Arduino Workshop

Hosted By

CLUB ELECTRO

Outline

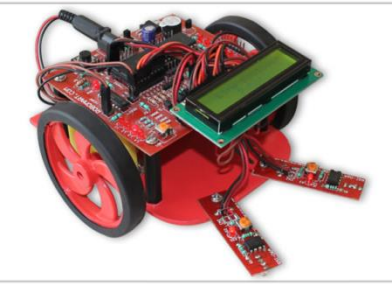
- What is a micro controller ?
- What is an Arduino dev board ?
- Basic Electronics concepts
- Basic programming concepts
- Getting your hands dirty ! (Practice)

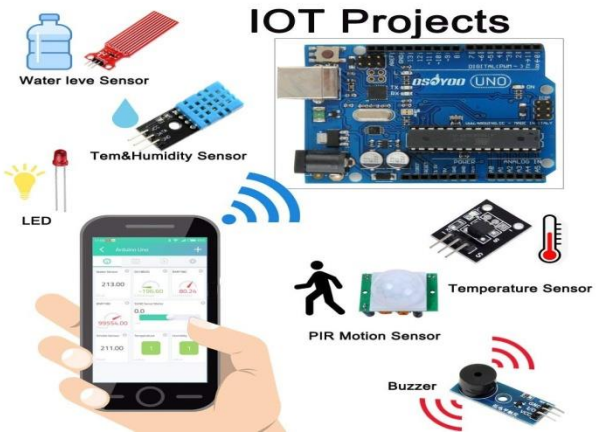
What’s a Microcontroller ?

A **microcontroller** is a **small computer** on a single Integrated circuit (IC). the **IC** contains : **CPU** , **RAM** , **ROM** (Read Only memory), **FLASH MEMORY** ,and **other Peripherals**...

Microcontrollers have very **limited hardware** (2 KB of RAM for the atmega328p), so they generally run only **a single program** and are used for **application specific purposes.**

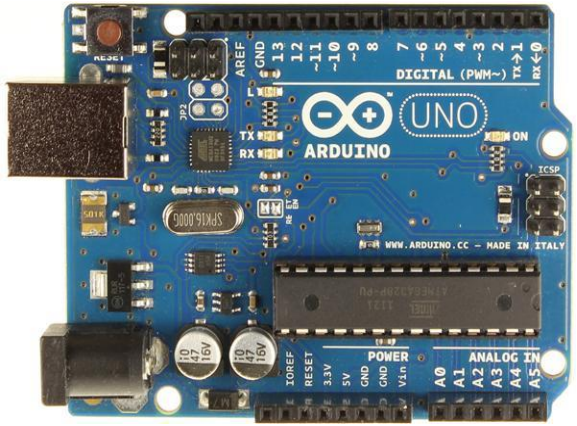
Applications of microcontrollers





Micro controllers are used even within computers themselves so..

You'll find them in almost every electronics appliance

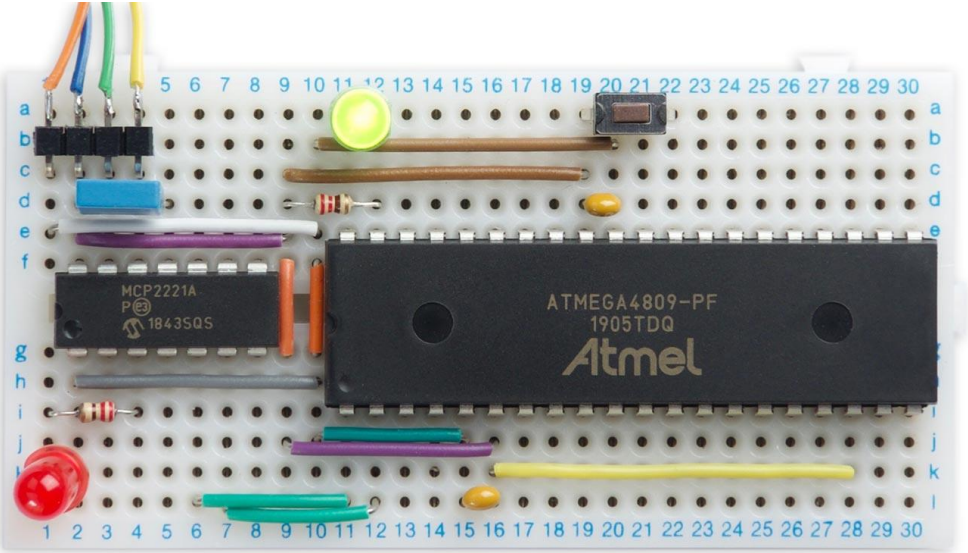


Arduino is an open-source electronics **platform** based on **easy-to-use** hardware and software. It's intended for anyone making interactive projects.

The **Arduino Uno board** Is based on the **Atmega328P** Microcontroller. And it is **not the microcontroller itself**.

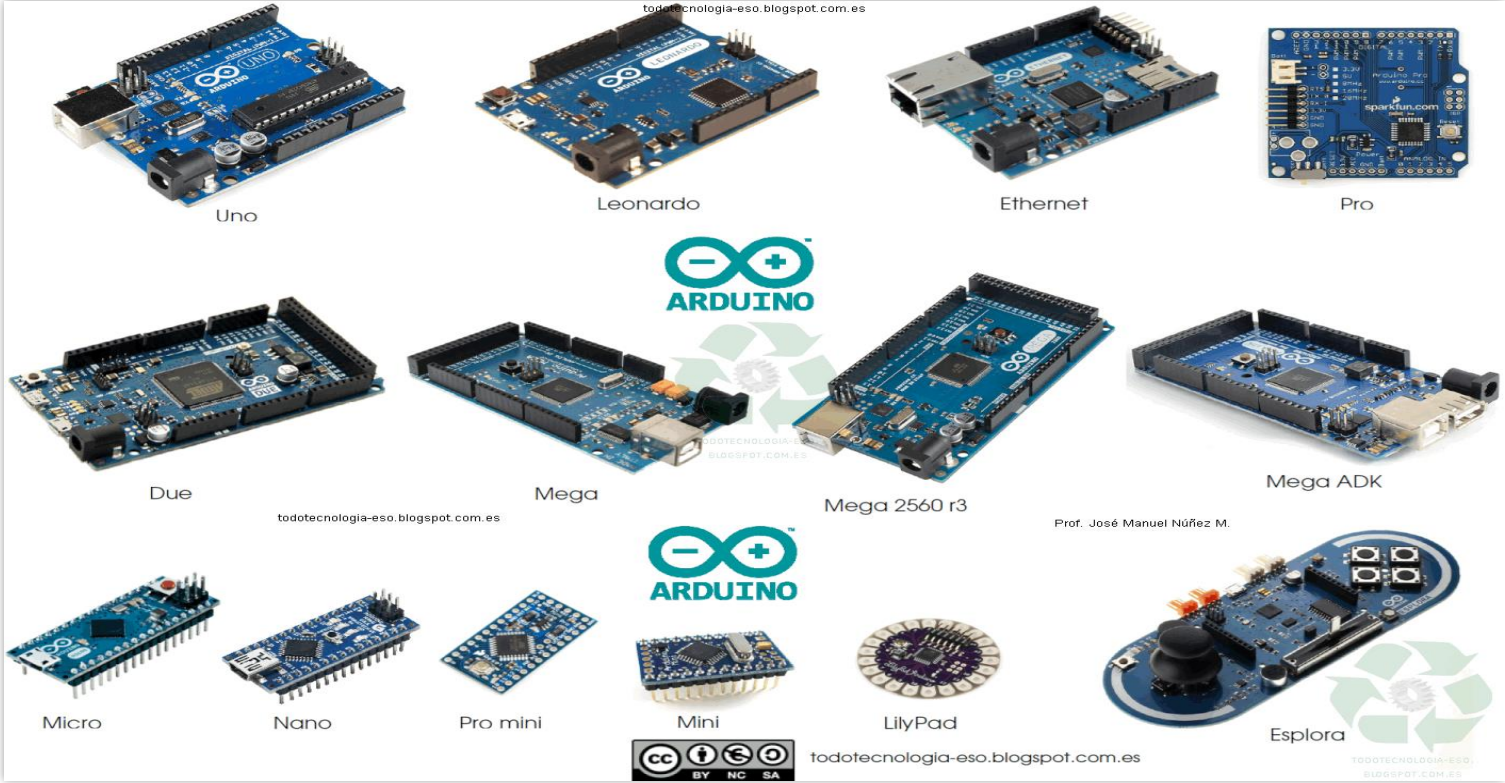
Why dev boards ?

The development board is meant for hobbyists and for people who don’t have a background in electronics. Without a dev board you will have to do this ...

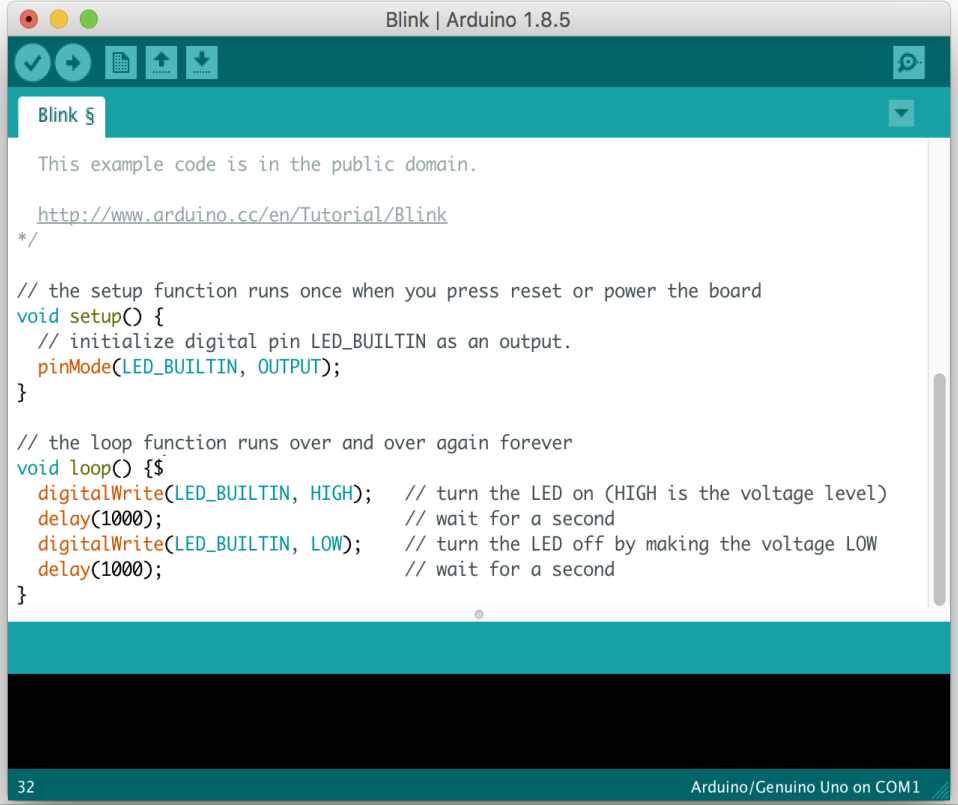


Which is already a too **steep learning curve** for a beginner

Arduino boards Family



Arduino IDE

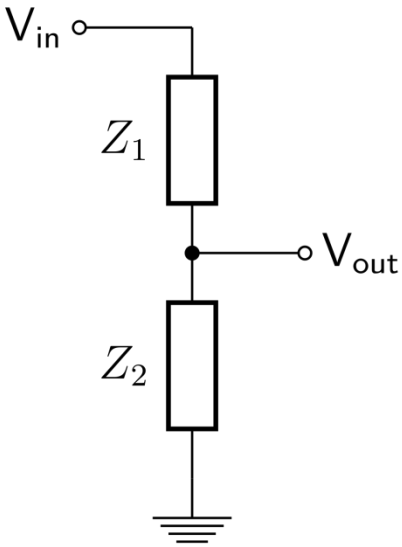


Arduino **I**ntegrated **D**evelopment **E**nvironment (**IDE**) makes it easy to write code and upload it to the board.

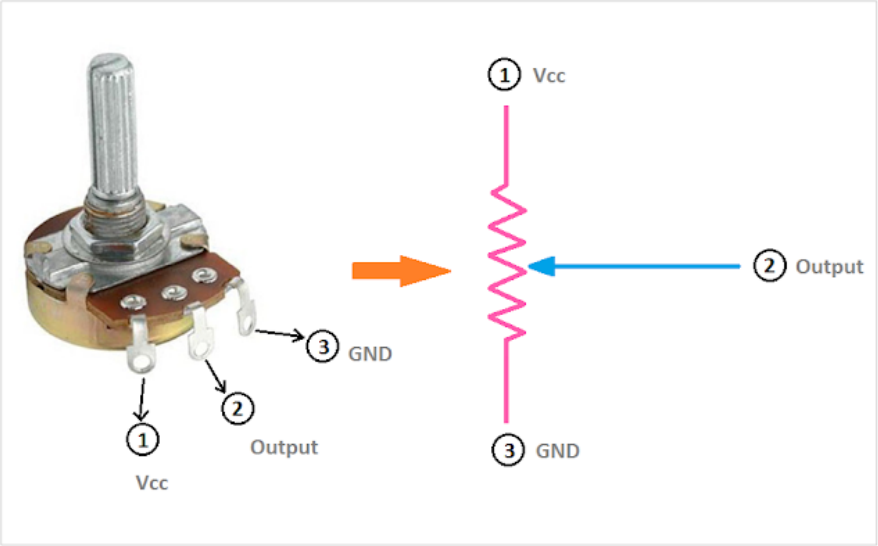
This software can be used with any Arduino board.

Basic Electronics Concepts

- Voltage divider

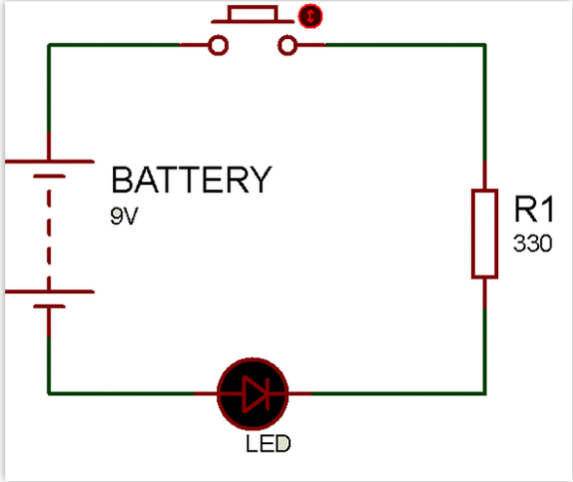


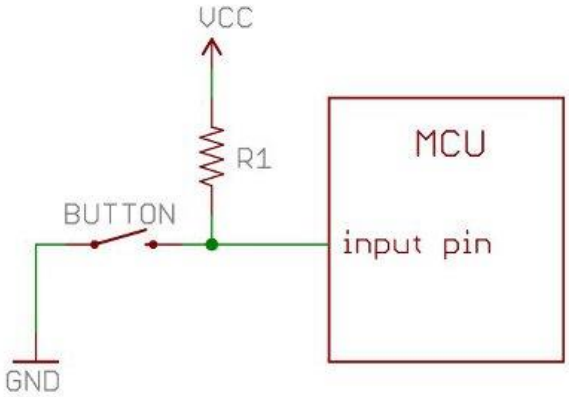
- Potentiometer (Variable resistor)



Basic Electronics Concepts

A **Push Button** is a switch based on the principle “**Push**-to-make”.
Initially, it leaves the circuit opened (no current), but when pressed the circuit is completed and current passes. To Use it with Arduino you are going to need what’s called a **pullup resistor**.

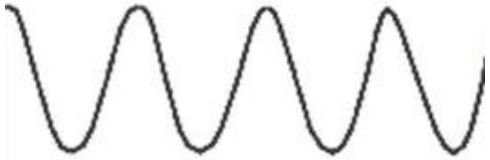




Digital & Analog

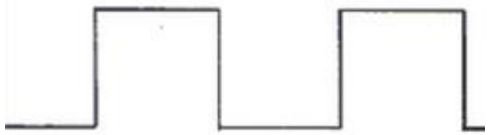
In electronics we have **Digital** and **analog** signals :

- Potentiometers for example provide **analog signals**. Since, varying the resistance results in voltage variatio



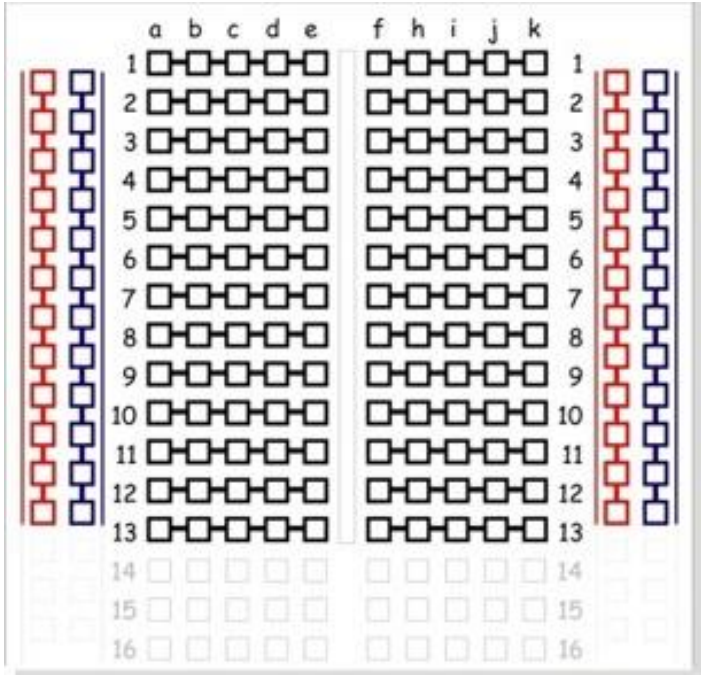
Analog Signal

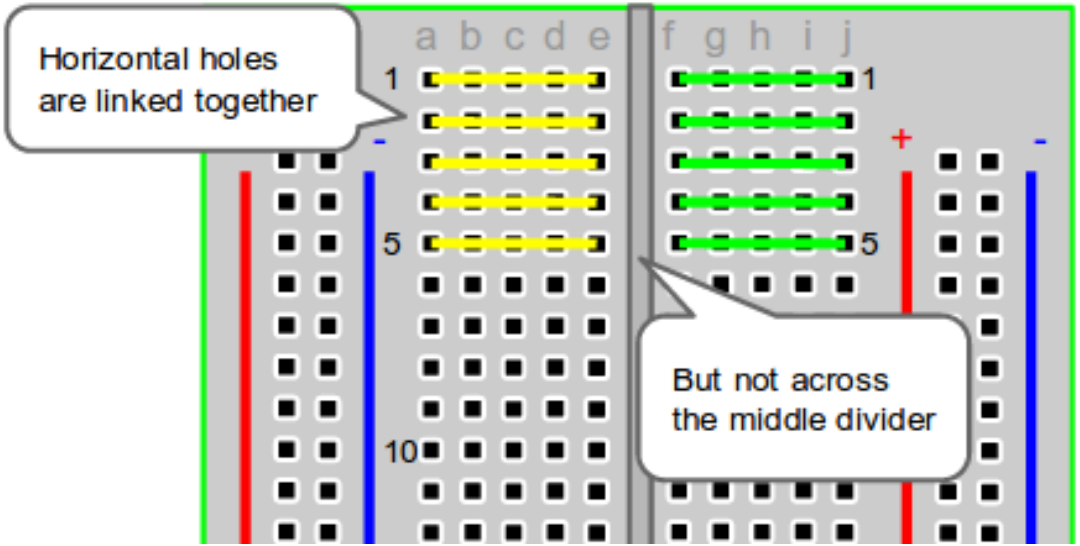
- **Push buttons** and switches provide a **digital signal** (either opened(0V) or closed(5V) ie 0 or 1)



Digital Signal

Breadboard layout





Basic Programming Concepts

In this workshop we are going to use 3 simple programming concepts:

- **Data types** and **Variables**
- **Functions**
- **Conditions**

A **variable** is a way to store data (value) and pair it with a symbolic name,, they can store

- Numbers, characters …
- Results of math operations
- Values returned from functions

Syntax:

datatype var_name = value ;

Basic Programming Concepts

Functions are modular pieces of code that can be defined ones and reused multiple times. (eg; digitalWrite(pin,value))

- In this workshop most used functions are **predefined** and **help us write code faster**
- Functions like ‘digitalRead() and analogRead() return values that can be stored in variables’

Conditions are a way to make the MCU take decisions and choices.

Syntax:

```
If ( condition ) {
...
Code
...
}
```

Setup() & Void() functions

Arduino framework makes your code **cleaner** by providing the following two functions:

```
void setup() {

  //Code here will run only ONES (setup code)

}

void loop() {

  //Code here will run REPEATEDLY (main code)
}
```

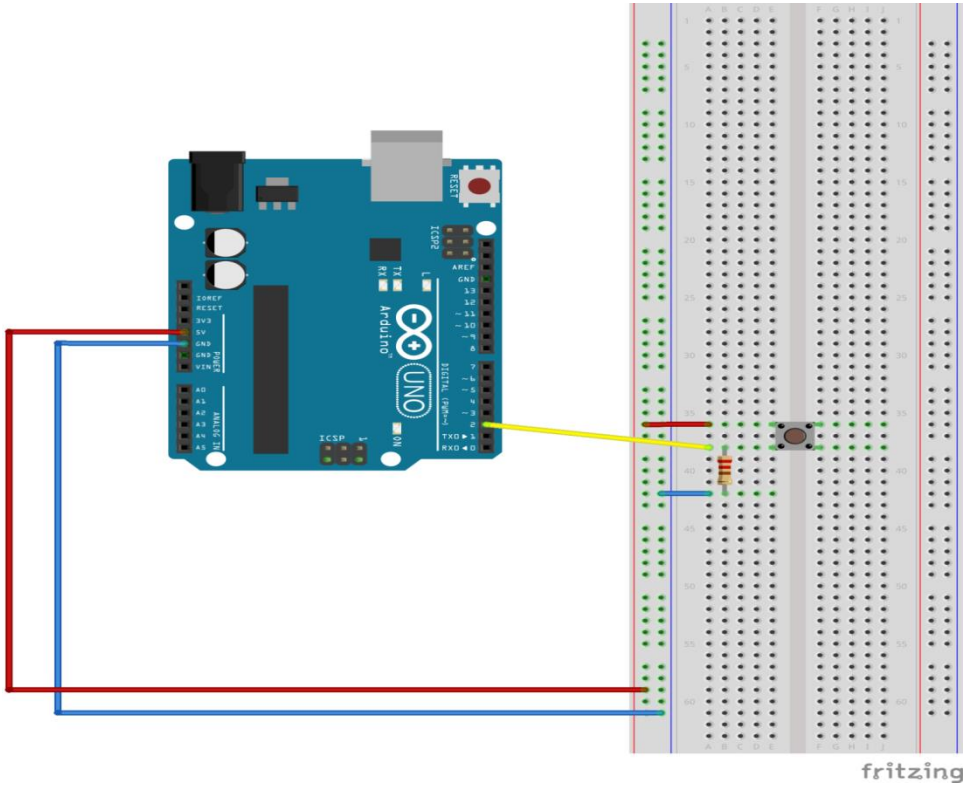
Getting hands dirty ! (Blink Me)

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, 1);
  delay(1000);
  digitalWrite(13, 0);
  delay(1000);
}
```

Arduino Uno has a **built-in** LED with a current limiting resistor attached to **PIN 13**, so we will use it directly !

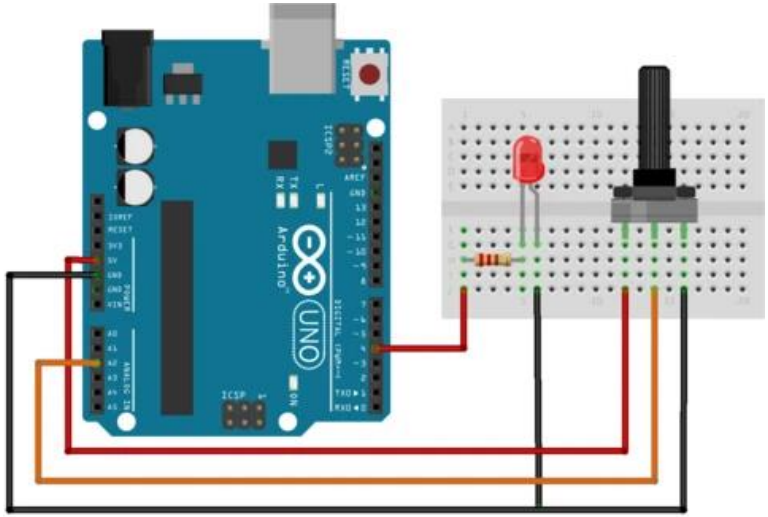
Push & digital read

```
void setup() {
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {

  bool switch_state = digitalRead(2);

  if ( switch_state == 1 ){
    digitalWrite(13, 1);
  }

  else if ( switch_state == 0 ){
    digitalWrite(13, 0);
  }

}
```



Pot and analog

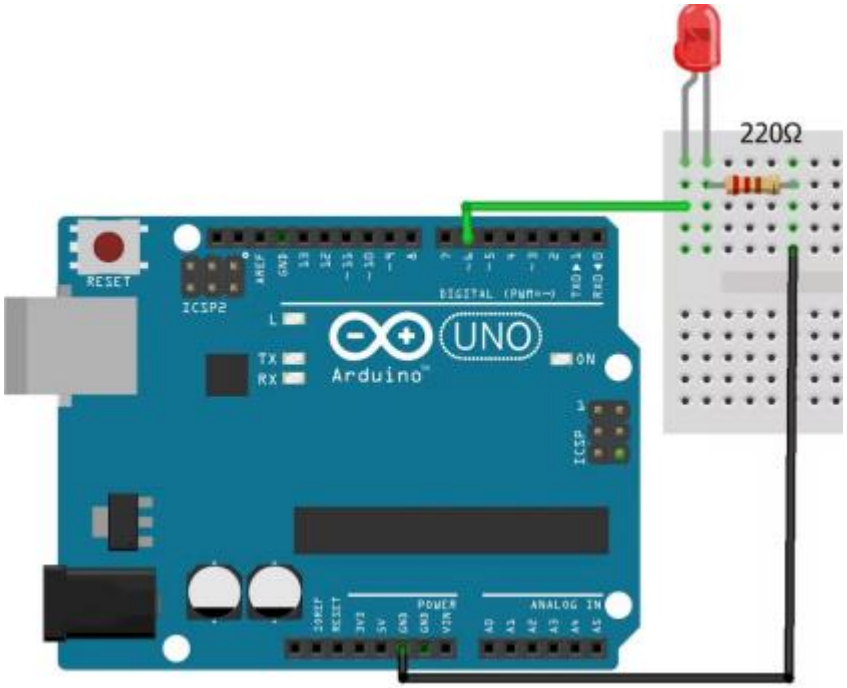
```
int potPin = A3;
int ledPin = 4;
int resolution = 1023;
int value = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  value = analogRead(potPin);
  value = map(value, 0, resolution, 0, 2000);
  digitalWrite(ledPin, HIGH);
  delay(value);
  digitalWrite(ledPin, LOW);
  delay(value);
}
```



- An alternative for the last 4 lines is to use the **analogWrite()** with pins marked with **PWM**

LED Fade with PWM

```
int intensity = 0;

void setup() {
  pinMode(6, OUTPUT);
}

void loop() {
  intensity = intensity + 1;
  analogWrite(6, intensity);
  delay(10);

  if (intensity == 255){
    intensity = 0;
  }
}
```



Short Break !
(15 Mins)

CLUB ELECTRO