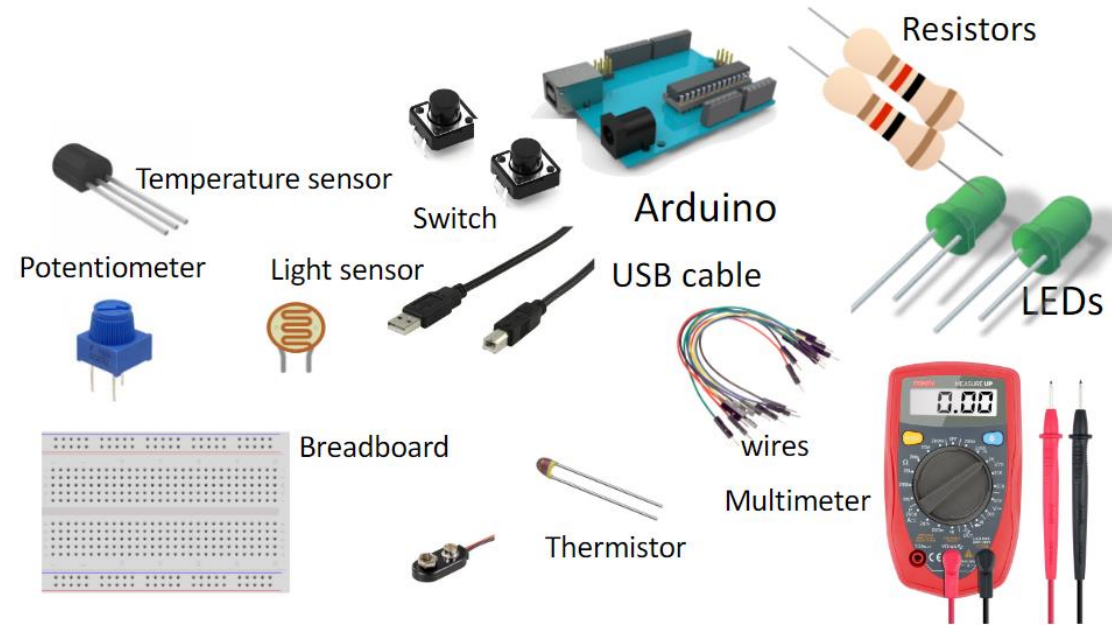


# IT.2406 - Introduction and Basics of Arduino

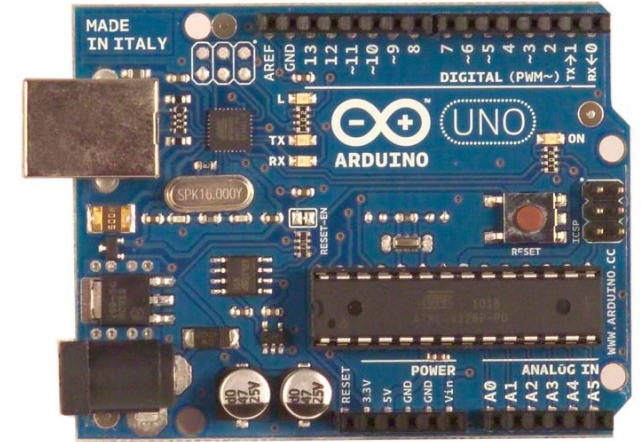
# What do you need ?

- To get started you need the following:
  - PC (Windows, Mac, Linux)
  - Hardware:
    - Arduino UNO Starter Kit
    - Electrical components: wires, resistors, etc.
    - Sensors / Actuators
  - Software: C programming background
  - Use cases: you imagination is the limit



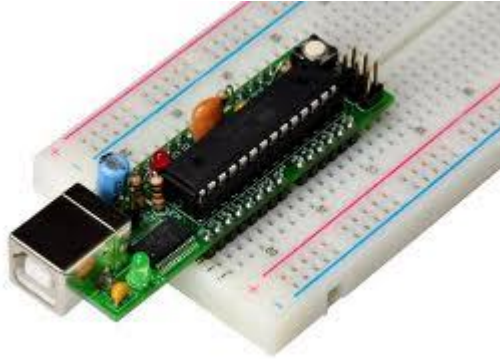
# Introduction

- What is Arduino ?
  - An **open-source** electronic prototyping platform based on flexible easy to use hardware and software.
- Basically: Arduino is **Microcontroller**.
- Microcontroller is **microprocessor** with **memory, RAM** and some other peripheral connected with it.
- The Arduino Uno is a microcontroller board based on the ATmega328. The ATmega328 has Flash memory of 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

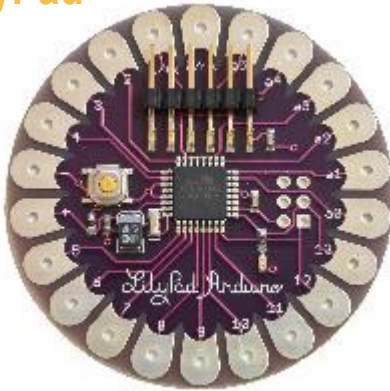


# Different types of Arduino

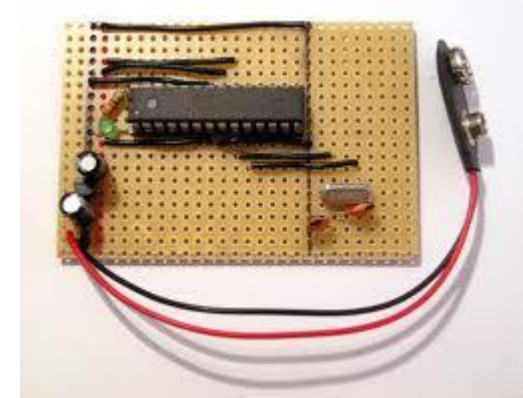
Boarduino Kit



Arduino LilyPad



Arduino Uno



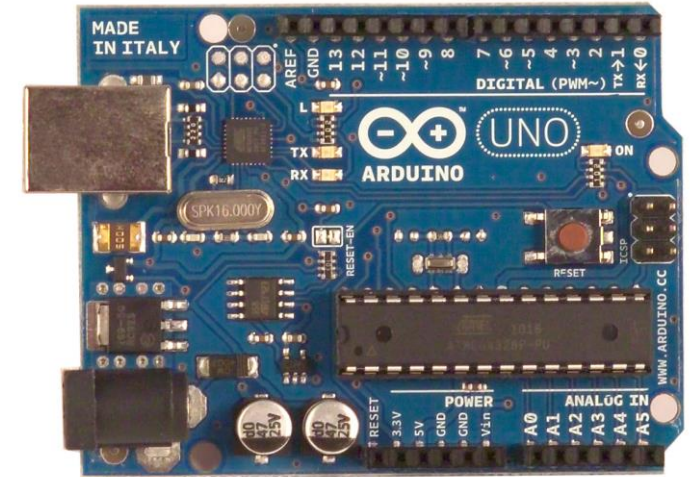
DIY Arduino



Arduino Mega 2560

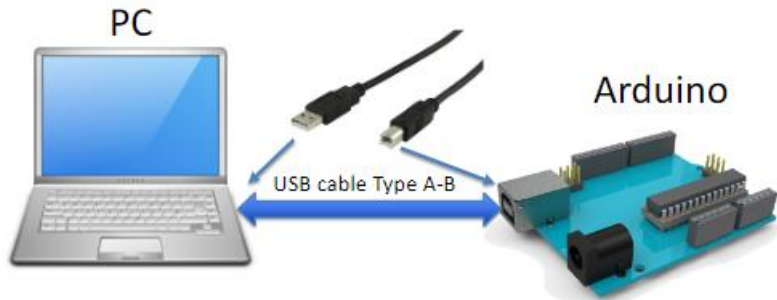
# Arduino Uno

- What does it have?
  - 14 Digital In/Out pins
  - 6 Analog Inputs
  - A USB Connection
  - A Power Jack
  - Reset Button
  - On-board LED
  - SCL/SDA pins (Serial Clock/ Serial Data pins)
- In short, it contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.





# Arduino Uno



**POWER**  
5V / 3.3V / GND

**Analog  
INPUTS**

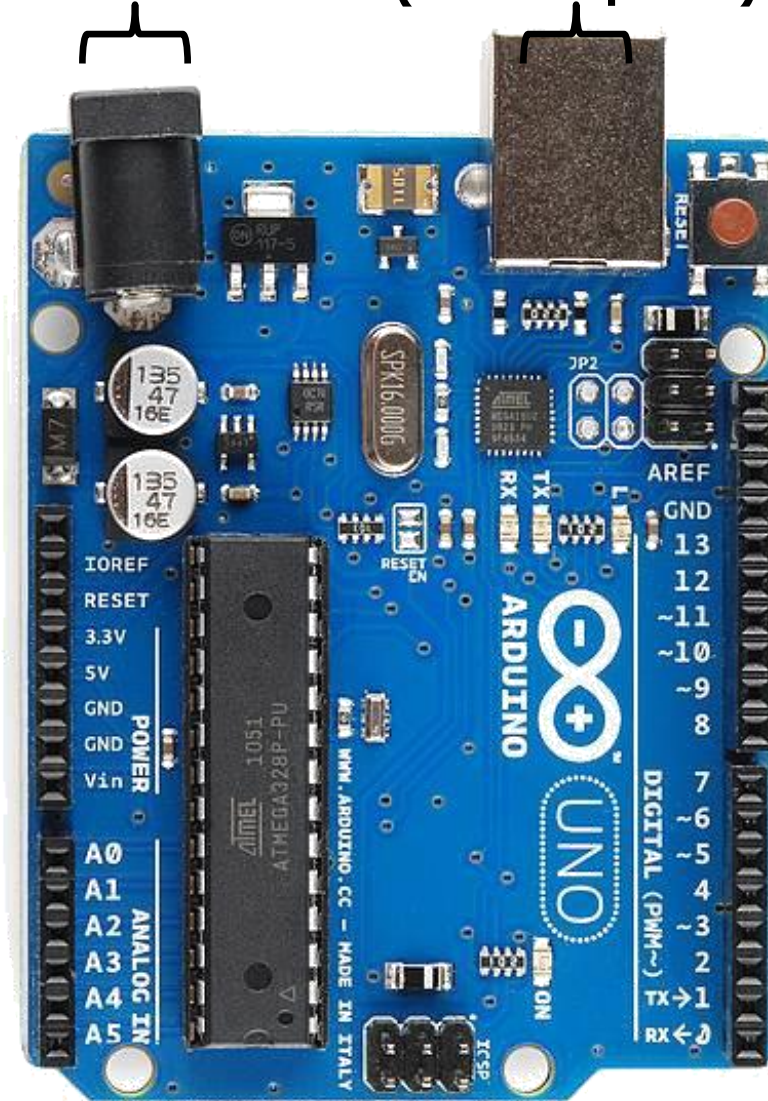
**PWR IN**

**USB  
(to Computer)**

**RESET**

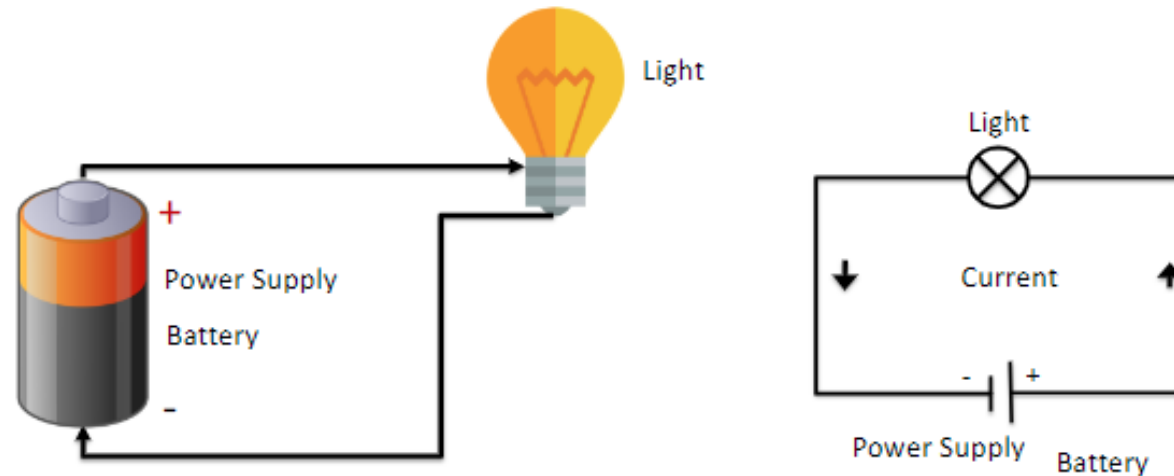
**SCL\SDA**  
(I2C Bus)

**Digital I\O**  
PWM(3, 5, 6, 9, 10, 11)



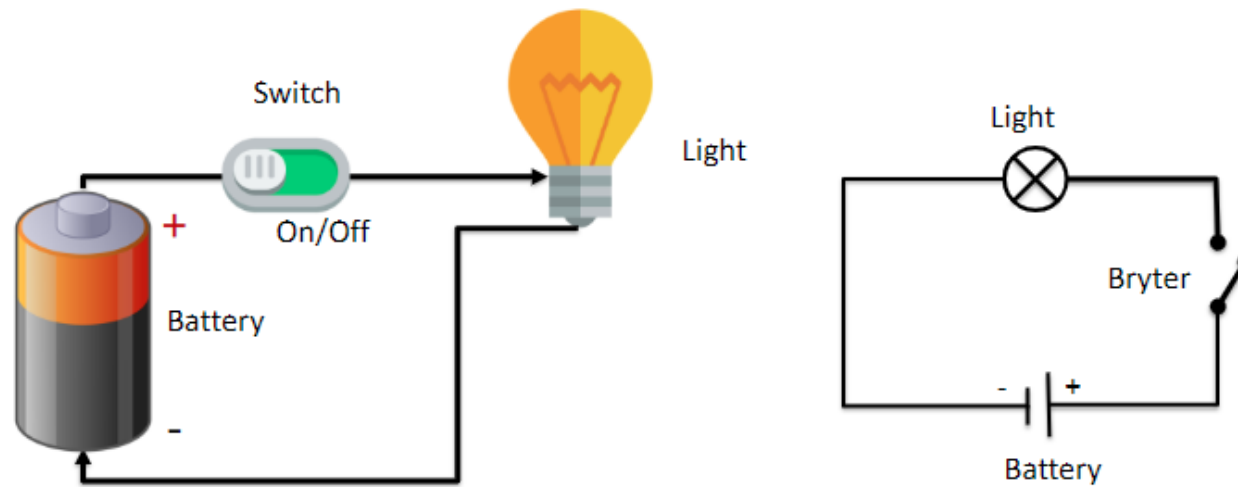
# Electrical Circuit

- Here you see a basic Electrical Circuit:



# Electrical Circuit

- Here you see a basic Electrical Circuit with a Switch:

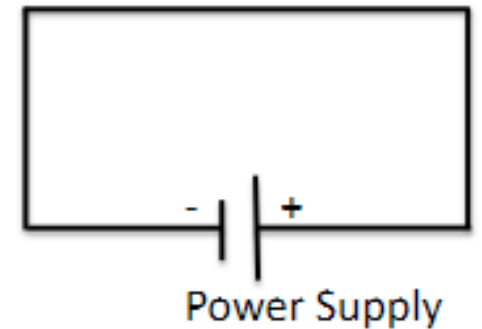




# Short Circuit

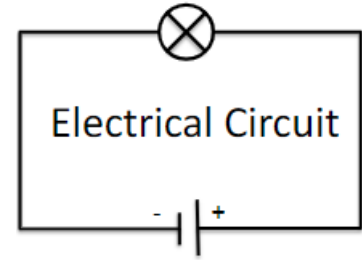
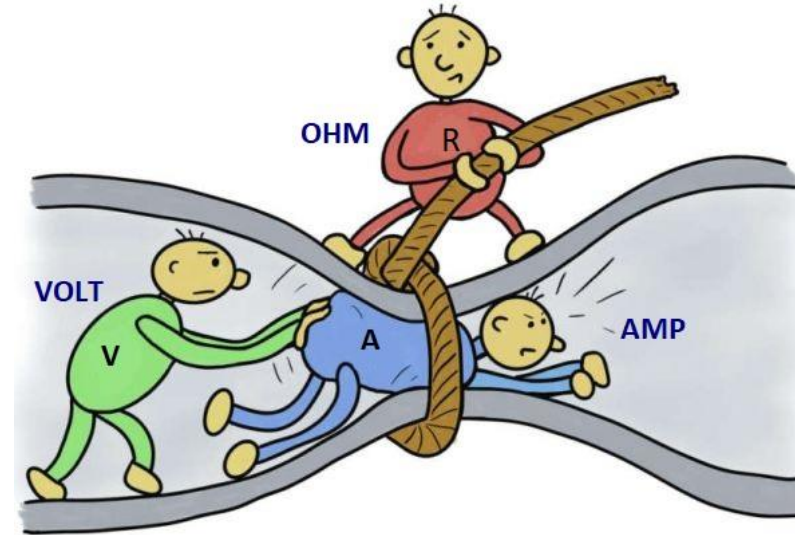


- We must never connect positive and negative side to a power source without having an electrical component in between.
- If you do, it is called a short circuit.
- For example, if you short circuit a battery, the battery will get very hot, and the battery will run out very quickly.
- Some batteries may also start to burn.
- When it starts to smoke from electrical components, it happens because it has become too hot.
- In most cases, it means that the component is broken.



# Ohms Law

- This is Ohms Law:
  - $U = RI$ 
    - U : Voltage [V]
    - R : Resistance [ $\Omega$ ]
    - I : Current [A]



# Tools

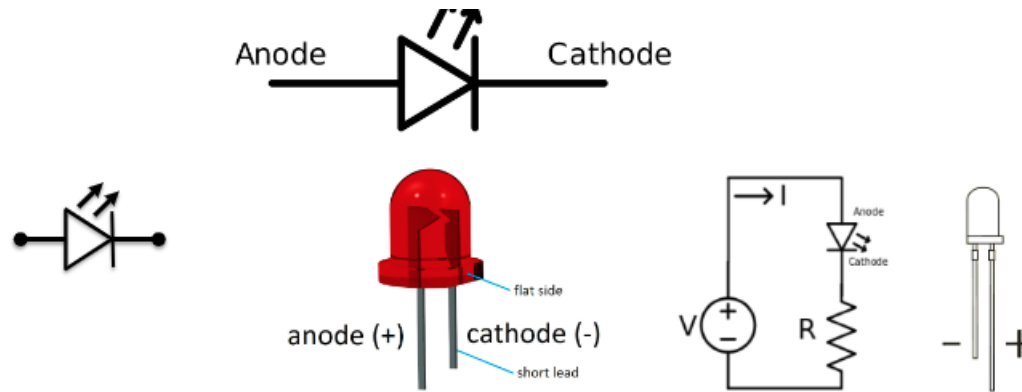
- Multimeter:
  - You can use a Multimeter to measure current, voltage, resistance, etc. in an electric circuit.



<https://learn.sparkfun.com/tutorials/how-to-use-a-multimeter>

# Equipements

- Light-Emitting Diode - LED



# Equipements

- Resistors:
  - Resistance is measured in Ohm ( $\Omega$ )
  - Resistors comes in many sizes, e.g.,  $220\Omega$  ,  $270\Omega$ ,  $330\Omega$ ,  $1k\Omega$   $10k\Omega$ , ...
  - The resistance can be found using Ohms Law  $U= RI$

<https://en.wikipedia.org/wiki/Resistor>

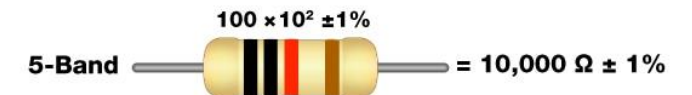
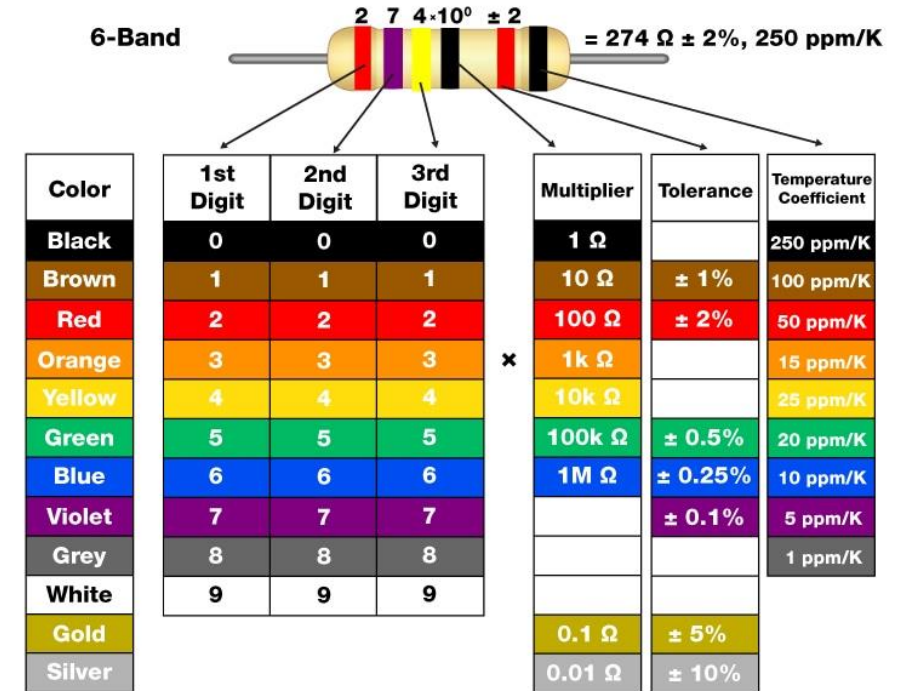


# Equipements

- Resistors:
  - Resistor Color Codes [1]
  - You can use the multimeter to check the resistance value.
  - Or Use a « Resistor Color Code Calculator » [2]

[1] <https://www.electronicshub.org/resistor-color-code/>  
[2] <http://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>

## How to Read Resistor Color Codes



# Equipements

- Resistors:

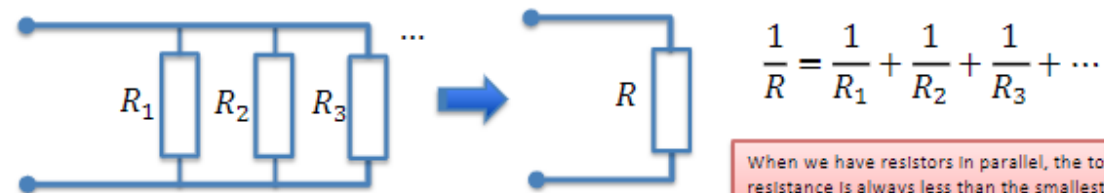
## Resistors in Series:



The total resistance of resistors connected in series is the sum of their individual resistance values.

When we have resistors in series, the sum of the sub-voltages is equal to the voltage of the voltage source

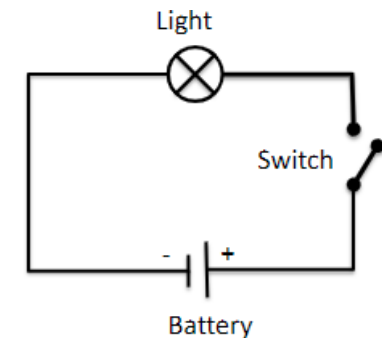
## Resistors in Parallel :



When we have resistors in parallel, the total resistance is always less than the smallest resistors

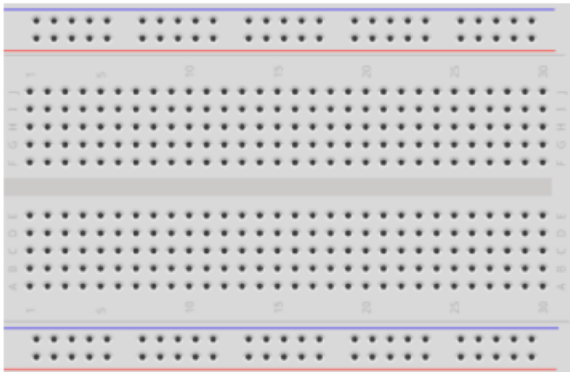
# Equipements

- Switch:
  - A switch breaks the flow of current through a circuit when open. When closed, the current will flow unobstructed through the circuit.



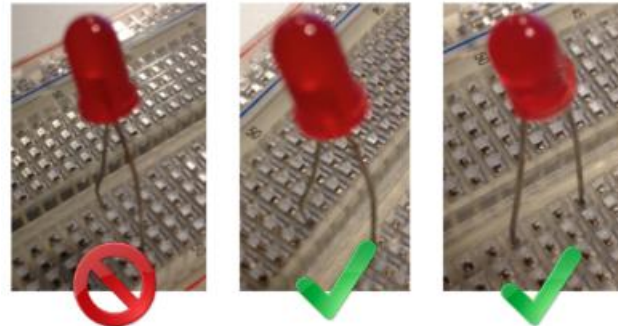
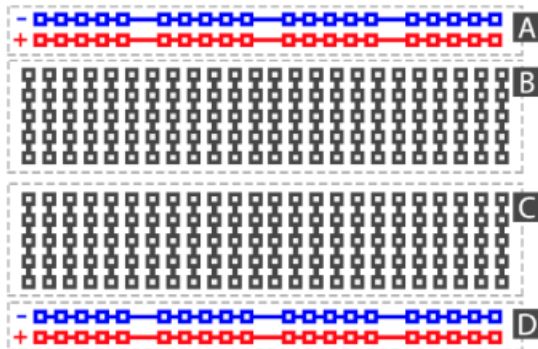
# Equipements

- Breadboard:
  - A breadboard is used to wire electric components together



# Equipements

- Breadboard:
  - Make sure not to short-circuit the components that you wire on the breadboard





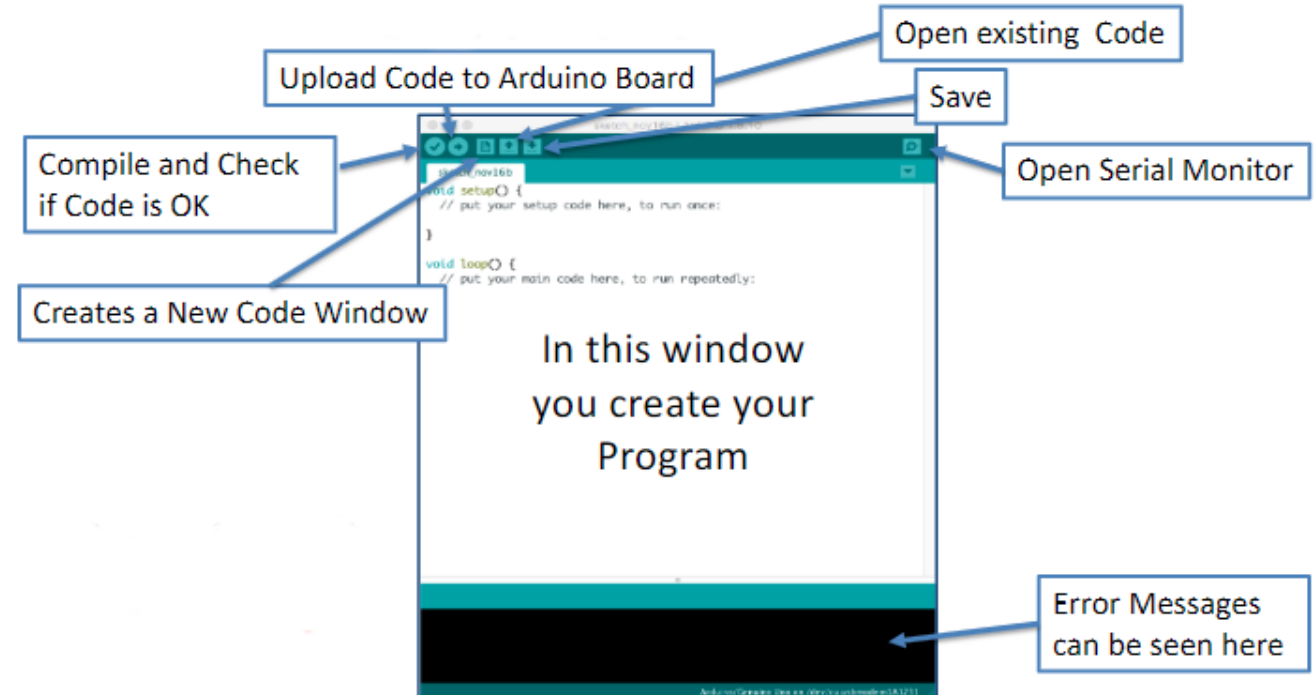
- **Fritzing** is an open-source hardware initiative that makes electronics accessible as a creative material for anyone.
- They offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional PCBs.

**\$ sudo apt-get install fritzing**

<https://fritzing.org/>  
Tutorials → <https://fritzing.org/learning/tutorials/>

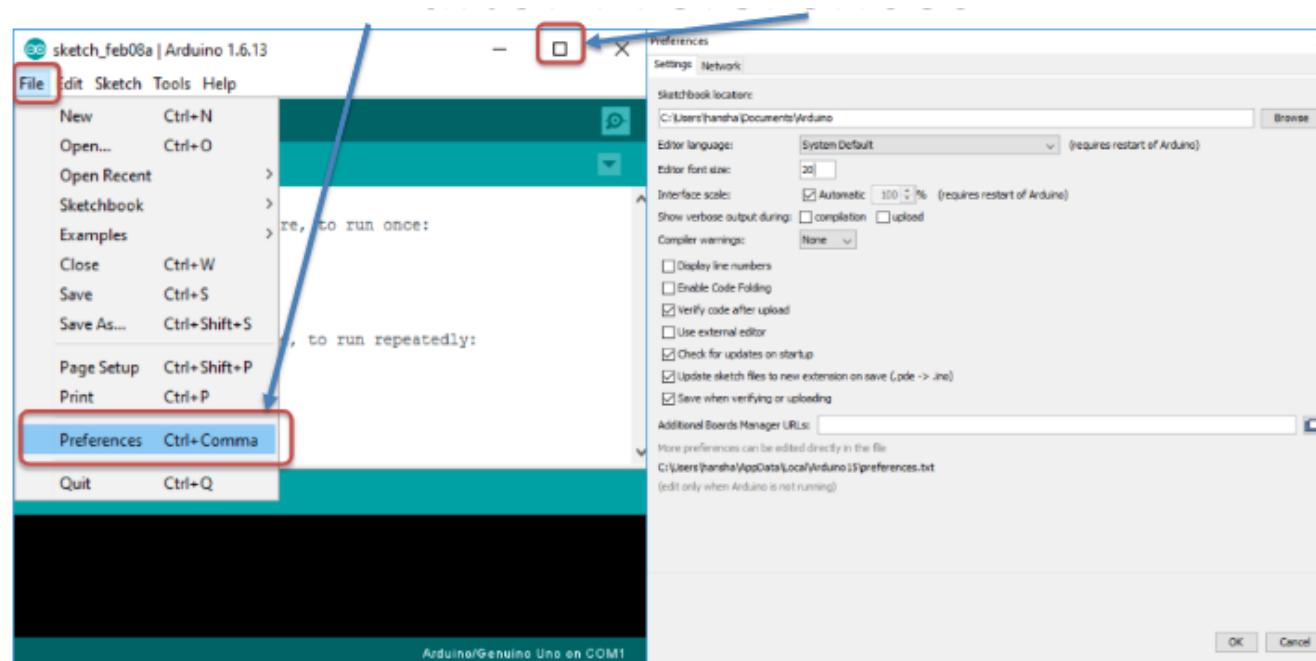
# Arduino Development Environment

- Arduino Software (IDE):
  - Can be downloaded for free:  
<https://www.arduino.cc/en/software>



# Arduino Development Environment

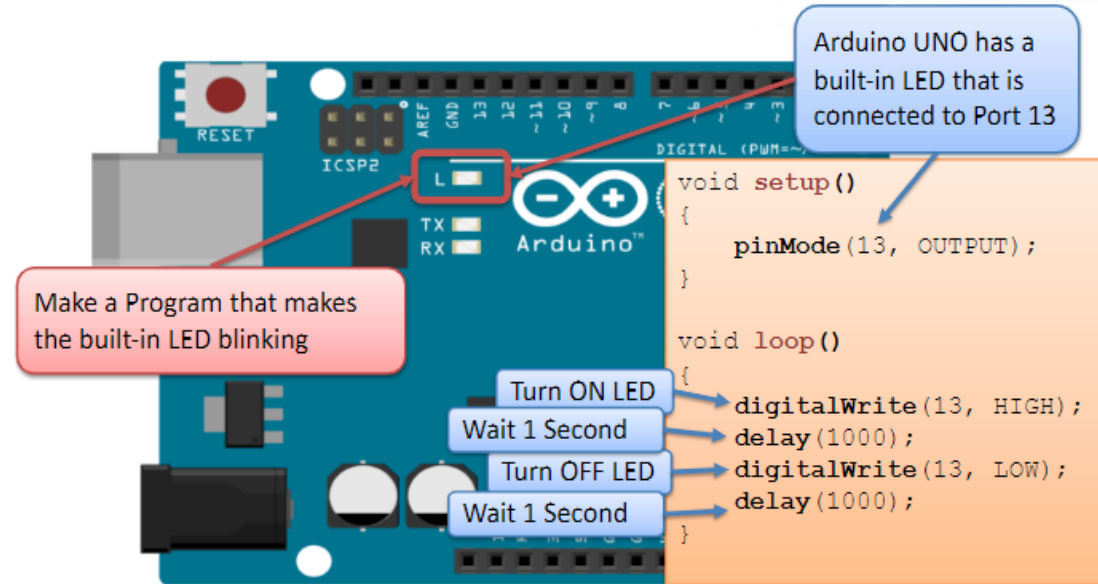
- Editor Preferences



# Arduino Development Environment

- Example: Blinking LED

Try it out



# Arduino Development Environment

- Example: Blinking LED

```
void setup()  
{  
    pinMode(13, OUTPUT);  
}  
  
void loop()  
{  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

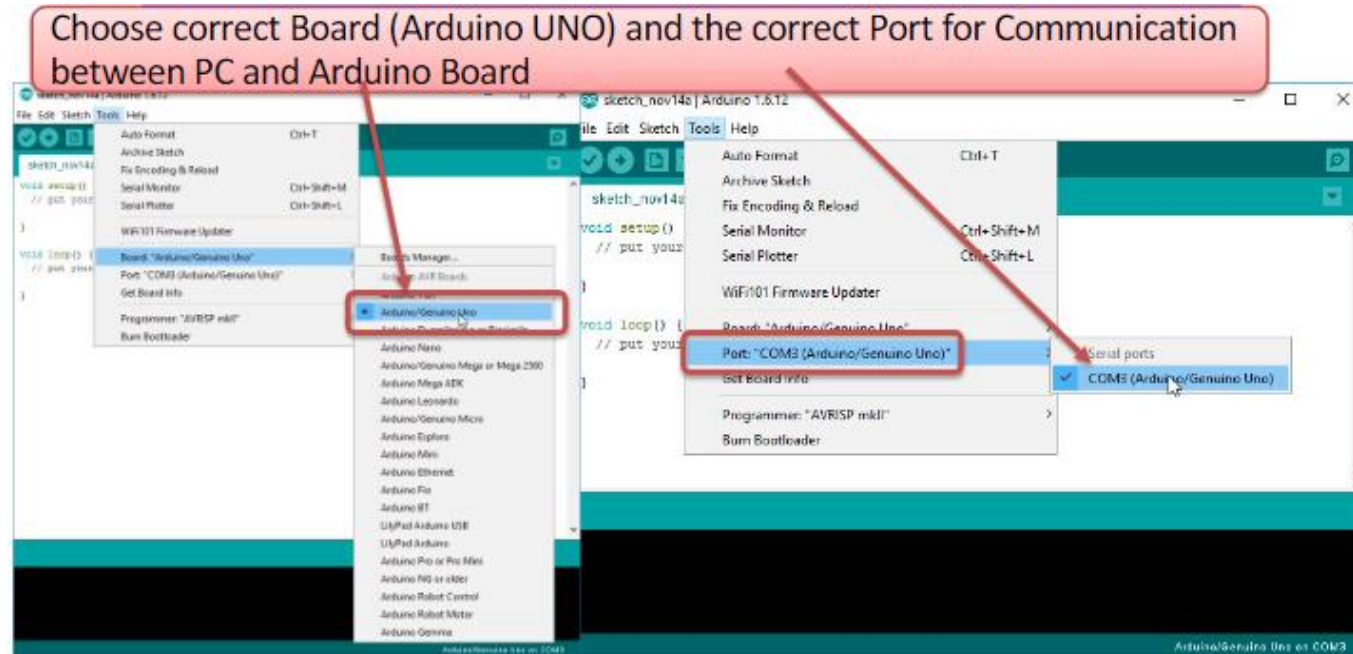
This Program makes the  
built-in LED blinking

Try to change from  
1000 to 100  
– What happens then?



# Arduino Development Environment

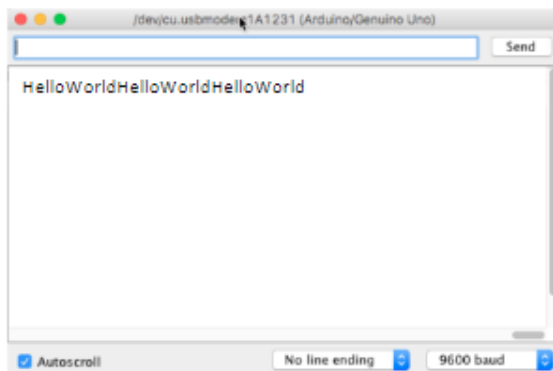
- Do you get an Error Message?



# Arduino Development Environment

- Serial Monitor:

- You use the Serial Monitor when Debugging Arduino programs or when you want to show data or values from your program.
- You need to have Arduino connected to your PC in order to use the Serial Monitor.



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

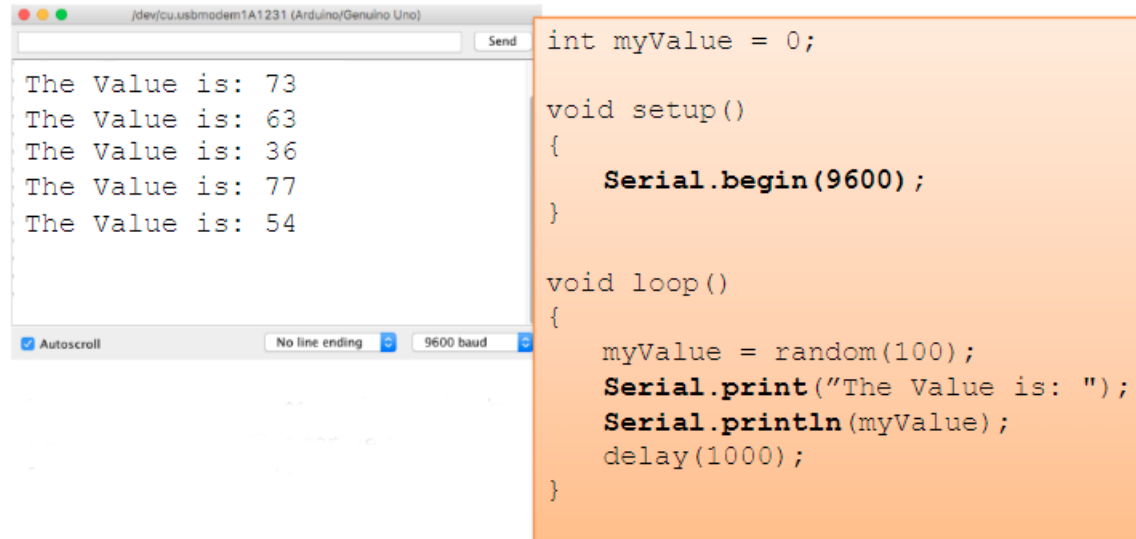
void loop()
{
    Serial.print("Hello World");
    delay(1000);
}
```

Try it out

# Arduino Development Environment

- Serial Monitor:

- Here you see how we can write a value to the Serial Monitor.
- This can be a value from a sensor, e.g., a temperature sensor.



The image shows a screenshot of the Arduino IDE. On the left, the Serial Monitor window is open, displaying five lines of text: "The Value is: 73", "The Value is: 63", "The Value is: 36", "The Value is: 77", and "The Value is: 54". The window has a "Send" button and a status bar at the bottom showing "Autoscroll" checked, "No line ending", and "9600 baud". On the right, the code editor shows the following C++ code:

```
int myValue = 0;

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

void loop()
{
    myValue = random(100);
    Serial.print("The Value is: ");
    Serial.println(myValue);
    delay(1000);
}
```

Try it out

# Arduino Development Environment

- Arduino Programs:
  - All Arduino programs must follow the following main structure:

```
// Initialization, define variables, etc.  
  
void setup()  
{  
    // Initialization  
    ...  
}  
  
void loop()  
{  
    //Main Program  
    ...  
}
```

# Arduino Development Environment

- Arduino Programs:

- Example:

```
void setup()
{
    pinMode(11, OUTPUT);      //Set the Pin as an Output
}

void loop()
{
    digitalWrite(11, HIGH);   // Turn on the LED
    delay(1000);              // Wait for one second
    digitalWrite(11, LOW);    // Turn off the LED
    delay(1000);              // Wait for one second
}
```



# Arduino Development Environment

- Arduino Programs:

- Example:
  - Using Comments

```
void setup()
{
    pinMode(11, OUTPUT);    //Set the Pin as an Output
}

void loop()
{
    digitalWrite(11, HIGH); // Turn on the LED

    /*
    ... This will not be executed by the program because
    it is a comment...
    */
}
```

# Arduino Development Environment

- Arduino Programs:
  - Example:
    - Creating and Using Functions

```
int z;  
  
void setup()  
{  
    }  
  
void loop()  
{  
    z = calculate(2,3);  
}  
  
float calculate(int x, int y)  
{  
    return (x + y);  
}
```

Using the Function

Creating the Function

# Arduino Development Environment

- Arduino Programs:
  - Example:
    - Creating and Using Functions → Sum

```
int z;int a;int b;
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    a = random(100);
    b = random(100);
    z = calculate(a,b); //Adding 2 Numbers

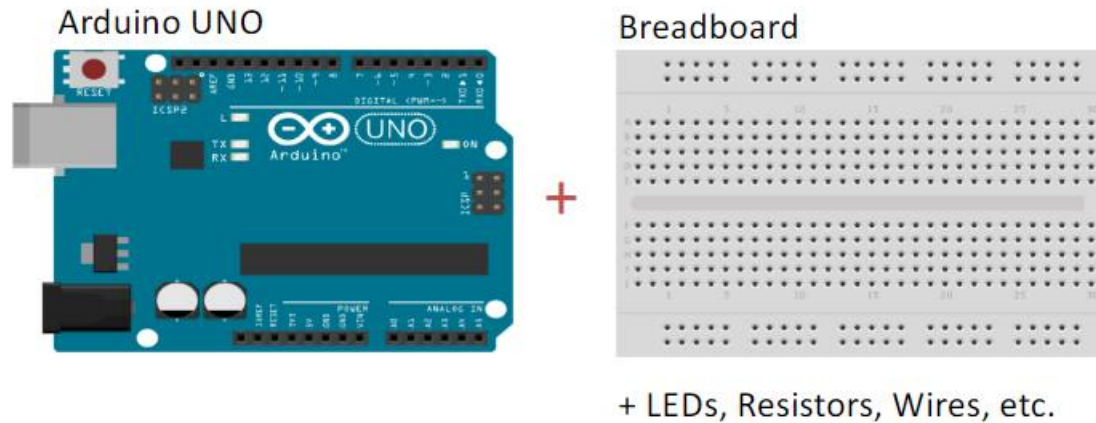
    //Write Values to Serial Monitor
    Serial.print(a);
    Serial.print(" + ");
    Serial.print(b);
    Serial.print(" = ");
    Serial.println(z);

    delay(1000);
}
float calculate(int x, int y)
{
    return (x + y);
}
```

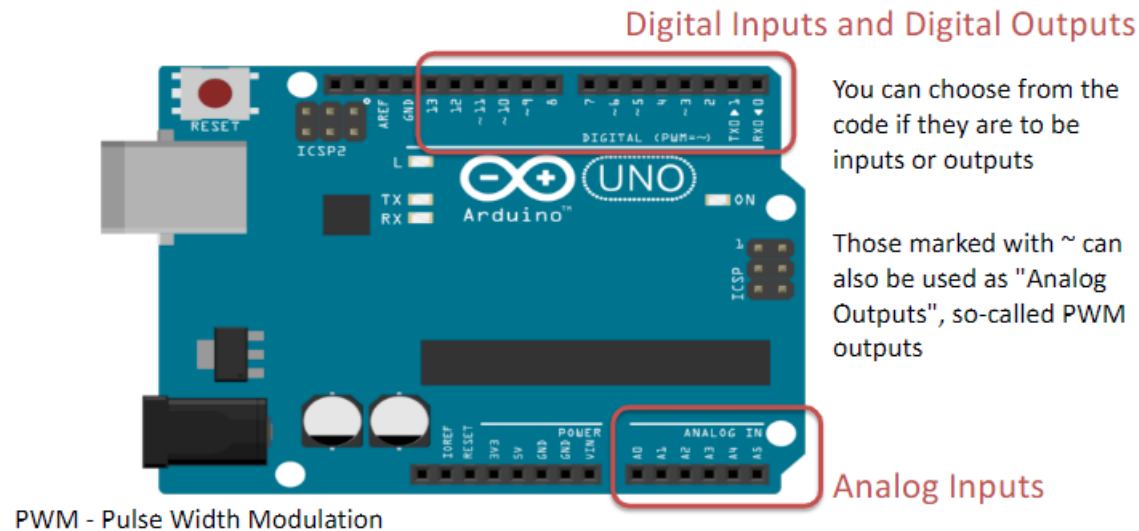
TRY IT OUT!

# Mini-projects

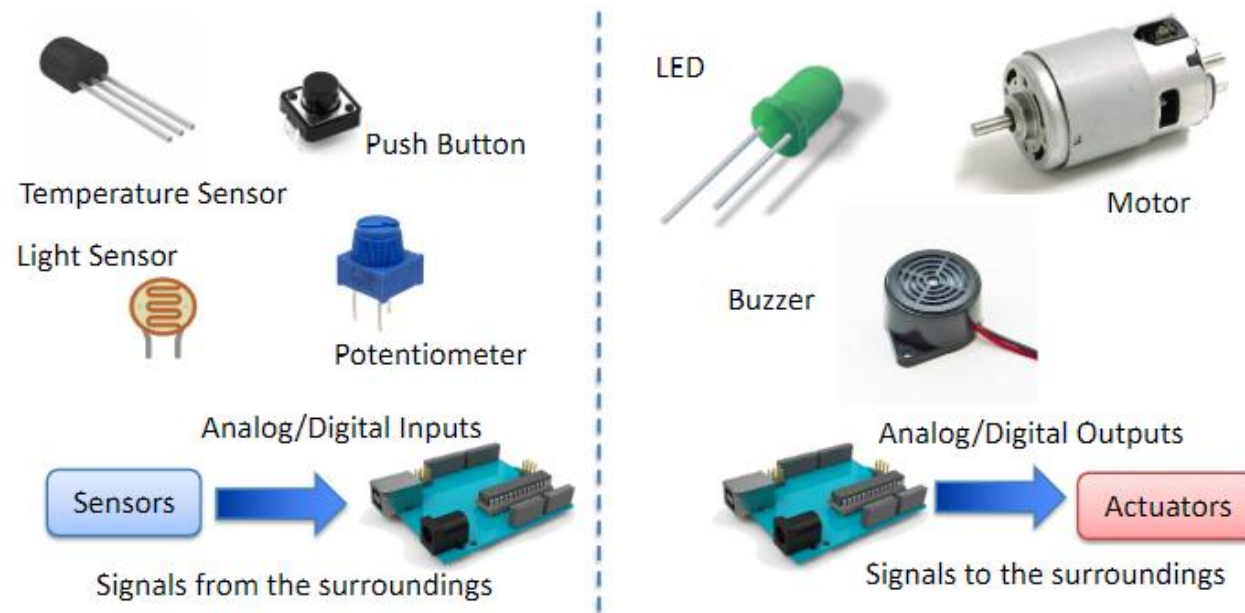
- You will need:



# Inputs and Outputs (Analog and Digital)



# Sensors and Actuators



# Sensors and Actuators

- A Sensor is a converter that measures a physical size and converts it to a signal that can be read by an instrument, data acquisition device, or an Arduino.
  - Examples: temperature sensor, pressure sensor, etc.
- An Actuator is a kind of motor that moves or controls a mechanism or system. It is powered by an energy source, typical electric current, hydraulic fluid pressure, or air pressure, and converts this energy into motion.
  - Examples: Engine, Pump, Valve, etc.

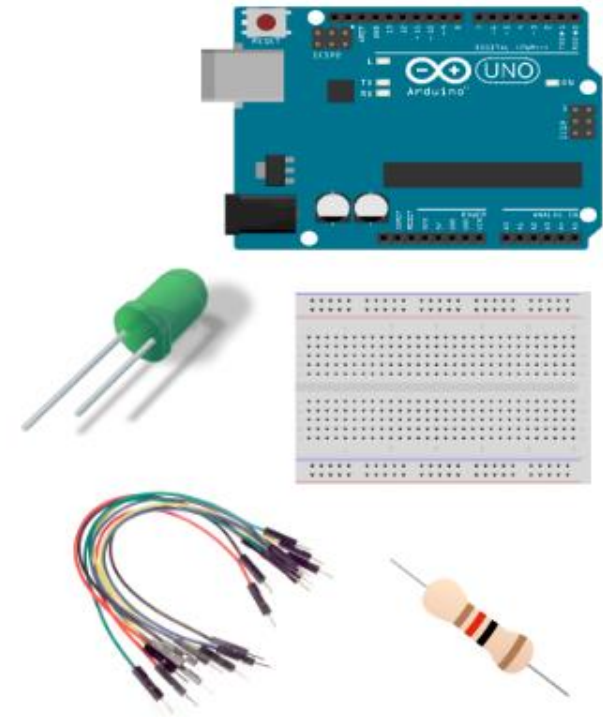
# Sensors and Actuators

- The sensors and actuators can be either digital or analog.
- Some sensors and actuators have been made for Arduino, while others need to be connected in some circuit to work properly with Arduino.
- Many of these come with ready-made libraries for Arduino, so they are easy to use.



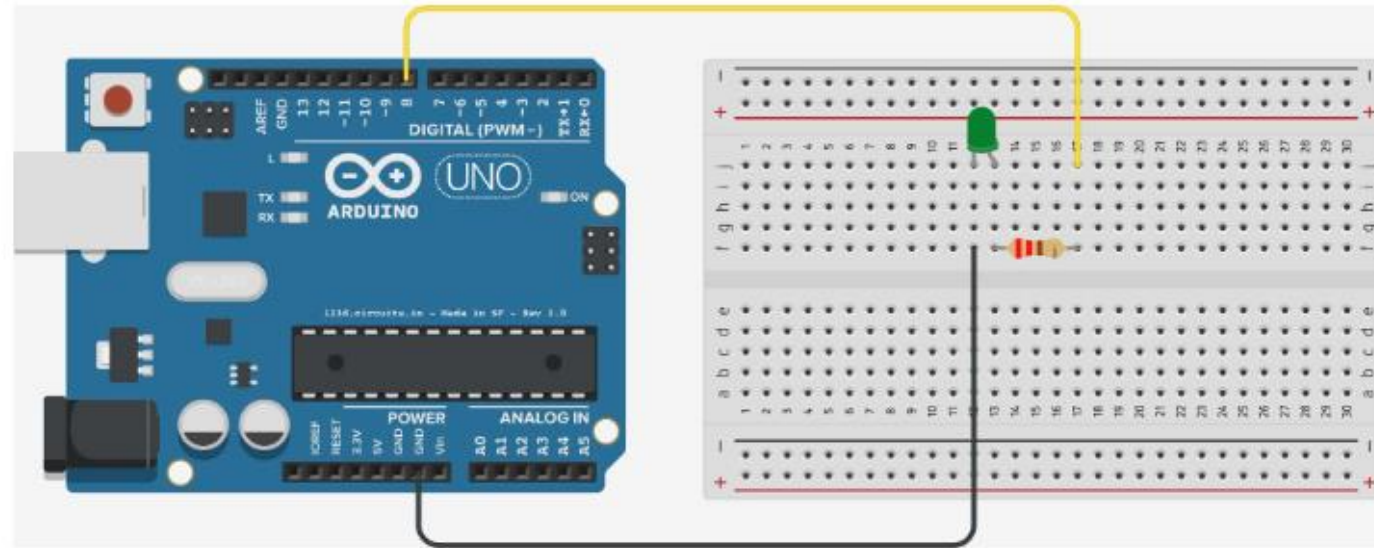
# Project example: External Blinking LED

- Equipment:
  - Arduino UNO
  - Breadboard
  - LED
  - Resistor,  $R = 270\Omega$
  - Wires (Jumper Wires)



# Project example: External Blinking LED

- Wiring



# Project example: External Blinking LED

- Programming

## Program Structure

```
//Globle variable
...

void setup()
{
    //Initialization
}

void loop()
{
    //Main Program
}
```

You need to use the following:

Which Pin (0, 1, 3, ...) are you using?

**pinMode**(pin, mode);

A Digital Pin can either be an INPUT or an OUTPUT. Since we shall use it to turn-on a LED, we set it to OUTPUT.

**digitalWrite**(pin, value);

Turn-on LED  
Turn-off LED

A Digital Pin can have 2 values, either HIGH or LOW

**delay**(ms);

The delay() function makes a small pause in milliseconds (ms), e.g., delay(1000) pause the program for 1 second

# Project example: External Blinking LED

- Program

Try it out

```
void setup()
{
    pinMode(8, OUTPUT);
}

void loop()
{
    digitalWrite(8, HIGH);    // Turn on the LED
    delay(1000);              // Wait for one second
    digitalWrite(8, LOW);    // Turn off the LED
    delay(1000);              // Wait for one second
}
```

# Practice

- Fritzing → Circuit Schema
- Arduino Kit

Your Turn

# References

- Book: Programming with Arduino, Hans-Petter Halvorsen, <https://www.halvorsen.blog/documents/technology/iot/arduino.php>
- Arduino IDE: <https://www.arduino.cc/en/Reference/HomePage>
- Fritzing tutorials : <http://projet.eu.org/pedago/sin/tutos/fritzing.pdf>
- Fritzing tutorials : <https://fritzing.org/learning/tutorials/>



[www.isep.fr](http://www.isep.fr)