Making sense of sensors

Fiore Basile

FABRICADEMY

textile and technology academy

Required materials

Option 1 - Buy it yourself or reuse arduino kits

Arduino Compatible Board
(any brand will work, but Leonardo can act as a keyboard)
LDR light sensor
(photoresistor)
Temperature sensor (i.e. TMP36)
Adafruit LIS3DH Triple-Axis Accelerometer (or similar)
NeoPixels

Option 2 - Flora Kits

FLORA Sensor Pack
LDR light sensor
NeoPixels

Option 3 - All in one
Cheaper & recommended

Adafruit Circuit playground

FABRICADEMY
textile and technology academy

What is a sensor?

Anything that can measure some natural phenomena

FABRICADEMY
textile and technology academy

Many different kind of sensors

Sensors can be classified based on:

- Which **natural quantity** is measured
- The measurement **precision**
- The kind of output: **analog** vs **digital**

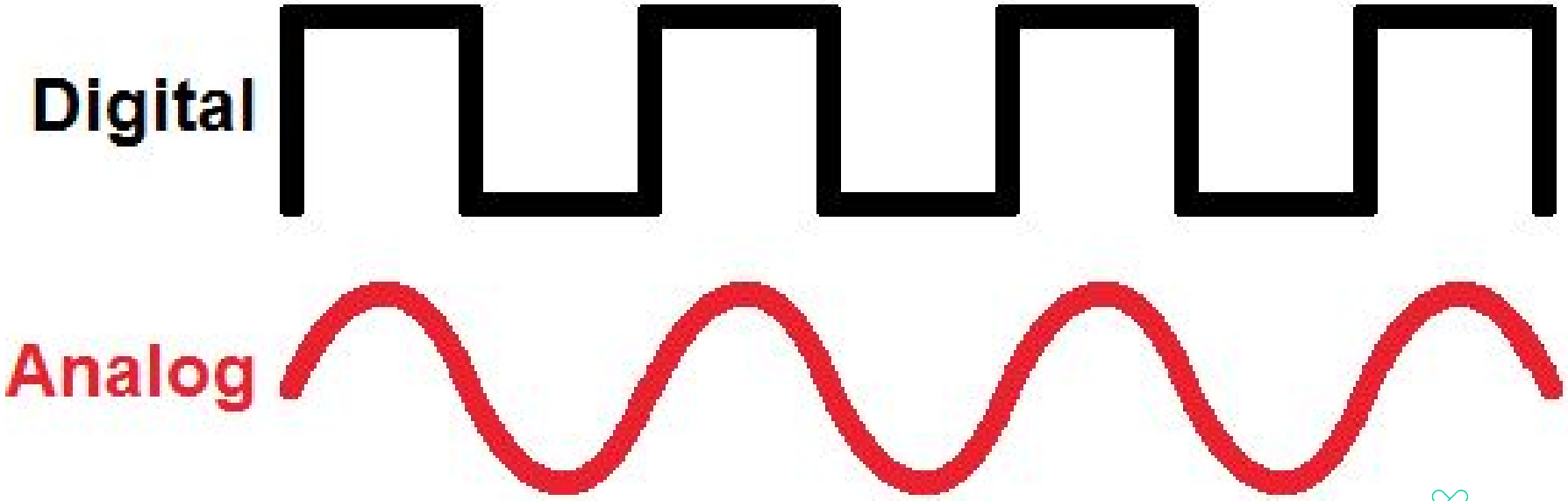
FABRICADEMY
textile and technology academy

Sensor types

- **Acoustic**, **sound**, **vibration**: microphone, etc
- **Automotive**: speed, air flow, knock
- **Chemical**: gas, etc
- **Electricity**, Fluids Flow, Radiation
- **Environment**: pressure, temperature, moisture, etc
- **Navigation**: gyroscope, compass, altitute
- **Position**: capacitive, accelerometer, photoelectric, rotary encoders, PIR, flexure, stretch, etc
- **Force**, **Proximity**, etc

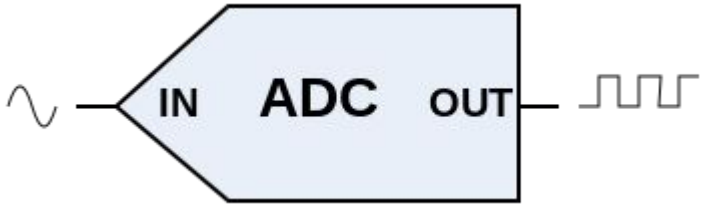
FABRICADEMY
textile and technology academy

Analog vs Digital Sensors



FABRICADEMY
textile and technology academy

Analog to digital conversion



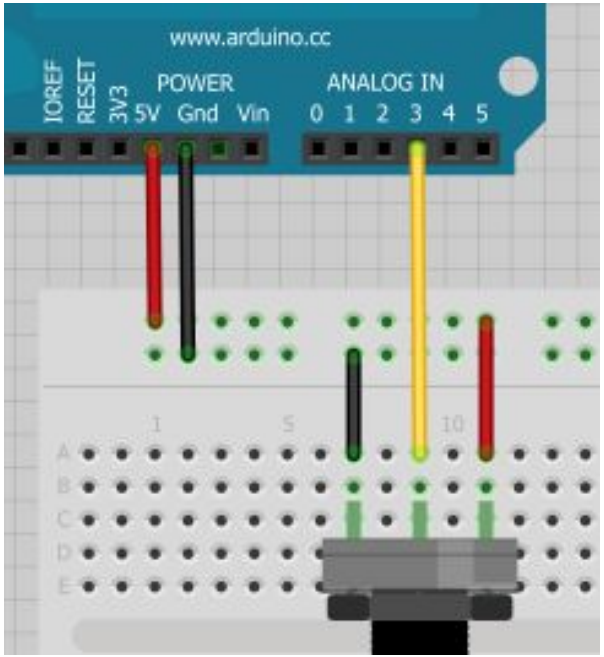
Analog to Digital Converter (ADC) is a part of almost any micro-controller.

It helps you to **convert analog inputs** (coming from a sensor) to **a digital stream** of 0 and 1.

It does this comparing the Voltage read on a PIN with the AREF (analog reference voltage)

FABRICADEMY
textile and technology academy

Analog sensor with Arduino: an example




```
// Set the PIN A3 as input
pinMode(A3, INPUT);
// Read in a value
int x = analogRead(A3);
// Print it to the serial console
Serial.println("Analog Reading")
Serial.println(x)
// will be a number from 0 to 1023
// representing the voltage we read
```

FABRICADEMY
textile and technology academy

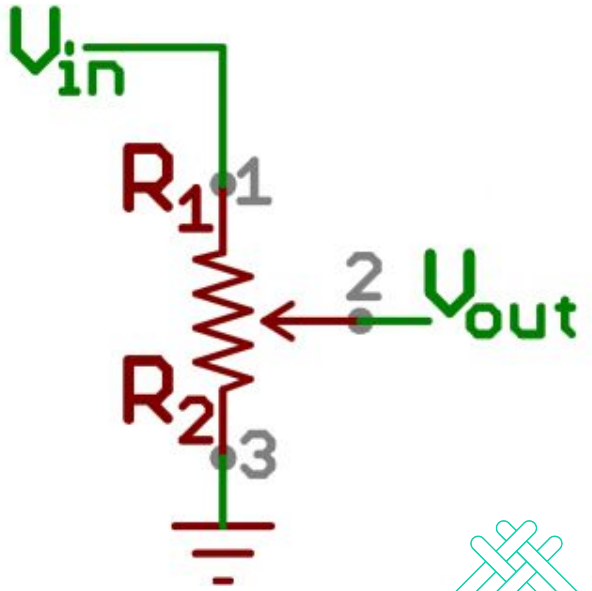
Resistive sensors: voltage divider

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

The ADC can only measure voltage, many sensors are Resistive.

Resistance is limiting the amount of current passing in a circuit.

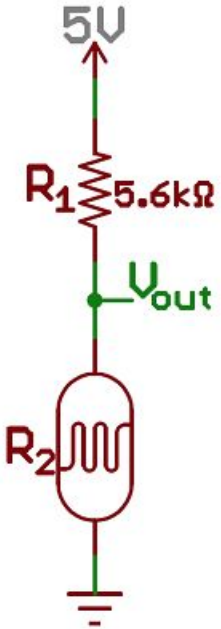
We can measure resistance using a **voltage divider**



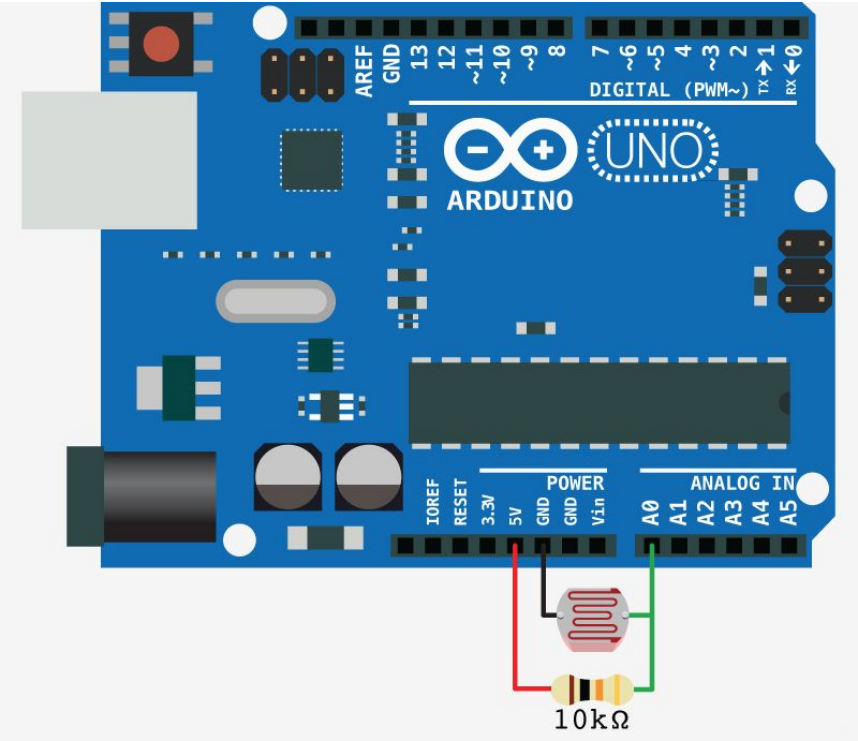
FABRICADEMY
textile and technology academy

Voltage divider example

We use a **photoresistor** which has a variable resistance based on the amount of light it is exposed



Light Level	R_2 (Sensor)	R_1 (Fixed)	Ratio $R_2/(R_1+R_2)$	V_{out}
Light	1kΩ	5.6kΩ	0.15	0.76 V
Dim	7kΩ	5.6kΩ	0.56	2.78 V
Dark	10kΩ	5.6kΩ	0.67	3.21 V



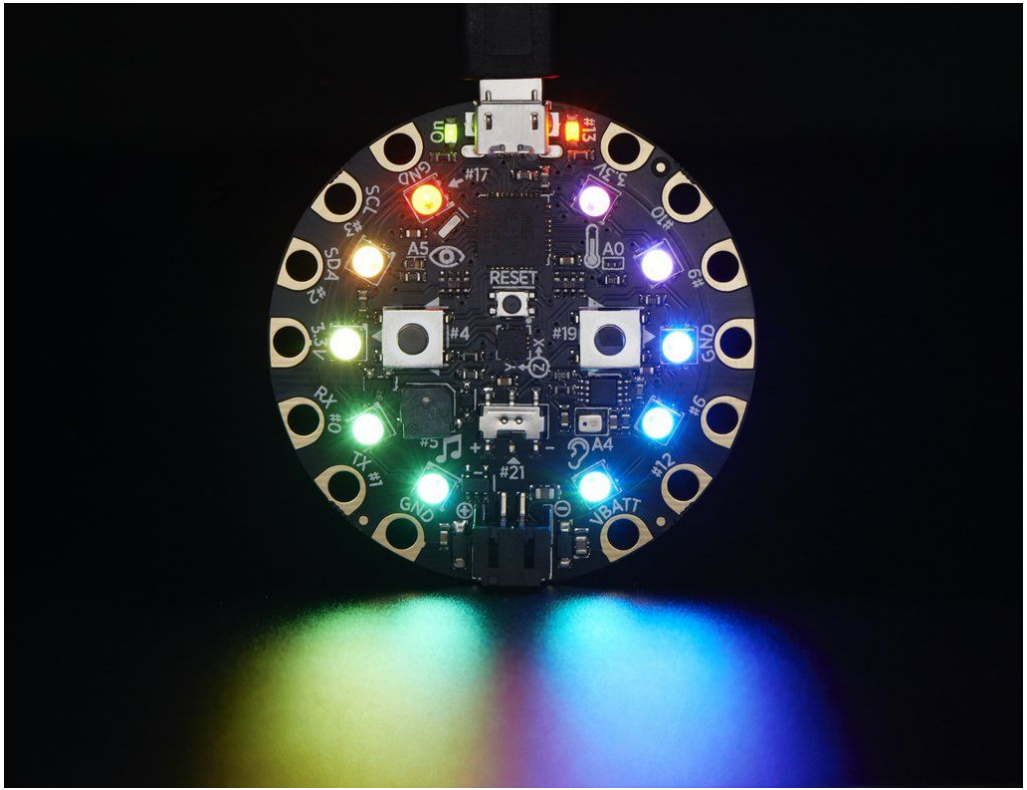

```
int LDR_Pin = A0; //analog pin 0

void setup(){
    Serial.begin(9600);
}

void loop(){
    int LDRReading = analogRead(LDR_Pin);
    Serial.println(LDRReading);
    delay(250);
}
```

Wiring & Arduino Code

Exercise:
Print “Dark” when lights go down,
Print “Light” when sensor is in bright light,
Print “Shade” otherwise

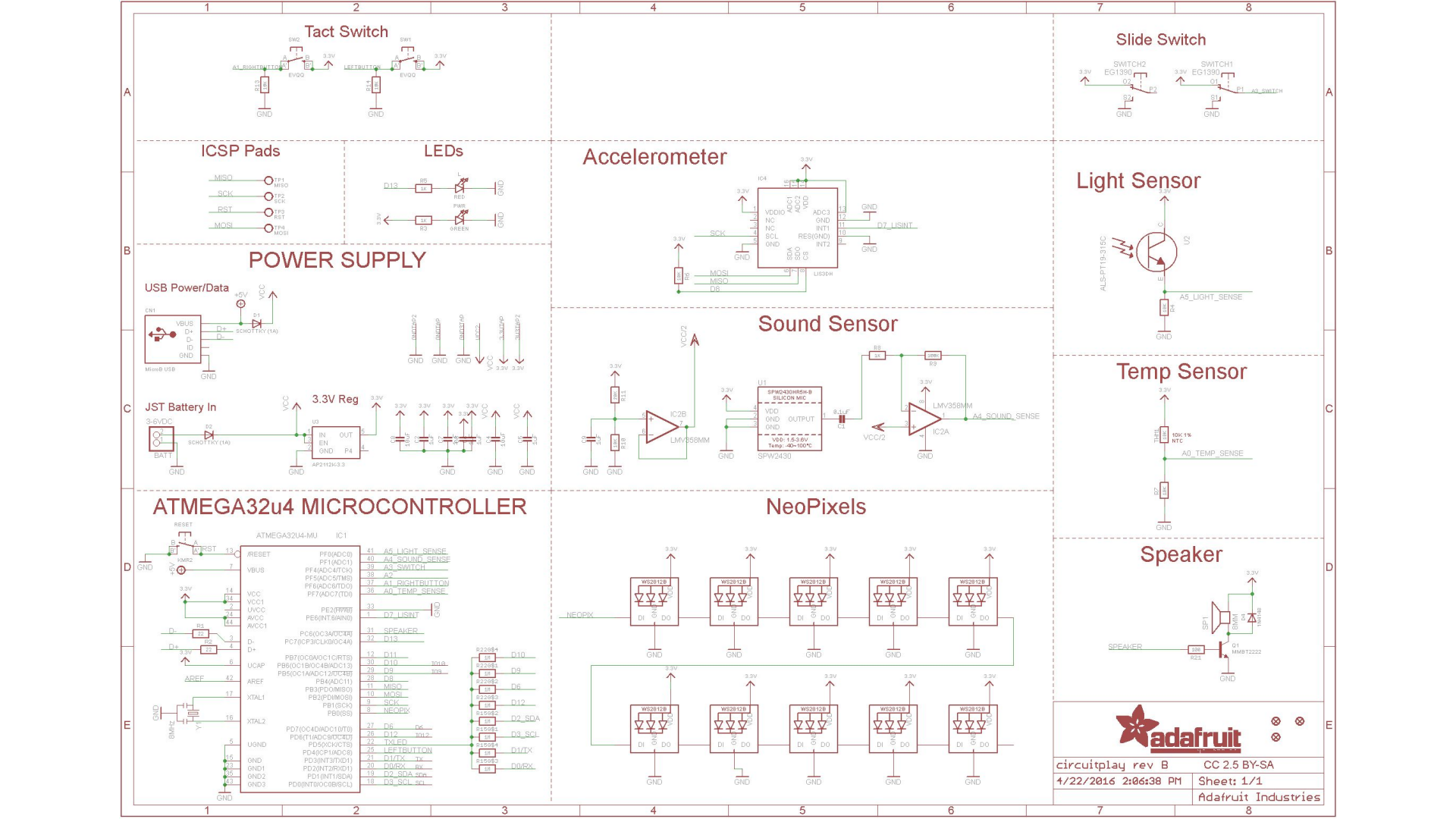
Adafruit Circuit Playground

Windows Drivers Here!

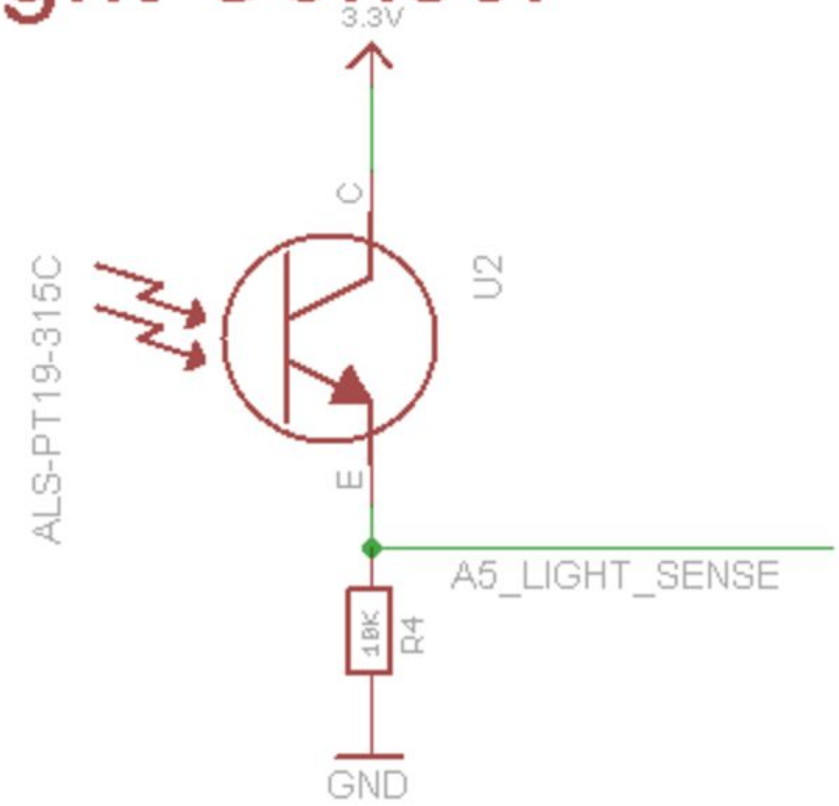


Arduino - compatible board, with many sensors integrated, no soldering required!

Accelerometer
Light sensor
Temperature
Microphone
Buzzer
RGB LEDs
Buttons

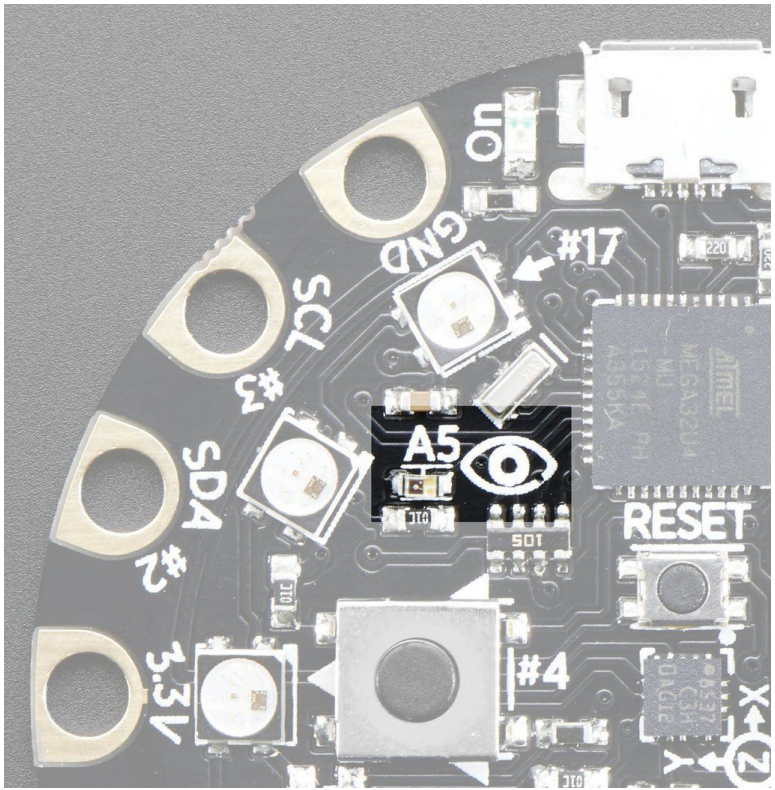

Tact Switch
Slide Switch
ICSP Pads
LEDs
Accelerometer
Light Sensor
POWER SUPPLY
USB Power/Data
JST Battery In
3.3V Reg
Sound Sensor
Temp Sensor
ATMEGA32u4 MICROCONTROLLER
NeoPixels
Speaker
adafruit
circuitplay rev B
CC 2.5 BY-SA
4/22/2016 2:06:38 PM
Sheet: 1/1
Adafruit Industries

Light Sensor



On the Circuit Playground schematics you can see that this **sensor is connected to analog pin #A5**

Light sensor



This sensor is connected to analog pin #A5

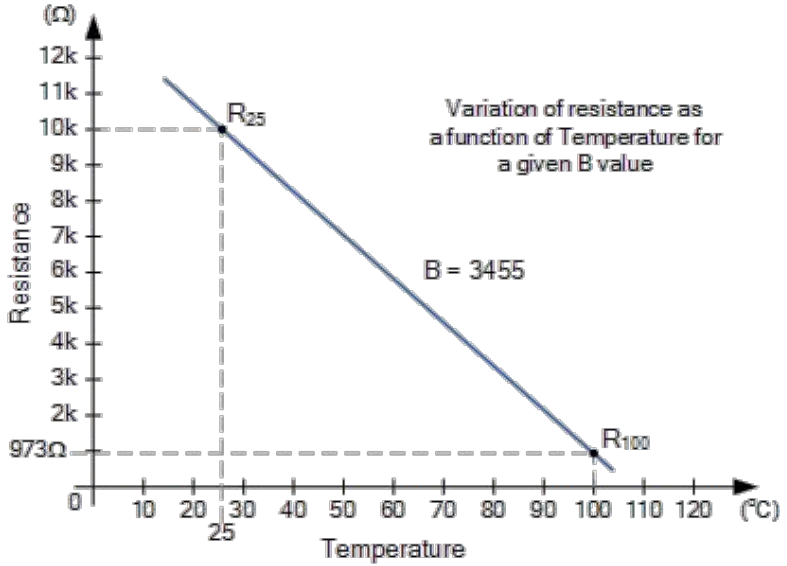
Will read between 0 and 1023 with higher values corresponding to higher light levels.

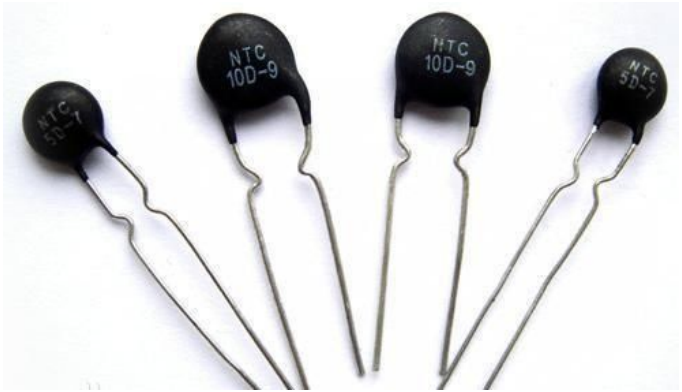
A reading of about 300 is common for most indoor light levels.

Sensing temperature

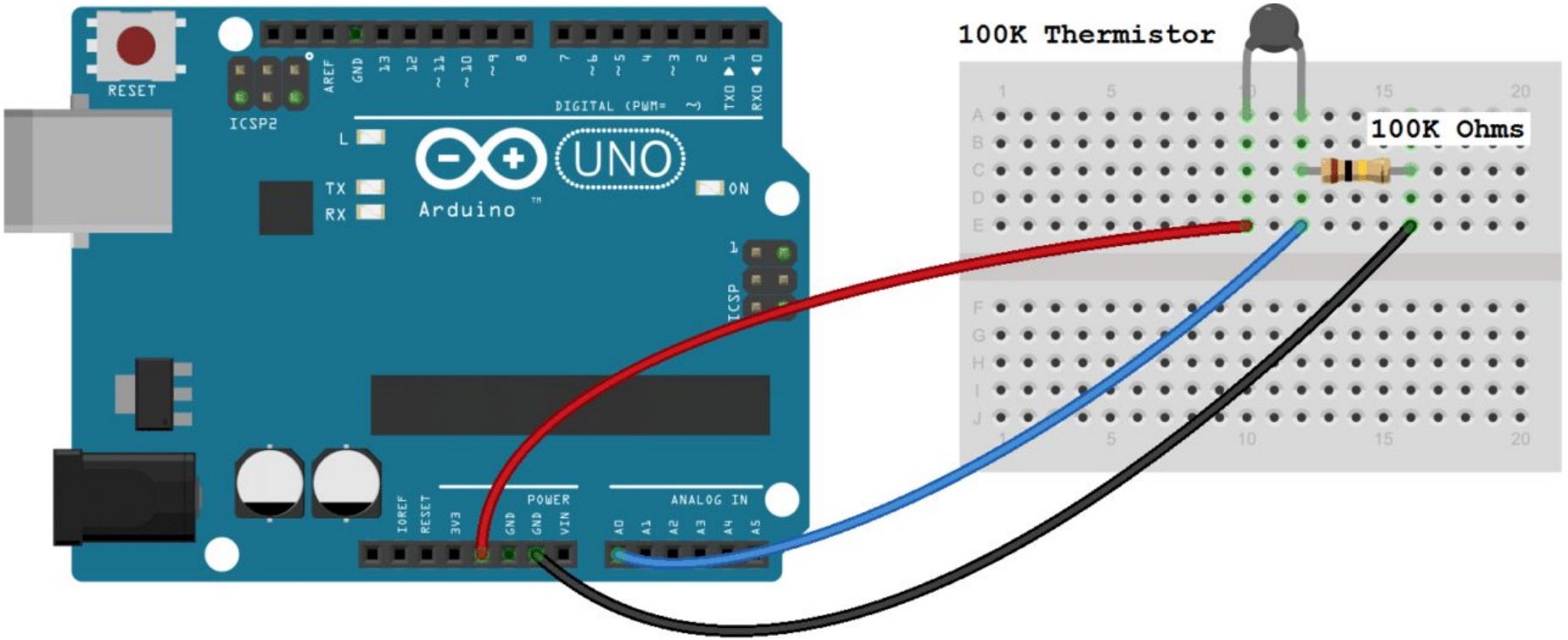
A thermistor is a resistor that changes it's resistance based on the temperature

The higher the temperature the lower the resistance

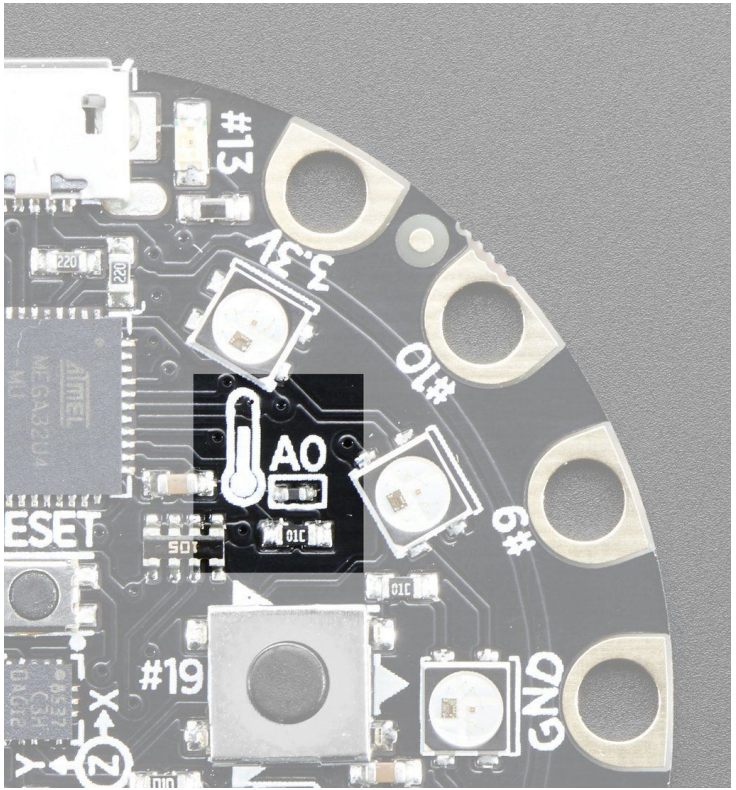


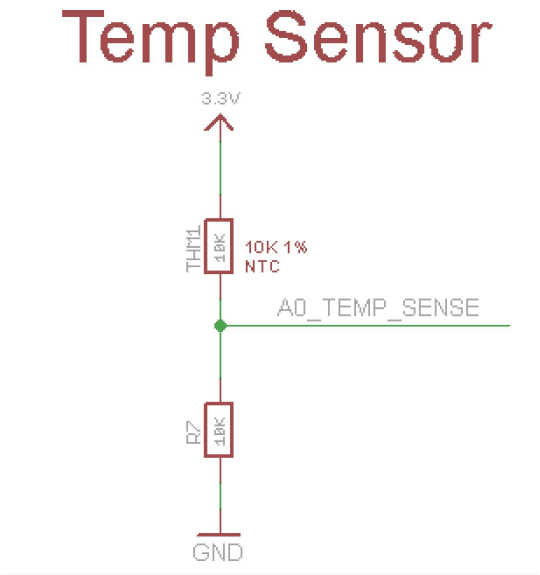


Wiring


100K Thermistor
100K Ohms
RESET
ICSP2
AREF
GND
DIGITAL (PWM=~)
TXD
RXD
L
TX
RX
UNO
Arduino
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
GND
GND
VIN
A0
A1
A2
A3
A4
A5

Temperature sensor





On the Circuit Playground schematics
you can see that this **sensor is connected to analog pin #A0**

Code

Code is the same as Light Sensor.

We have a linear voltage reading that can be mapped to C or F with ±0.25°C accuracy

But we can also use the voltage only less precise reading: i.e. hot / cold

```
int NTC_Pin = A0; //analog pin 0

void setup(){

   Serial.begin(9600);

}

void loop(){

   int NTCReading = analogRead(NTC_Pin);
   Serial.println(NTCReading);
   delay(250);

}
```

Checking the actual temperature

HARD WAY

1) Read the sensor analog voltage
2) Convert it to resistance, using the voltage divider formula
3) Take several readings and make an average
4) Optionally use 3V3 as AREF, for less noisy reading
5) Map the resistance to the C or F using the Steinhart-Hart equation

Check out this article for more details

Checking the actual temperature

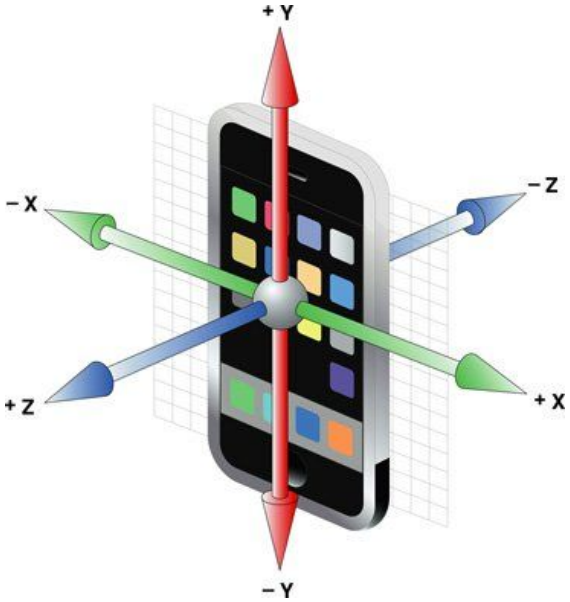
EASY WAY

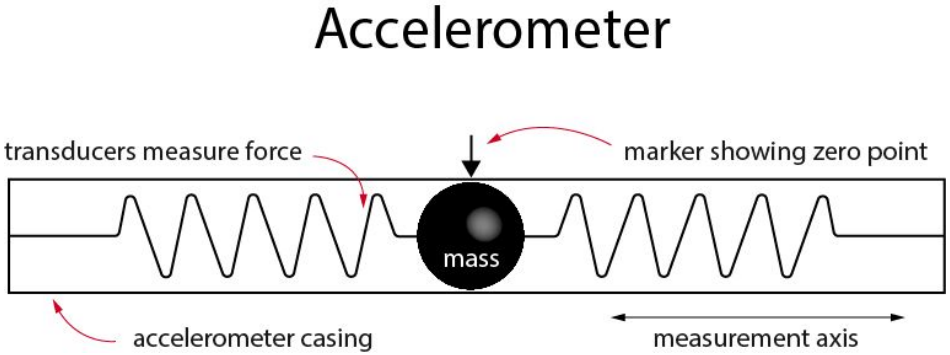
```
#include <Adafruit_CircuitPlayground.h>

void setup() {
   Serial.begin(9600);
   CircuitPlayground.begin();
}

void loop() {
  Float temp;
  temp = CircuitPlayground.temperature();
  Serial.println(temp);
}
```

Thermistor Arduino Library for other boards, and generic sensors

Accelerometers

Accelerometers are devices that measure acceleration, which is the rate of change of the velocity of an object.



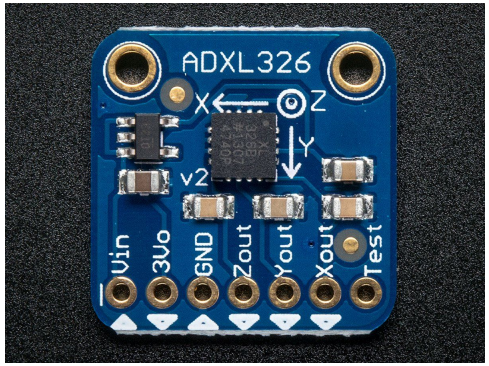


Micro-Electro-Mechanical Sensors (MEMS)
Inside there's usually a mass that moves and transducers that move the force of movement

Choosing an accelerometer

Analog vs digital: Analog accelerometers output a constant **variable voltage** depending on the amount of acceleration applied. Digital ones output their value with **PWM** or using multi-wire digital protocols such as **I2C or SPI**.

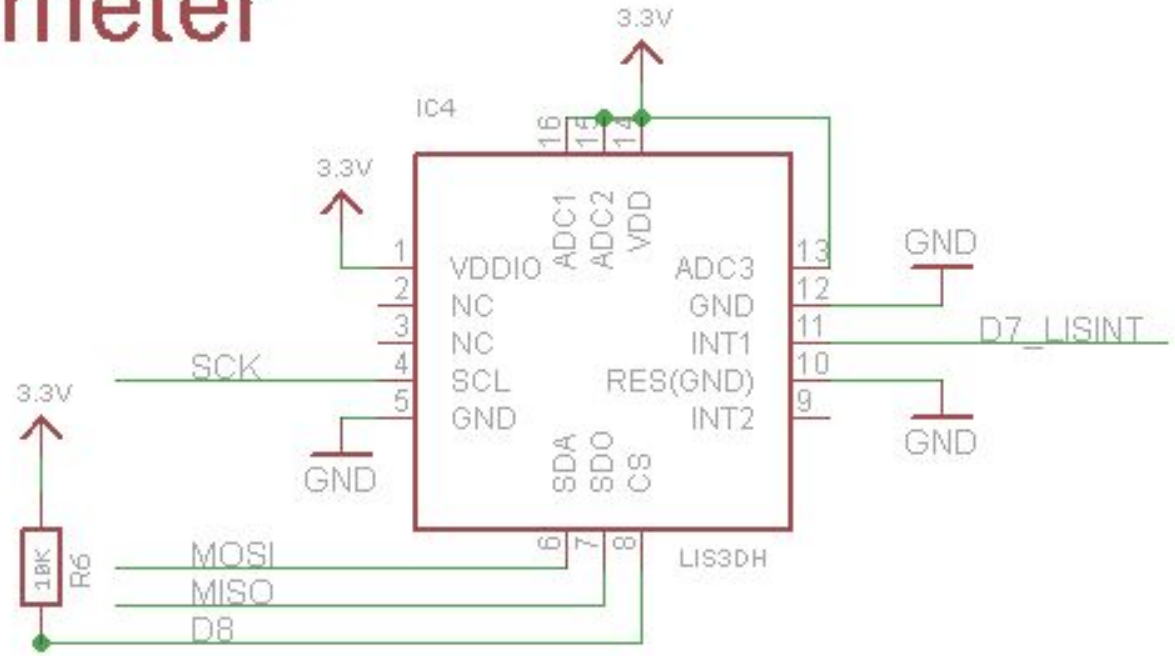
Precision range, Number of Axes, bandwidth, etc



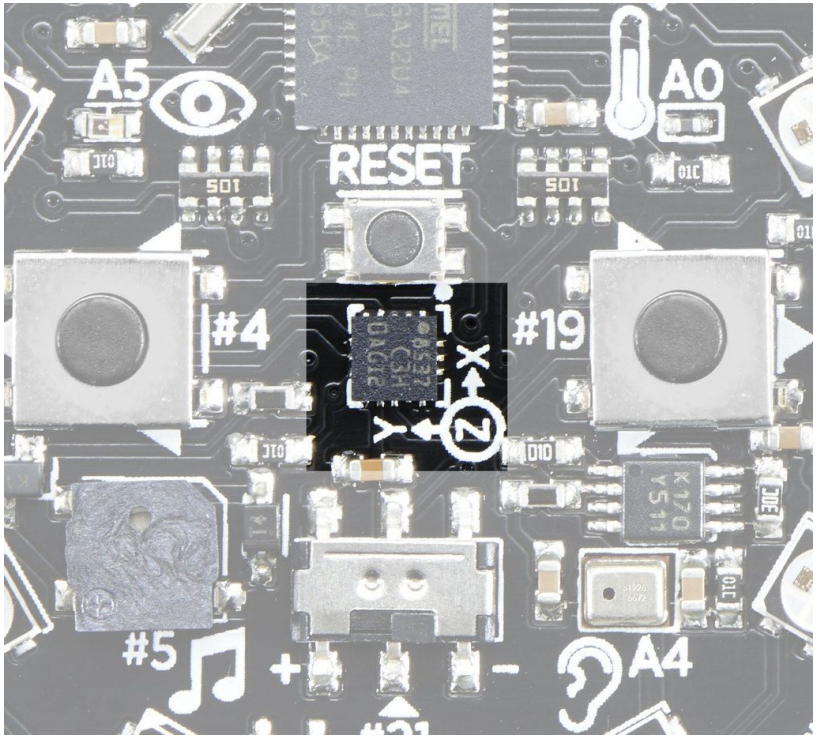
Analog accelerometer ADXL326

Digital accelerometer ADXL345 I2C

Accelerometer



Accelerometer

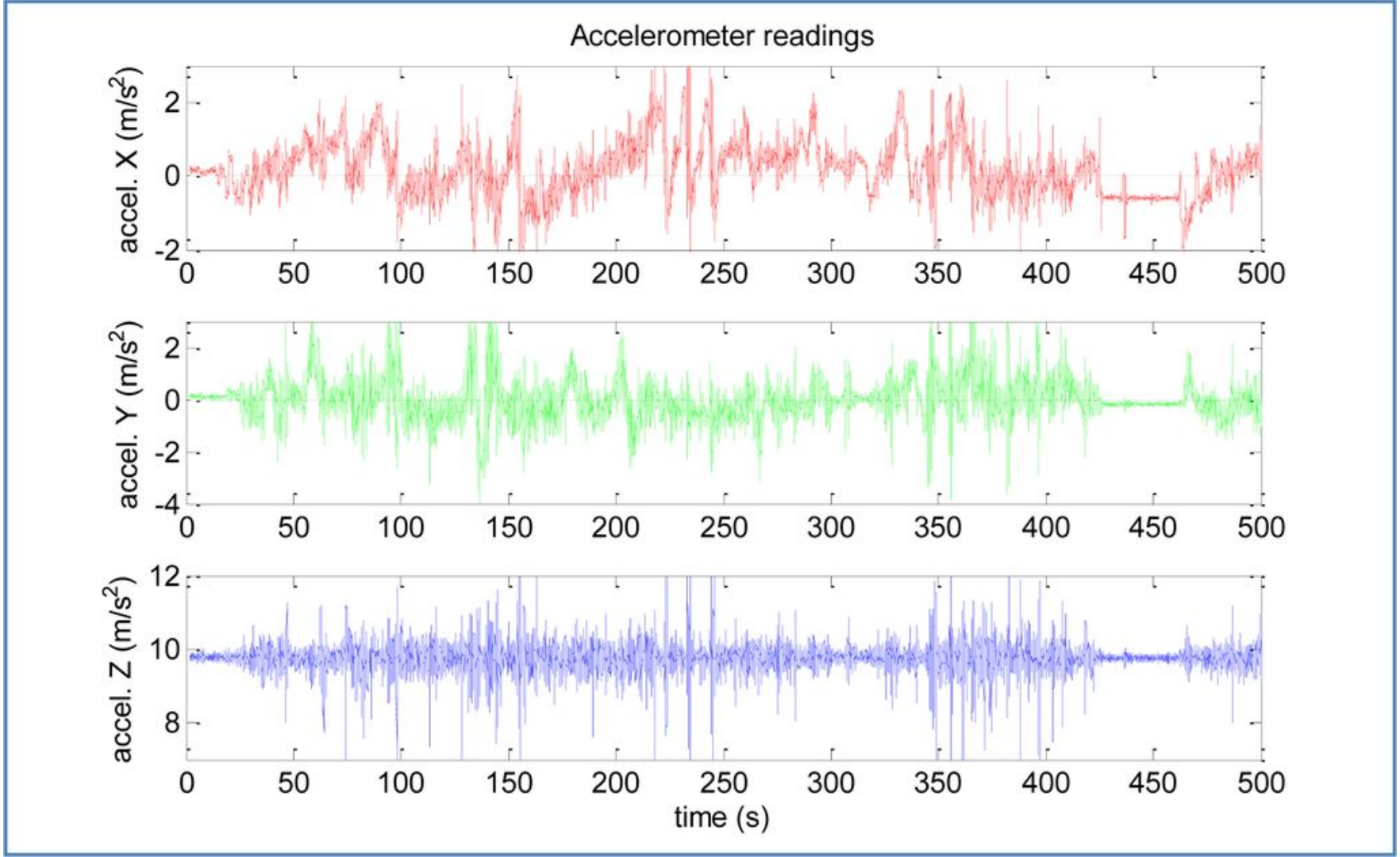


The CircuitPlayground has a model that can sense ±2g, 4g, 8g (g = 9.8 meters/s2)

The sensor is **more precise** when you set it to 2g, less when you set it to 8g

It also has **tap detection** (1 or 2 taps)

So you can easily detect these small movements in hardware


Accelerometer readings
accel. X (m/s2)
2
0
-2
0
50
100
150
200
250
300
350
400
450
500
accel. Y (m/s2)
2
0
-2
-4
0
50
100
150
200
250
300
350
400
450
500
accel. Z (m/s2)
12
10
8
0
50
100
150
200
250
300
350
400
450
500
time (s)

Interpreting readings

Any accelerometer will give you a **stream of x,y,z readings**.

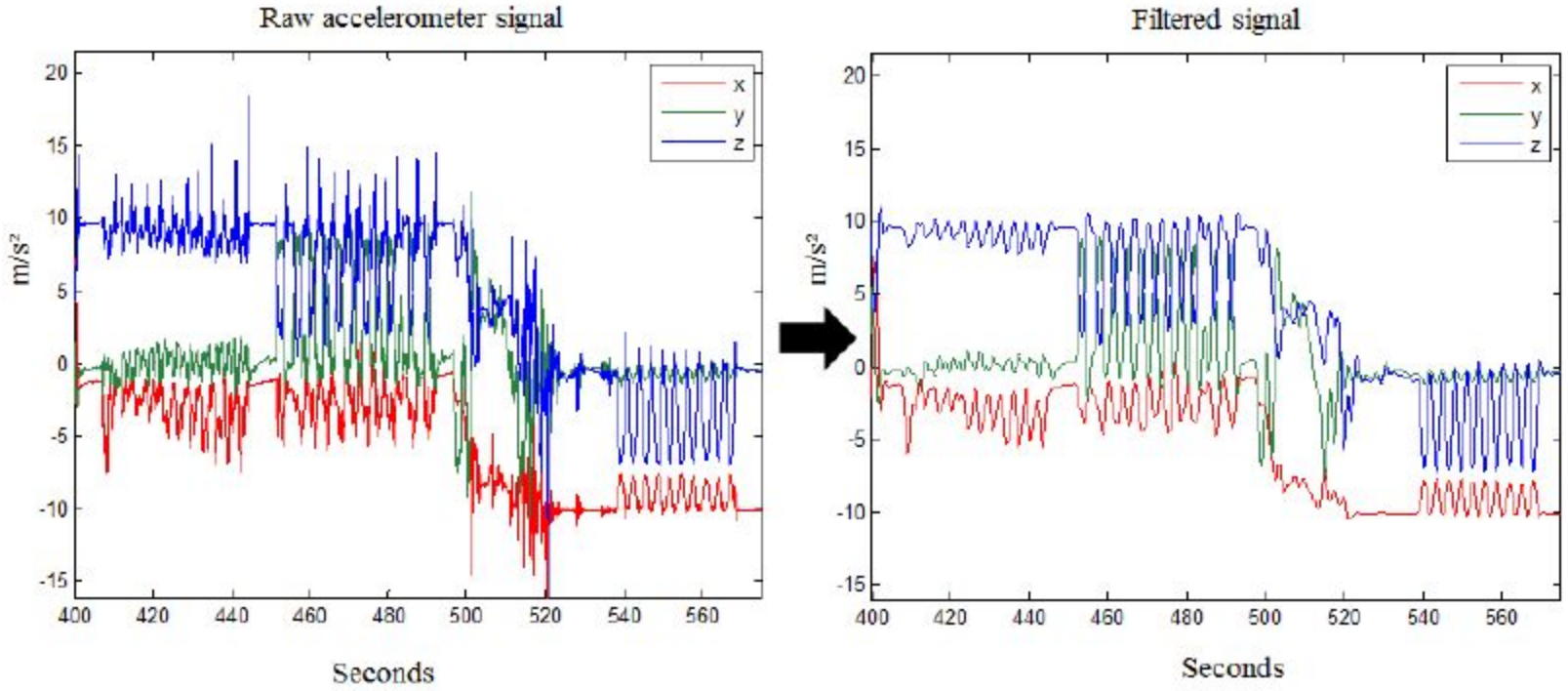
This will be **very noisy**, as the accelerometer is **very sensitive**.

Only **standing still** will get you **(0,0,0)**

Also we must **remove gravity** from our readings.

This can be done with a **low-pass filter**, which uses previous readings and a threshold to smooth out our signals

Raw vs filtered



Basic low pass filter

```
filtered_output[i] = α*raw_input[i] + (1-α)*filtered_output[i-1]
```

α is the threshold between 0 and 1 and we can use it to adjust the sensitivity of the filter

This is an exponential weighted moving average, read more here

Arduino function

```
int smooth(int data, float filterVal, float smoothedVal){

  // check to make sure param's are within range

  if (filterVal > 1){
    filterVal = .99;
  }
  else if (filterVal <= 0){
    filterVal = 0;
  }

  smoothedVal = (data * (1 - filterVal)) + (smoothedVal  *  filterVal);

  return (int)smoothedVal;
}
```

Detecting motion

Once we smooth the data we can detect motion in different ways:

- **Amplitude of change** - how much acceleration changes on one or more axis during a period of time, good for shake
- **Comparing** data taken from samples (training corpus) to the one we read from the sensors

The second method is more complex and require some processing power.

If you need to recognize gestures, the following methods are proven to work:

- Hidden Markov Models
- Dynamic time warping
- 1$ Gesture recognizer