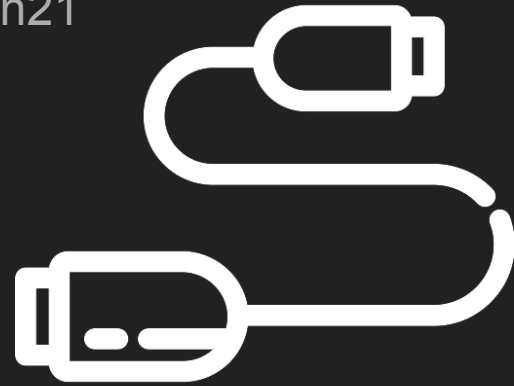


Physical Computing

with Casey Hunt

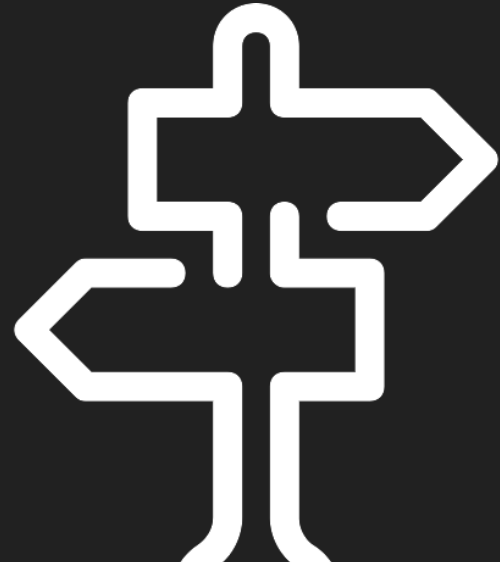
Before we start

1. Install Arduino IDE (if you haven't yet) - <https://www.arduino.cc/en/software>
2. Find a micro USB cable
3. Get your kit
4. <https://github.com/caseyhunt/Creative-Tech-Jan21>



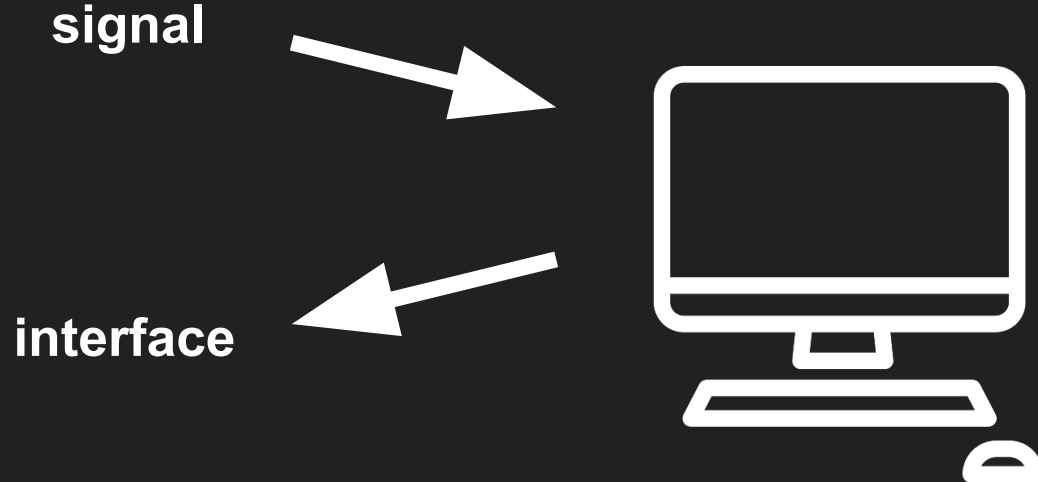
Where we're going

1. What is Physical Computing?
2. Circuit Basics
3. Board Tour
4. Tutorial: Moisture Sensor



What is Physical Computing?

Physical computing refers to computer systems that **sense** and **respond** to the physical world around them.



Arduinos are Microcontrollers

Microcontrollers are small computers that are dedicated to a single task.

Microcontrollers are good for low complexity processes and use little power

Microcontrollers can connect to many inputs/outputs

Microcontrollers consist of a CPU and a memory (usually RAM and ROM)

Microcontrollers do not have an operating system

RasPis are Single Board Computers

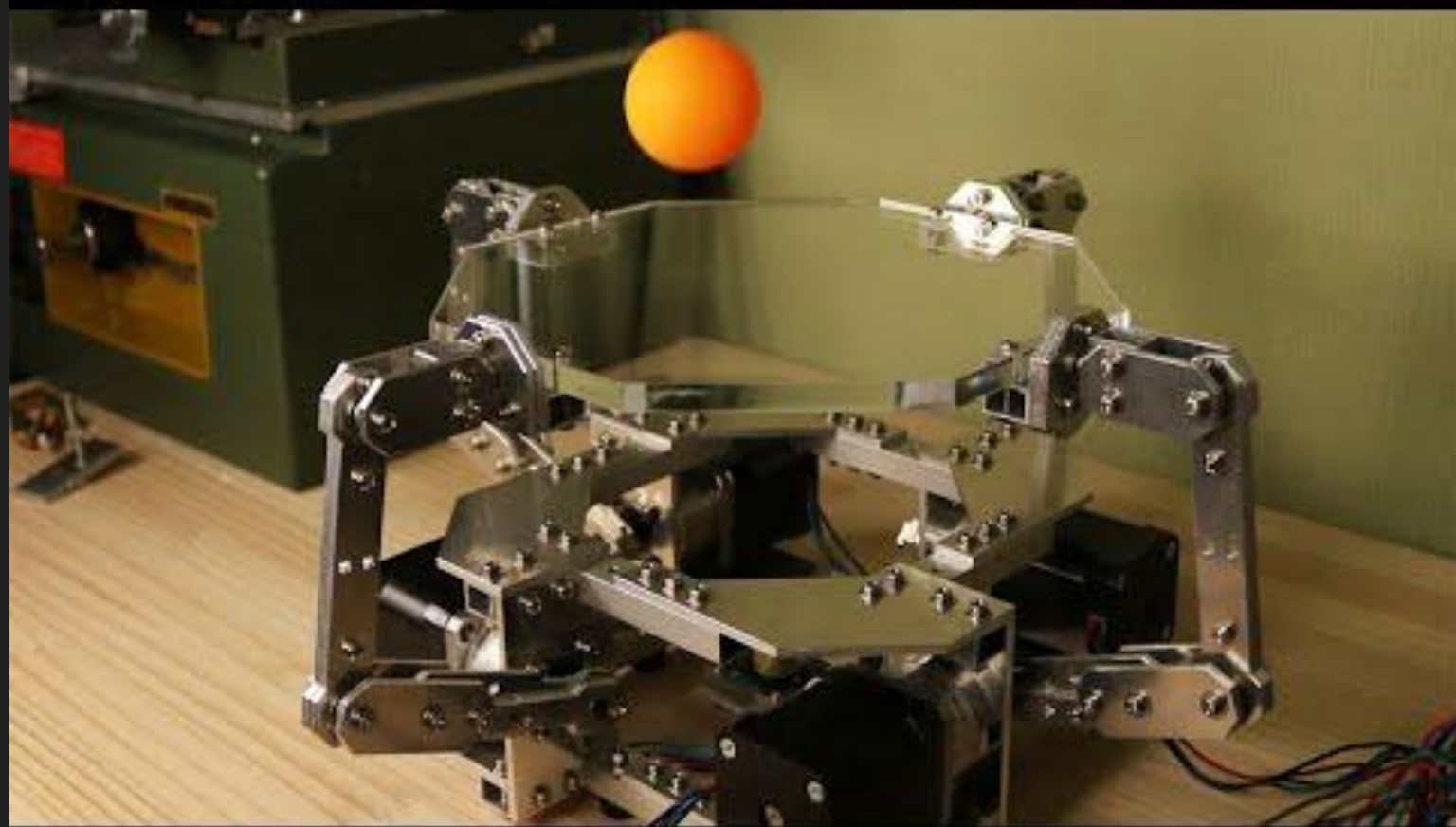
SBCs are meant to run many processes at one time

SBCs usually have more onboard features like WiFi and Bluetooth

SBCs use more power than microcontrollers (but less than larger CPUs)

SBCs run operating systems and can easily install drivers to run USB hardware like displays and webcams

Arduinos are Prototyping Machines



TFT LCD Touch Screen



M How To
MECHATRONICS
www.HowToMechatronics.com

Tutorial



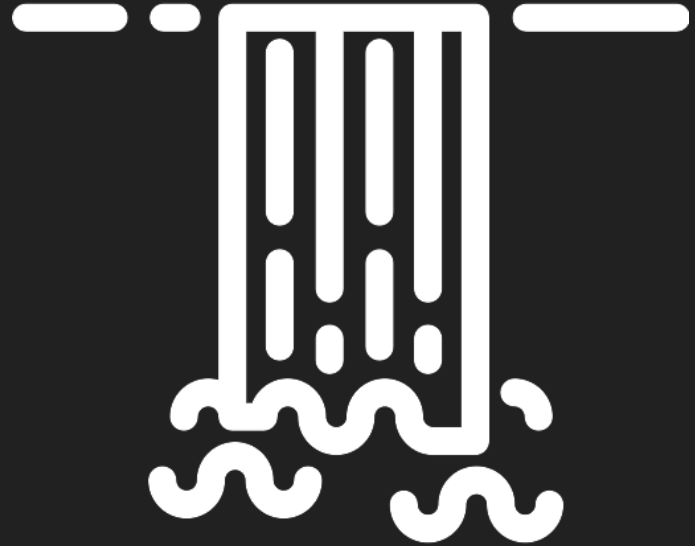
Circuit Building Basics

DC Current is Unidirectional

Power flows from power to ground

Voltage (V) = the potential energy of the flow

Current (A) = the speed of the flow



Always Connect Ground First

Ground is usually brown, green, or black

Power is usually red



Short Circuits

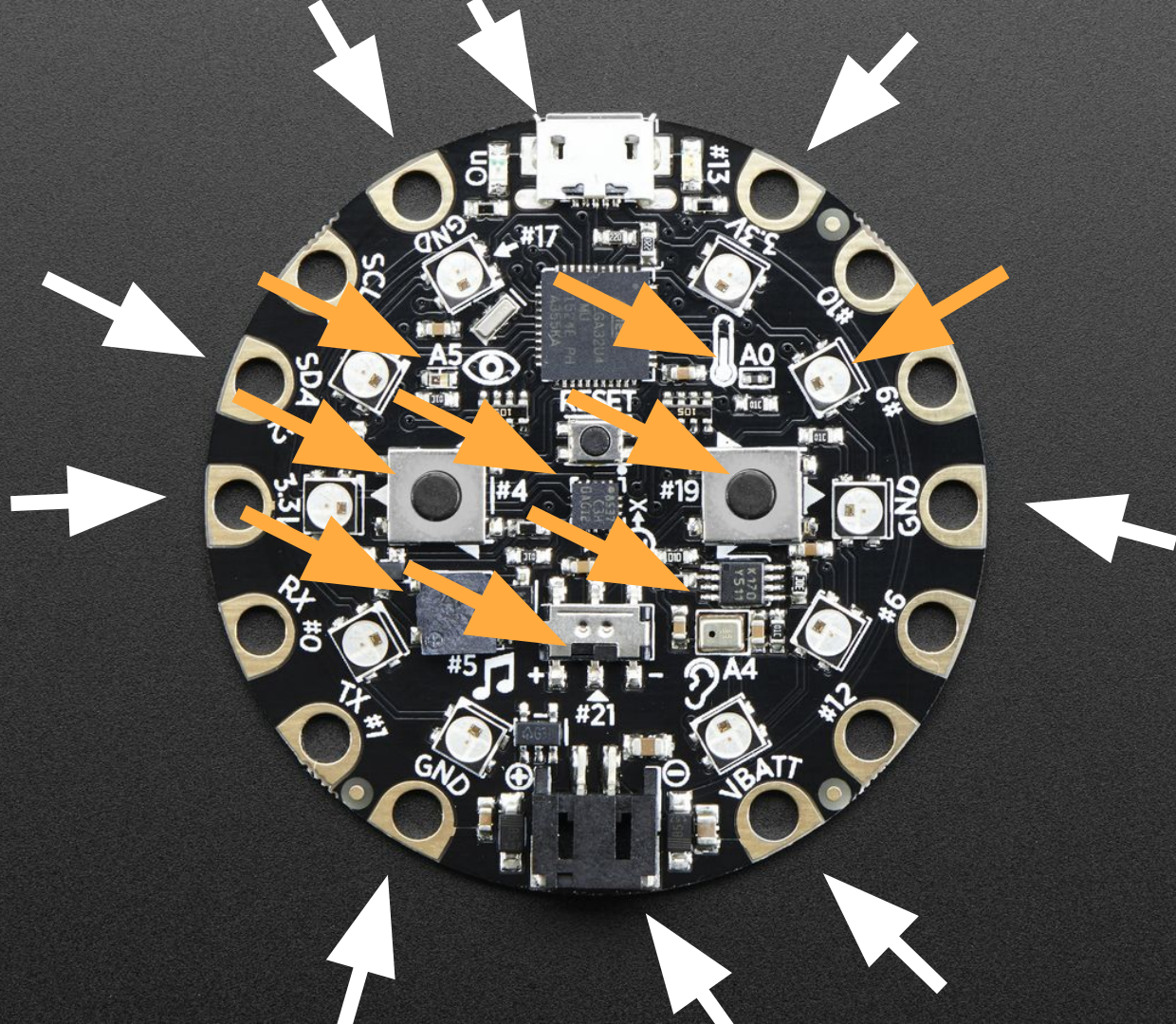
Shorts occur when power is connected directly to ground

Shorts happen when there is no voltage drop across a connection

Shorts can fry a board or cause a system to overheat and catch fire

Prevent short circuits by adding resistance to your system

Board Tour



Arduino IDE uses Java

MakeCode uses block code or JavaScript: <https://makecode.adafruit.com/>

CircuitPython uses Python: <https://learn.adafruit.com/welcome-to-circuitpython/>

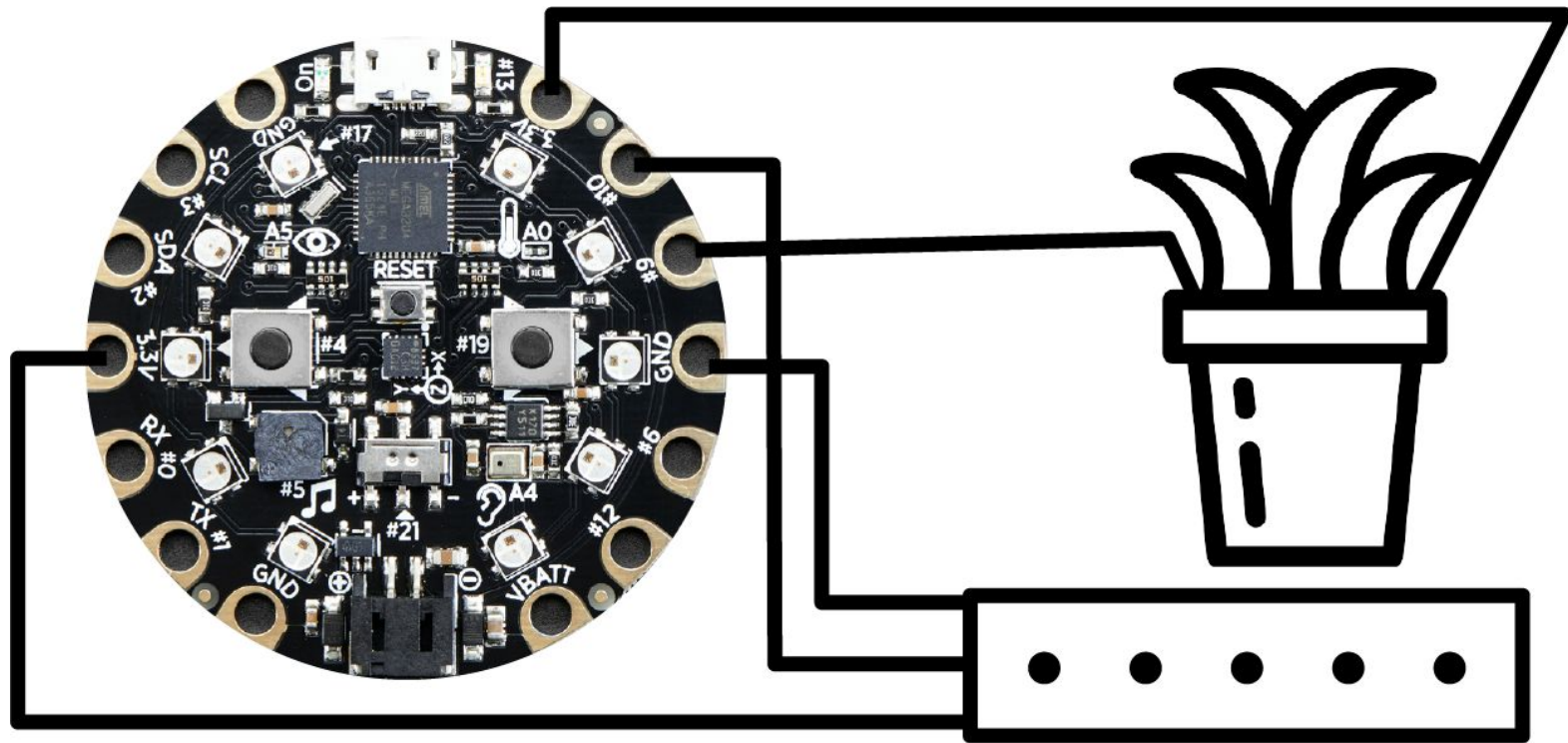
Demo Time!

Before we start

Take a break!

1. Get a plant or glass of water
2. Get a paper towel
3. Optional: something conductive for sensing nodes





Design Challenge

More Technical:

Explore one of the sensors on the Circuit Playground. Adapt example code to make a smart home object or game.

More Designed:

Improve on the plant sensor design, maybe try adding a servo to water your plant?
Focus on the aesthetic/user experience of this object