

APSC 1001 & CS 1010

Deep dive into Raspberry Pi with Python

Prof. Kartik Bulusu, MAE Dept.

From LED Madness to
Ultrasound Cacophony

Teaching Assistants:

Sara Tenaglio, BME Dept.

Catherine Karpova, BME Dept.

Zachary Stecher, CEE Dept.

Learning Assistants:

Jonathan Terry, CS Dept.

Ethan Frink, MAE Dept.

Jack Umina, CS Dept.

Olivia Legault, CS Dept.

Alexis Renderos, MAE Dept.



School of Engineering
& Applied Science

THE GEORGE WASHINGTON UNIVERSITY

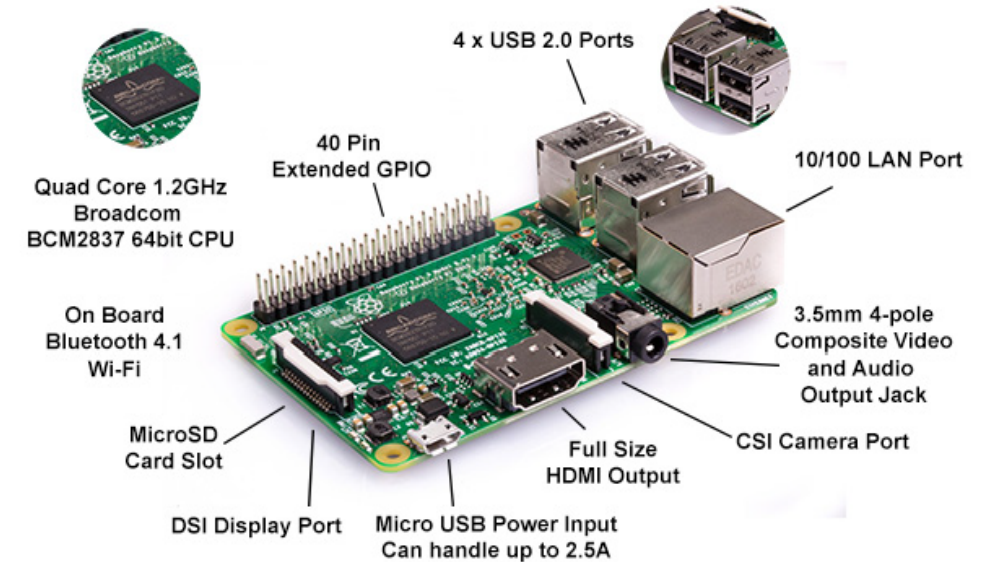
Fall 2021

Photo: Kartik Bulusu

Raspberry Pi Hardware and Connections

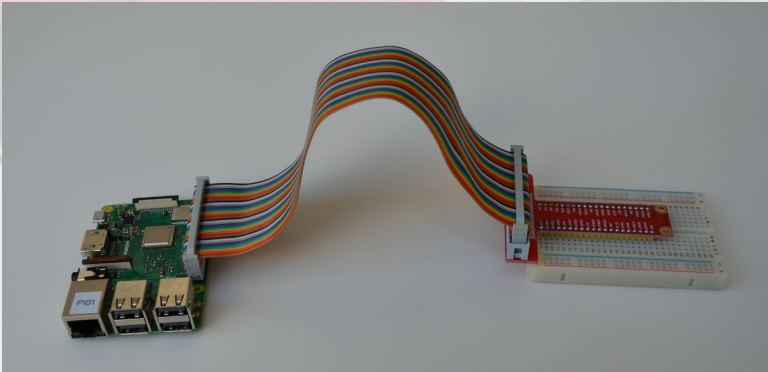


Source: <https://www.raspberrypi.org/help/>

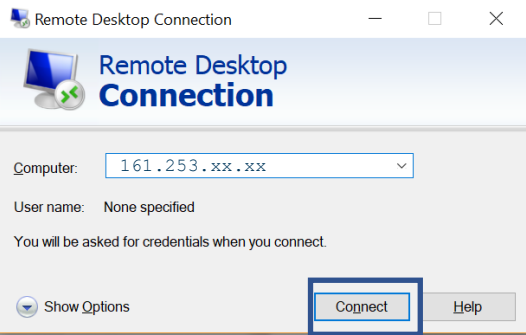


Source: <https://opensensorhub.org/2019/05/19/kinect-support-on-raspberrypi-3b/>

Connect the Raspberry Pi Model 3 B+ (RPI) to a bread board

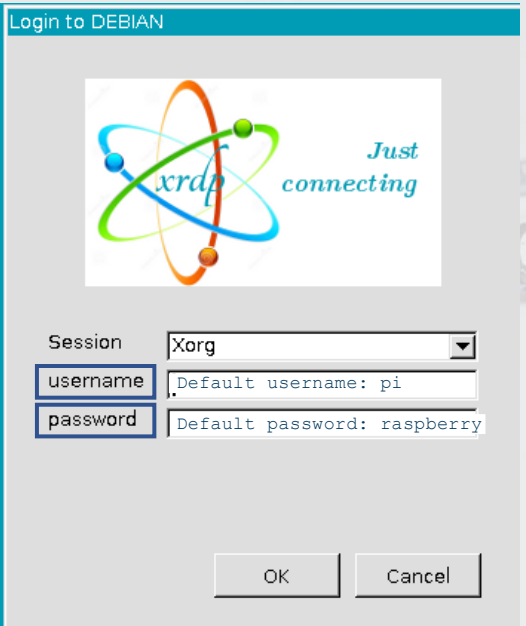


Access to the RPi in the laboratory



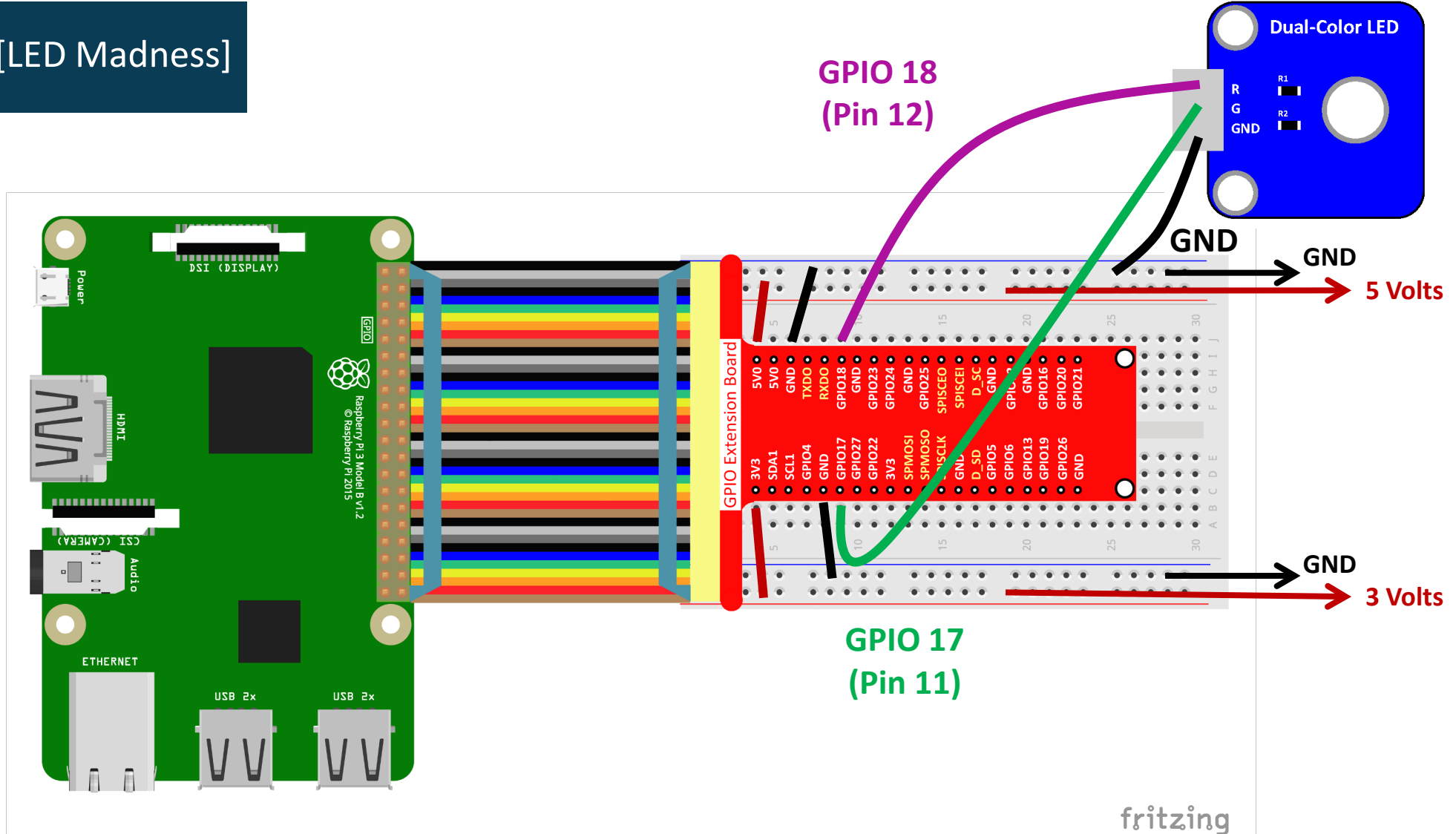
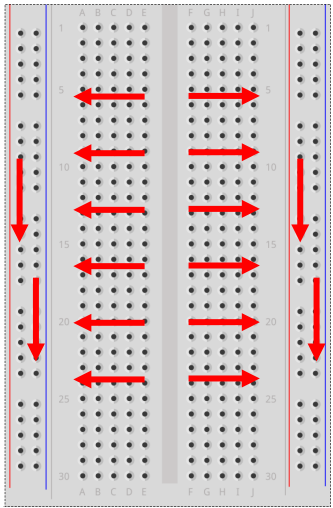
Each RPi is assigned a unique

- IP address <161.253.xx.xx>
- *username & password*

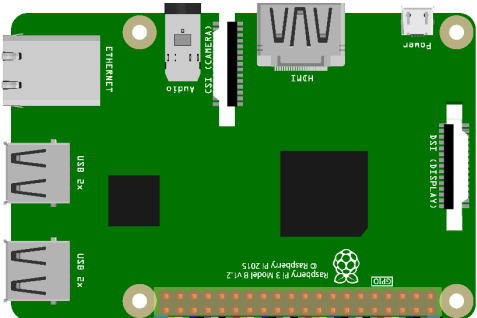


Source: https://upload.wikimedia.org/wikipedia/commons/f/f1/XRDP_Screenshot.png

Recap from last week [LED Madness]



How a python code lit up your LED with Raspberry Pi Model 3 B+ (RPi)



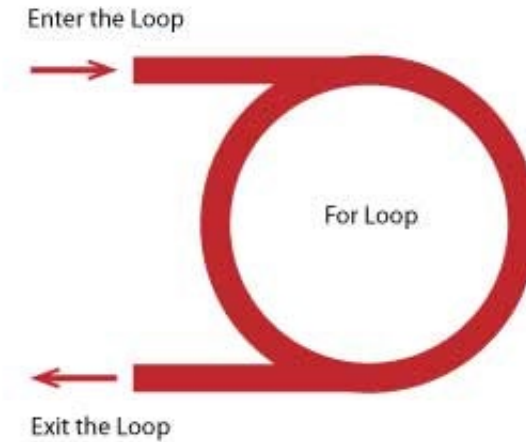
```
import RPi.GPIO as GPIO
import time
```

```
GPIO.setmode(GPIO.BOARD)
```

GPIO Extension Board			
1	3V3	5V0	2
3	SDA1	5V0	4
5	SCL1	GND	6
7	GPIO4	TXD0	8
9	GND	RXD0	10
11	GPIO17	GPIO18	12
13	GPIO27	GND	14
15	GPIO22	GPIO23	16
17	3V3	GPIO24	18
19	SPMOSI	GND	20
21	SPMOSO	GPIO25	22
23	SPISCLK	SPISCEO	24
25	GND	SPISCEI	26
27	D_SD	D_SC	28
29	GPIO5	GND	30
31	GPIO6	GPIO12	32
33	GPIO13	GND	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	GND	GPIO21	40

```
GPIO.setup(12, GPIO.OUT)
```

(For) How many times do you want to execute a piece of code ?



```
for i in range(0,15):

    GPIO.output(12, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(12, GPIO.LOW)
    time.sleep(0.5)
    print(i)

GPIO.cleanup()
```

Ultrasound Signals and its Applications



Source: <https://youtu.be/Rr9RaisO11E>

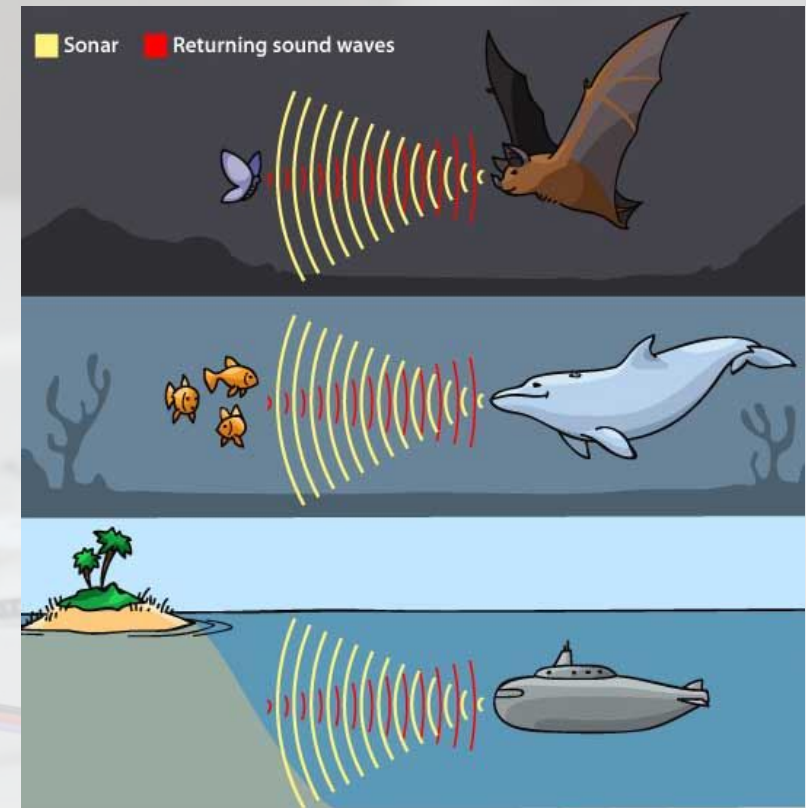
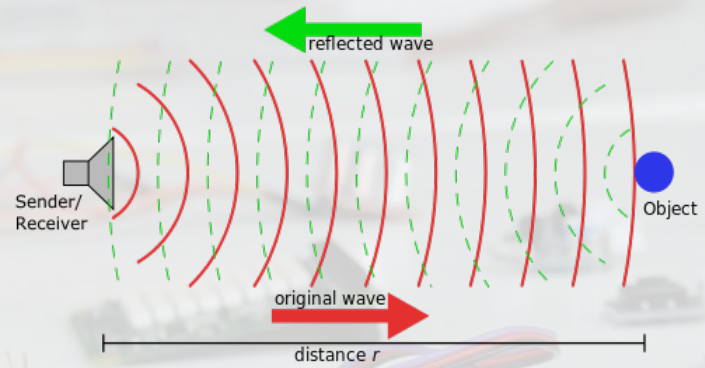
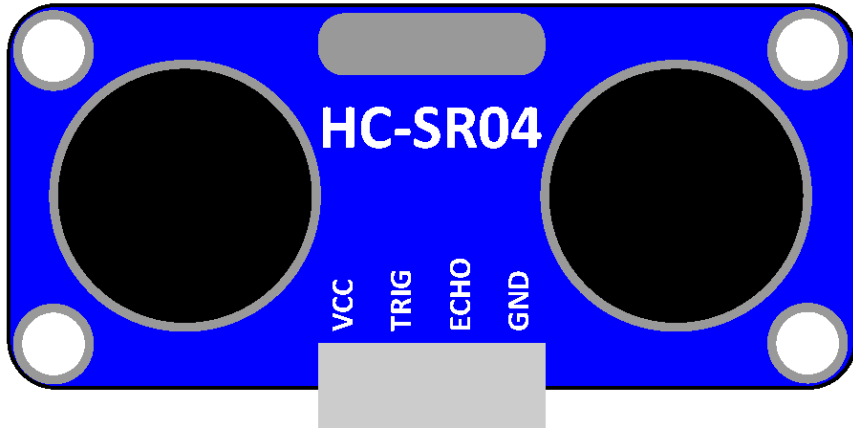


Image Credit:
<http://www.robaid.com/bionics/bat-biosonar-biomimicry-for-improved-sonar-technology.htm>

$$Distance\ traversed = (Speed\ of\ sound) \times (Time\ elapsed/2)$$

Know your Ultrasonic Sensor



The Ultrasonic sensor sends out ultrasonic waves to detect objects and measure distances.

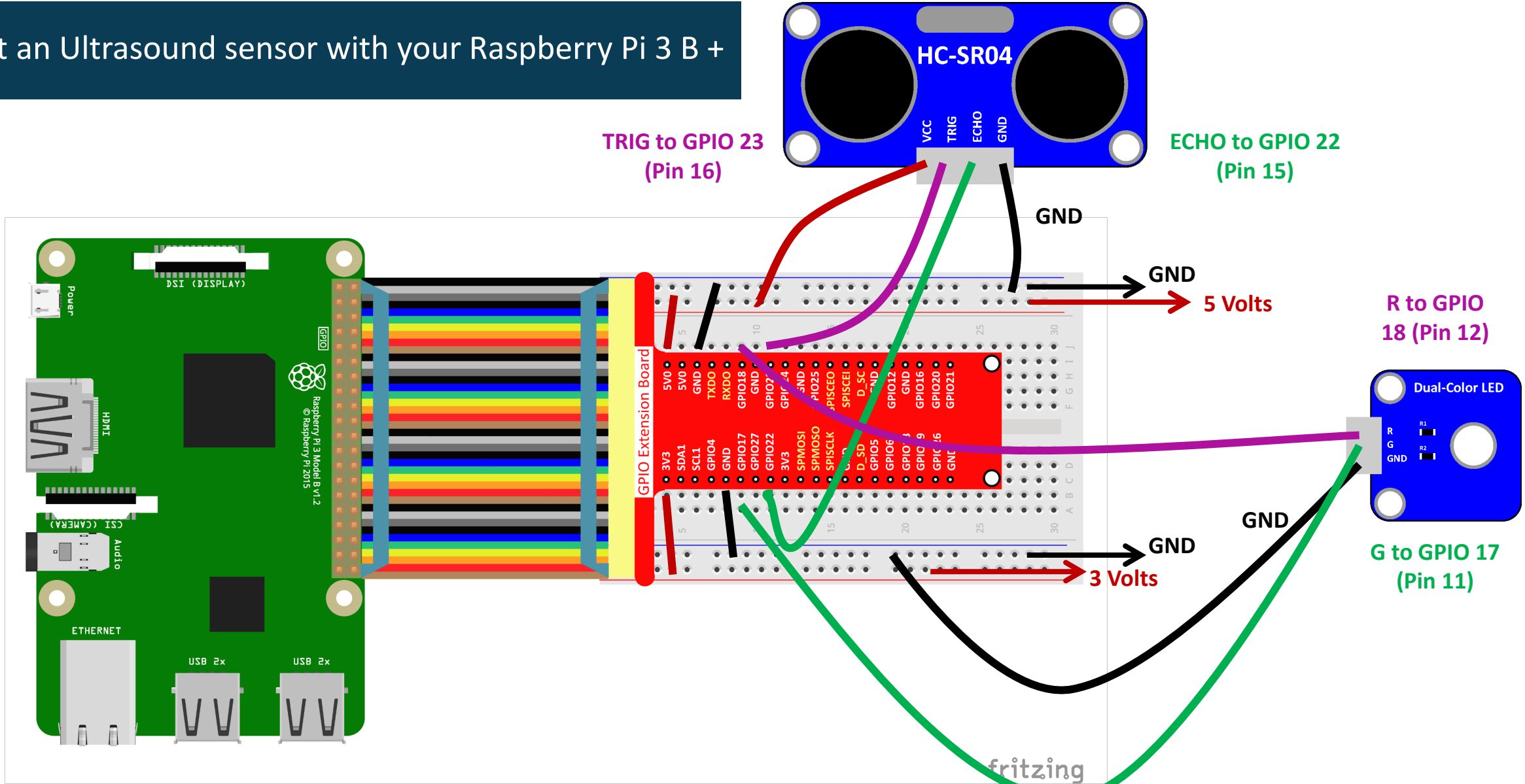
Connector:

4-pin anti-reverse cable

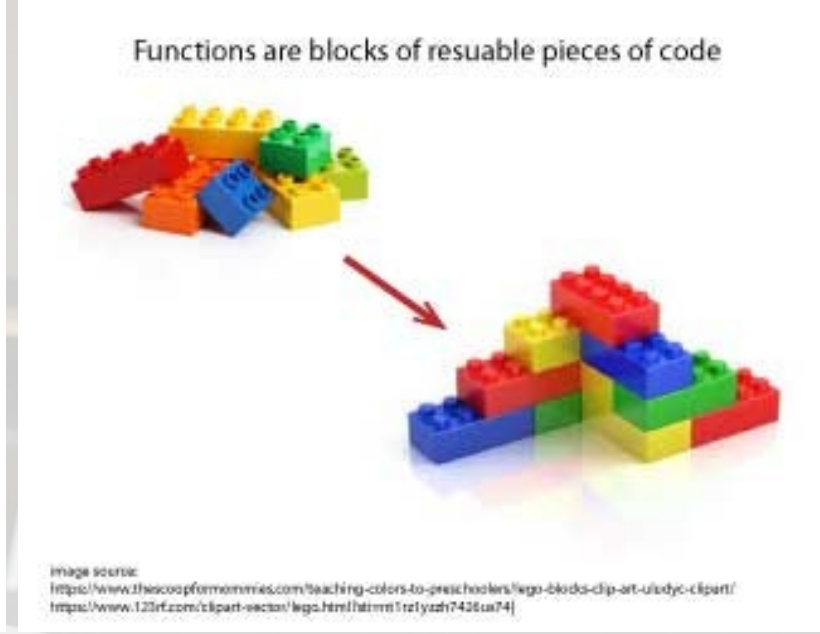
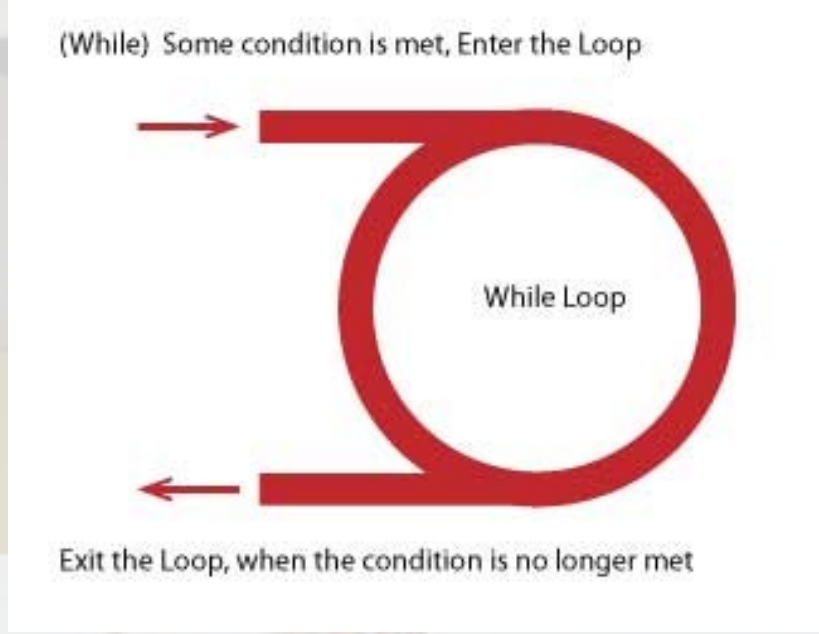
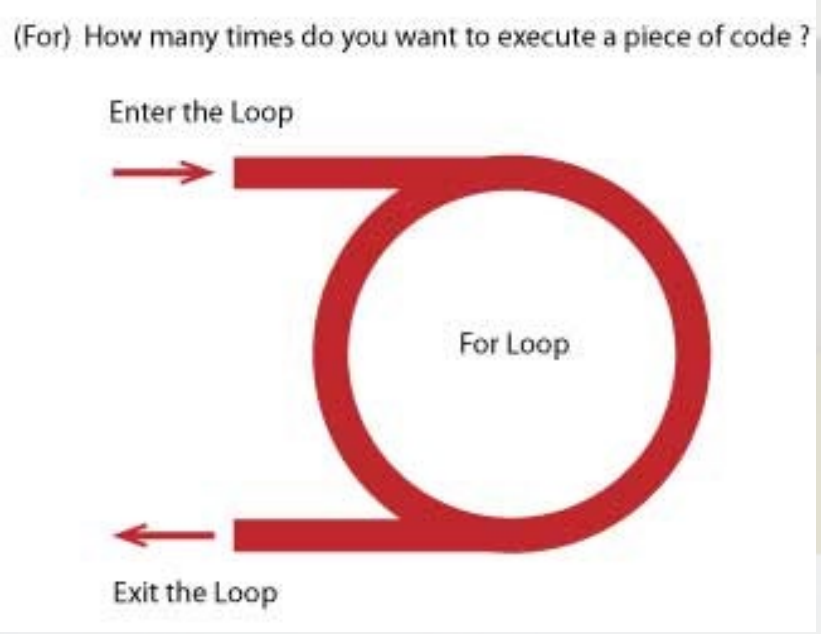
Goal of the lab segment:

- **Co-work**
 - Observe, ask and try in groups
- **Make**
 - Build-a-hack
 - Ultrasound sensors and Raspberry Pi 3B + boards
- **Analyze data using Python**
- **Think about**
 - Challenges, Opportunities, Gaps and Surprises

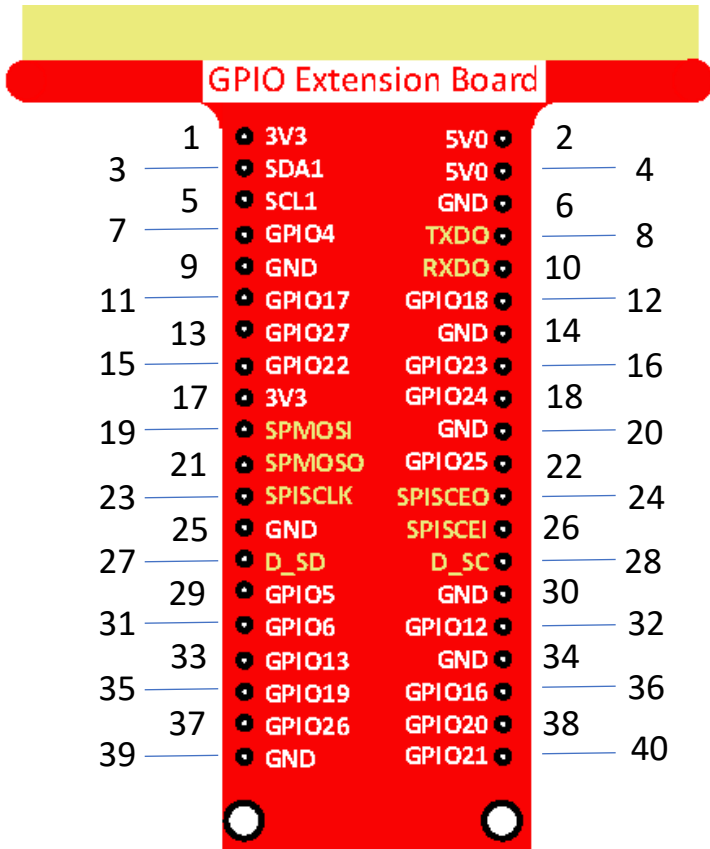
Start an Ultrasound sensor with your Raspberry Pi 3 B +



Know some programming paradigms



Pseudo-code to kick start your Raspberry Pi Model 3 B+ (RPi)



```
import LIBRARY as NAME
import ANOTHER_LIBRARY
```

INITIALIZE GPIO CHANNELS

DEFINE SETUP FUNCTION

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(CHANNEL-1, GPIO.OUT)
GPIO.setup(CHANNEL-2, GPIO.IN)
```

DEFINE DISTANCE FUNCTION

```
return (TIME_ELAPSED / 2) * 340 * 100
```

DEFINE LOOP FUNCTION

```
while True:
    ...
```

DEFINE DESTROY FUNCTION

CLEAN UP GPIO CHANNELS

```
if __name__ == "__main__":
    setup():
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```

User defined functions

Functions are blocks of reusable pieces of code

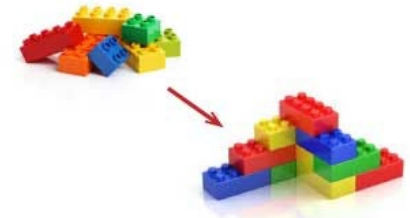
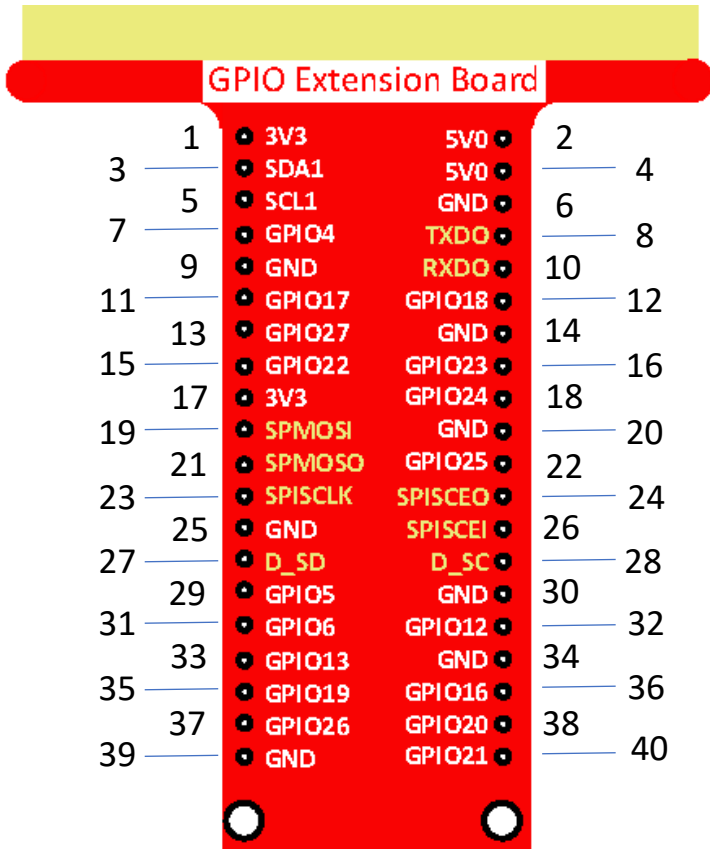


image source:
<https://www.thecodingmummies.com/teaching-coders-to-press-hobbies/lego-blocks-to-art-activity-clipart/>
<https://www.123kit.com/lego.html#/art/lego-art-7426a74>

Entry point into the program – pulls in all user defined functions

A simple python code to kick start your Raspberry Pi Model 3 B+ (RPi)



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

```
TRIG = 16
ECHO = 15
def setup():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.setup(ECHO, GPIO.IN)
```

```
def distance():
    GPIO.output(TRIG, 0)
    time.sleep(0.000002)
    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)

    while GPIO.input(ECHO) == 0:
        time1 = time.time()

    while GPIO.input(ECHO) == 1:
        time2 = time.time()

    during = time2 - time1
    return (during / 2) * 340 * 100
```

```
def loop():
    while True:
        dist = distance()
        print(dist, 'cm')
        print('')
        time.sleep(0.1)
```

```
def destroy():
    GPIO.cleanup()
```

```
if __name__ == "__main__":
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```