- HC-SR04 Ultrasonic sensor
- resistors (1k Ohm & 2.2k Ohm)
- extra wires

Ultrasonic RangeFinder



The HC-SR04 ultrasonic sensor uses sonar signals to determine distance to an object



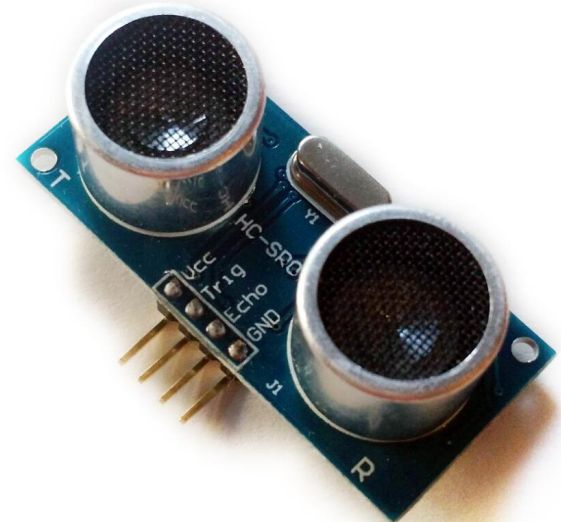
Ultrasonic RangeFinder



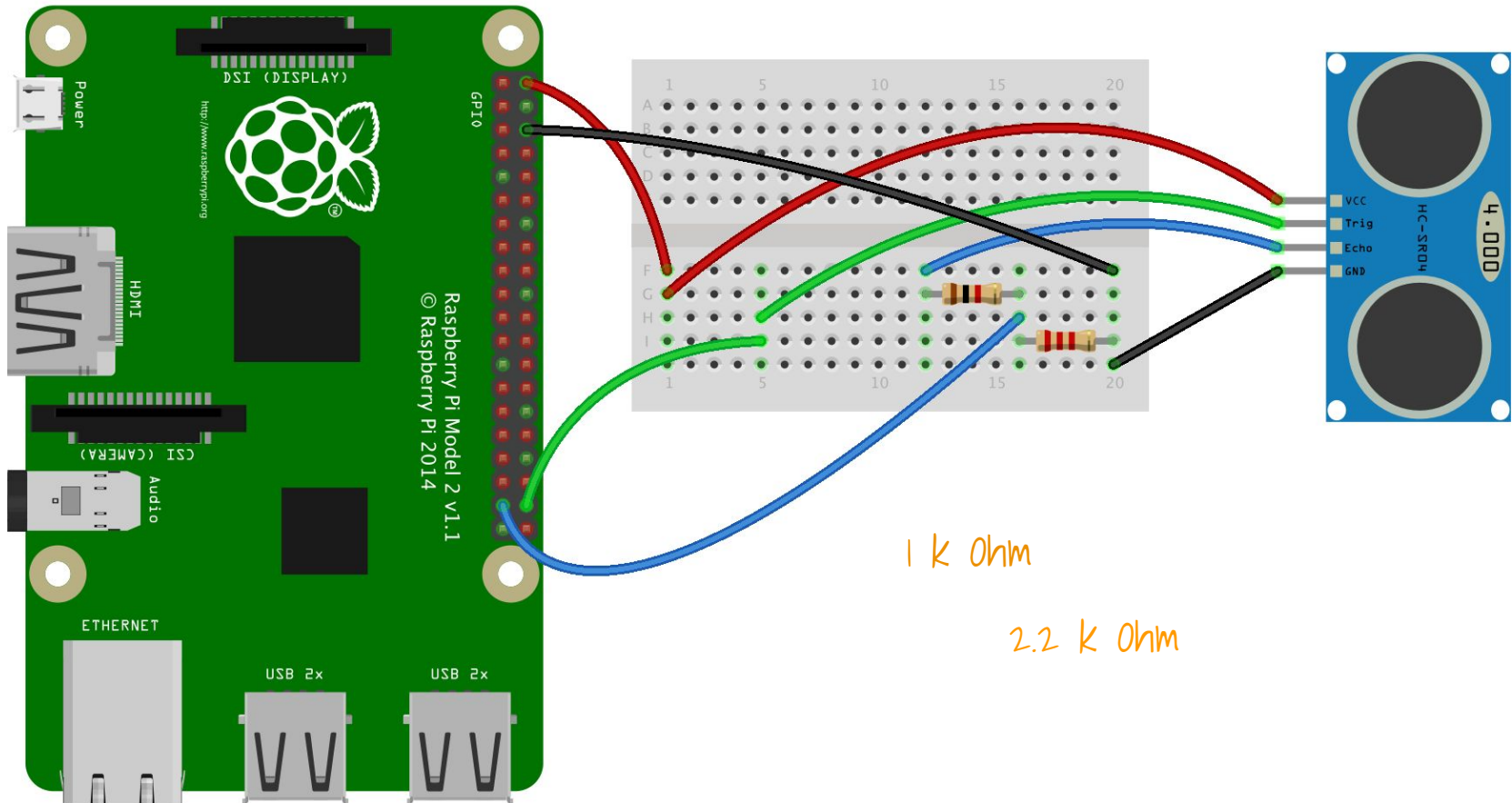
<https://github.com/pubnub/workshop-raspberrypi/blob/master/projects-python/range-finder/>



<http://pubnub.github.io/workshop-raspberrypi/web/range.html>



Ultrasonic RangeFinder



1 k Ohm

2.2 k Ohm

Thank you :-)



Creative Commons Attributions

- LED circuit: Wikimedia
- PIR Sensor: Wikimedia / Oomlout
- Ultrasonic: Wikimedia / Georg Wiora (Dr. Schorsch)
- GPIO Pins: RaspberryPi-Spy.co.uk

Also, great public domain images from Pixabay!



Please give us feedback at:

<https://goo.gl/0WeZQb>

A hand-drawn pink arrow originates from the word "zero" and points upwards to the first character of the URL, which is a zero.

zero