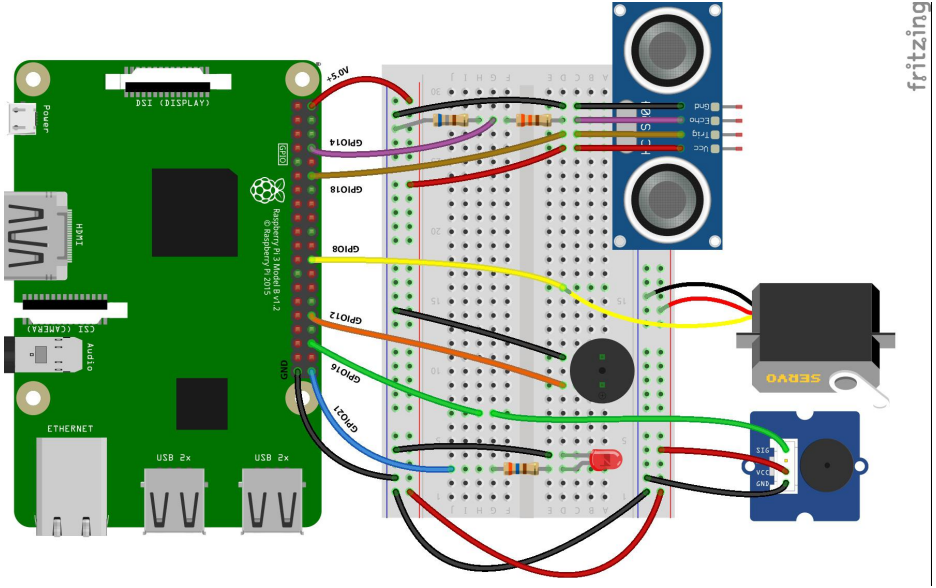
Making Sense



Taught by Colin Wang

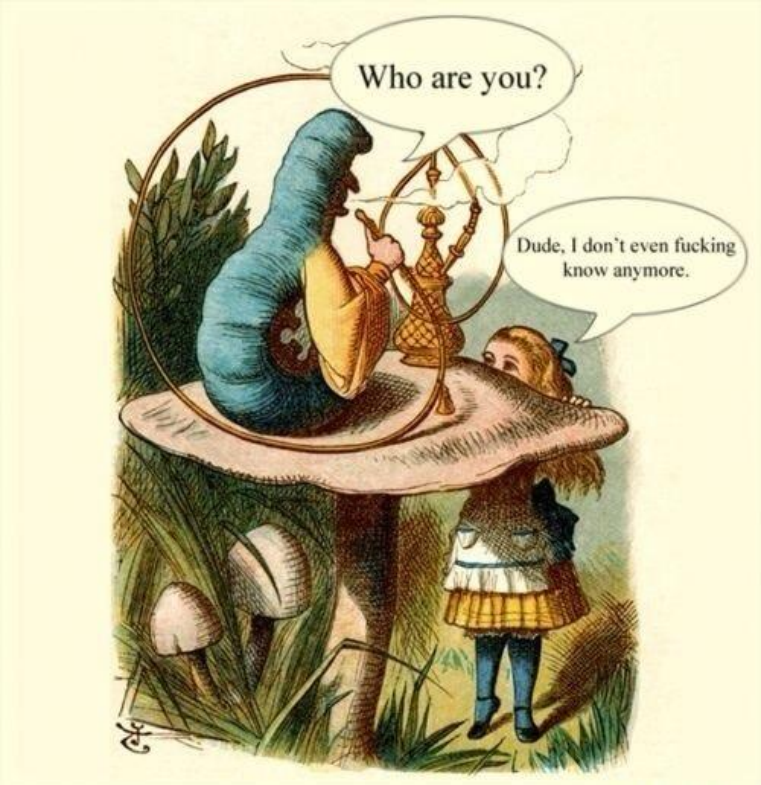
School for Poetic Computation Detroit 2019

Tonight’s Agenda

- Introduction
- Pre-class activity
- What is physical computing?
- Code and connect along
- Break
- Playtime
- Share out
- Wrap up

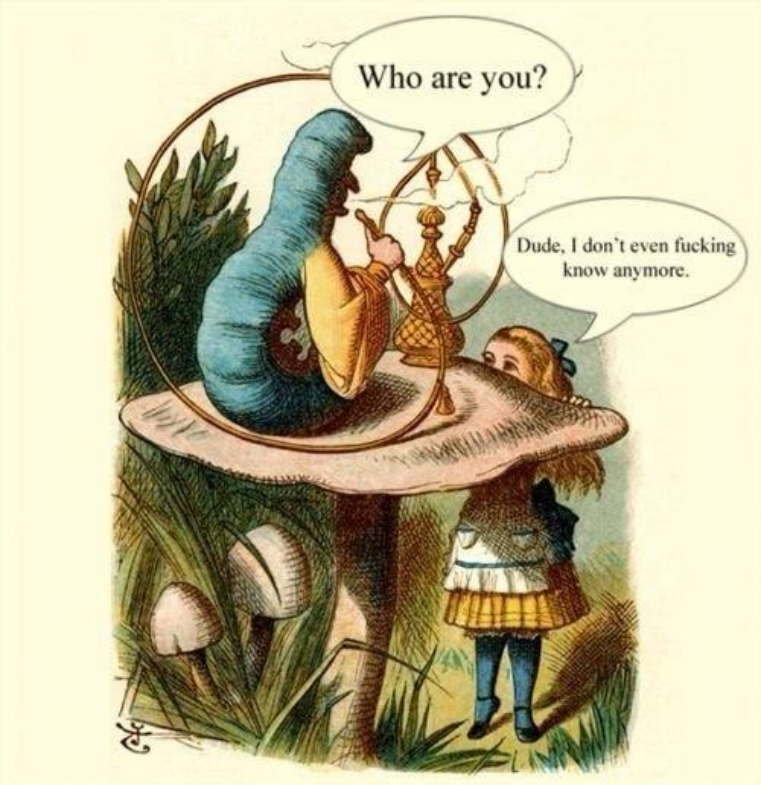
Who Are You?

- ? Was not down with computation
- ? Shunned and scoffed at it for years
- ? Took some years to get over that
- ? Computation is so cool
- ? Why would I shut myself out of that community?
- ? SFPC



Who Are You?

- ! Mechanical Engineer
- ! Computer cables
- ! Furniture
- ! Antique coin operated arcade
- ! Makerspace
- ! Prototyping
- ! Science museum

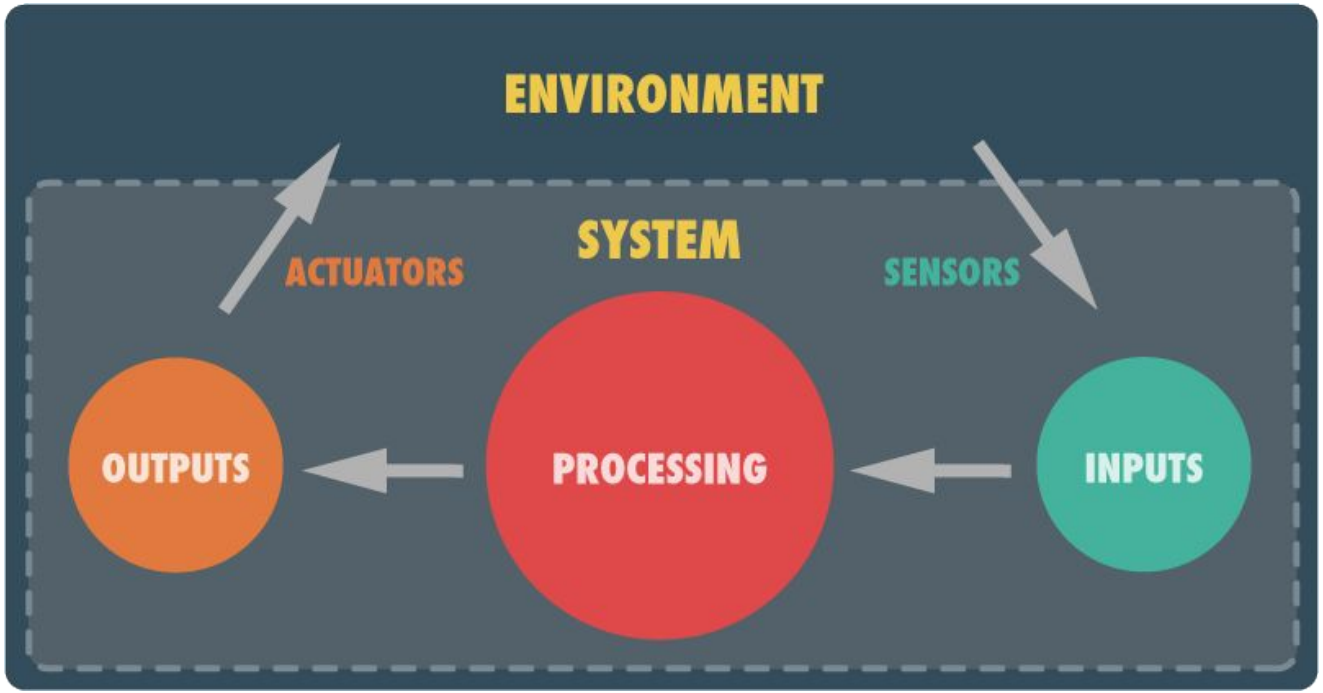


Getting in Touch with Our Senses




The Senses, Rembrandt, 1647

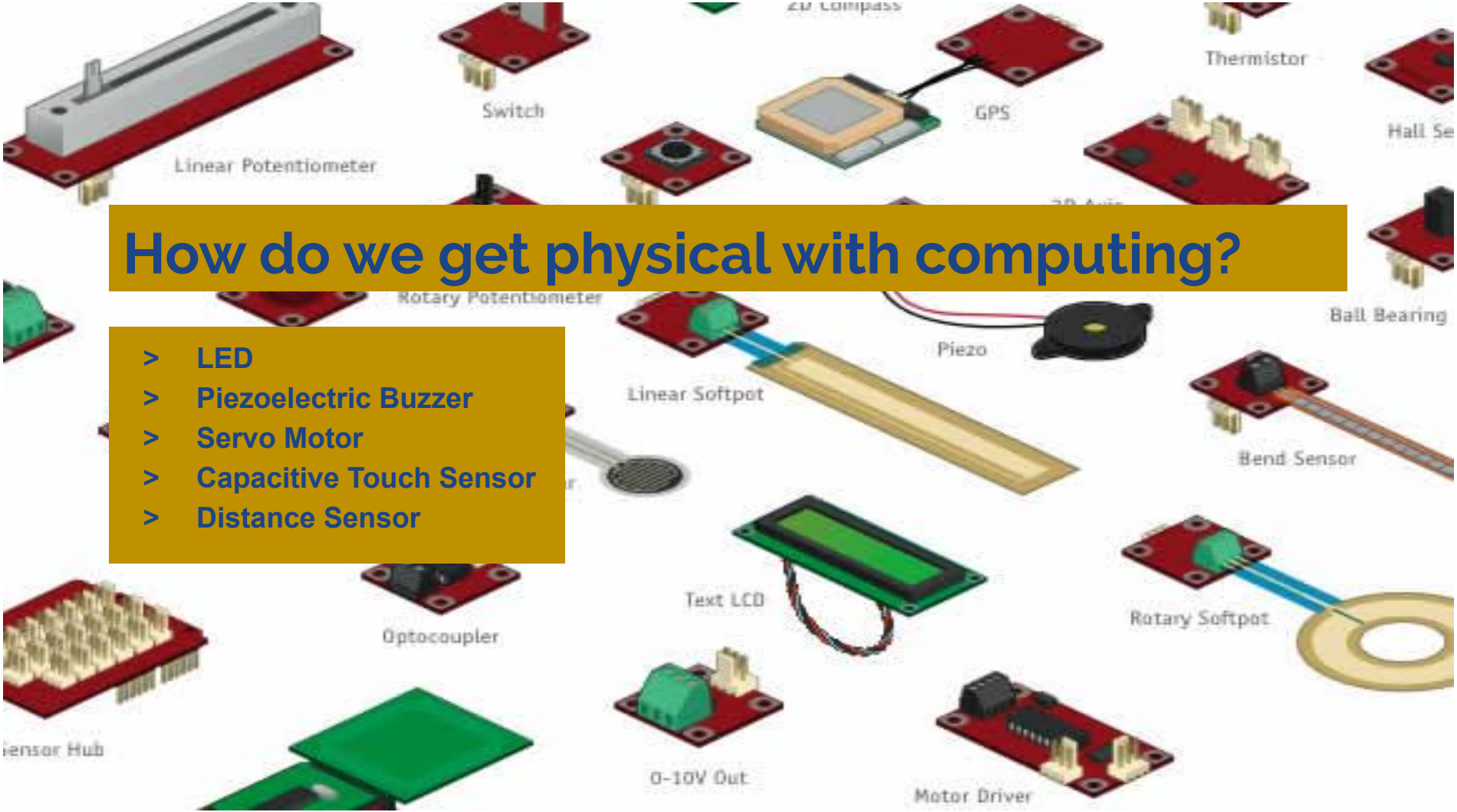
Physical Computing



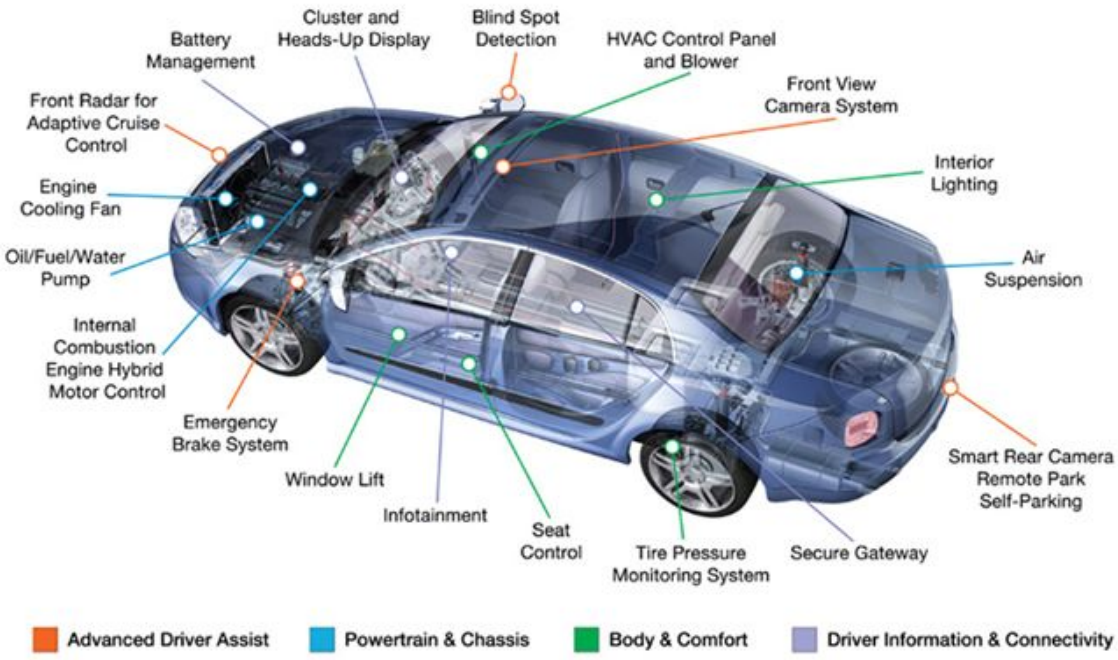
How did we get physical with computation?

How do we get physical with computing?

- > LED
- > Piezoelectric Buzzer
- > Servo Motor
- > Capacitive Touch Sensor
- > Distance Sensor



Is Physical Computation Poetic?



School For Poetic Physical Computation

SAM Sound Activated Mobile by Edward Ihnatowicz, Cybernetic Serendipity 1968
Mutual Air by Rosten Woo 2019

Ground Rules

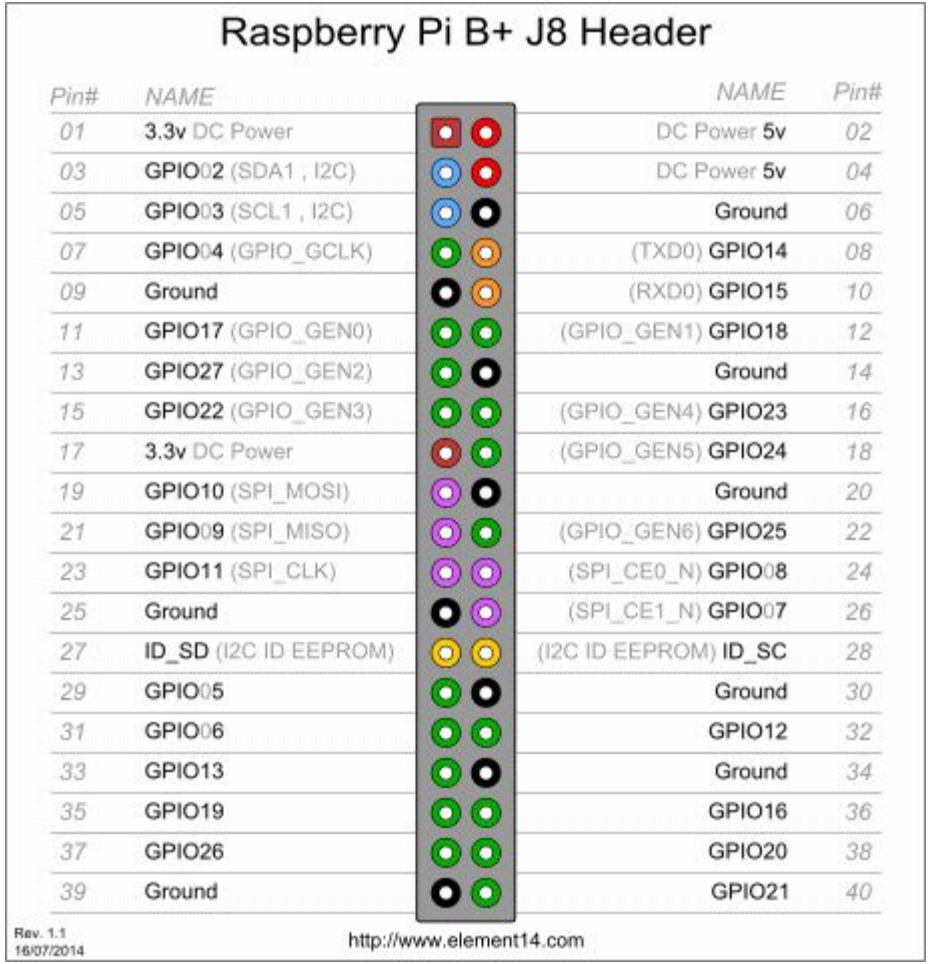
- Ask questions! Be curious!
- Speak up if you're stuck or want a double check
- If you've got it down, help your neighbor
- Err on the side of caution, don't damage your Pi
- Have fun! Jam, Remix, Improv

Grounding ⏚urselves

- [ ] Touch something metal and relatively large
- [ ] Two feet on the ground and a deep breath
- [ ] Stretch and limber up your hands and fingers

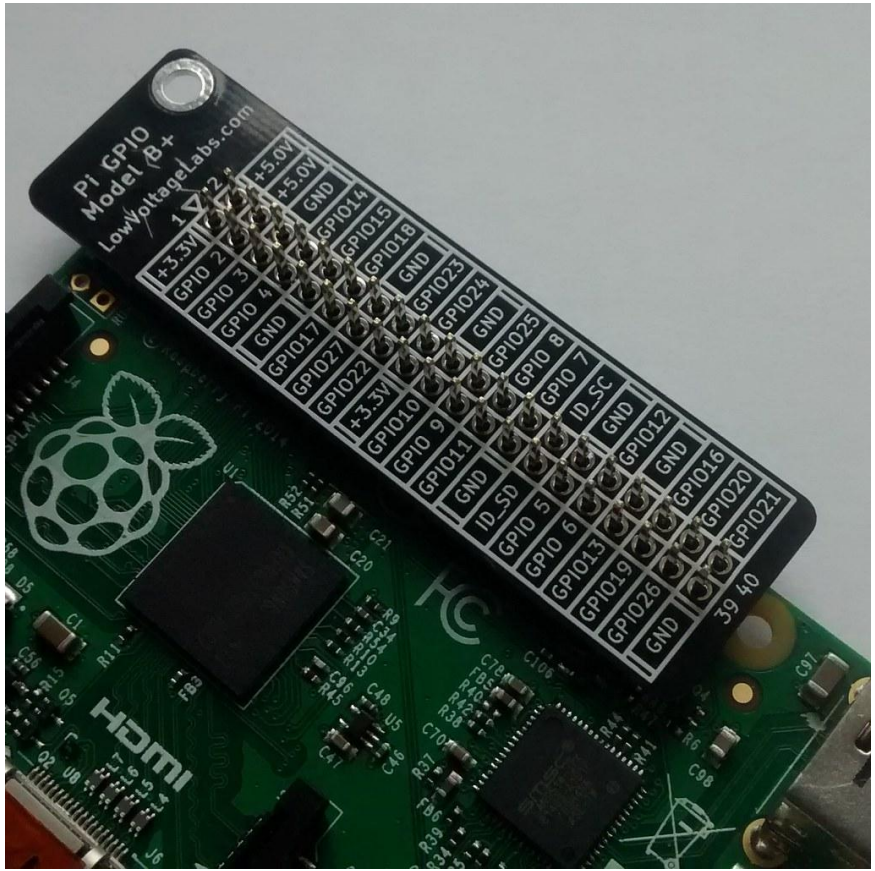
GPIO

- General Purpose Input / Output
- Can read voltages when set to input
- Can output 0V up to 3.3V

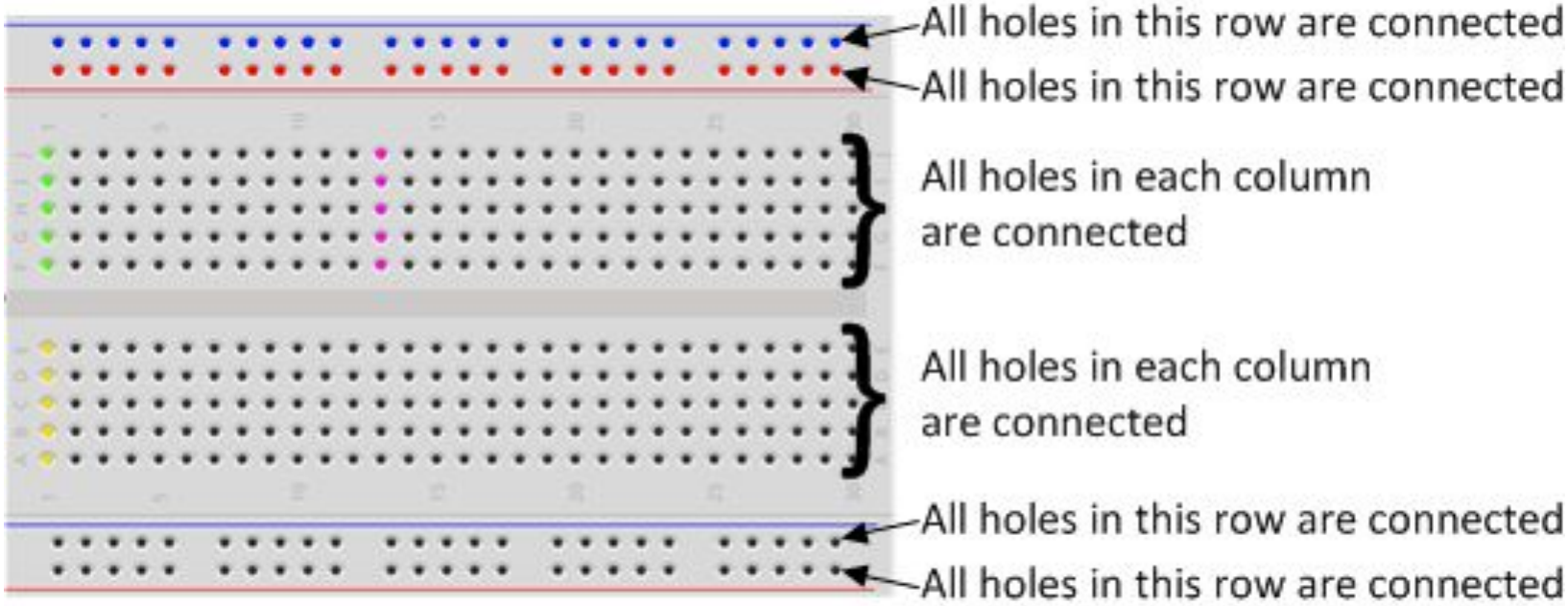


GPIO Reference Card

- Big hole towards the end of the board
- Both sides are identical

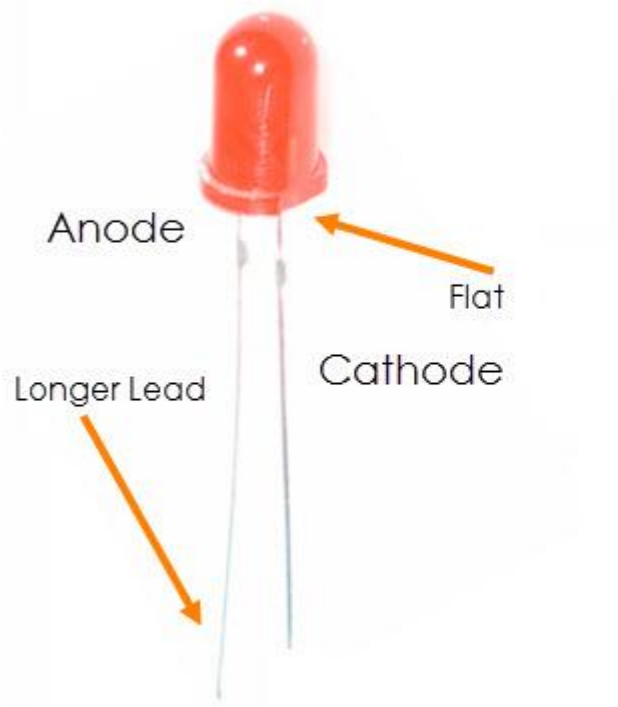


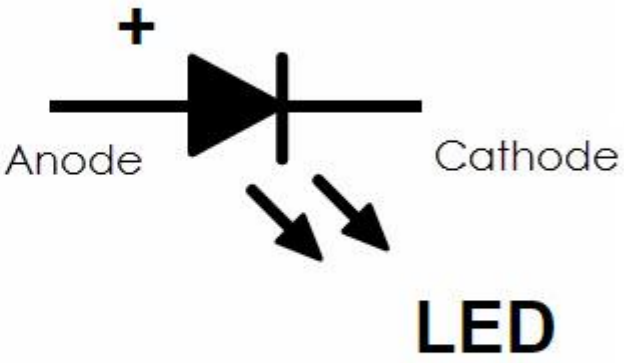
Breadboard


All holes in this row are connected
All holes in this row are connected
All holes in each column
are connected
All holes in each column
are connected
All holes in this row are connected
All holes in this row are connected

Breadboard

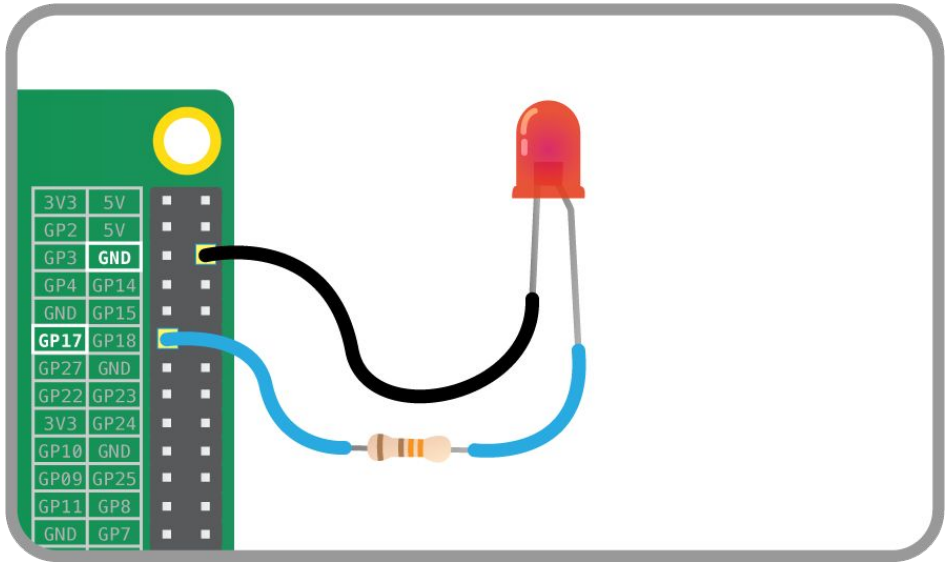
LED - Light Emitting Diode





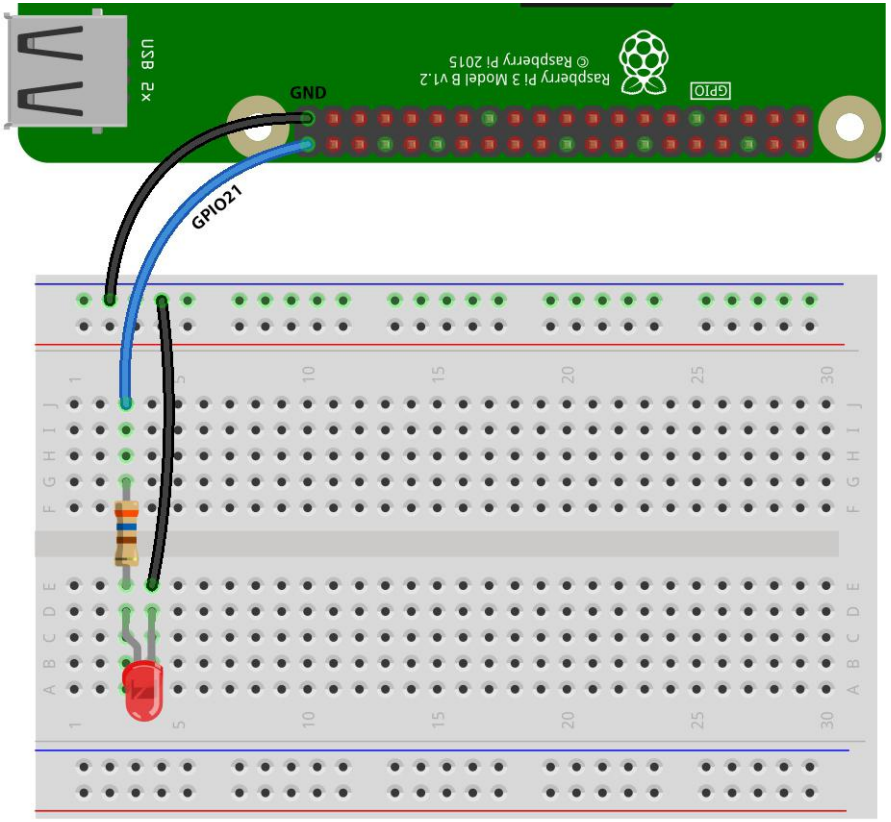
Always use a resistor with an LED

- LEDs can only draw up to a certain amount of current before burning out
- But they are not complex enough to prevent this
- Using Ohm's law we can calculate the resistance needed to limit the current draw



LED Circuit

- 💡 Connect negative lead (flat side) to ground
- 💡 Connect positive lead (long) to 360 Ohm resistor and then to GPIO21



gpiozero library

`from gpiozero import LED`	Imports the LED functions from the GPIO library
`from time import sleep`	Imports the sleep functions from the time library

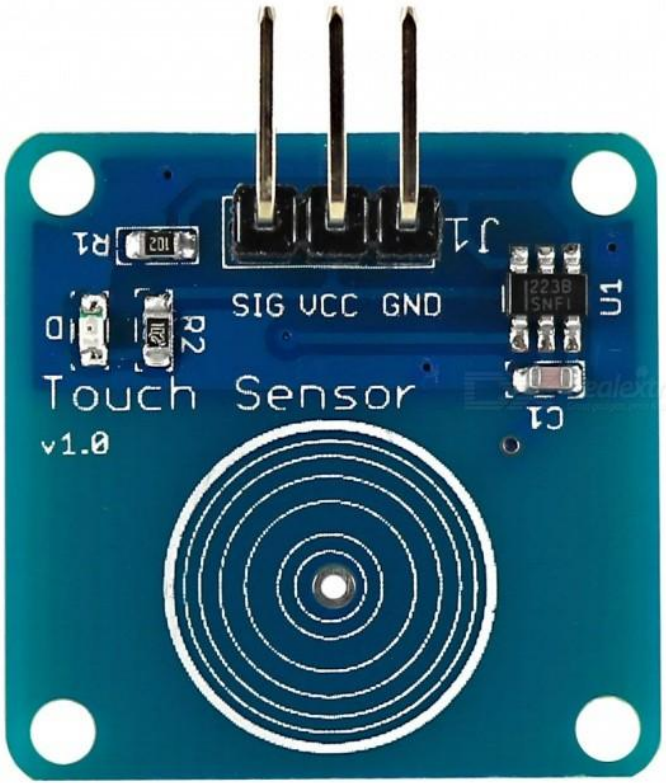
https://gpiozero.readthedocs.io/en/stable/

Flashing LED

```python
from gpiozero import LED
from time import sleep

led = LED(21)

while True:
    led.on()
    sleep(.5)
    led.off()
    sleep(.5)
```

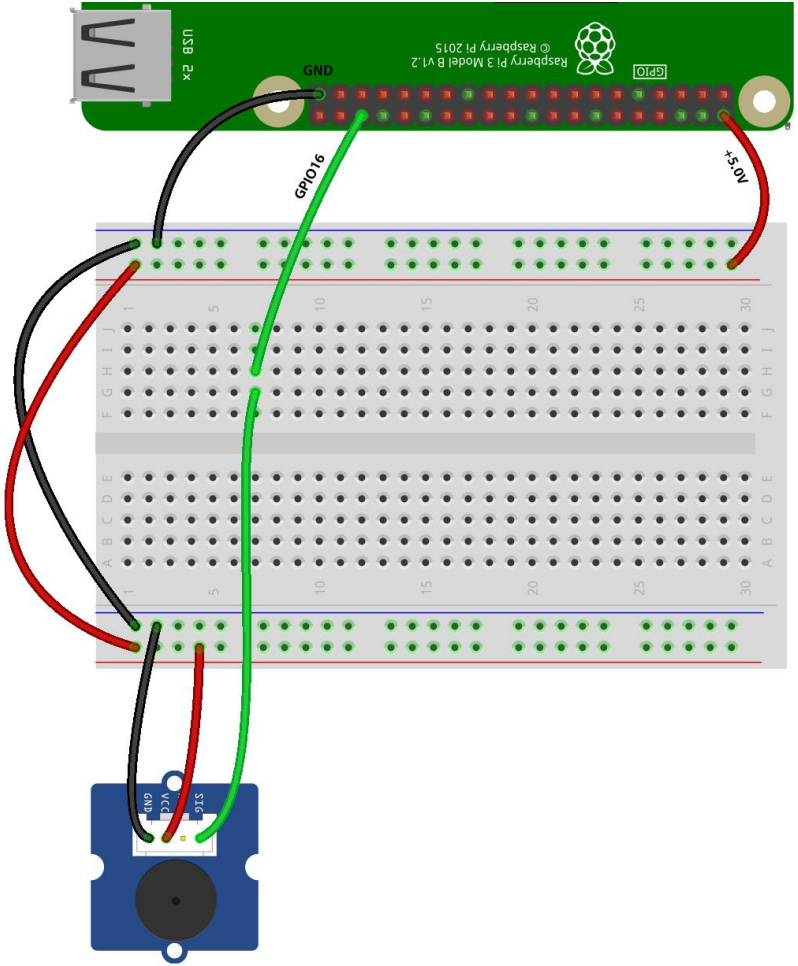
Capacitive Touch Sensor

- Works like a button but opposite
- When touched, your finger breaks the circuit
- Can be used through material
- Touch pads are capacitive sensors

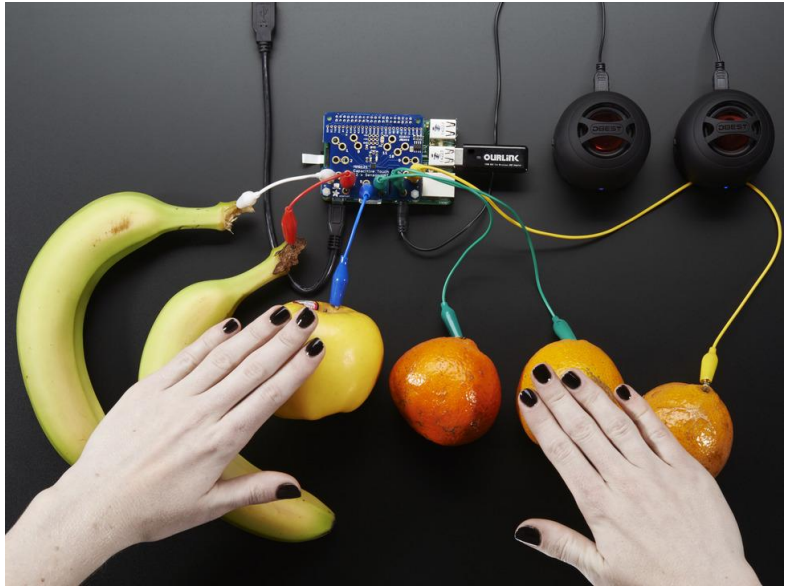


Touch Sensor Circuit

- Connect GND to GND
- Connect SIG to GPIO16
- Connect Vcc to +5.0V
- Signal Output is only 0.3V-0.8V



Turning on an LED with a touch sensor

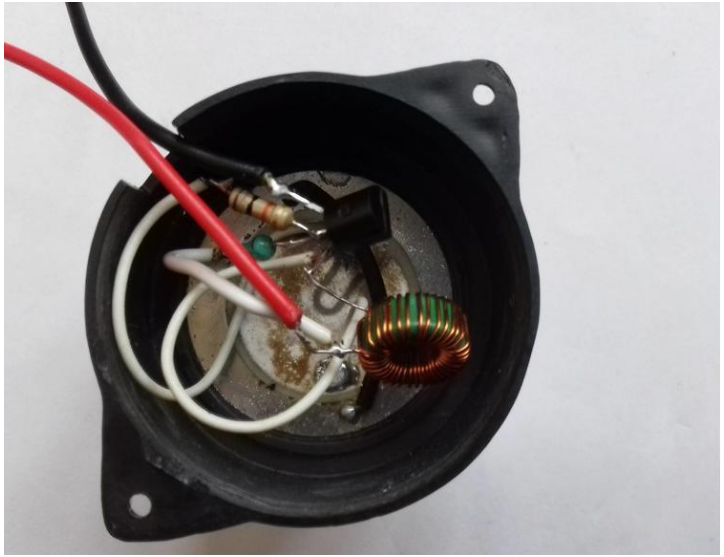



```
from gpiozero import LED, Button
from time import sleep

led = LED(21)
button = Button(16)

button.when_pressed = led.off
button.when_released = led.on
```

Piezoelectric Buzzers

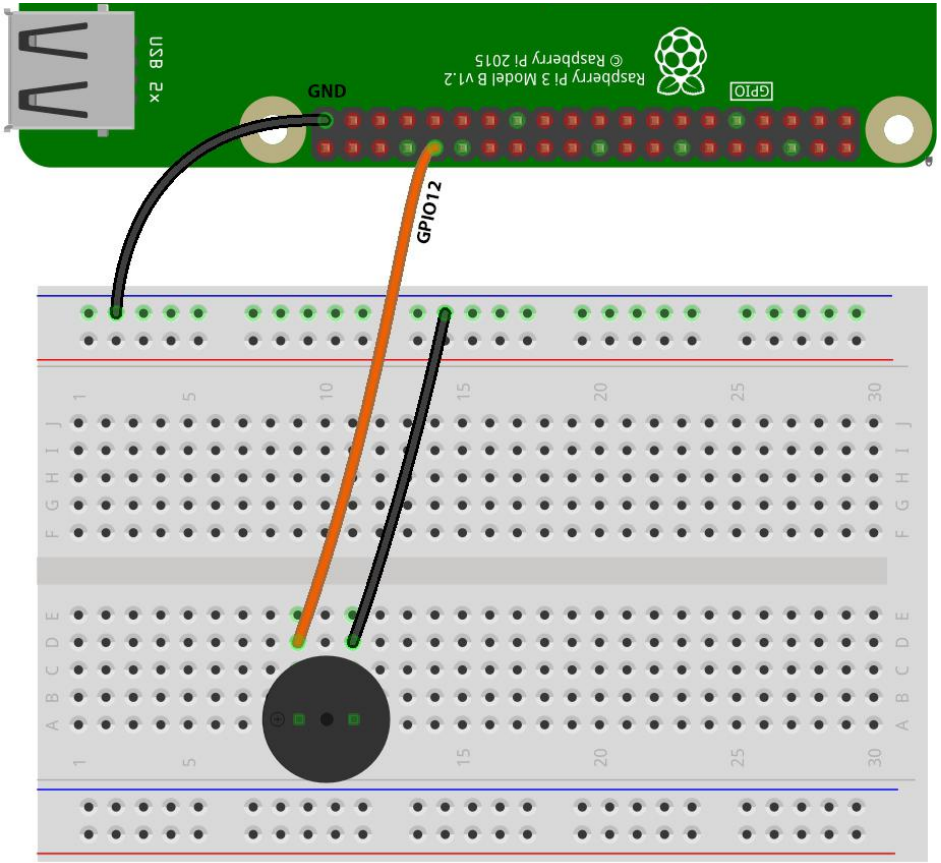
- A crystal sandwiched between two pieces of metal
- The crystal changes shape and vibrates when current is applied to it
- The metal vibrates and projects sound
- Can work backwards as a microphone, and generate electricity

Piezos



Piezo Circuit

- ☐ Disconnect +5.0V connections
- ☐ Connect non +side to GND
- ☐ Connect +side to GPIO12
- ☐ Don’t need a resistor because piezos draw very little current
- ☐ Reconnect +5.0V connections if necessary

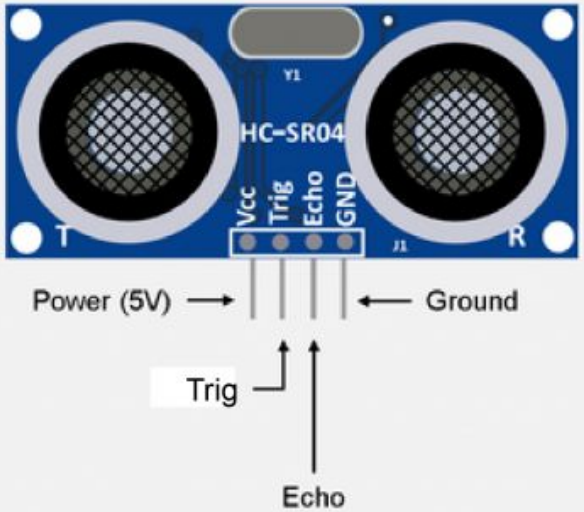


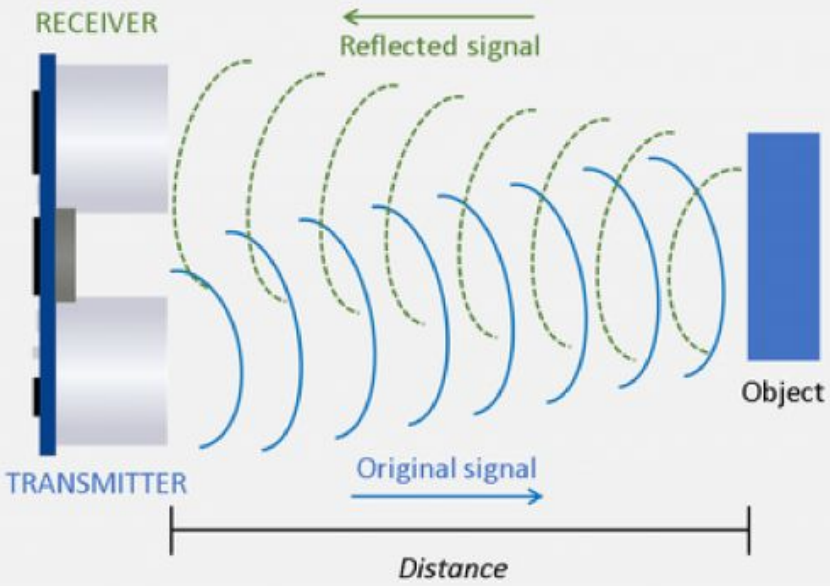
Using our button to buzz

- Running our piezo using the LED function
- gpiozero does have buzzer and tonalbuzzer functions

```
from gpiozero import LED, Button
from time import sleep

led = LED(21)
button = Button(16)
piezo = LED(12)

button.when_pressed = piezo.off
button.when_released = piezo.on
```

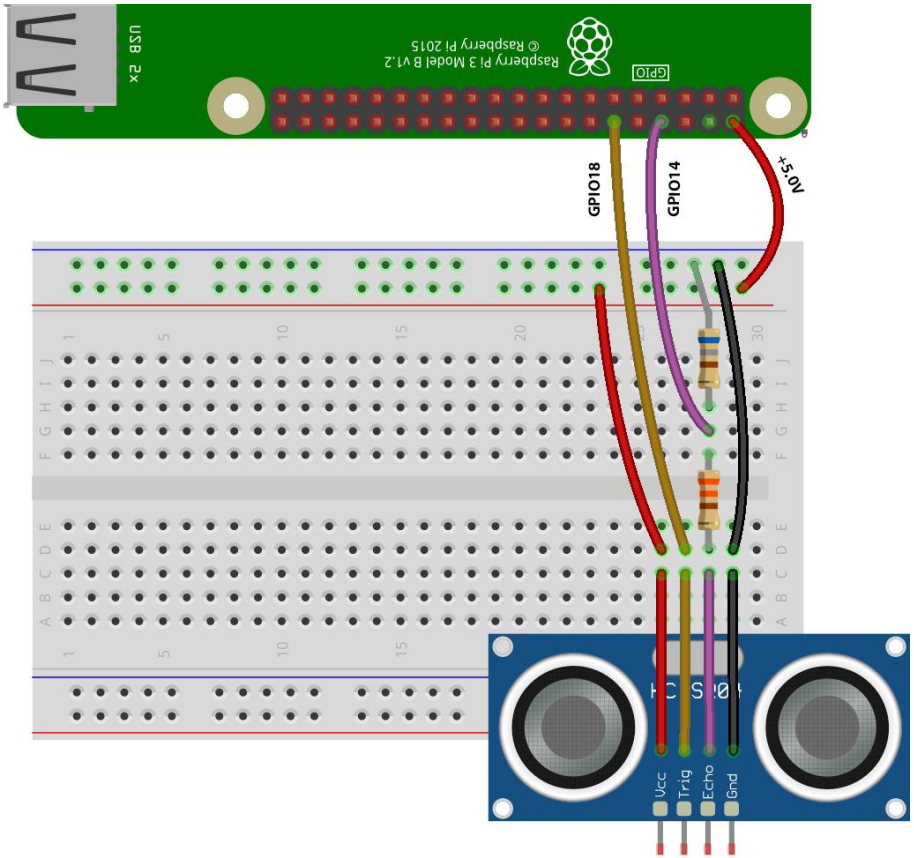
Distance Sensor





Distance Sensor Circuit

- Disconnect +5.0V connections
- Connect GND to GND
- Connect TRIG to GPIO 18
- ECHO outputs 5V, which is too much for the GPIO pins and will fry your board
- Use a voltage divider circuit between ECHO and GPIO14 to decrease to 3.3V
- Use 330Ohm and 680Ohm resistors
- Reconnect +5.0V connections



Printing distance

```
from gpiozero import LED, Button, DistanceSensor
from time import sleep

distanceMeters = DistanceSensor(trigger = 18, echo = 14)

while True:
    distanceInches = distanceMeters.distance * 39.37
    distanceInches = round(distanceInches, 1)

    print('Distance: ', distanceInches, 'in')

    sleep(.1)
```

Turn on LED at distance

`led = LED(21)`	Initializes pin 21 as an LED and defines variable name "led"
`distanceMeters = DistanceSensor(trigger = 18, echo = 14, threshold_distance = 0.15)`	Initializes pins 8 and 11 as a DistanceSensor echo and trigger, respectively, and defines variable name "distanceMeters". Also sets threshold distance to 0.15m (~6in)
`distanceMeters.when_in_range = led.on`	When distance sensed is less than threshold distance, turn on led
`distanceMeters.when_out_of_range = led.off`	When distance sensed is greater than threshold distance, turn off led

Turn on LED and Piezo at distances

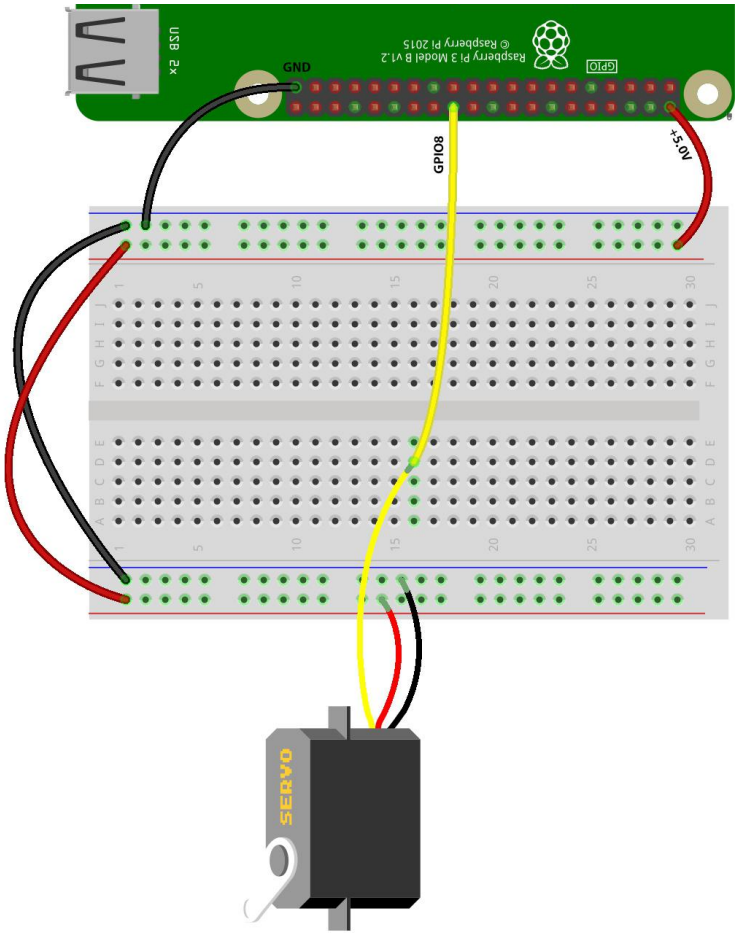
`piezo = LED(12)`	Initializes pin 12 as an LED and defines variable name "piezo", we'll use the LED functions to run our piezo
`    if distanceInches <= 3:`	If distance is less than or equals to 3 inches, run the indented code below, otherwise skip to the indented code below else
`        piezo.on()`	Turn piezo on
`    else:`	If if statement is not satisfied, run the indented code below
`        piezo.off()`	Turn piezo off

Servo

- A motor that allows you to control its angular position
- Uses PWM Pulse Wave Modulation signal to go between max and min positions
- Similar but different from a stepper motor

Servo Circuit

- [ ] Disconnect +5.0V connections
- [ ] Connect brown wire to GND
- [ ] Connect orange wire to GPIO8
- [ ] Connect red wire to 5V rail
- [ ] Reconnect +5.0V connections



Using the button to go from servo min to max

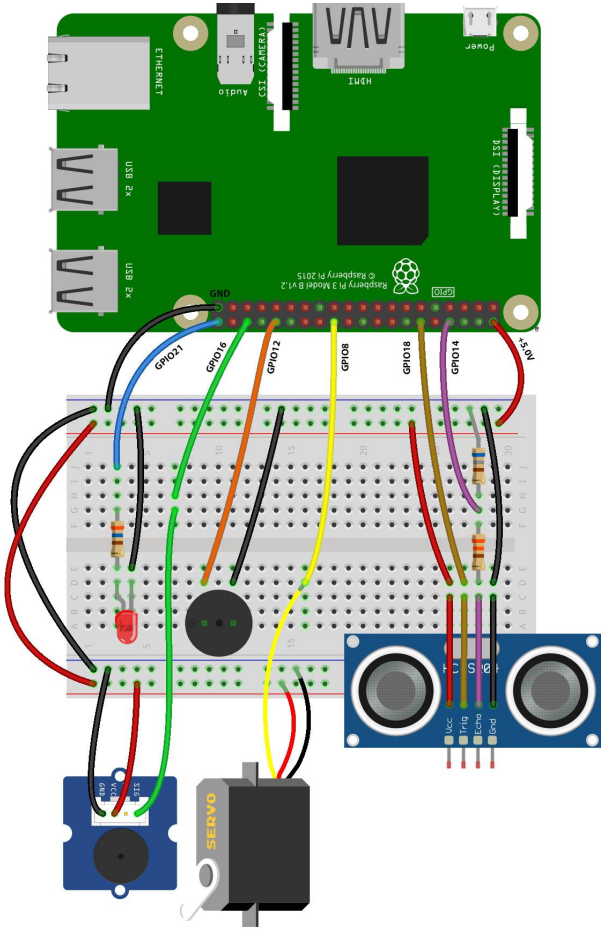
```
from gpiozero import LED, Button, DistanceSensor, Servo
from time import sleep

button = Button(16)
servo = Servo(8)

while True:
    if button.is_pressed == False:
        servo.max()
    else: indented code below
        servo.min()
```

Hello world servo wave

```
from gpiozero import LED, Button, DistanceSensor, Servo
from time import sleep

button = Button(16)
servo = Servo(8)

while True:
    if button.is_pressed == False:
        servo.max()
        sleep(.5)
        servo.min()
        sleep(.5)

    else: indented code below
        servo.mid()
```

Jam Session

- Mix up your code and try different combos
- Translate your pre-class poem
- Explore the gpiozero library for more possibilities
- Communicate with your neighbor’s device



Wrap Up

- Share your device
- What about physical computing or working on your device did you like
- What did you find difficult or frustrating?
- Any new ideas?