

Physical Computing

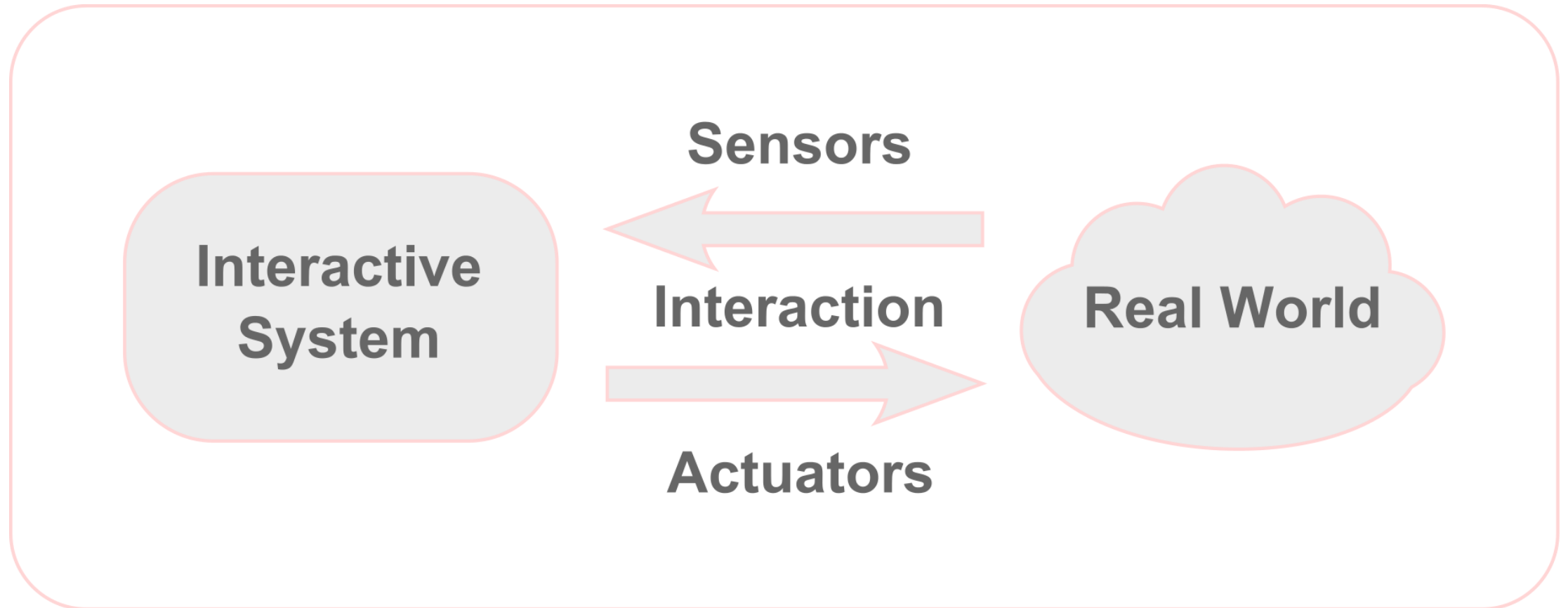


Agenda

- Physical Computing
- Electricity and Circuits 101
- Electronic Components
- Programming with the GPIO
- Lab Scenario (SMTP)
- PiCamera – Images and Streaming

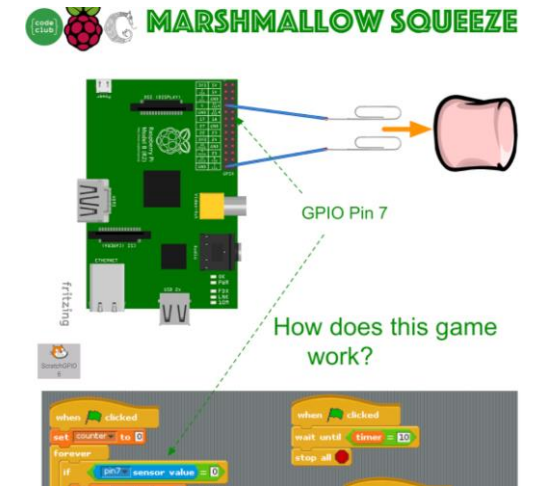
Physical Computing

“A creative framework for understanding human beings' relationship to the world”



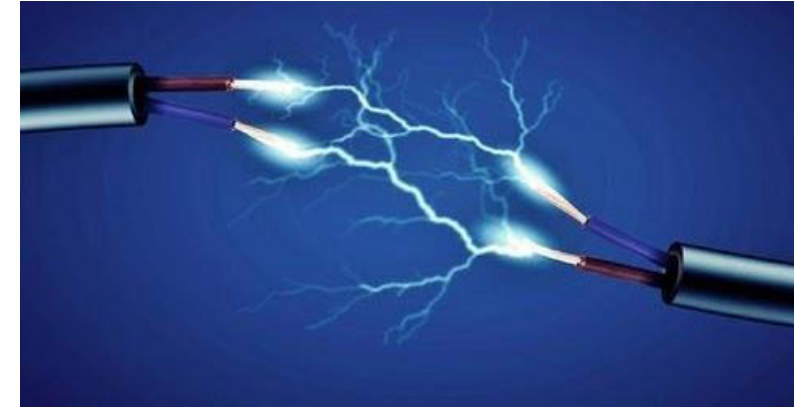
Physical Computing

- Break away from conventional input/output peripherals:
 - Keyboard, mouse, touch screen
- Think about how human/environment signals can be captured and changed into electronic signals that can be interpreted by a computing device.
- Physical computing applications tend to depend on people for input (and sometimes output), and transform that input into another form, like an animation, a sound, or motion.
- Sometimes powerfully fuses art and technology.



Electricity

- **Electricity** is the flow of electrical energy through some conductive material.
- **Sensors** are components that convert other forms of energy into electrical energy so we can read the changes in those other forms.
 - Transduction (e.g. microphone)
- **Voltage** is a measure of the difference in electrical potential energy between two points in a circuit. It is measured in **Volts**.
- **Current** is a measure of the magnitude of the flow of electrons through a particular point in a circuit. It is measured in **Amperes**, or **Amps**.
- **Resistance** is a measure of a material's ability to oppose the flow of electricity. It is measured in **Ohms**.



Electricity: Relationship

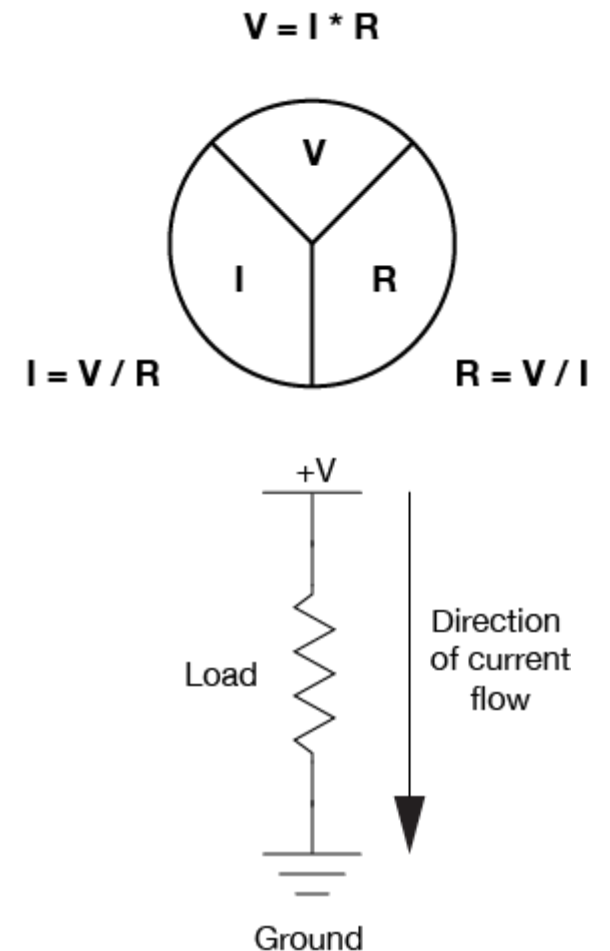
- **Voltage (V), Current (I), and Resistance (R)** are all related, by the following formula:

$$V = I \times R$$

- **electrical power (P)** (measured in watts), as follows:

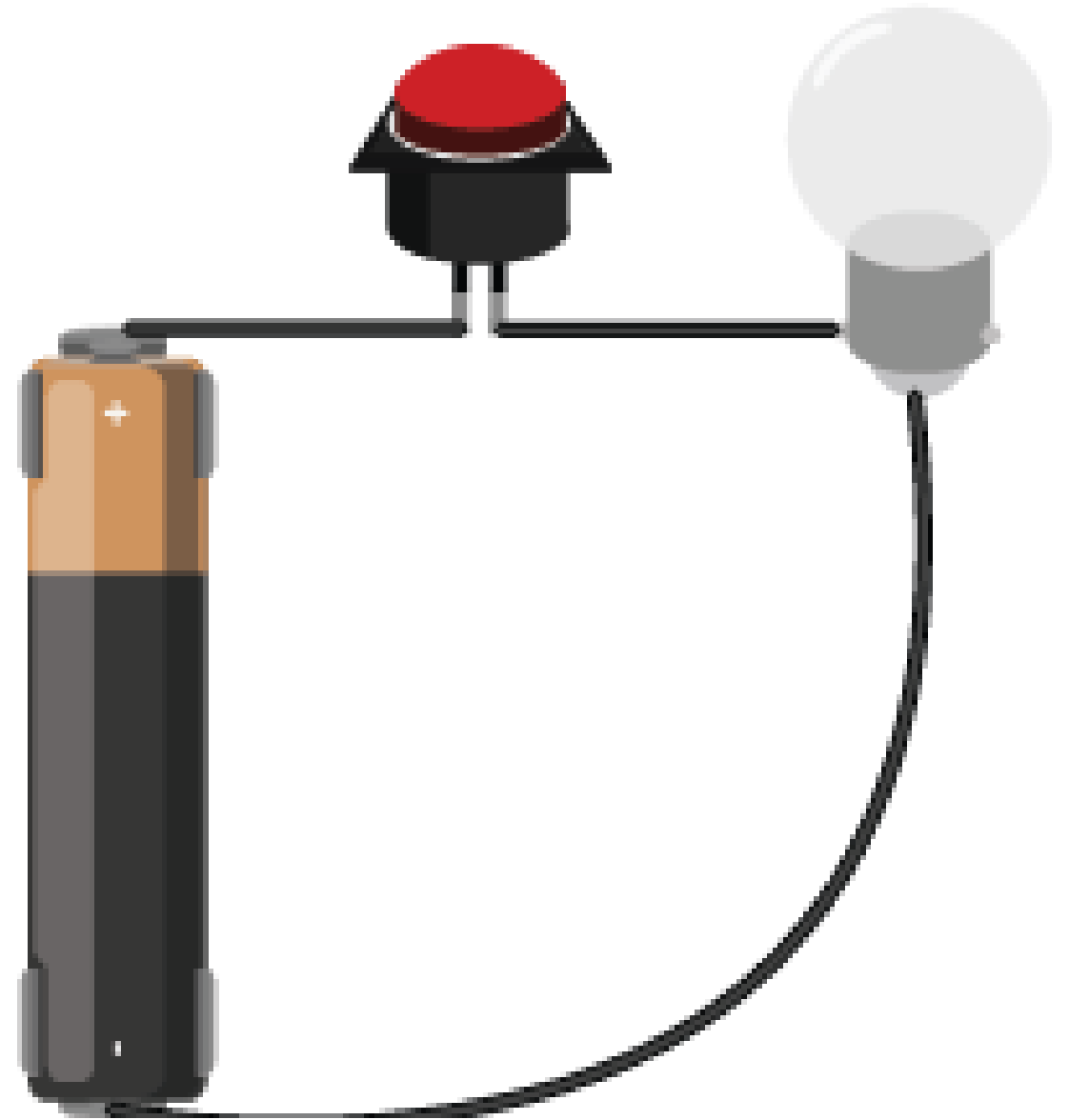
$$\text{Watts} = \text{Volts} * \text{Amps}$$

$$P = V * I$$



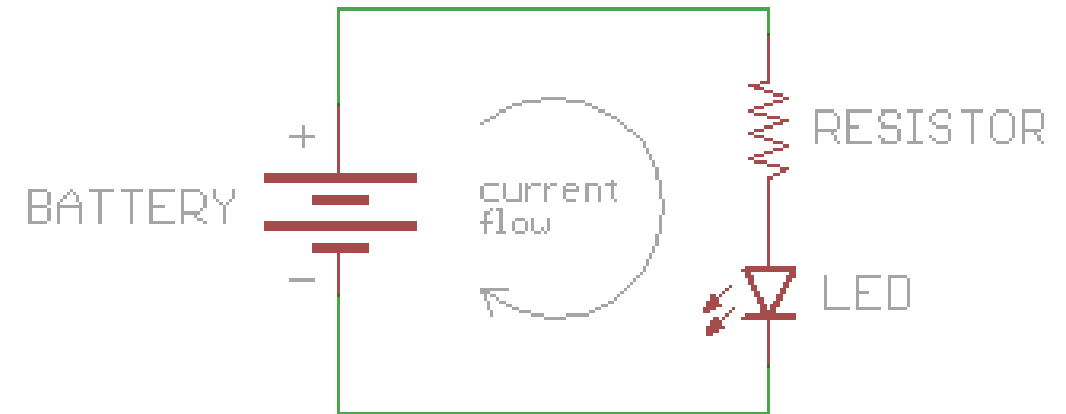
Circuits 1

- Physical Computing usually involves interfacing a computing device with electrical circuit(s).
- A circuit needs a:
 - Electrical Energy Source (e.g. battery)
 - Load (Converts elec energy to something else)



Circuits 2

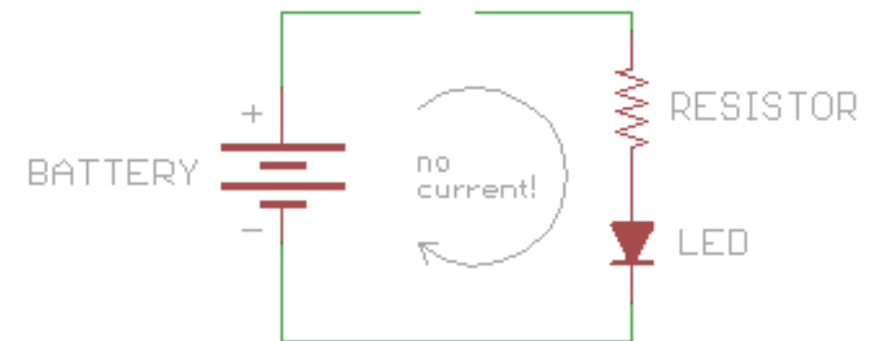
- Electricity needs to flow to do anything useful.
 - Needs a path to flow through, which must be an electrical conductor (like copper)
- Electricity (Current) will flow from a higher voltage (+) to a lower voltage (-) or ground.



Short Circuit - Bad



oops!

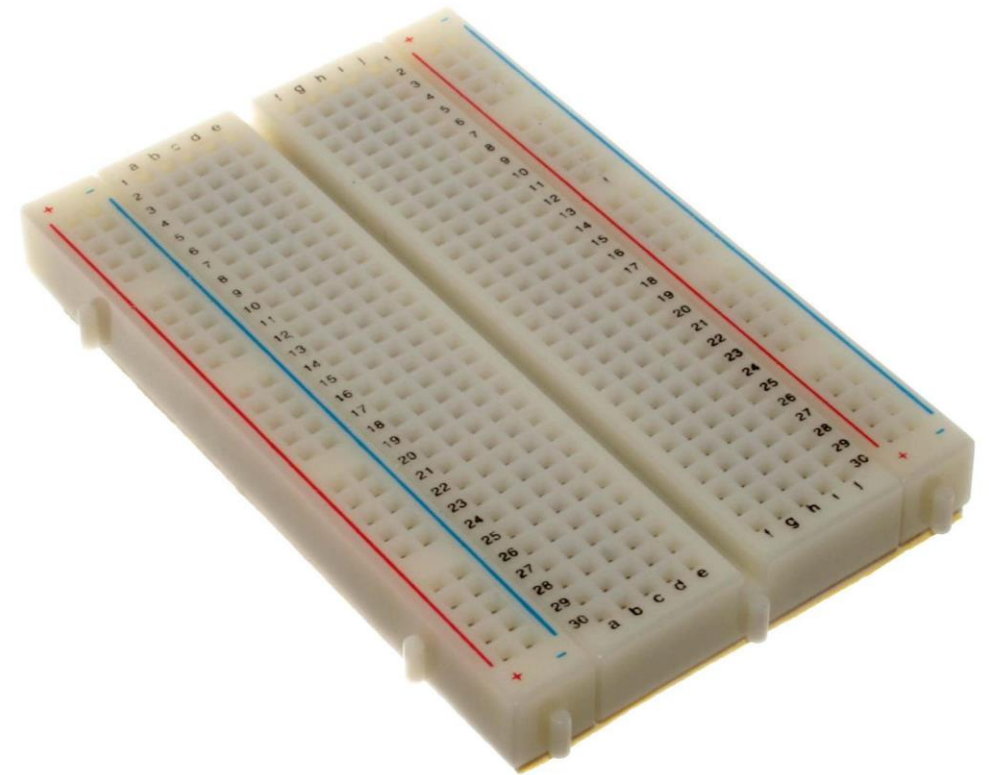
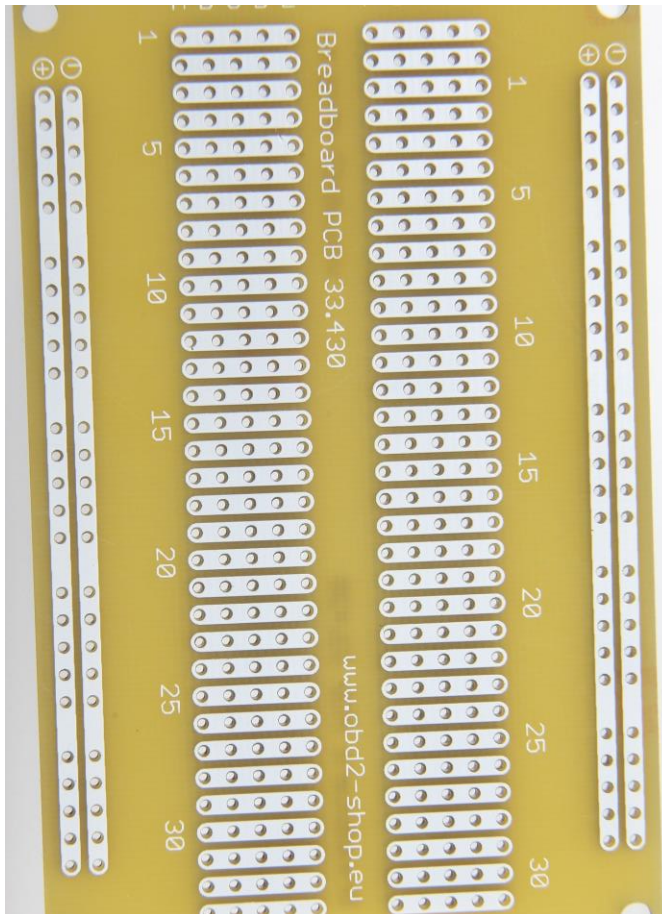




Electronic Components

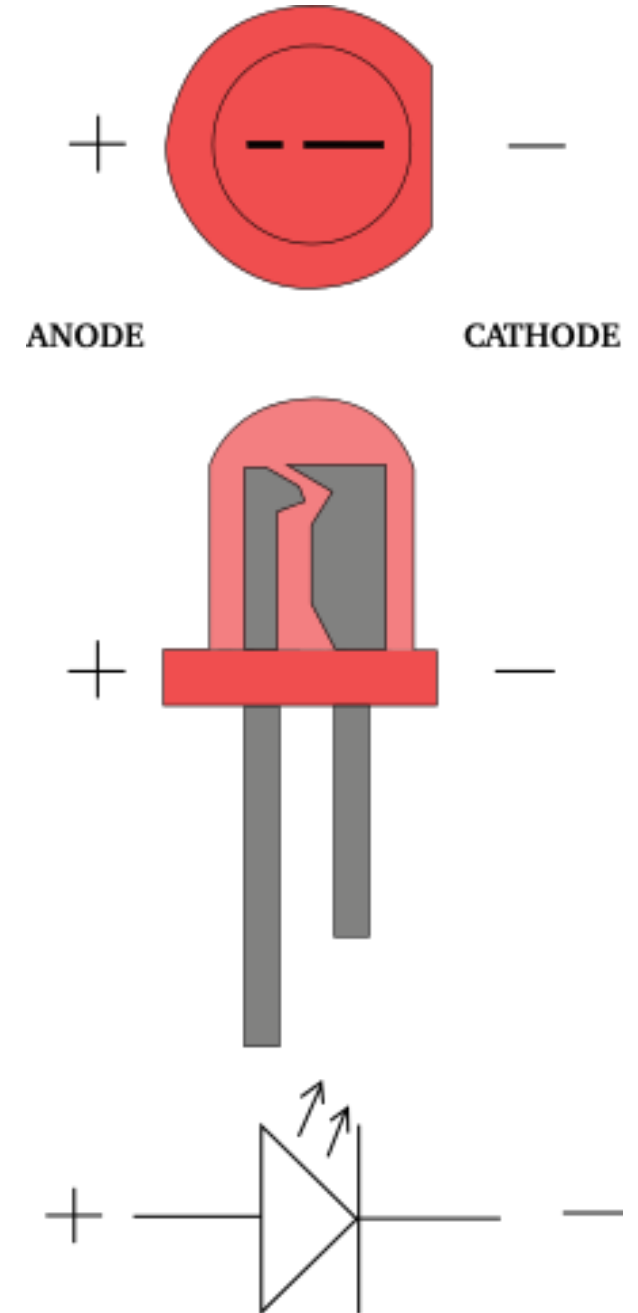
Electronics: Breadboard

Used for prototyping electronic solutions



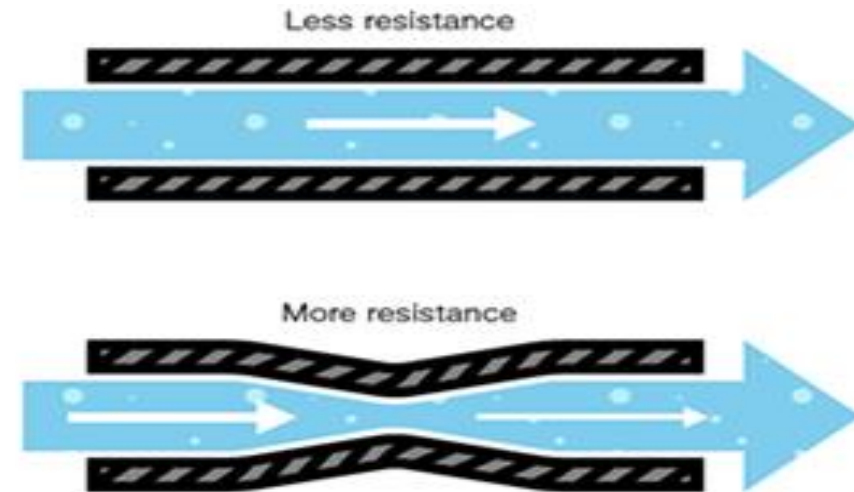
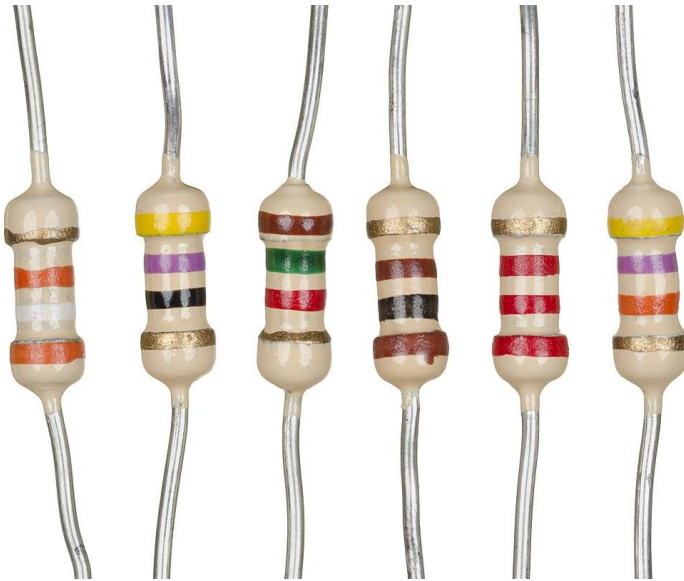
LEDs

- Emits light when current passes through
- Typical LED requires 2V at 20mA
- RGB LEDs, LED strips, IR LEDs, Ultrabright



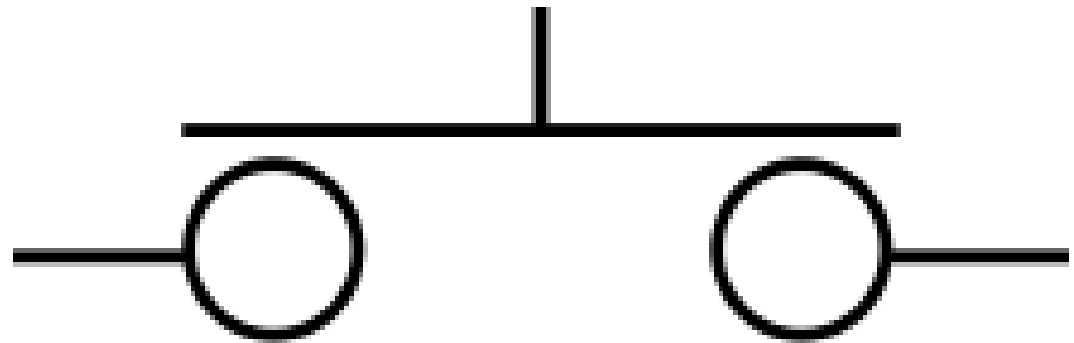
Resistors

- Reduce current and voltage to components
- Ohm Law: $I = V/R$



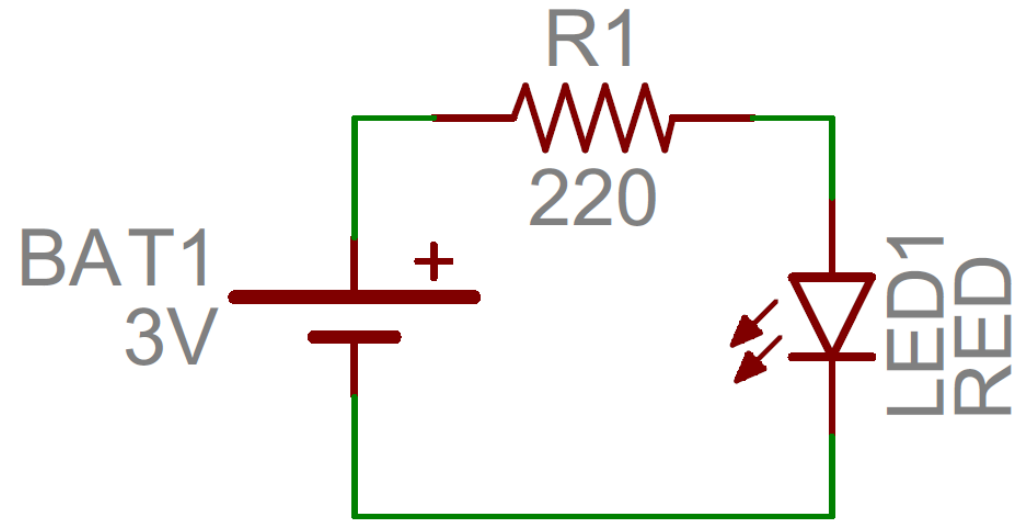
Push Button

- **pushbuttons** control the flow of current through a junction in a circuit



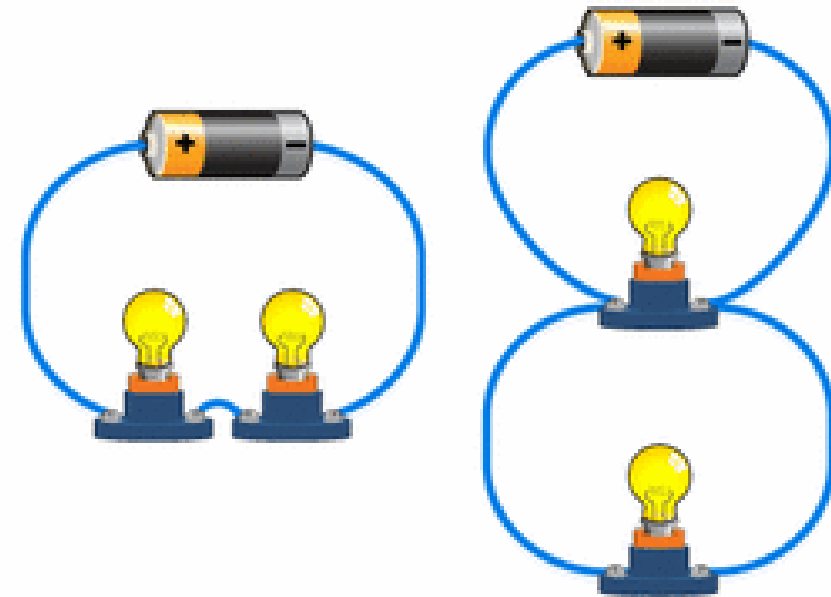
Circuit Schematics

- Circuit Schematic illustrate you how components are connected in a circuit.
- For more info follow this [link](#)

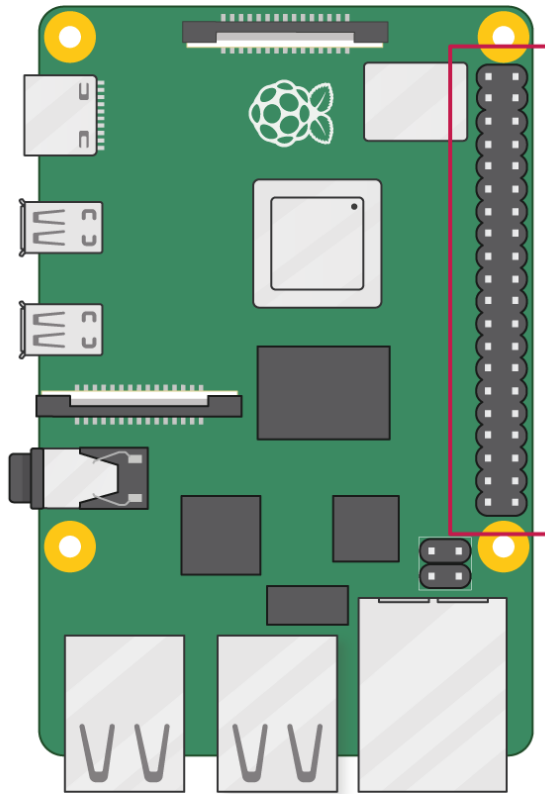


Electrical Flow

- Components can be arranged in series/parallel
- Current tends to follow the path of least resistance to the ground
- In any given circuit, the total voltage around the path of the circuit is zero
- The amount of current going into any point in a circuit is the same as the amount coming out of that point.



Physical Computing with Rpi: GPIO



3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

Physical Computing with RPi: Build Circuit

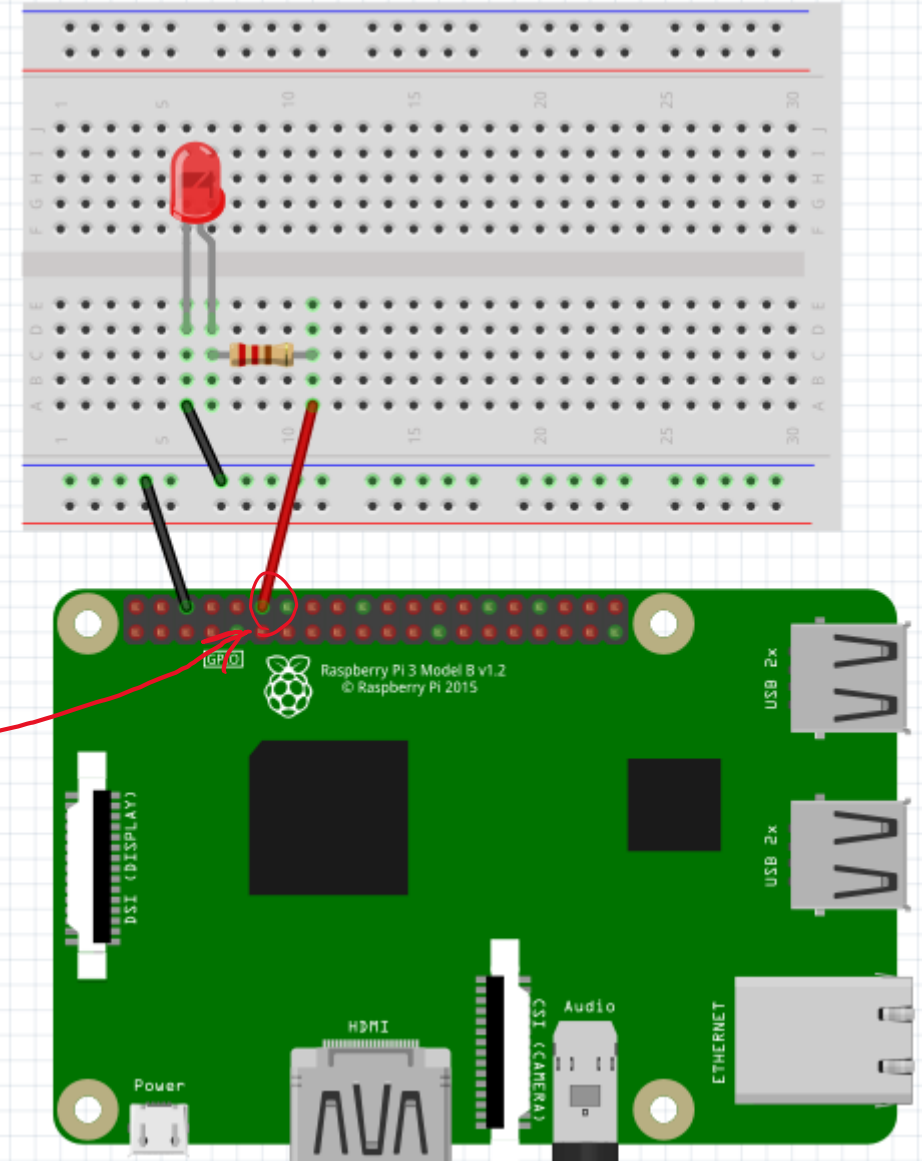
- Use GPIO Headers to create circuit
- Write program to Read/Write to GPIO headers

```
from gpiozero import LED
from time import sleep

red = LED(18)

while True:
    red.on()
    sleep(1)
    red.off()
    sleep(1)
```

GPIO 18



Physical Computing with RPi: GPIOZero

- Python API for physical computing
- Accessible and intuitive:
 - Interfaces for common physical computing components

GPIOZero

- Multi paradigm (programming):

- Polling
- Blocking
- Callbacks
- Declarative

```
while True:
    if button.is_pressed:
        led.on()
    else:
        led.off()
```

```
while True:
    button.wait_for_press()
    led.on()
    button.wait_for_release()
    led.off()
```

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

```
led.source = button.values
```

```
from gpiozero import LED, Button
```

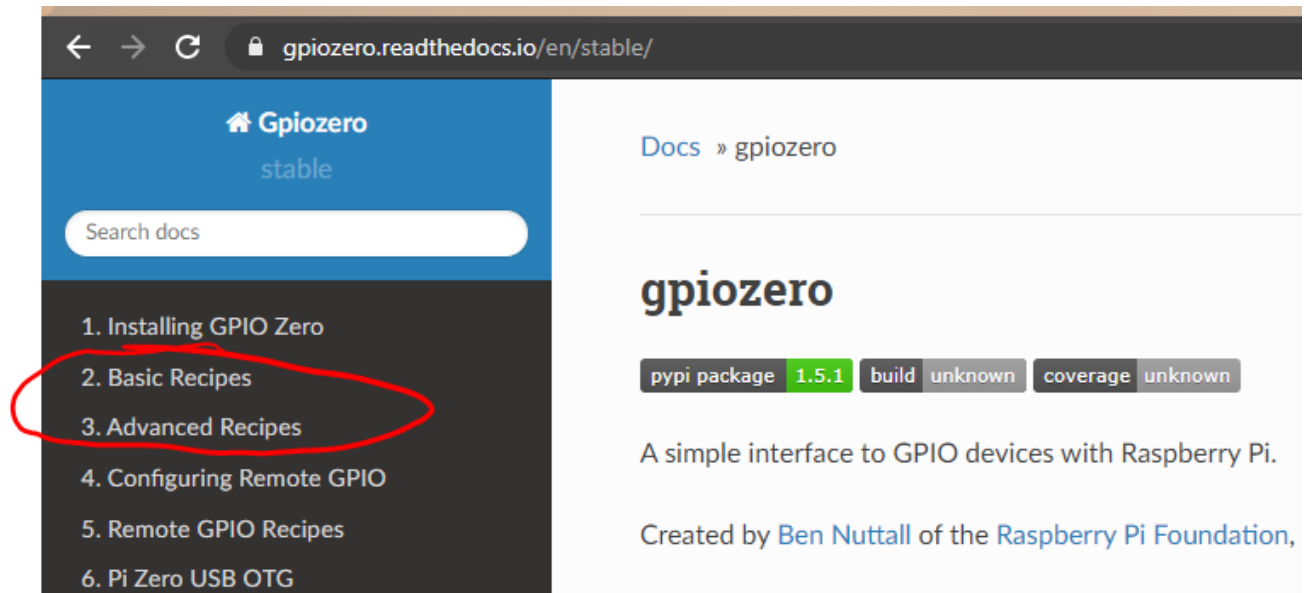
```
led = LED(17)
button = Button(4)
```

```
led.source = button.values
```

GPIO: Supported Devices

- LED (also PWM LED allowing change of brightness)
- RGB LED
- Buzzer
- Motor
- Button
- Motion Sensor
- Light Sensor
- Analogue-to-Digital converters MCP3004 and MCP3008

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



Physical Computing Lab: RPi Doorbell

- Abstract Idea: Send email on button push event
- Technologies:
 - Button and Breadboard connected to GPIO
 - PiCamera
 - SMTP server/email service to send email
- Use Case
 1. Button pushed
 2. PiCamera takes photo and stores on Rpi
 3. Image attached to email and sent using SMTP
 4. Recipient accesses image using mail client

