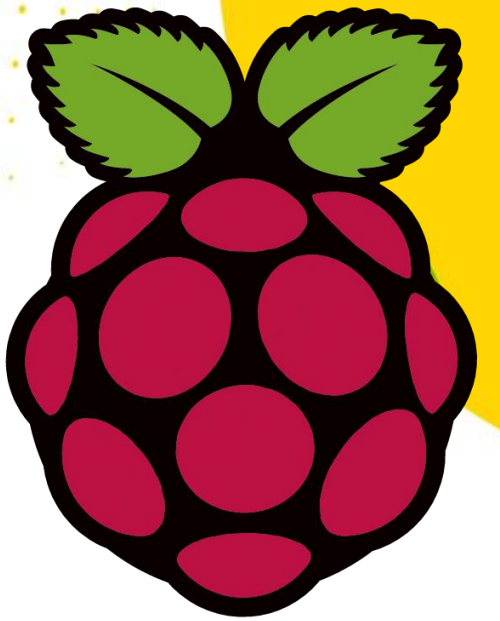
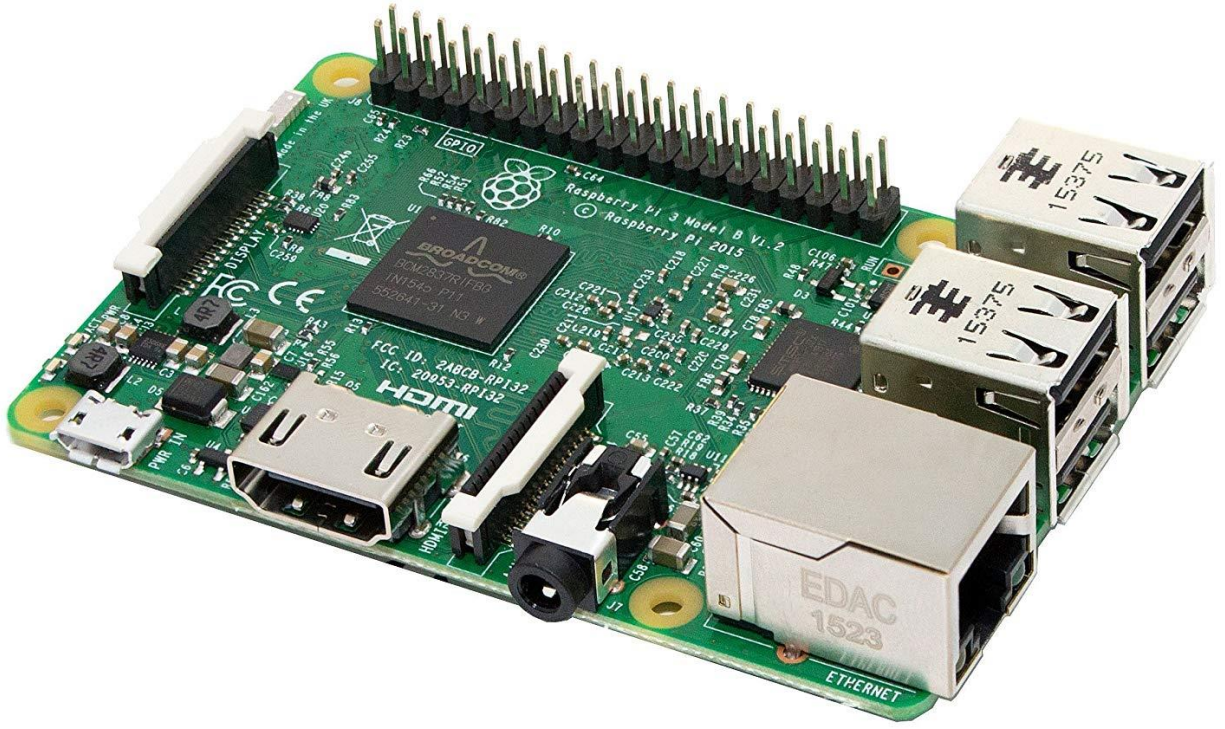
Intro to Raspberry Pi



Agenda

- Welcome
- Topics
- Structure
- Common Questions and Pitfalls
- GPIO

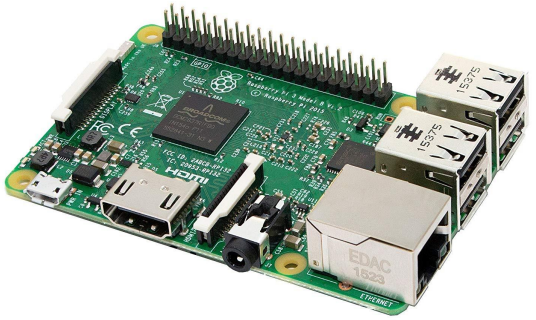


Welcome!

- We want to hear from you:
  - Your feedback is appreciated
  - Don’t be shy — ask away!
  - Additional supporting material if you need a refresher
- This workshop initially developed by students, for students:
  - IEEE student branch is a student run organization
  - Ties to the broader IEEE organization
  - Many Opportunities for involvement

What is a Raspberry Pi?

- A small, cheap computer
  - Good for prototypes
  - Can do most things a regular computer can do…
  - …at a fraction of the cost
- Easy to use
- Easy to connect to other devices
- Runs linux (Raspbian)
- Multiple versions with different use cases

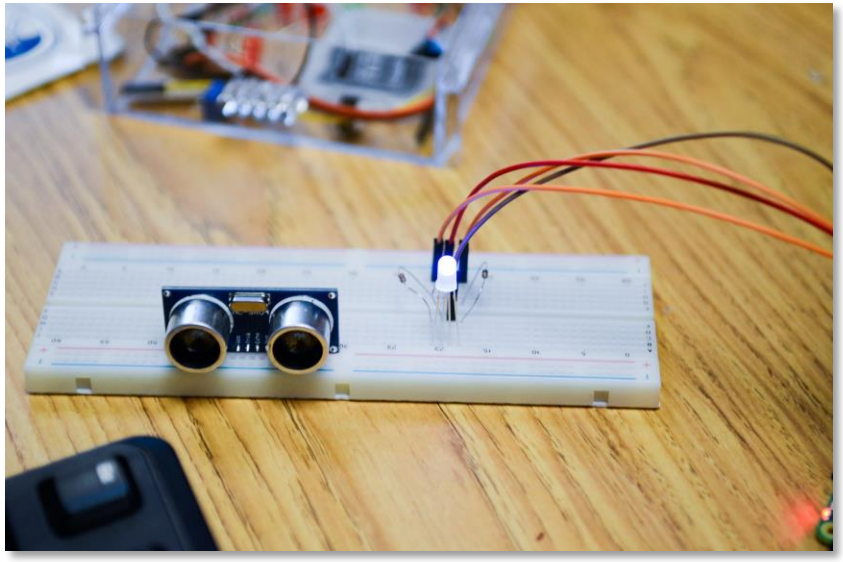


Topics

What Are We Doing Today?

- GPIO (we all start with this) – This is what you need to get started

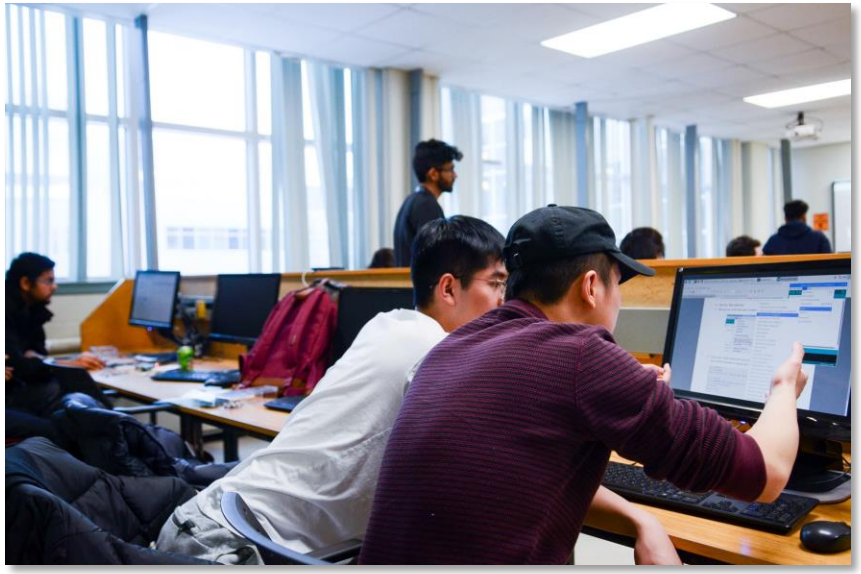
- Then, instructors will rotate between sections:
  - PWM and distance sensor
  - Working with a Camera Module
  - ADC and Serial Communication (using Arduino)
  - Audio Playback with RPi



Structure

Broad overview of each station

- GPIO is the basic building block common to all activities
  - We will discuss this now
  - We all go through exercises simultaneously
- For each station:
  - 5 minute explanation
  - 45 minutes to work through the exercises
  - 10 minutes – wrap-up and transition

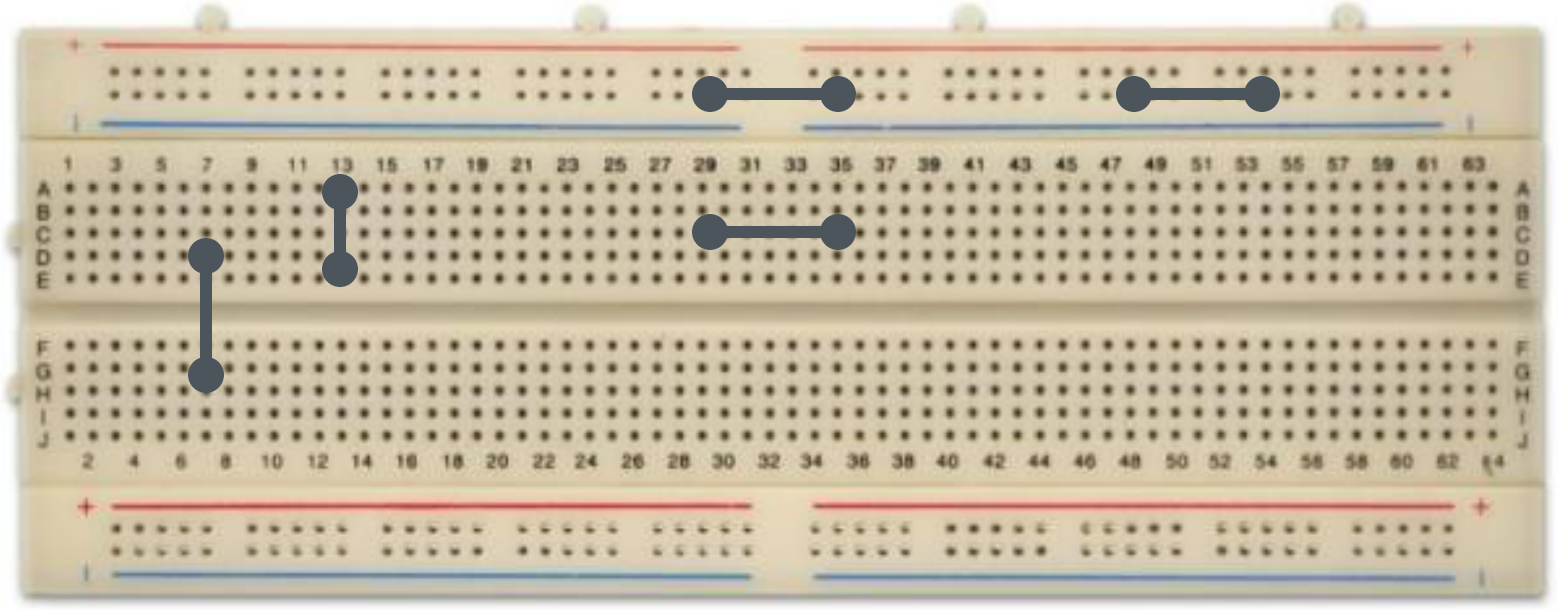


Volunteers will move between stations.

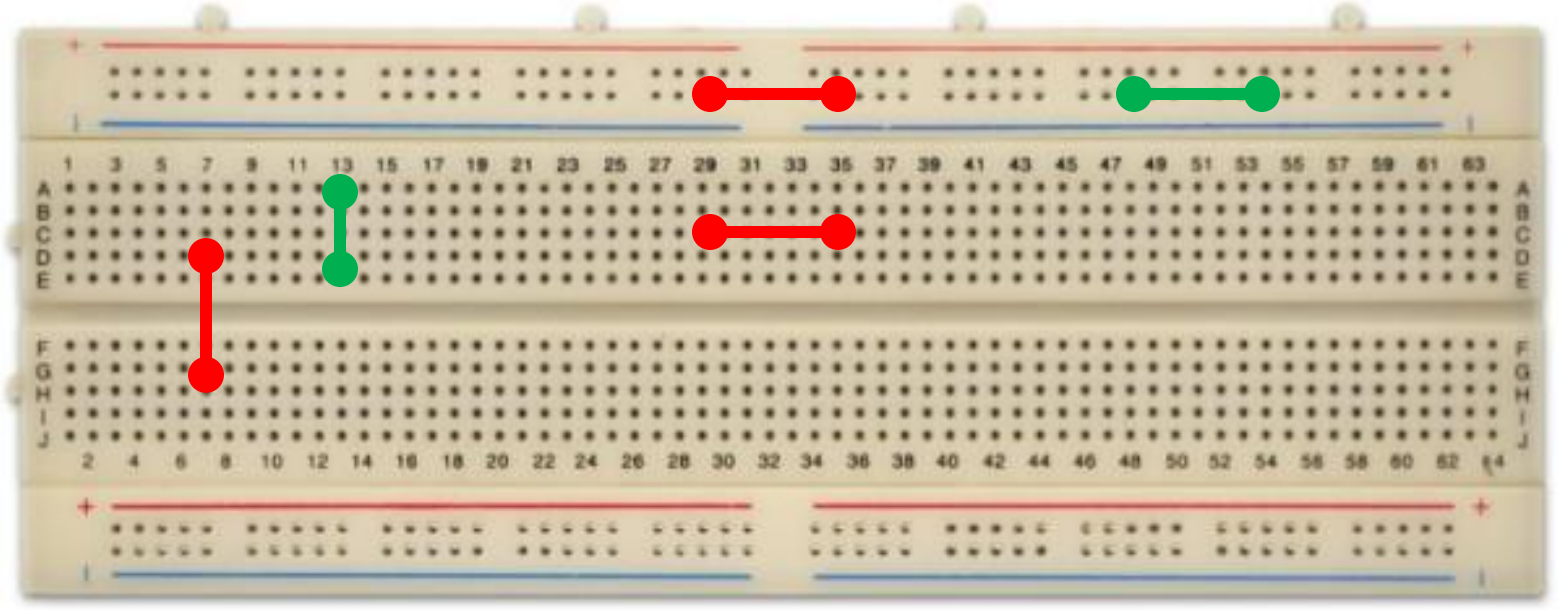
Common Questions and Pitfalls

Breadboards

Which of these points are connected?



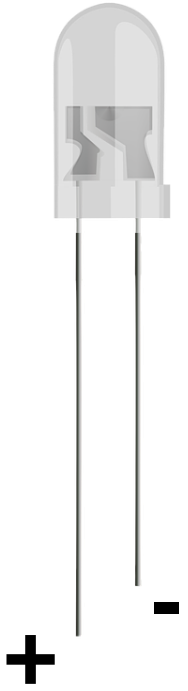
The GREEN ones.

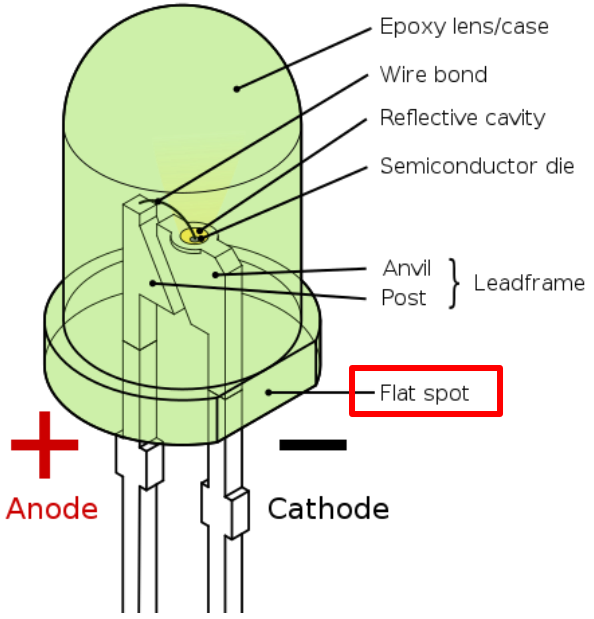


Common Questions and Pitfalls

LED's

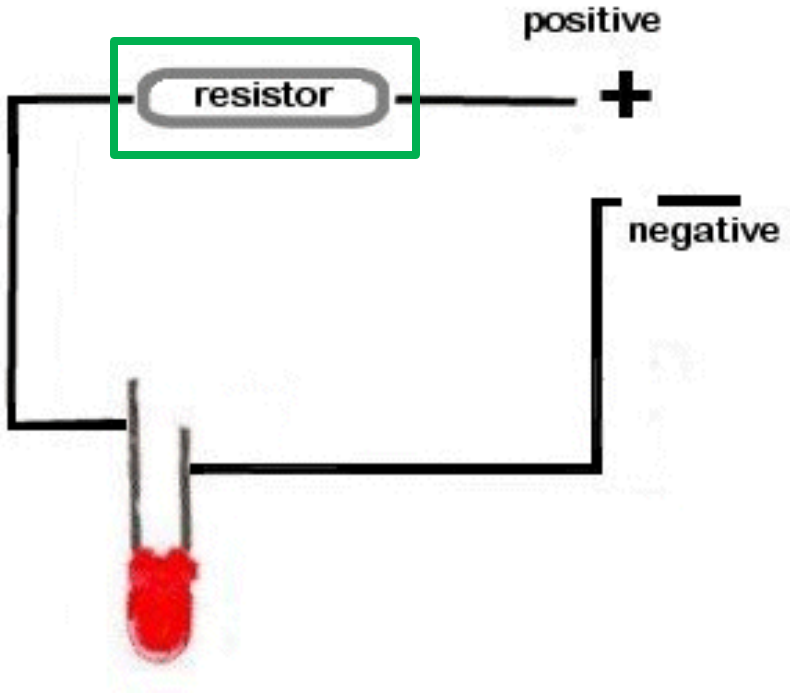
Two ways to tell which way they go…





If it doesn't work ☹ → try flipping it around ☺

LED's can burn up if too much current is passed through them



<u>Always</u> use a resistor to limit the current

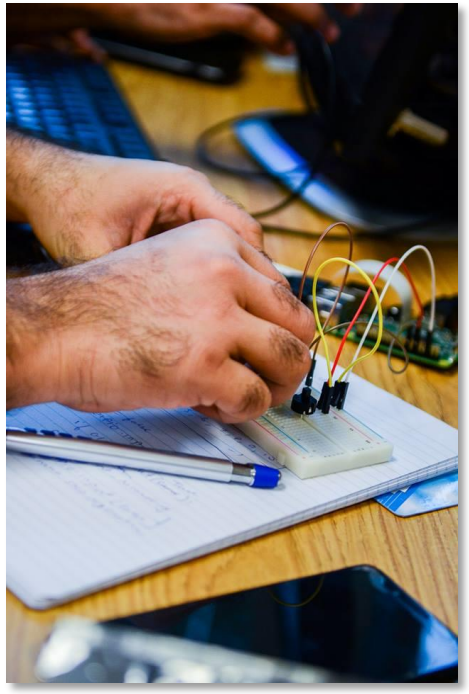
Don’t operate your RPi or Arduino on these bags!

They can short the pins on the bottom

Common Questions and Pitfalls

Taking Care of the RPi

- Don’t pull out the SD card or camera module. (Especially the camera!)
- **RPi is a 3.3V device**. Don’t apply voltages above 3.3V to the pins
- Don’t short the 5V and 3.3V pins
- Don’t pull out the power without clicking “shutdown” (in start menu)

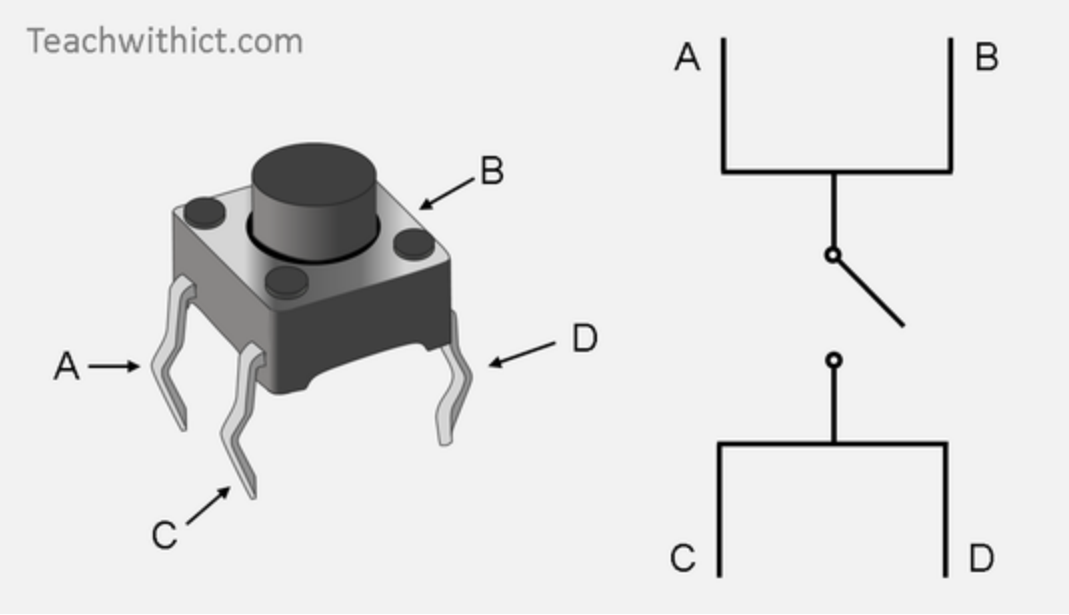


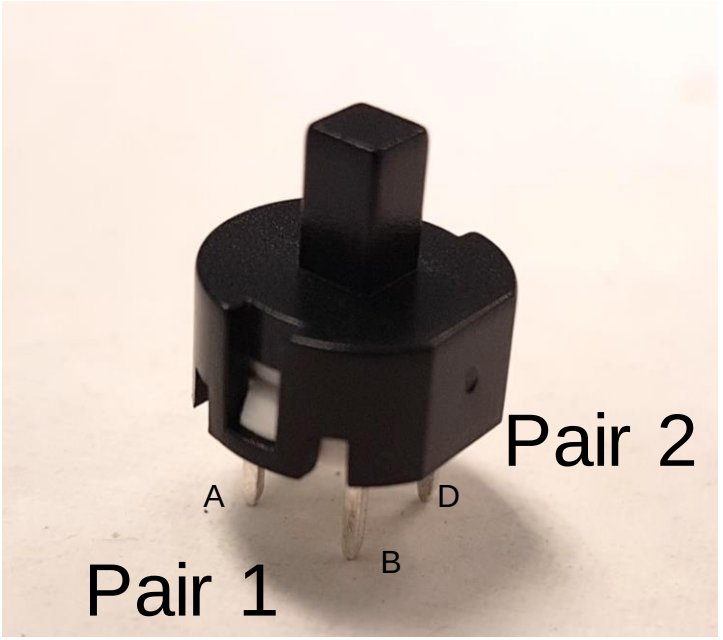
Common Questions and Pitfalls

Taking Care of the RPi…

- But also…don't worry too much…
- It is safe to use
- Quite robust
- Built in protections

Pins are wired in pairs…



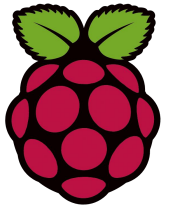


- The pair “becomes connected” when you press the button
- The two pairs are connected together (if wrong legs connected, button appears always pushed)

If you forget this, just use an LED to test which pairs are connected…

Common Questions and Pitfalls

Linux Terminal Tips



- Use IDLE IDE, **not THONNY**
- When in doubt:

Use “sudo” keyword before all commands
(not recommended with in a more sensitive setting, but it’s OK here)

- Syntax:

program flags arguments

Python -a –b –c “my_python_file.py”

- Break out of an operation:

Ctrl + C

- Press “up arrow” to get last command
- Press “Tab” to complete a command

i.e. type “Pyt” in the terminal then tab “Python”

- Questions?

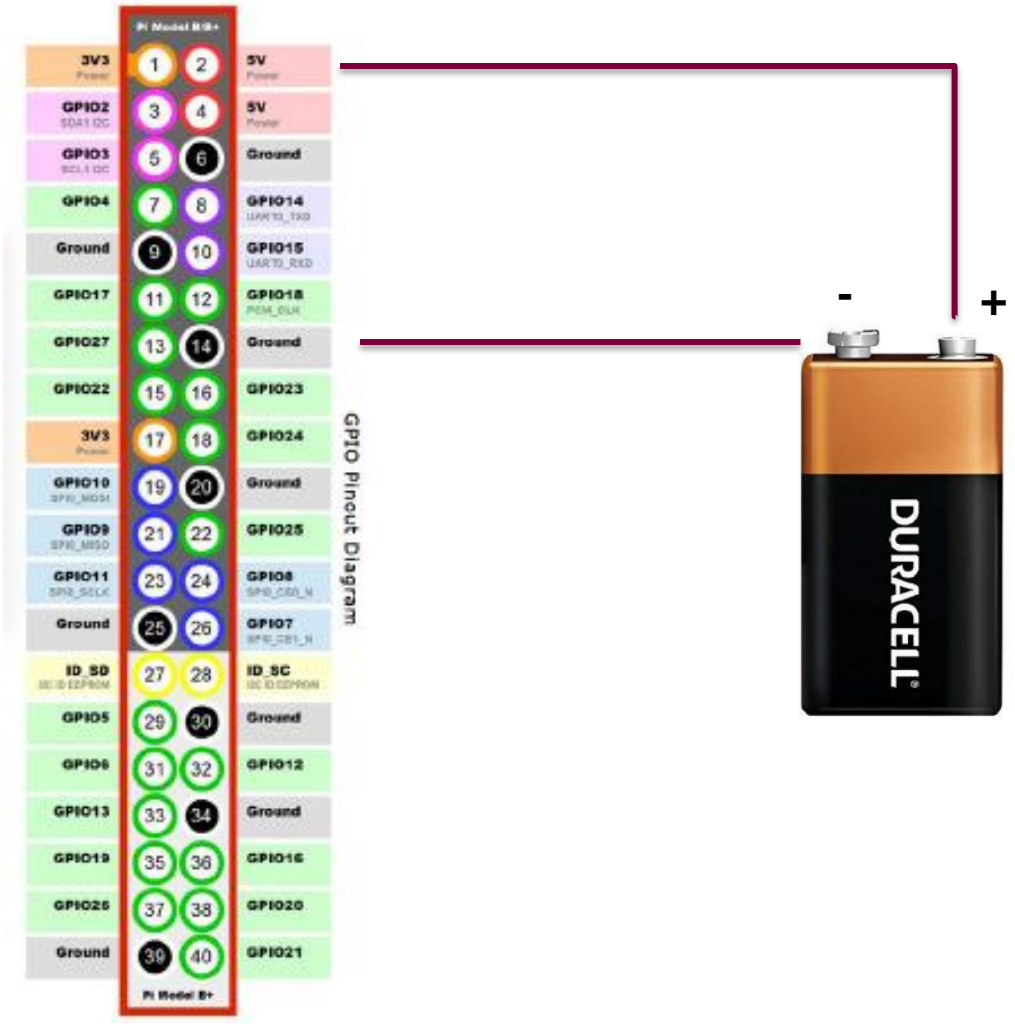
use “-h” with any command

(h stands for help!)

GPIO

What does it mean?

- General Purpose Input Output
- “Bidirectional”
- Pin numbering schemes:
  - BCM
  - Board 👍
- Important pins:
  - Ground (negative)
  - 3.3V (positive)
  - 5V pins (positive)



GPIO

Configuring for output

Configuring pins for output

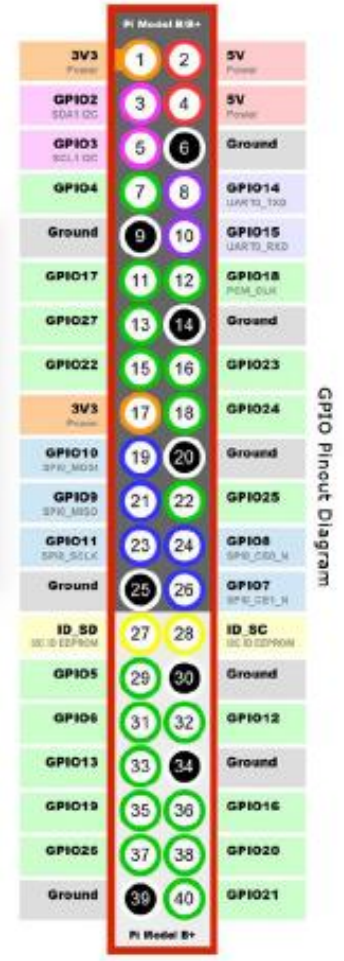
```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)   #physical numbering scheme

led = 11                   #connect LED to pin 11
GPIO.setup(led, GPIO.OUT)  #set pin 11 as output
```

Writing a value to a pin…

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



GPIO

Configuring for Input

Configuring Pins for Input

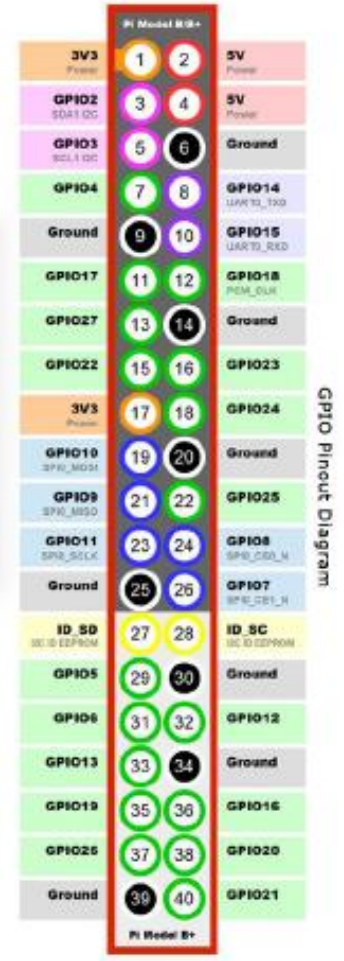
```
GPIO.setup(echo, GPIO.IN)

#Declare the pins as inputs and "attach" the pull up resistors to them
GPIO.setup(start, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(stop, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

Reading from Input Pins

```
if GPIO.input(start)==0:
    print('Start was pressed')
    sleep(0.5)
```

"start" is the **pin number** in this example



GPIO

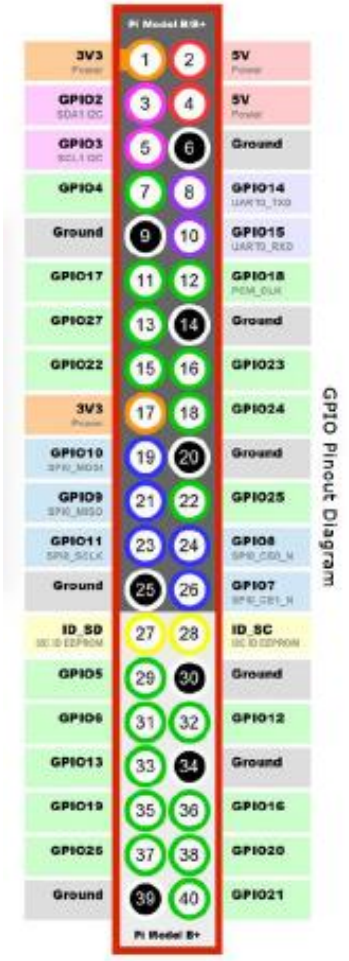
"Housekeeping"

Import statements

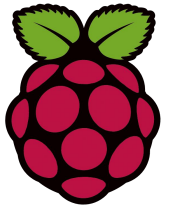
```
import RPi.GPIO as GPIO
from time import sleep
```

Clearing pin assignments

```
GPIO.cleanup()
```



Have Fun!



Let’s get started!