

A paved road stretches into the distance through a dry, scrubby landscape under a clear blue sky. The word "GOAL" is painted in large, white, sans-serif letters across the center of the road.

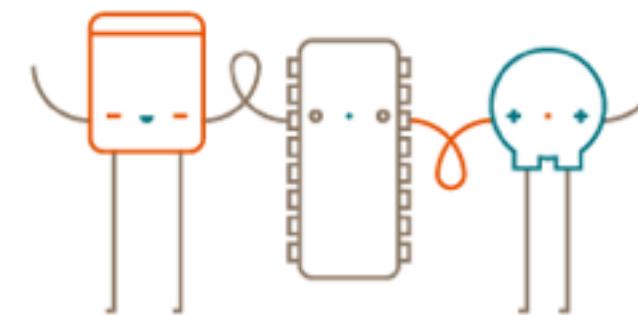
GOAL

# Leçon 0 : Installation d'IDE

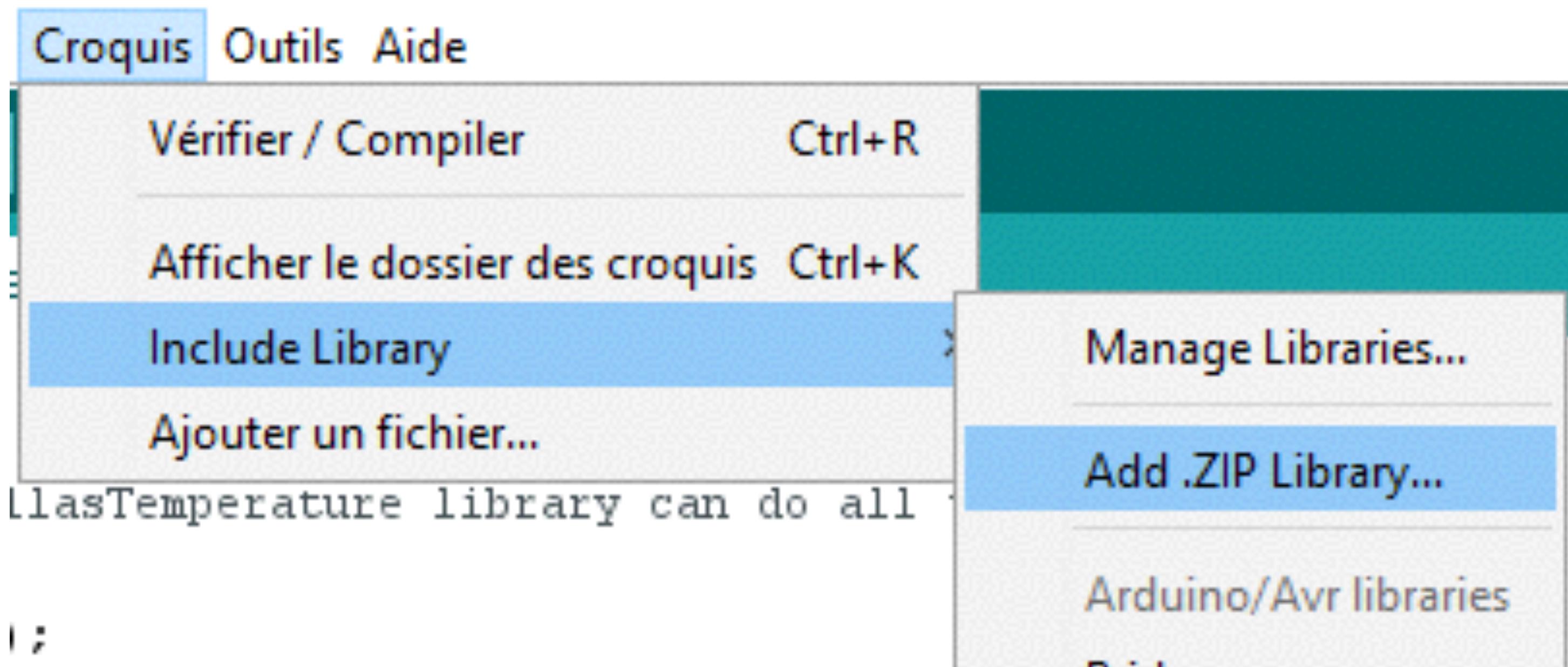


AN OPEN PROJECT WRITTEN, DEBUGGED,  
AND SUPPORTED BY ARDUINO.CC AND  
THE ARDUINO COMMUNITY WORLDWIDE

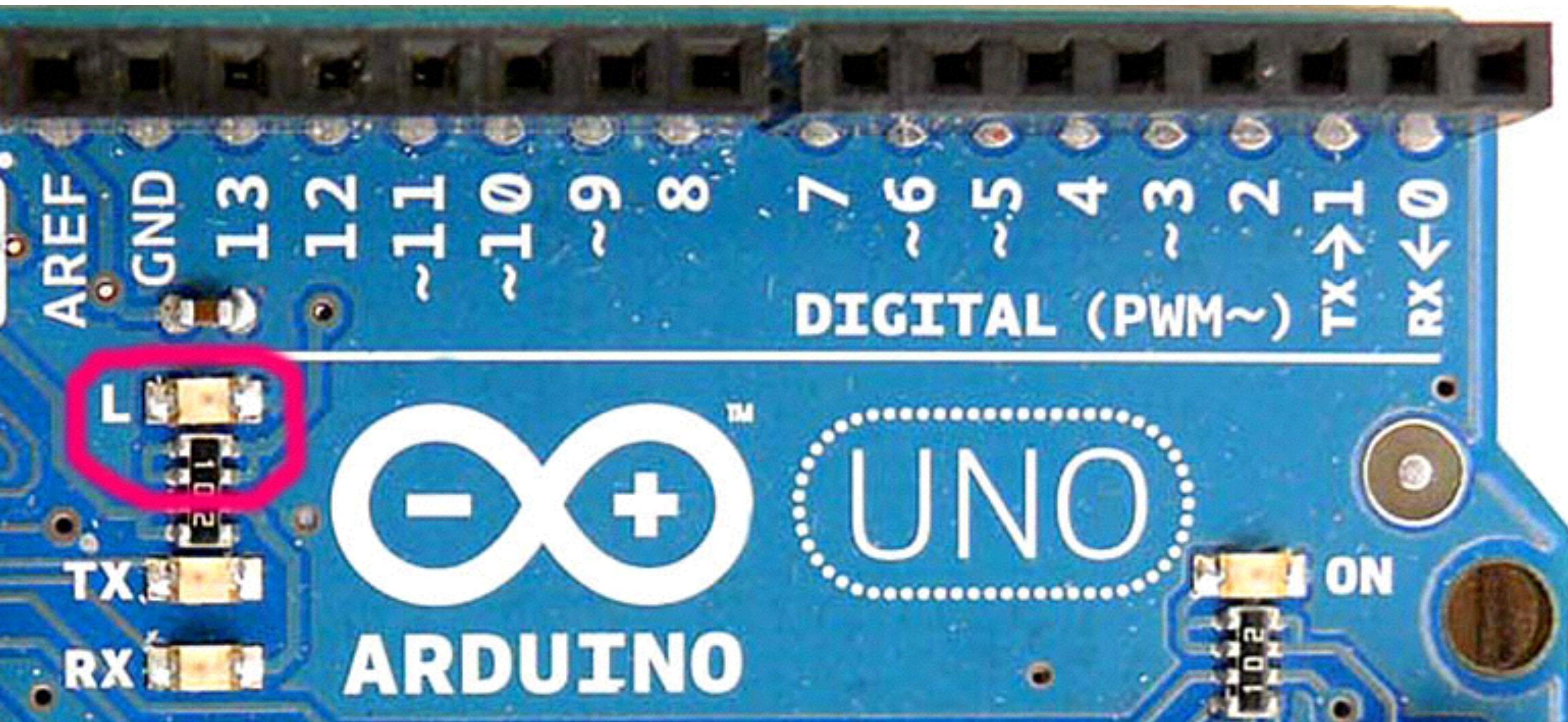
LEARN MORE ABOUT THE CONTRIBUTORS  
OF [ARDUINO.CC](http://ARDUINO.CC) on [arduino.cc/credits](http://arduino.cc/credits)



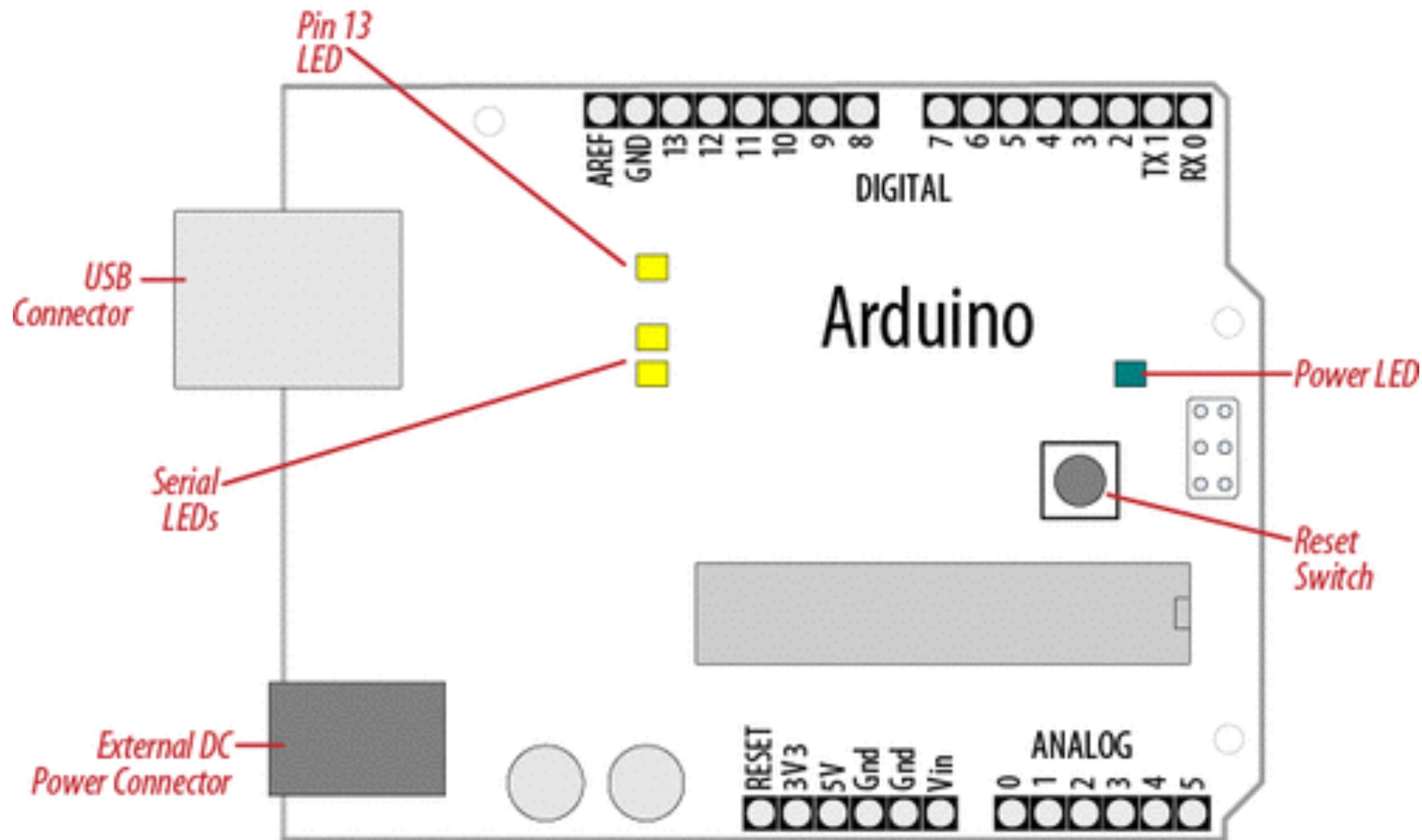
# Leçon 1: Ajouter les Bibliothèques



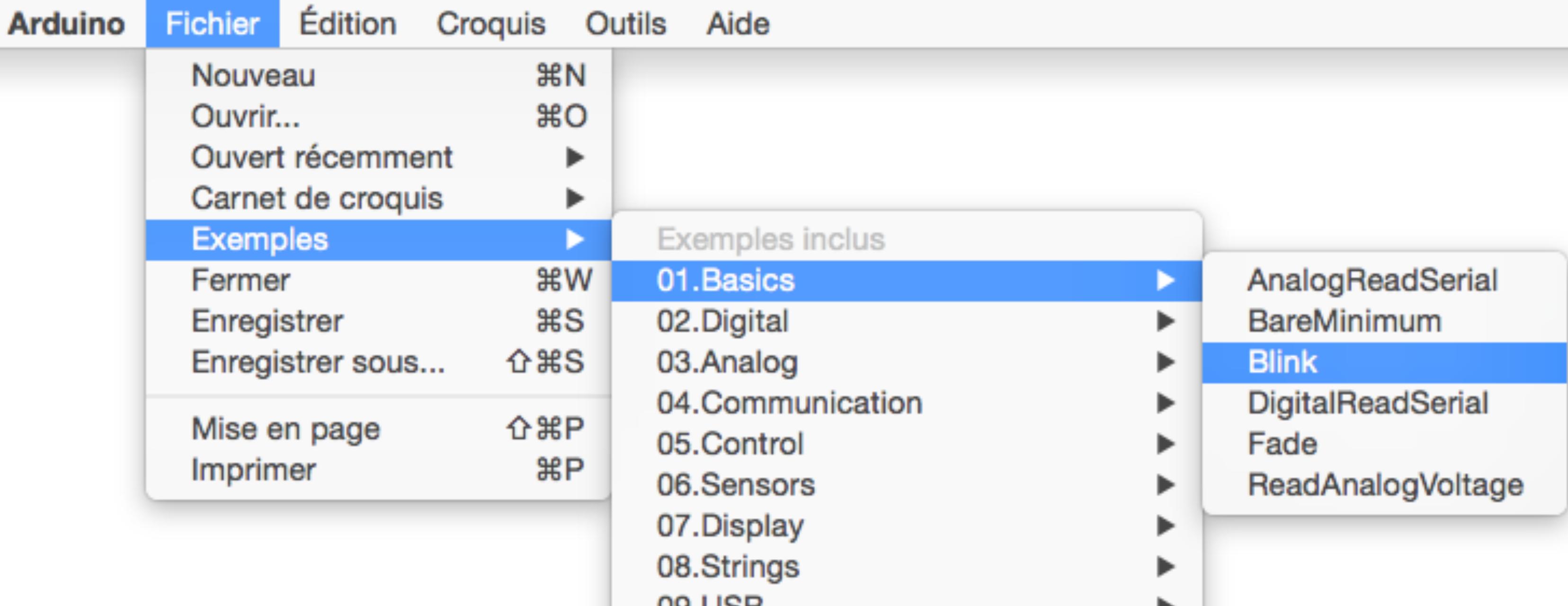
# Leçon 2: Clignotant



# Leçon 2: Clignotant



# Leçon 2: Clignotant



# Leçon 2: Clignotant



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.3". The code editor contains the "Blink" sketch, which blinks an LED connected to pin LED\_BUILTIN. The code is as follows:

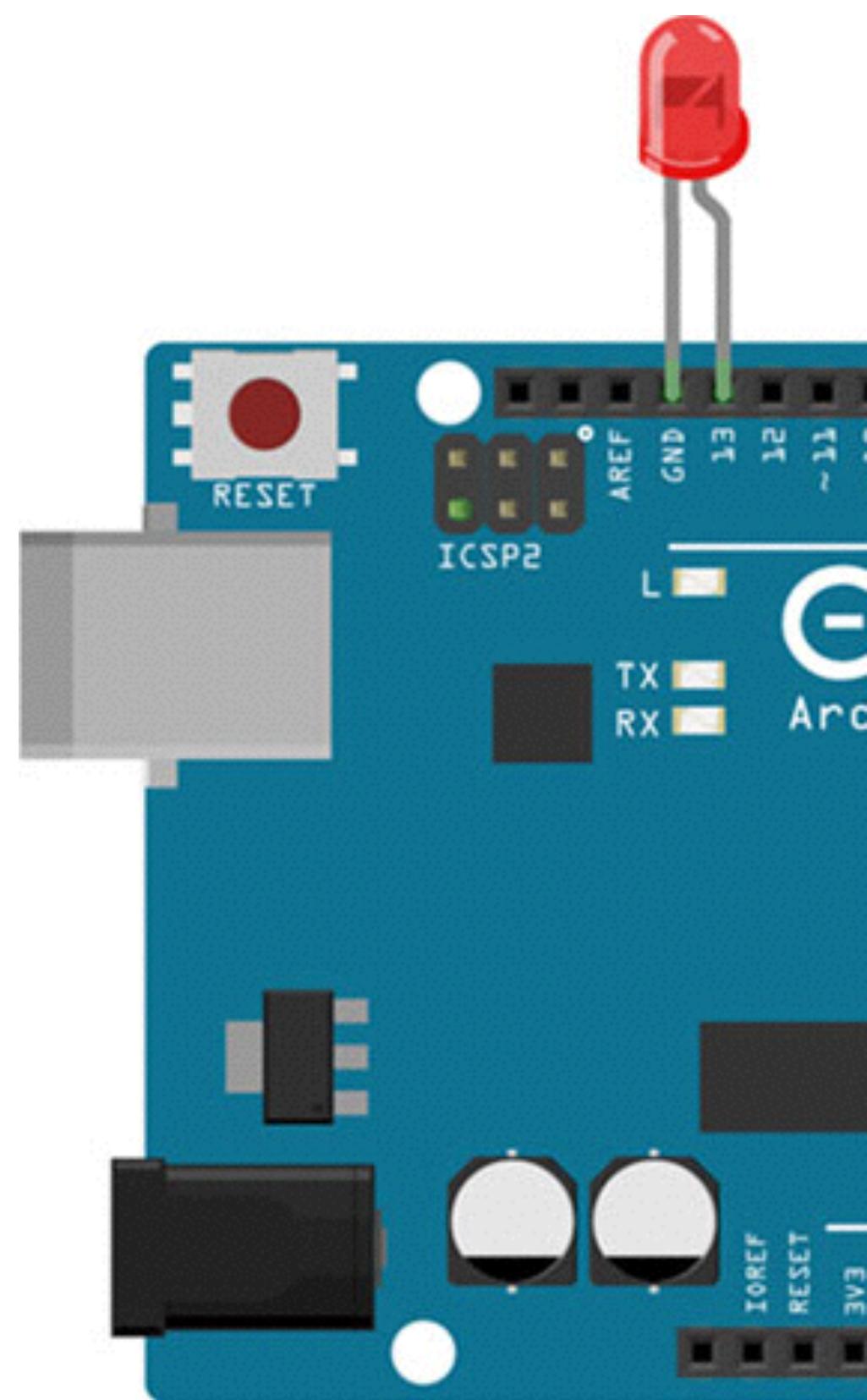
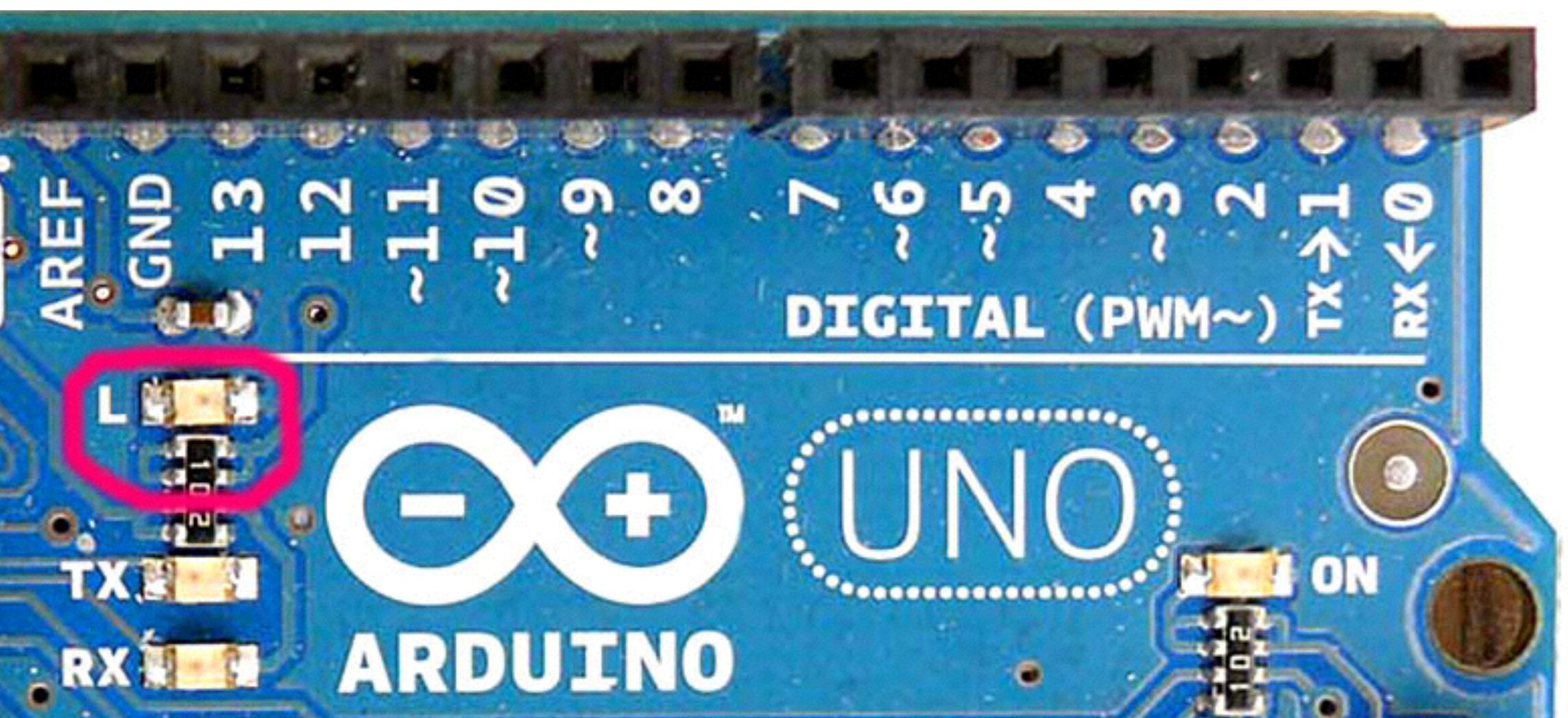
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

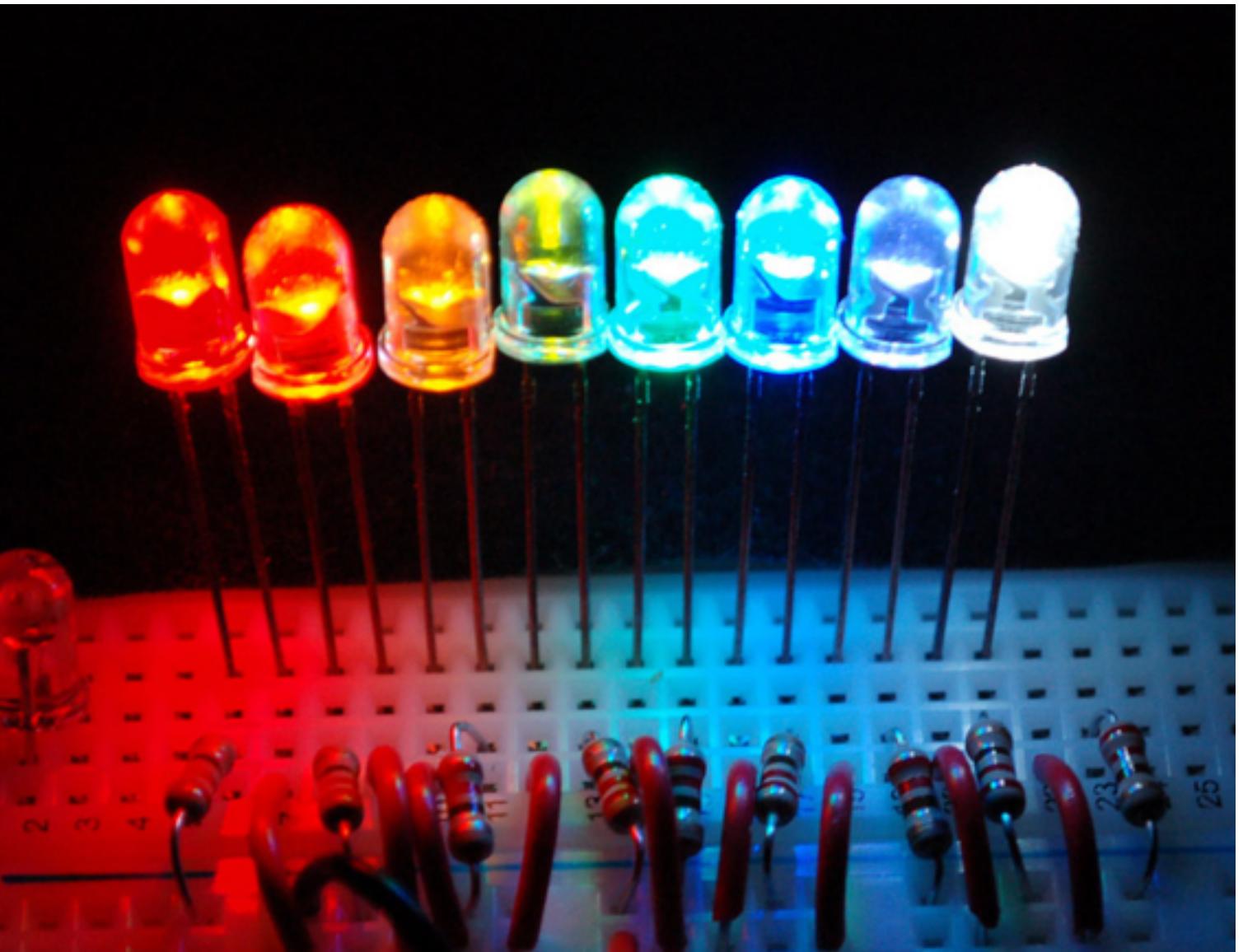
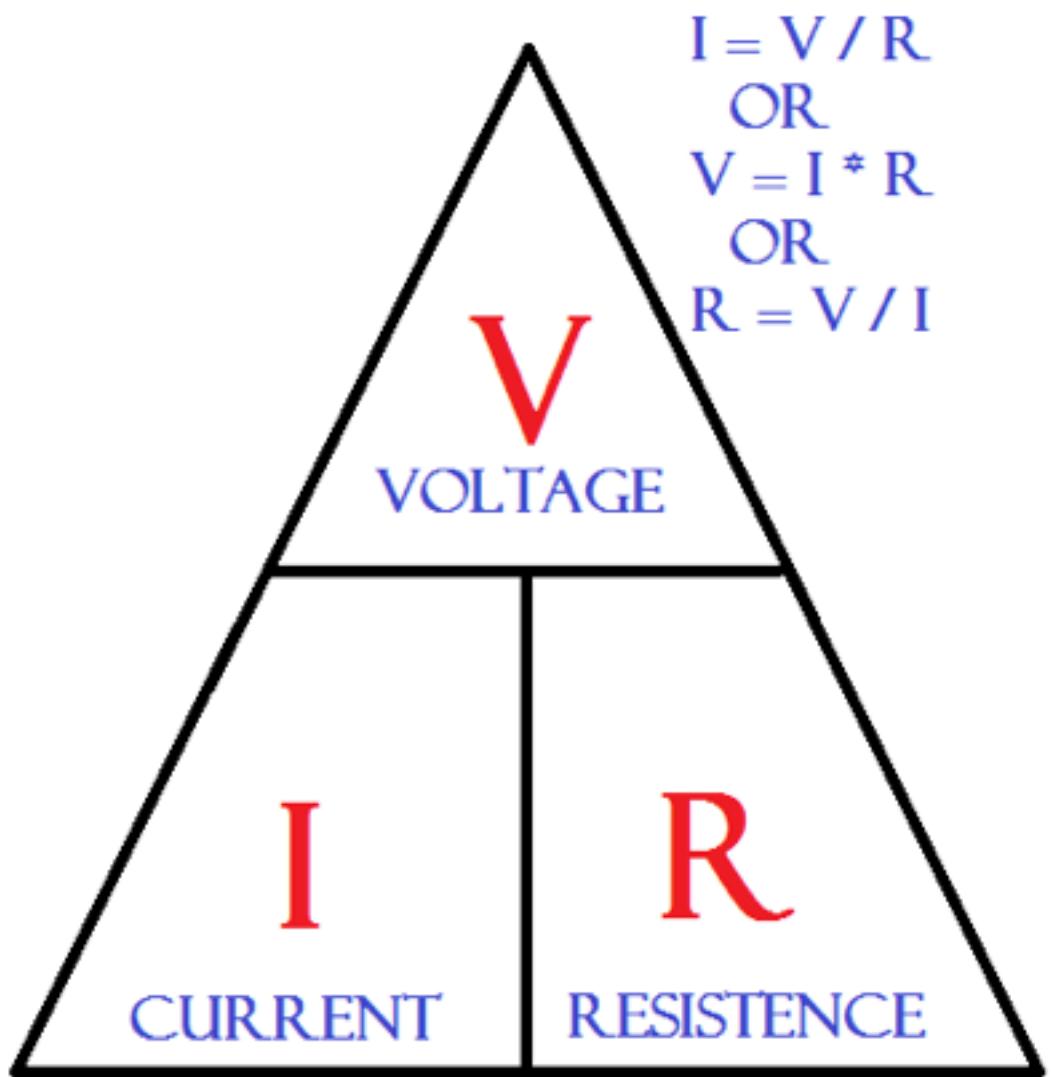
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                         // wait for a second
  digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage LOW
  delay(1000);                         // wait for a second
}
```

# Leçon 2: Clignotant

**LED\_BUILTIN** et/ou une **LED** (mais)



# Leçon 3: LED



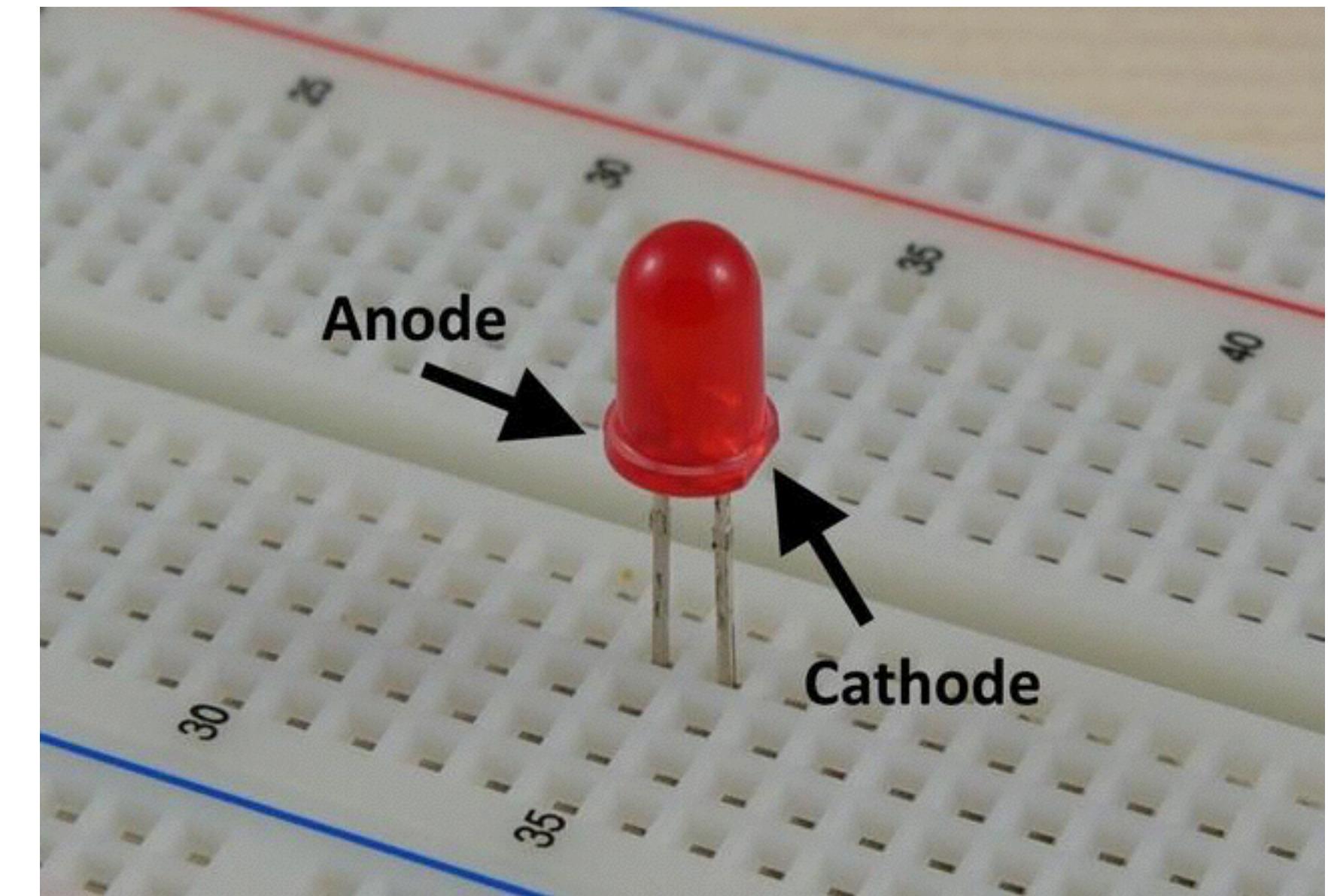
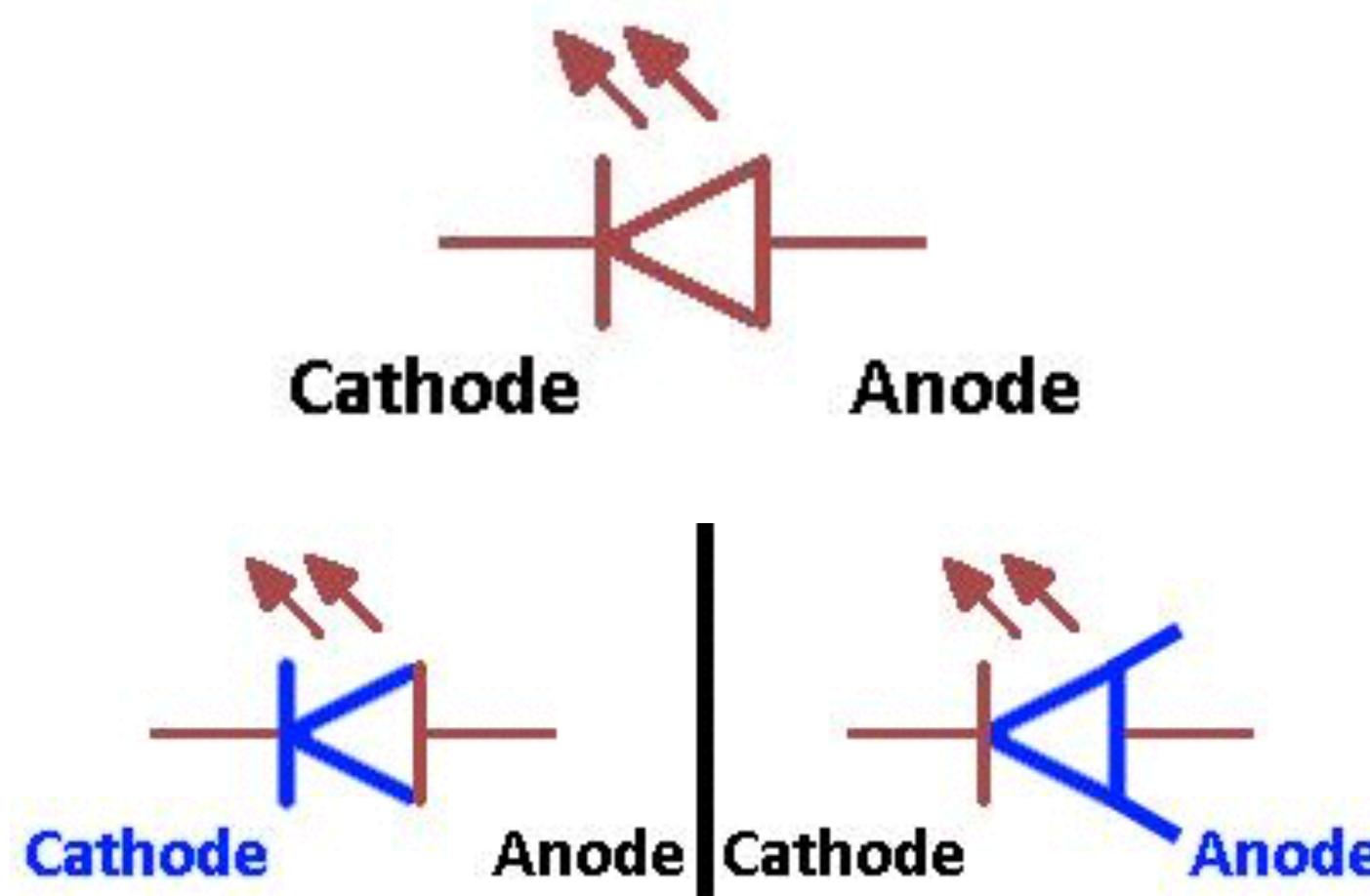
<https://learn.adafruit.com/all-about-leds/overview>

<https://www.carnetdumaker.net/articles/utiliser-les-entrees-sorties-numeriques-dune-carte-arduino-genuino/>

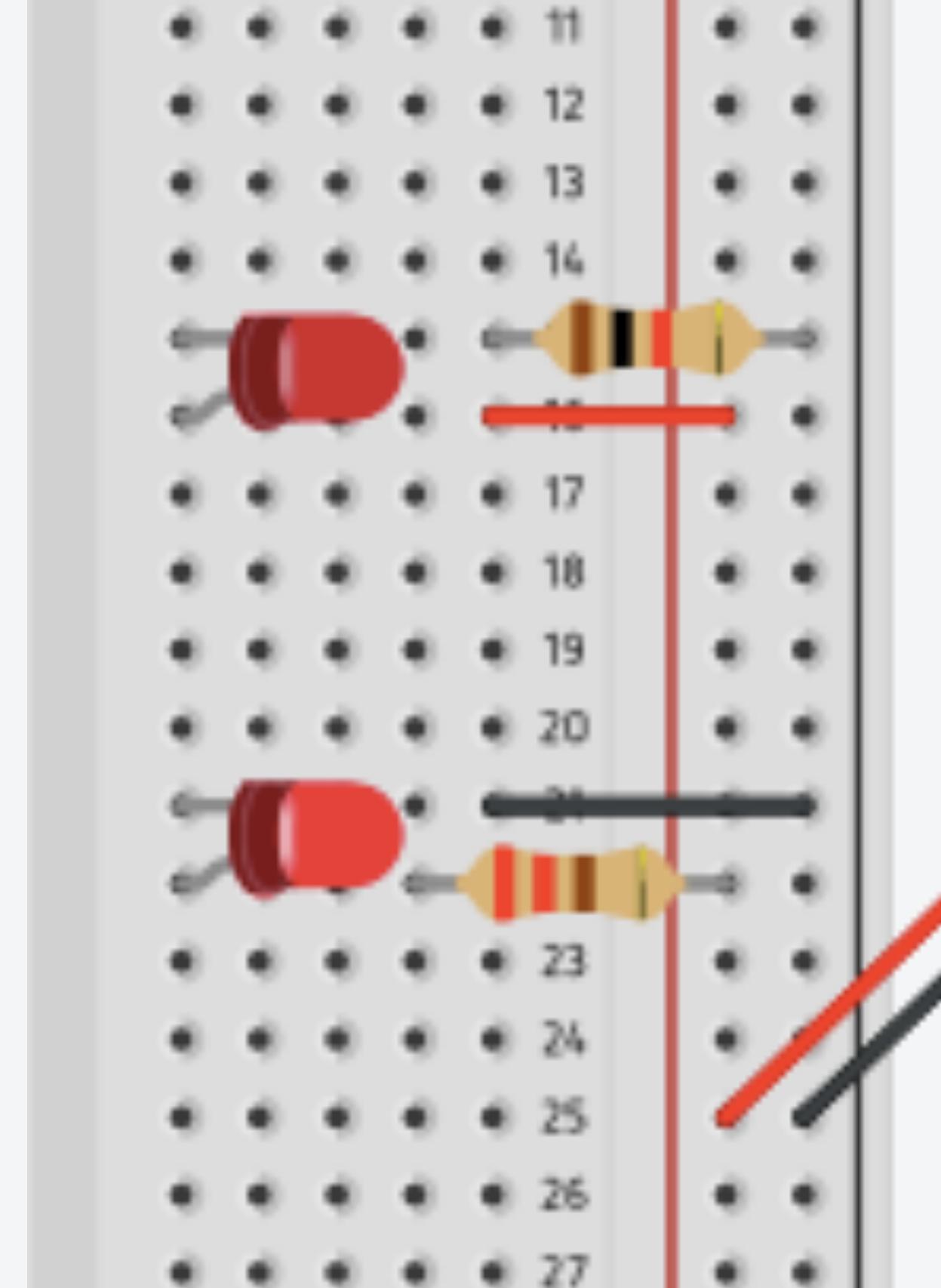
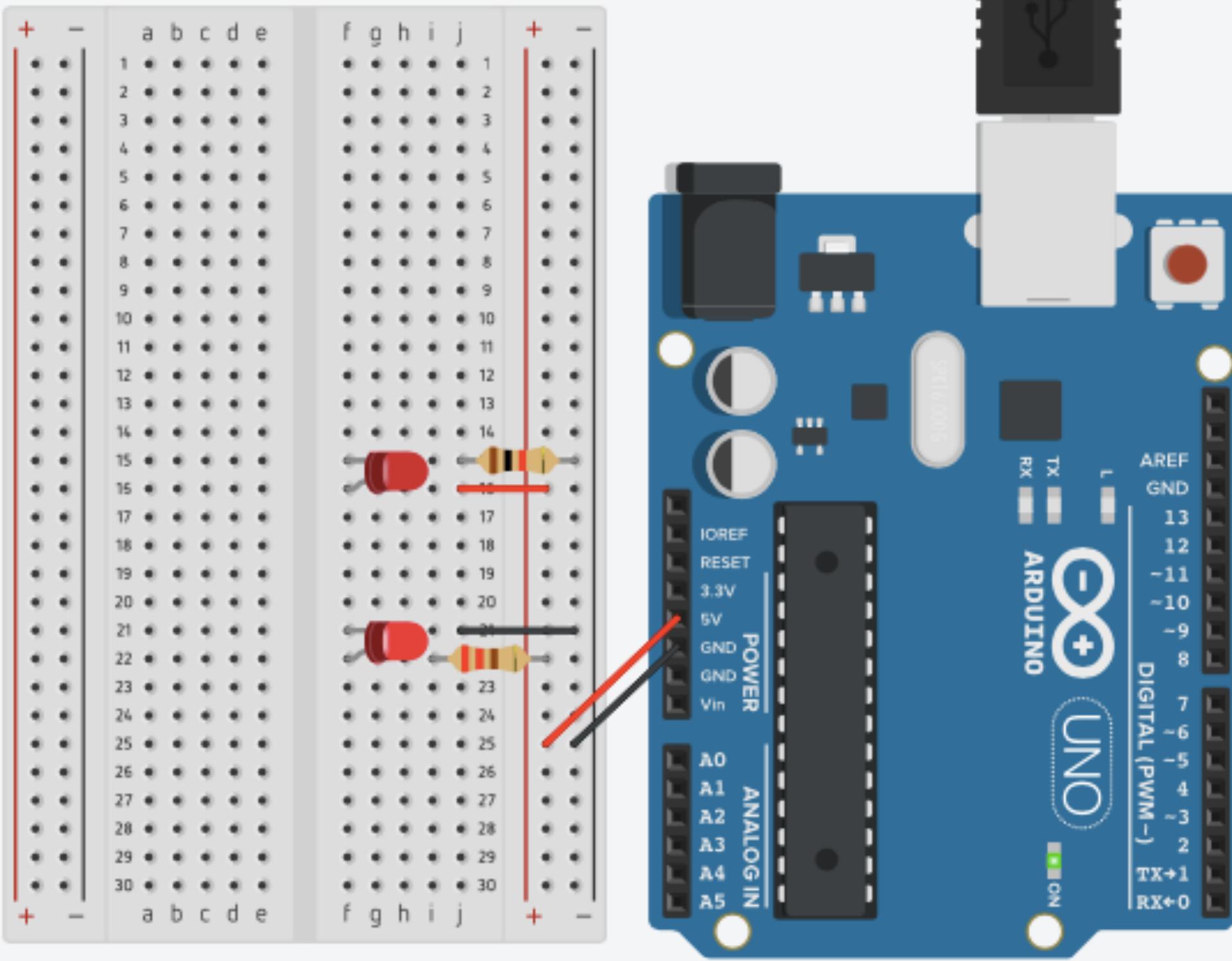
<http://www.filearchivehaven.com/2014/05/25/arduino-calculating-required-resistor-values/>

<https://www.hacktronics.com/Tools/led-resistor-calculator.html>

# Leçon 3: LED



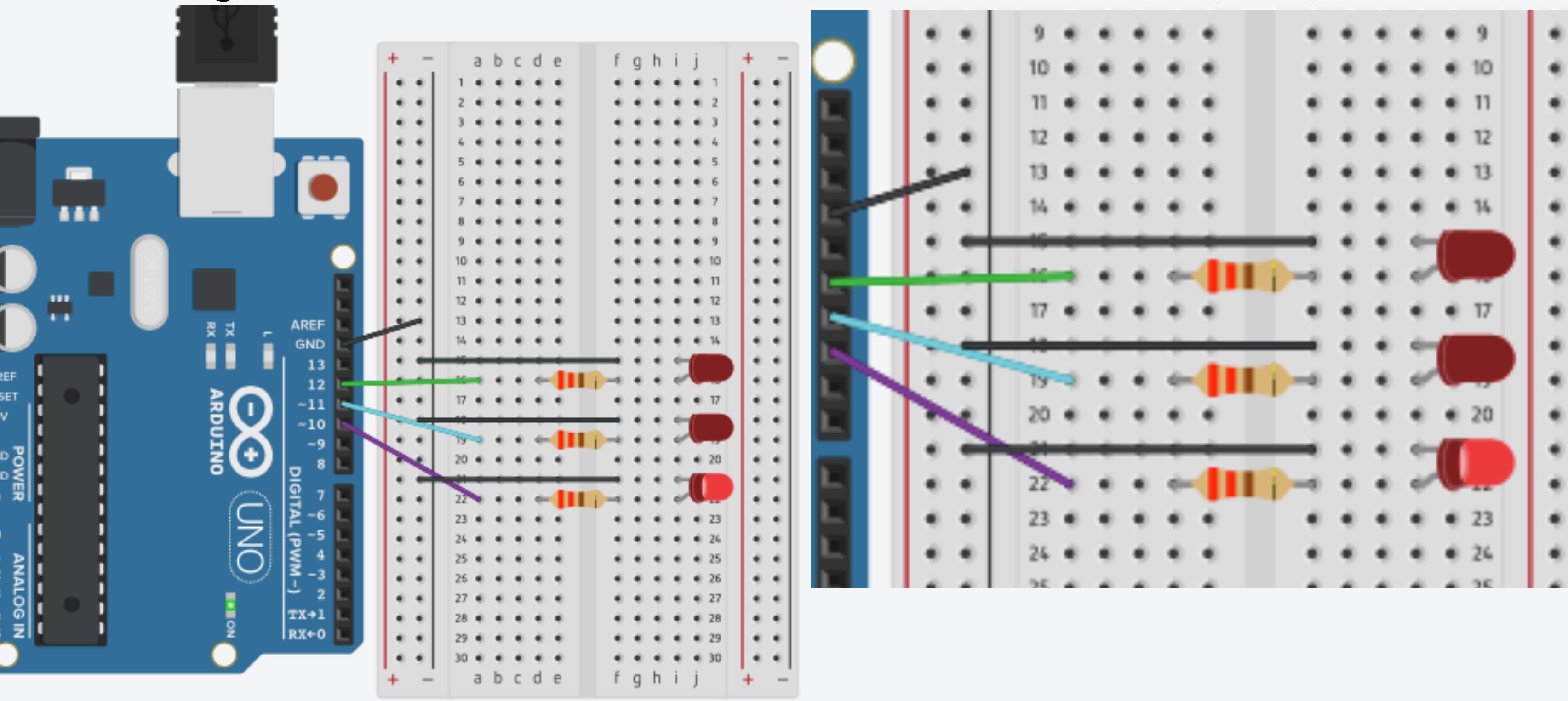
# Leçon 3: LED



# Leçon 3bis: Multi Blinked LED (3x)



# Leçon 3bis: Multi Blinked LED (3x)



<https://www.tinkercad.com/things/15emAOPRr3x-lecon-3bis-multi-led-blink/editel>

# Leçon 3bis: Multi Blinked LED (3x)

1 (Arduino Uno R3) ▾

Upload & Run

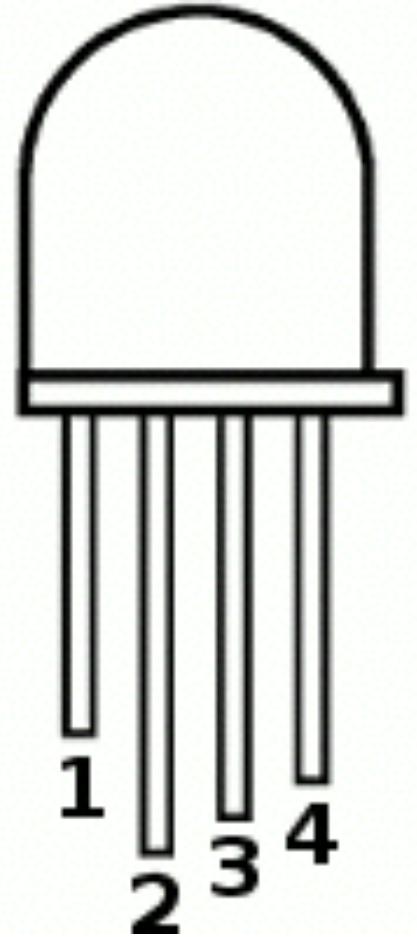
Libraries

```
1 // Pin 13 has an LED connected on most Arduino boards.
2 // give it a name:
3 int led1 = 12;
4 int led2 = 11;
5 int led3 = 10;
6
7 // the setup routine runs once when you press reset:
8 void setup() {
9     // initialize the digital pin as an output.
10    pinMode(led1, OUTPUT);
11    pinMode(led2, OUTPUT);
12    pinMode(led3, OUTPUT);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17     digitalWrite(led1, HIGH);
18     digitalWrite(led2, LOW);
19     digitalWrite(led3, LOW);
20     delay(1000);
21     digitalWrite(led1, LOW);
22     digitalWrite(led2, HIGH);
23     digitalWrite(led3, LOW);
24     delay(1000);
25     digitalWrite(led1, LOW);
26     digitalWrite(led2, LOW);
27     digitalWrite(led3, HIGH);
28     delay(1000);
29 }
```

# Leçon 3bis: Multi Blinked LED Inverse(3x)

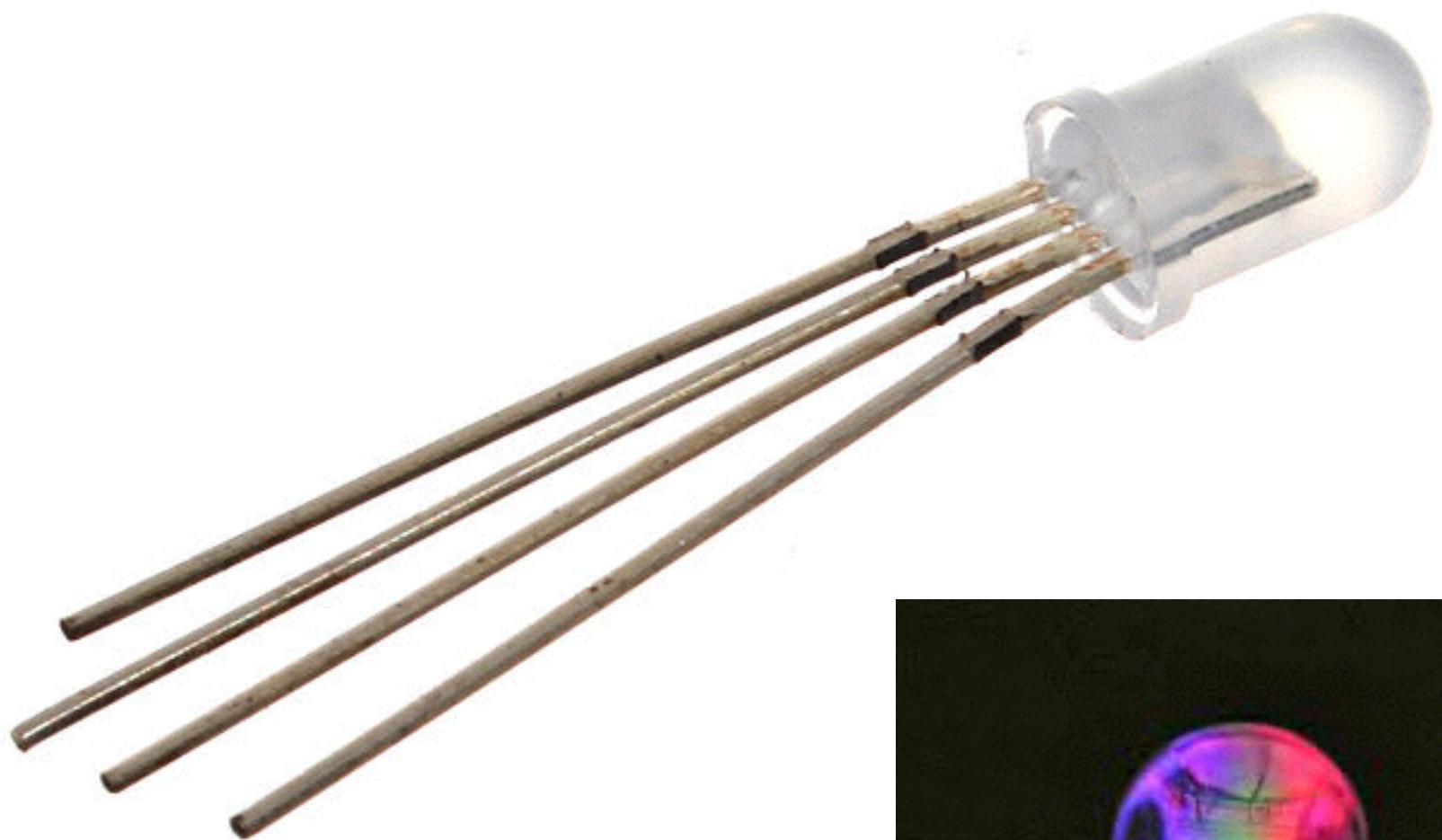
<https://www.tinkercad.com/things/8YaoMgFZYpy-lecon-3bis-multi-led-blink-inverse-resistances/editel>

# Leçon 4: RGB LED



**RGB LED**

- 1: Green (+)
- 2: Ground (-)
- 3: Blue (+)
- 4: Red (+)



# Leçon 4: RGB LED (facile)



**RELEVEZ LE DÉFI**

Réalisez un feu rouge

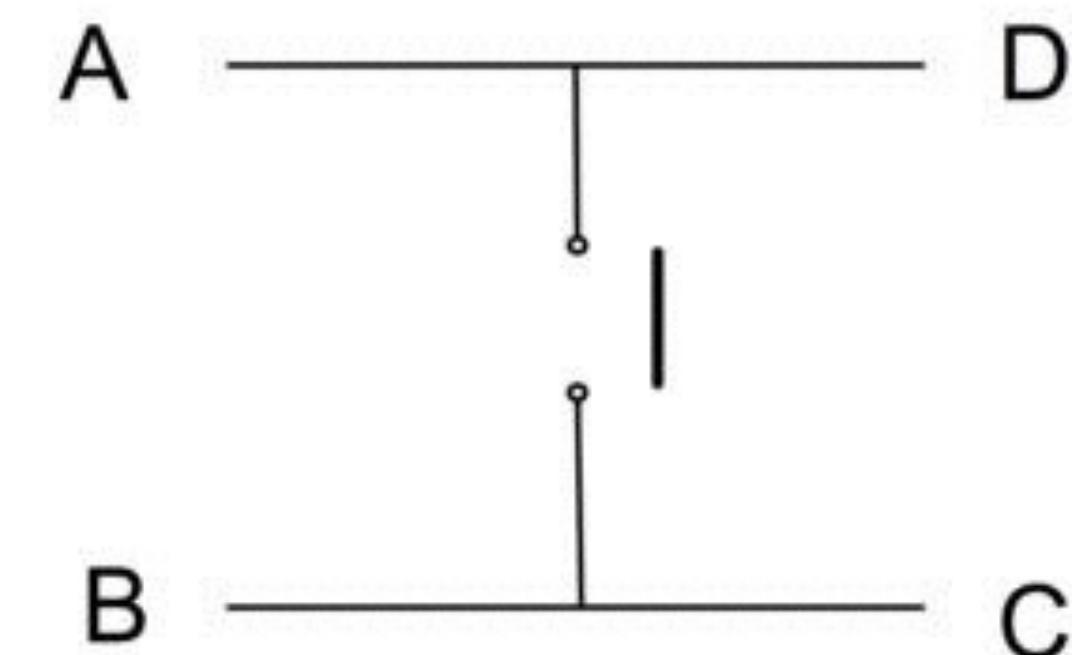
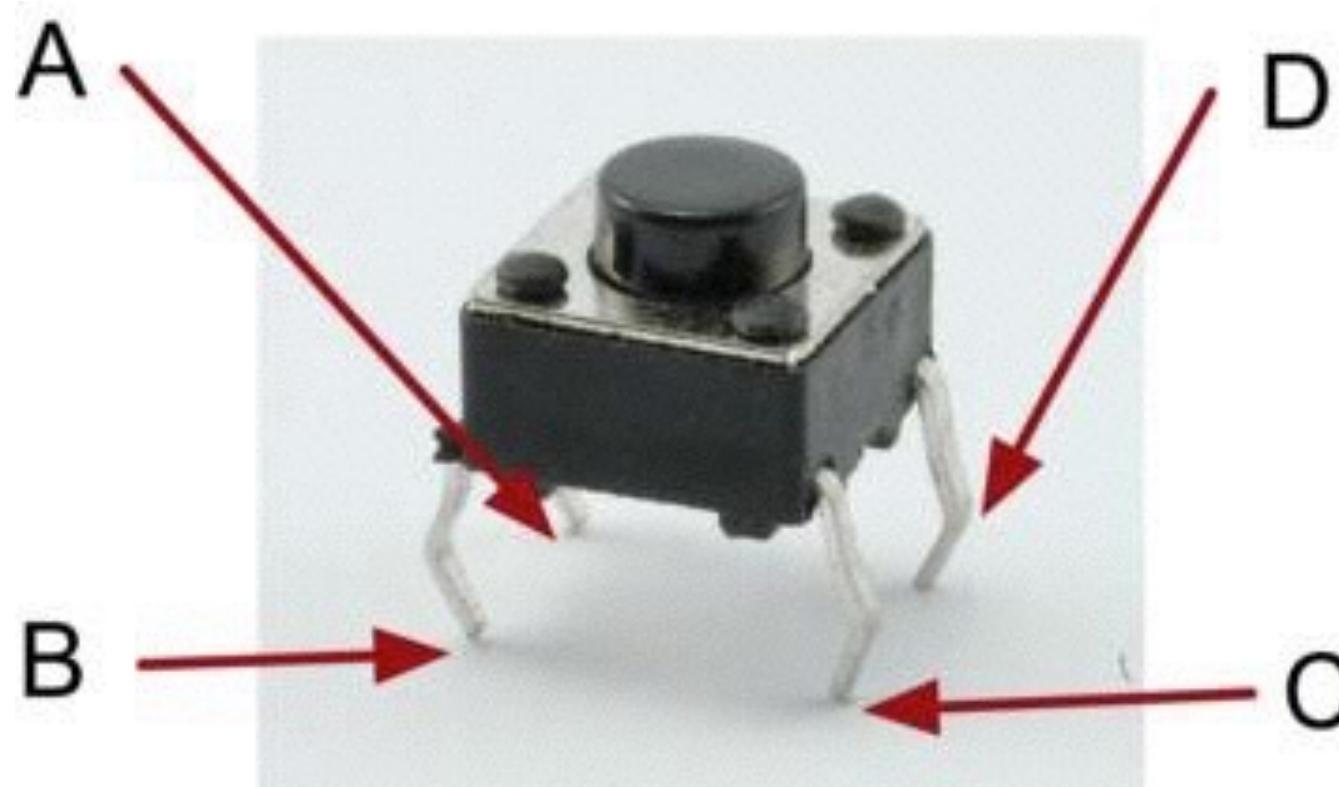


# Leçon 4: RGB LED (moyen)



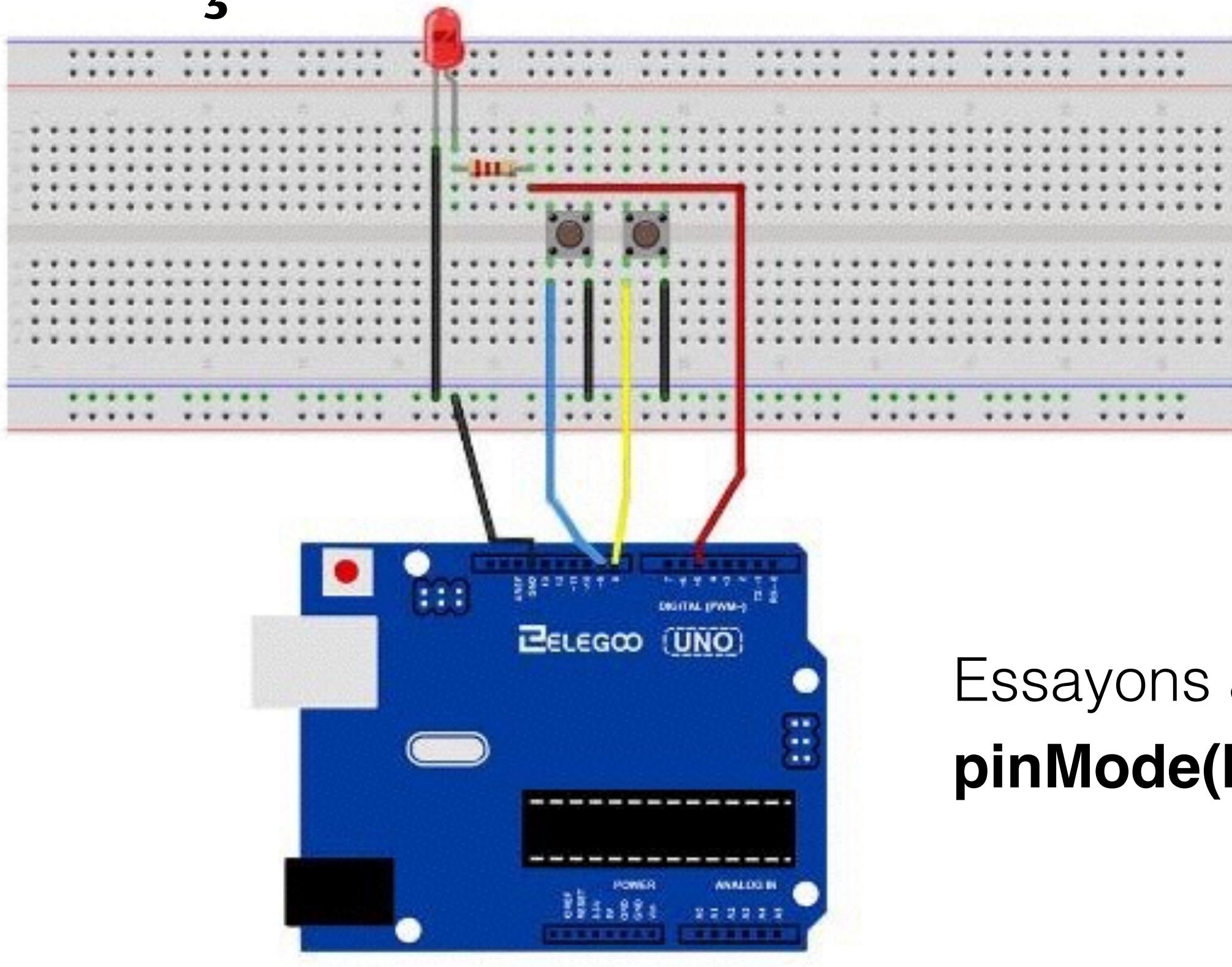
Réalisez un arc en ciel

# Leçon 5: Entrées Numériques



Essayons avec un **pinMode(bouton, INPUT);**

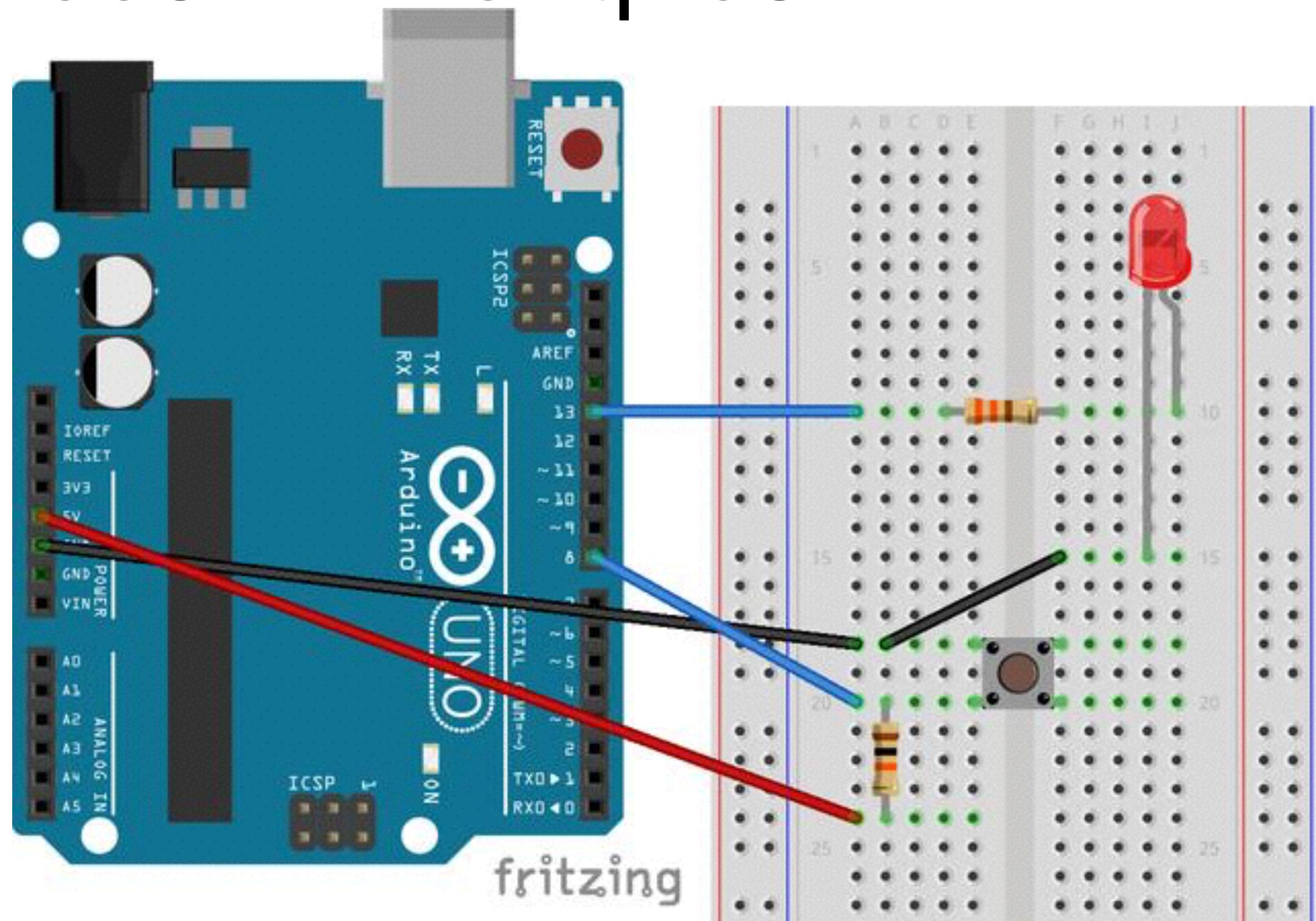
# Leçon 5: Entrées Numériques



Essayons avec un  
**pinMode(bouton, INPUT\_PULLUP);**

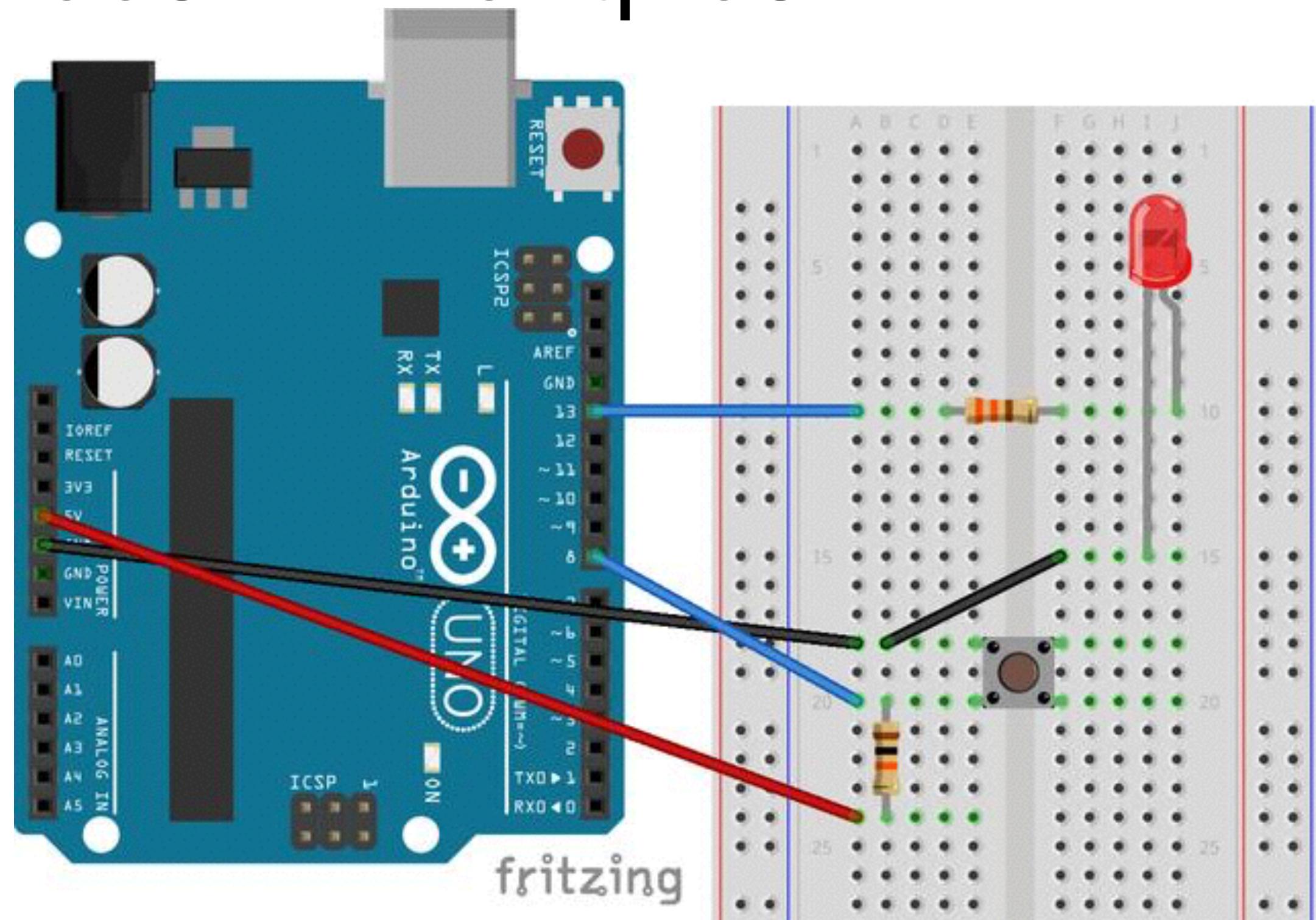
# Leçon 5: Entrées Numériques

Les résistances de tirage à la rescouasse



# Leçon 5: Entrées Numériques

Les résistances de tirage à la ressource

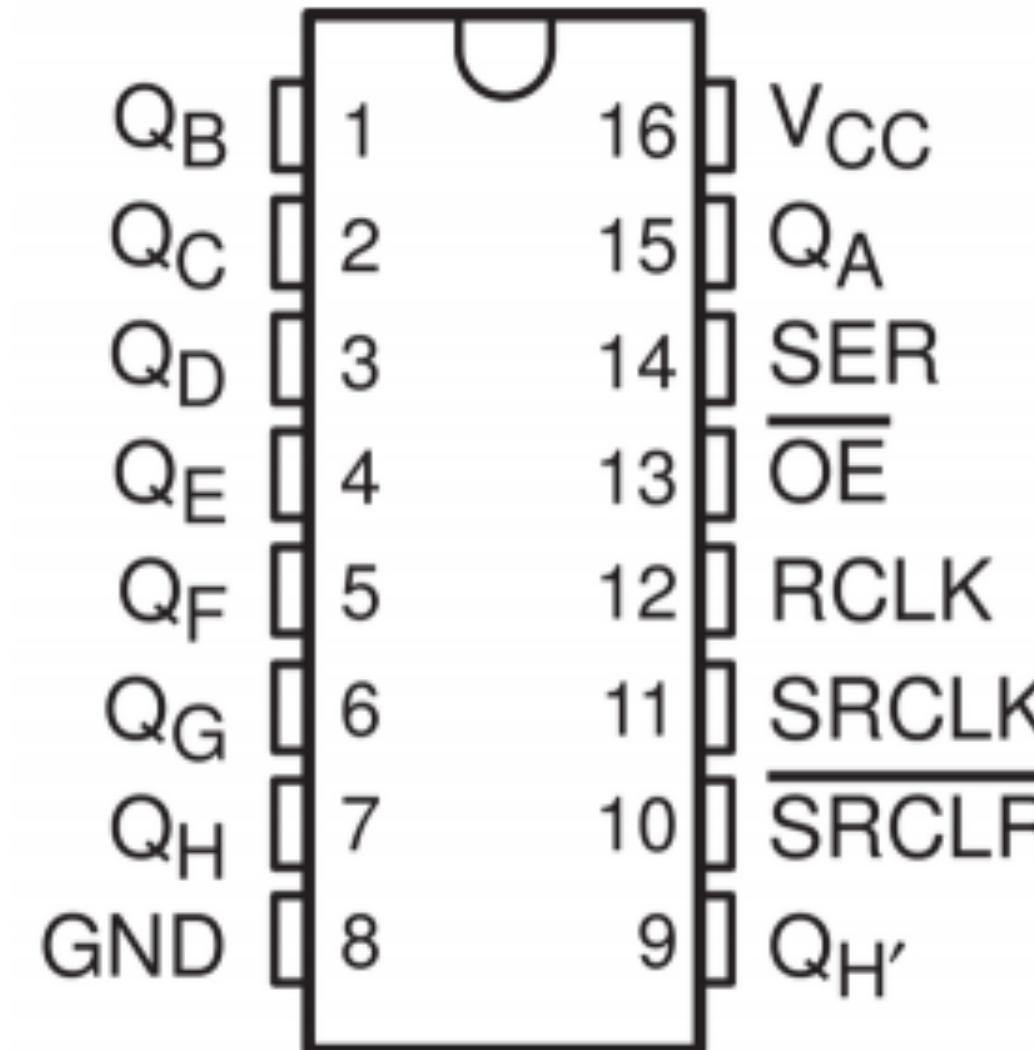
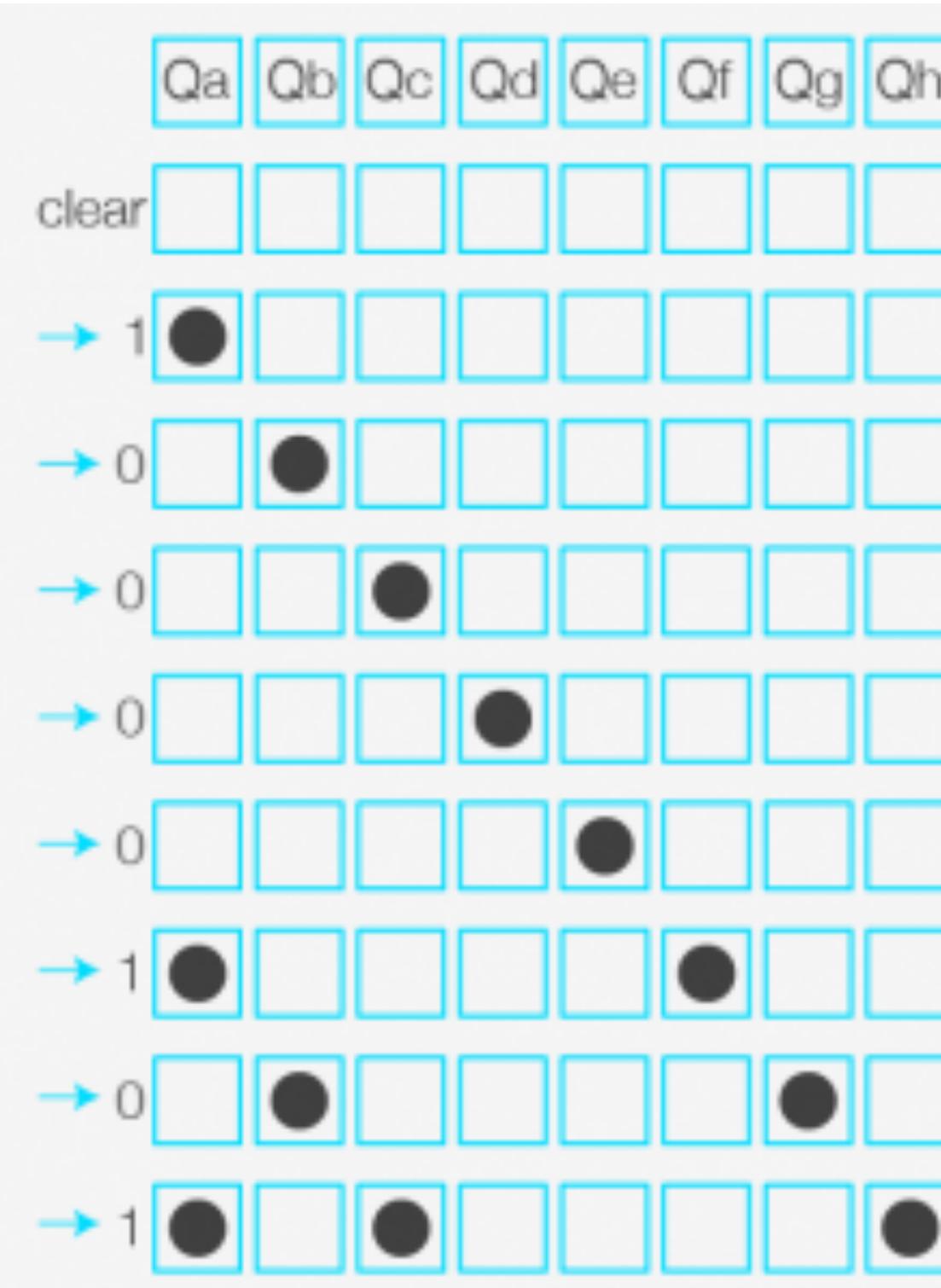


# Leçon 5: Entrées Numériques

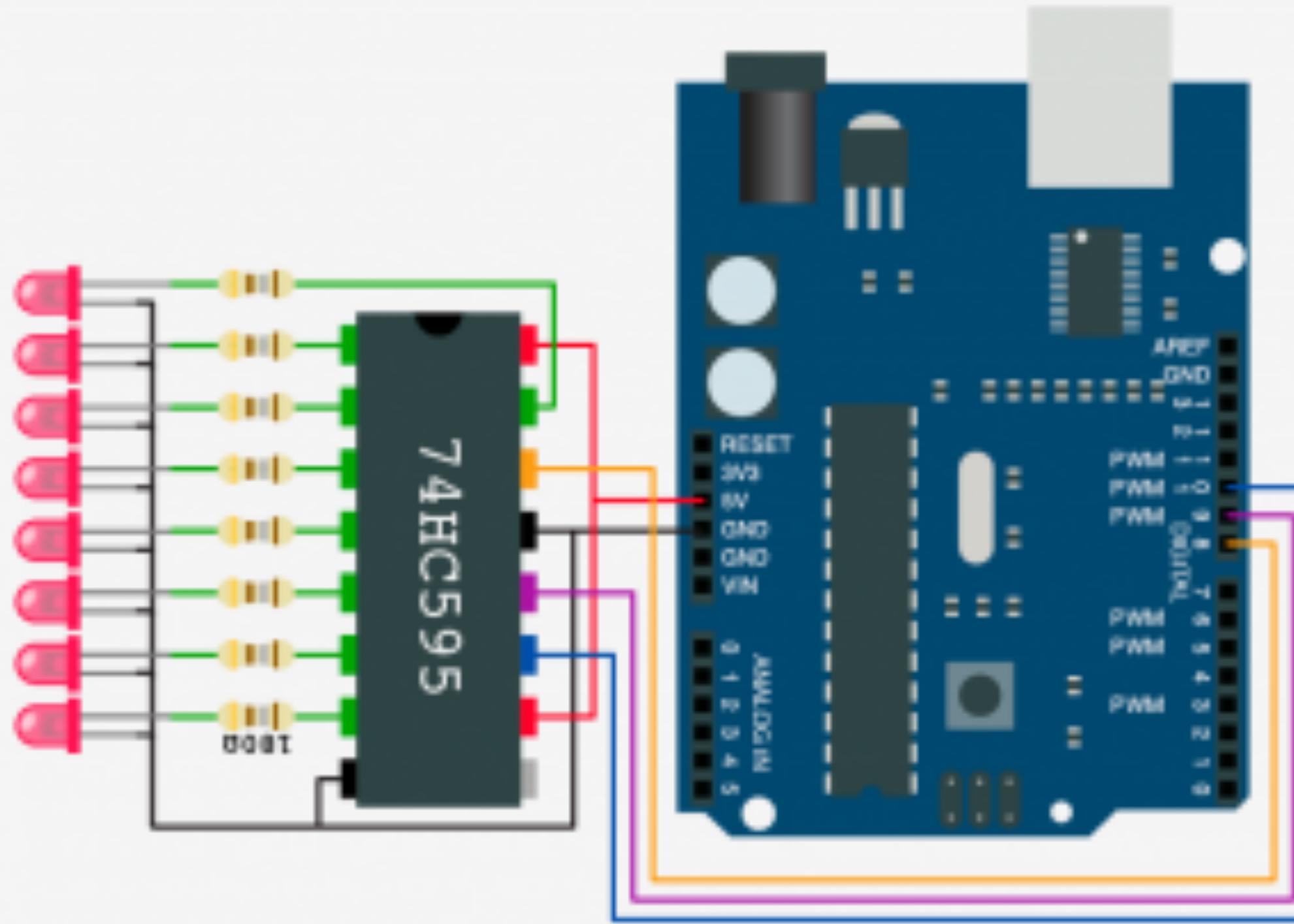


Je voudrais que la LED s'allume quand **j'appuie** sur le bouton

# Leçon 6: Huit LEDs avec 74HC595



# Leçon 6: Huit LEDs avec 74HC595



Tout allumer  
Tout éteindre  
1 sur 2



**RELEVEZ LE DÉFI**  
**Attention G>D>G**





**RELEVEZ LE DÉFI**  
**Avec 16 LEDs**



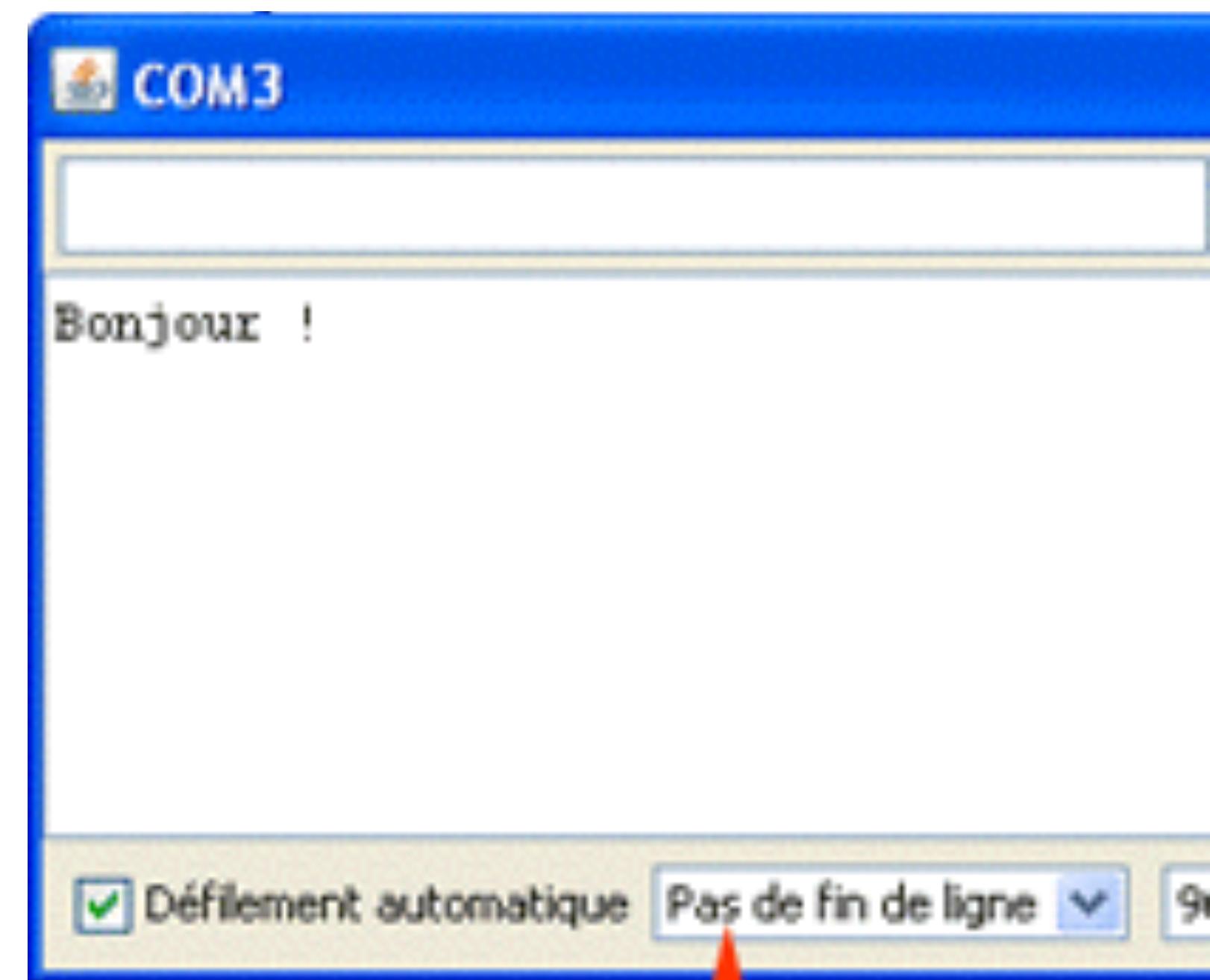
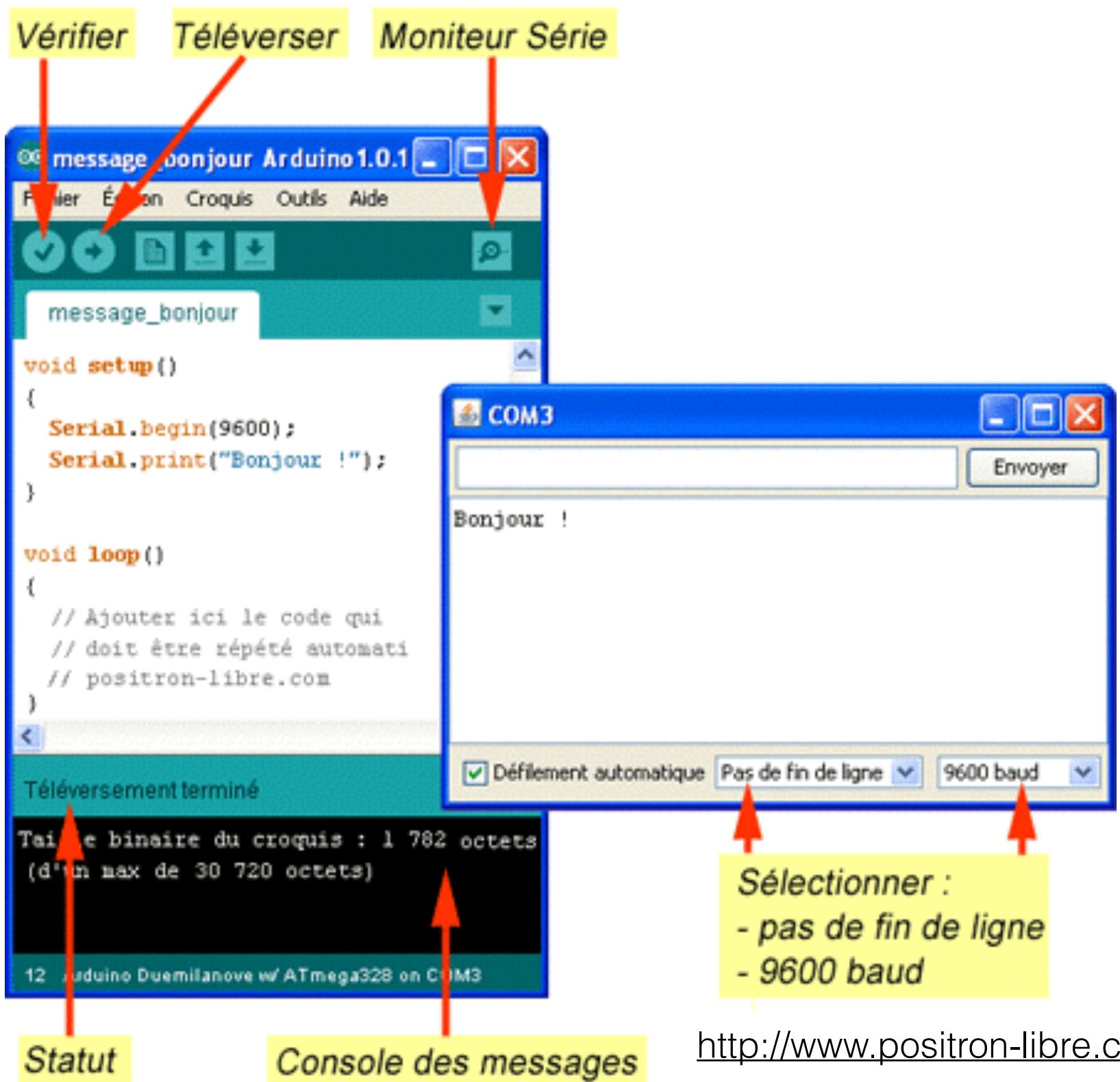
# Leçon 7: Moniteur Série

Hello, World.

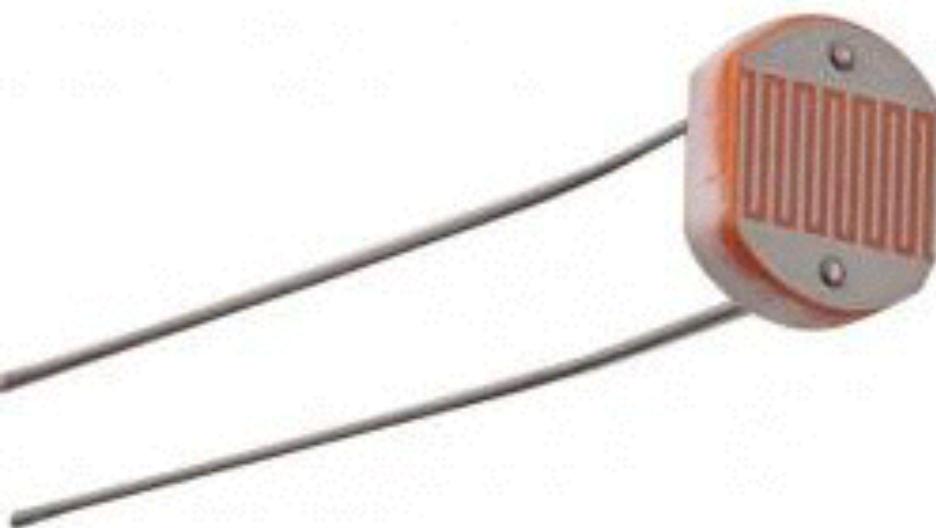


RELEVEZ LE DÉFI

# Leçon 7: Moniteur Série



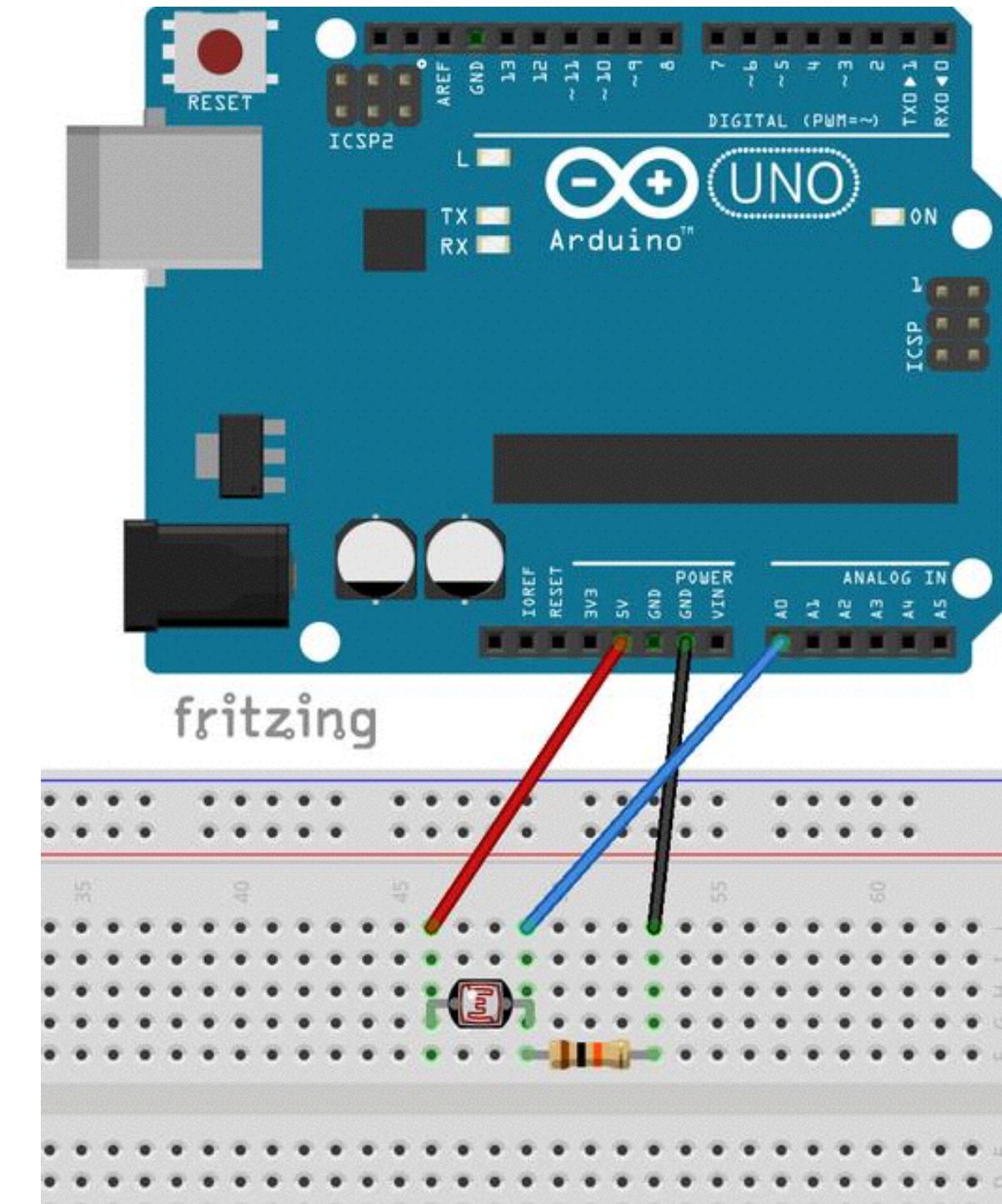
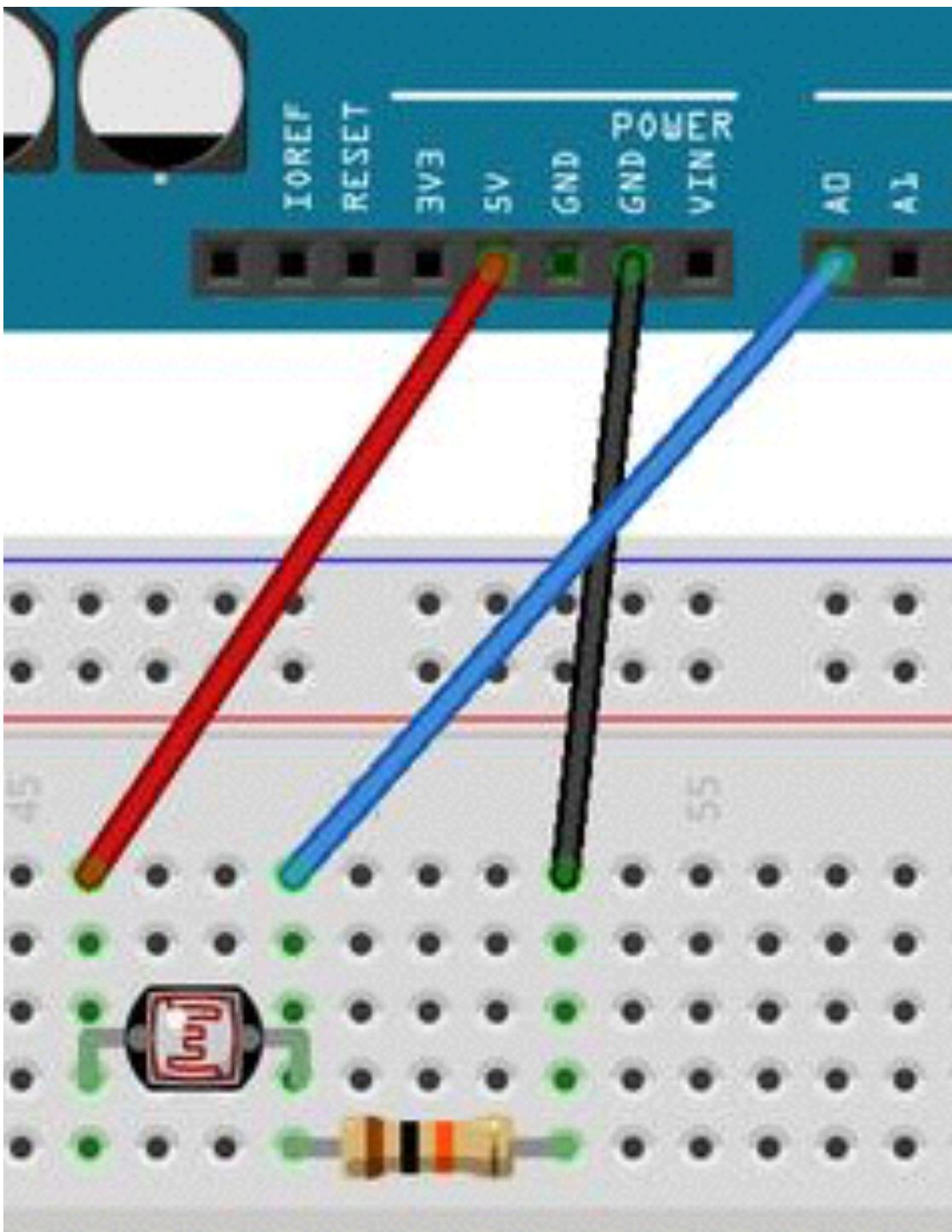
# Leçon 8: Cellule photorésistante



(variante)



# Leçon 8: Cellule photorésistante



# Leçon 8: Photorésistante (velleman VMA 407)



<http://www.velleman.eu/products/view/?country=be&lang=fr&id=435572>

[http://www.velleman.eu/downloads/29/vma407\\_a4v01.pdf](http://www.velleman.eu/downloads/29/vma407_a4v01.pdf)

# Leçon 8: Photorésistante (velleman VMA 407)

## 5. Overview

Measure light levels with this sensor module.

This module uses a Light Dependant Resistor (LDR) to produce an output voltage that varies with light level.

Connect it straight to your Arduino® to build a night/day sensor, a sun tracker or combine it with a laser module to make a laser trip wire.

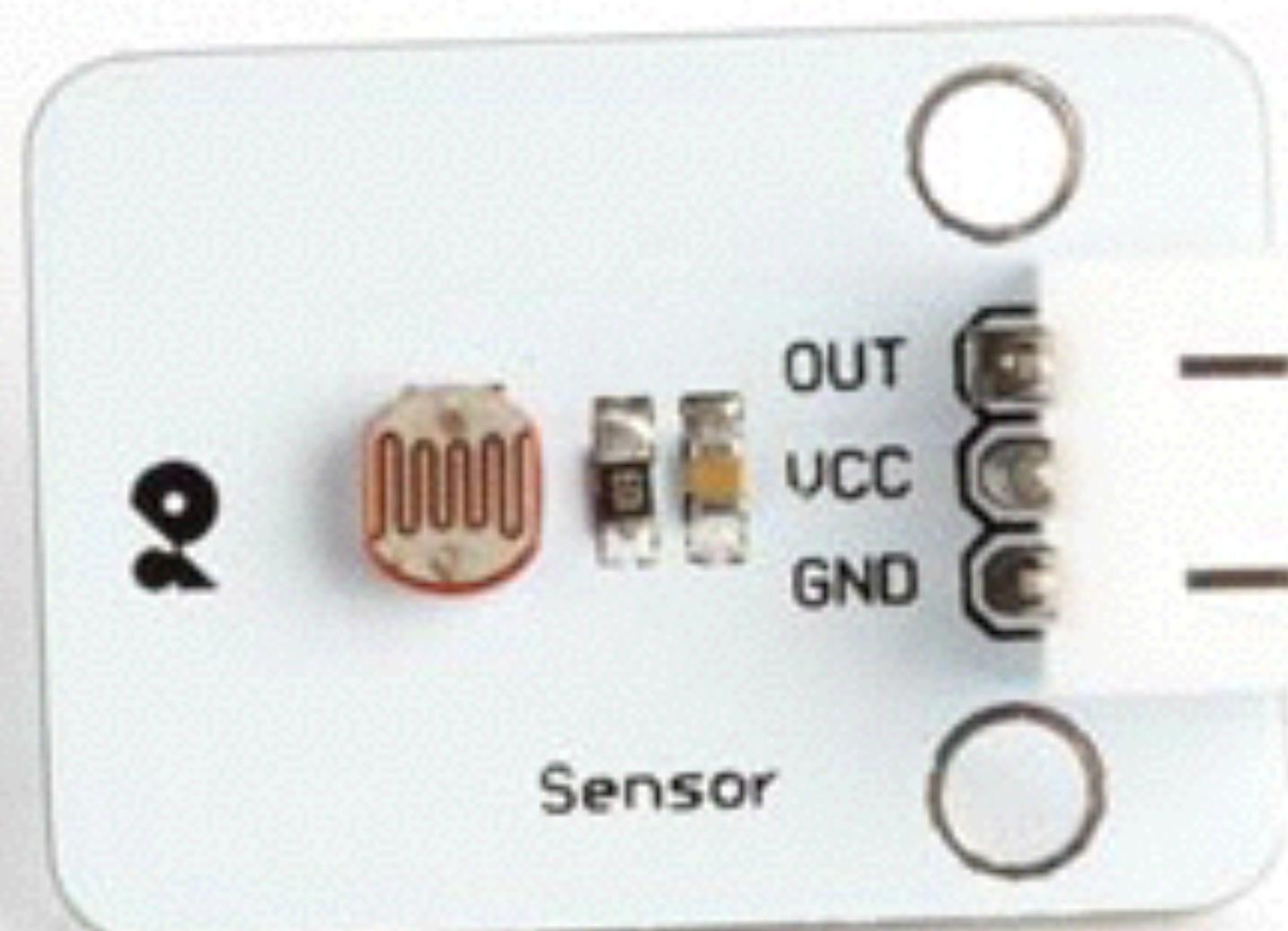
VMA407	►	Arduino®
OUT	►	A0
VCC	►	+5 V
GND	►	GND

operating voltage.....	3.3 VDC or 5 VDC
output .....	analogue voltage
pull-down resistor.....	10 kΩ, on board
dimensions .....	25 x 15 mm

<http://www.velleman.eu/products/view/?country=be&lang=fr&id=435572>

[http://www.velleman.eu/downloads/29/vma407\\_a4v01.pdf](http://www.velleman.eu/downloads/29/vma407_a4v01.pdf)

# Leçon 8: Photorésistante (velleman VMA 407)



<http://www.velleman.eu/products/view/?country=be&lang=fr&id=435572>

[http://www.velleman.eu/downloads/29/vma407\\_a4v01.pdf](http://www.velleman.eu/downloads/29/vma407_a4v01.pdf)

# Leçon 8: Photorésistante (velleman VMA 407)

Fichier > Exemples > 01.Basics > **AnalogReadSerial**

```
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize serial communication at 9600 bits per second:  
    Serial.begin(9600);  
  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    // read the input on analog pin 0:  
    int sensorValue = analogRead(A0);  
    // print out the value you read:  
    Serial.println(sensorValue);  
    delay(1);           // delay in between reads for stability  
}
```

# Leçon 8: Photorésistante (velleman VMA 407)

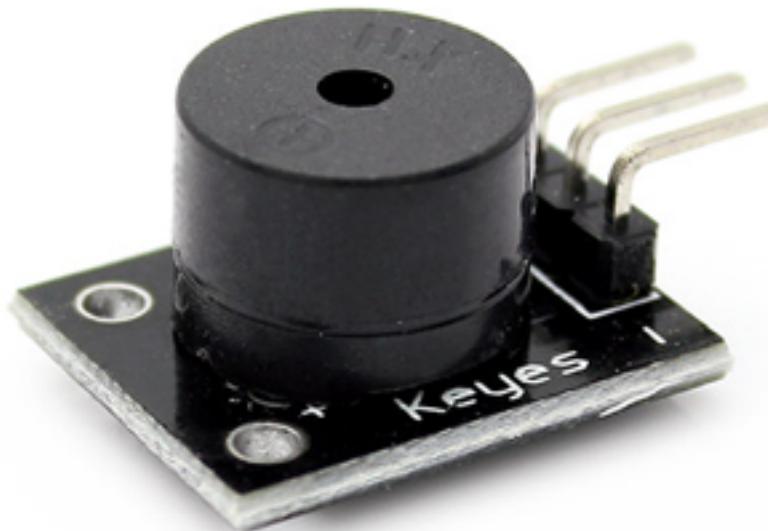


Allumez une led quand il fait nuit

Réglez **l'intensité** d'une LED avec une cellule photorésistante

Faites un **afficheur** avec 3 leds, du niveau de luminosité

# Leçon 9: Prise de sons (buzzer actif)



# Leçon 9: Prise de sons (buzzer actif)

The image shows two staves of musical notation. The top staff is in treble clef and common time (indicated by a '4'). It consists of two measures of music, each featuring eighth-note chords. The bottom staff is in bass clef and common time. It also consists of two measures, featuring eighth-note chords. The notes are represented by vertical stems with horizontal dashes indicating pitch and duration. The arrangement is attributed to Nguyen Maxime, with a link provided.

Arrangement : Nguyen Maxime  
<http://maxxxoo.free.fr/blog/>

Faire de la musique avec **tone()**

# Leçon 9: Prise de sons (buzzer actif)



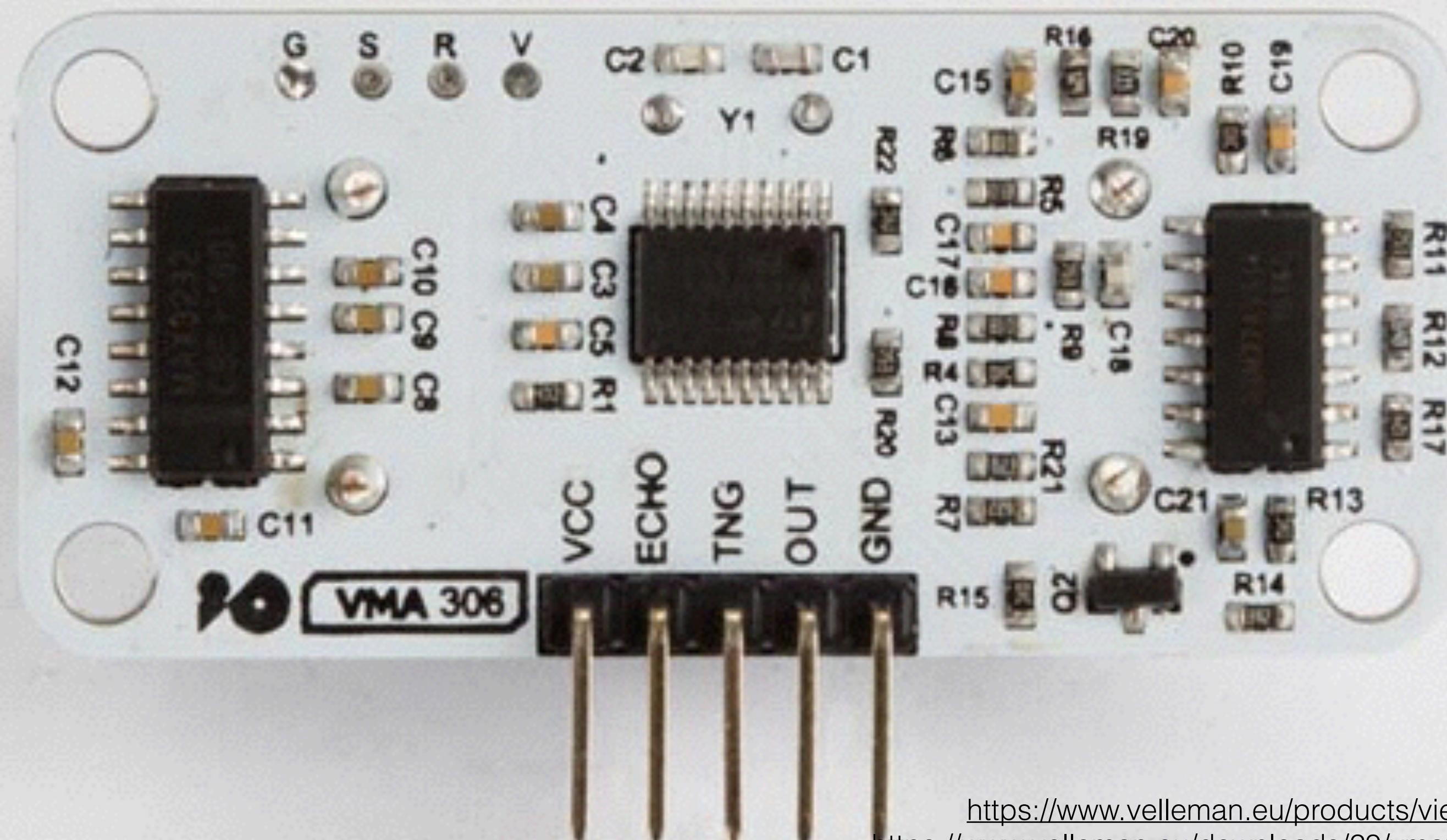
Composez une mélodie ?

# Leçon 18: Capteur à ultrasons (Velleman VMA 314)



<https://www.velleman.eu/products/view/?id=435542>  
[https://www.velleman.eu/downloads/29/vma314\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma314_a4v01.pdf)

# Leçon 18: Capteur à ultrasons HC-SR05 (VMA306)



<https://www.velleman.eu/products/view/?id=435526>  
[https://www.velleman.eu/downloads/29/vma306\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma306_a4v01.pdf)

# Leçon 18: Capteur à ultrasons HC-SR05 (VMA306)

The VMA306 ultrasonic distance sensor provides an easy way for your projects to measure distances up to 4.5 m.

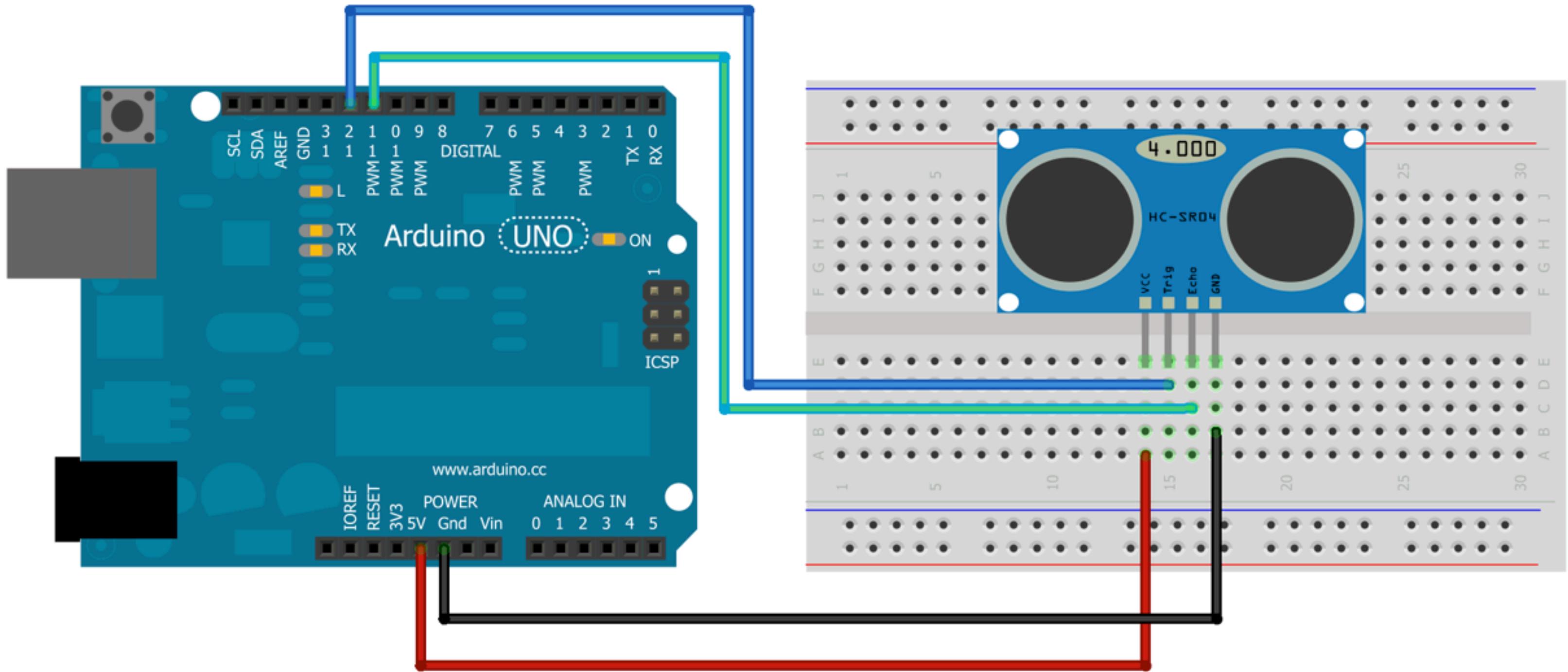
Arduino®	VMA306
+5 V	► VCC
D4	► ECHO
D2	► TNG
GND	► GND

voltage.....	4.5 to 5.5 VDC
sound frequency .....	40 KHz
measurement resolution .....	0.3 cm
measurement angle .....	15°
supply current .....	10 to 40 mA
trigger pin format .....	10 µS pulse
connector .....	5-pin male
detection distance.....	2 to 450 cm
dimensions .....	45 x 20 x 13 mm

<https://www.velleman.eu/products/view/?id=435526>

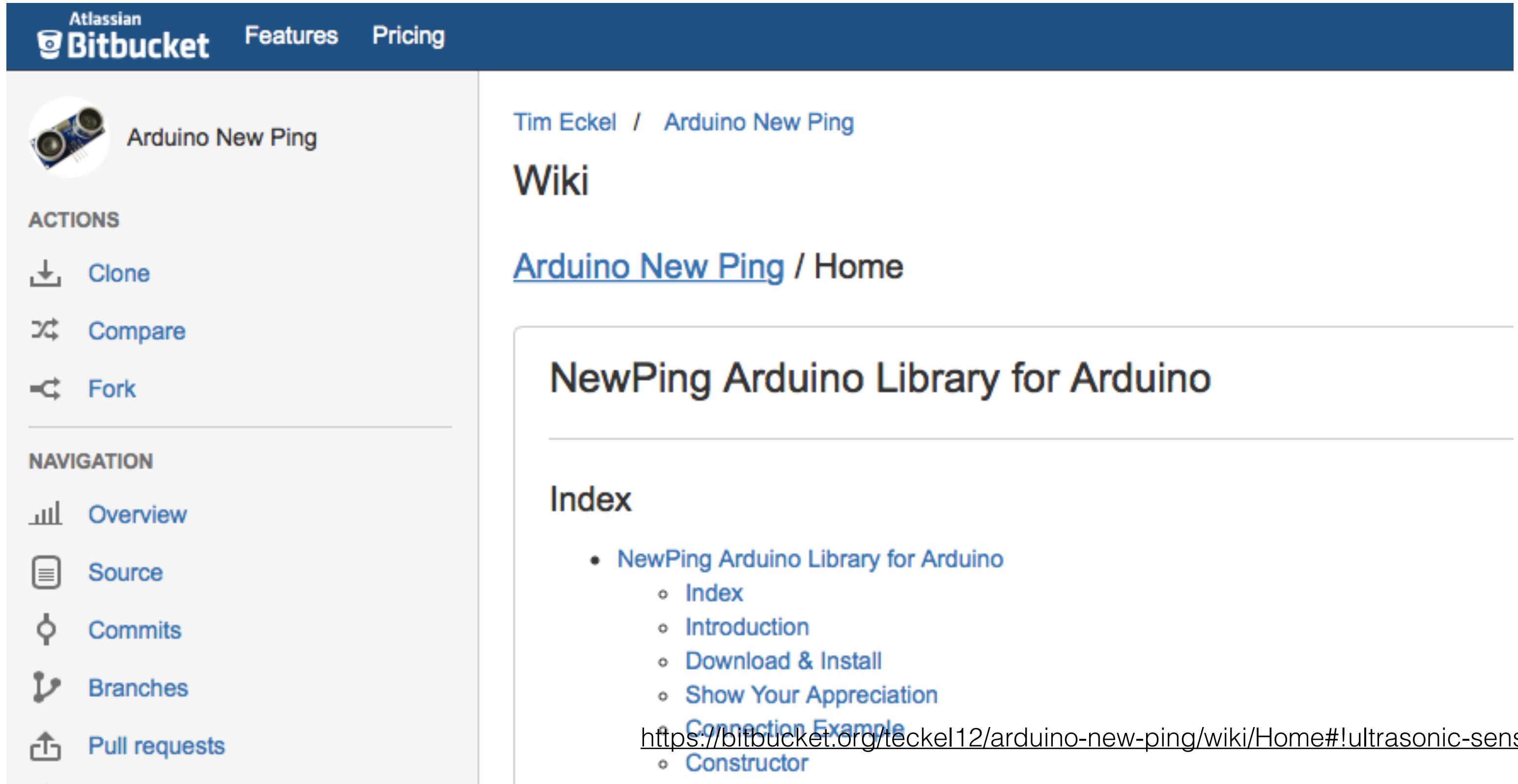
[https://www.velleman.eu/downloads/29/vma306\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma306_a4v01.pdf)

# Leçon 18: Capteur à ultrasons HC-SR04 (VMA306)



# Leçon 18: Capteur à ultrasons HC-SR05 (VMA306)

Atlassian  
**Bitbucket** Features Pricing



The screenshot shows a Bitbucket repository page for 'Arduino New Ping'. The repository was created by Tim Eckel. The main content area displays the 'Wiki' section, which includes a link to the 'Home' page of the NewPing Arduino Library. The 'Index' page is currently selected, showing a navigation tree for the library. The left sidebar contains links for 'Actions' (Clone, Compare, Fork) and 'Navigation' (Overview, Source, Commits, Branches, Pull requests).

Tim Eckel / Arduino New Ping

Arduino New Ping

Wiki

[Arduino New Ping / Home](#)

## NewPing Arduino Library for Arduino

### Index

- NewPing Arduino Library for Arduino
  - Index
  - Introduction
  - Download & Install
  - Show Your Appreciation
  - Connection Example
  - Constructor

<https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home#!ultrasonic-sensors>

ACTIONS

Clone

Compare

Fork

NAVIGATION

Overview

Source

Commits

Branches

Pull requests

# Leçon 18: Capteur à ultrasons HC-SR05 (VMA306)

Fichier > Exemples > New Ping > **NewPingExample**

```
// -----
// Example NewPing library sketch that does a ping about 20 times per second.
// -----



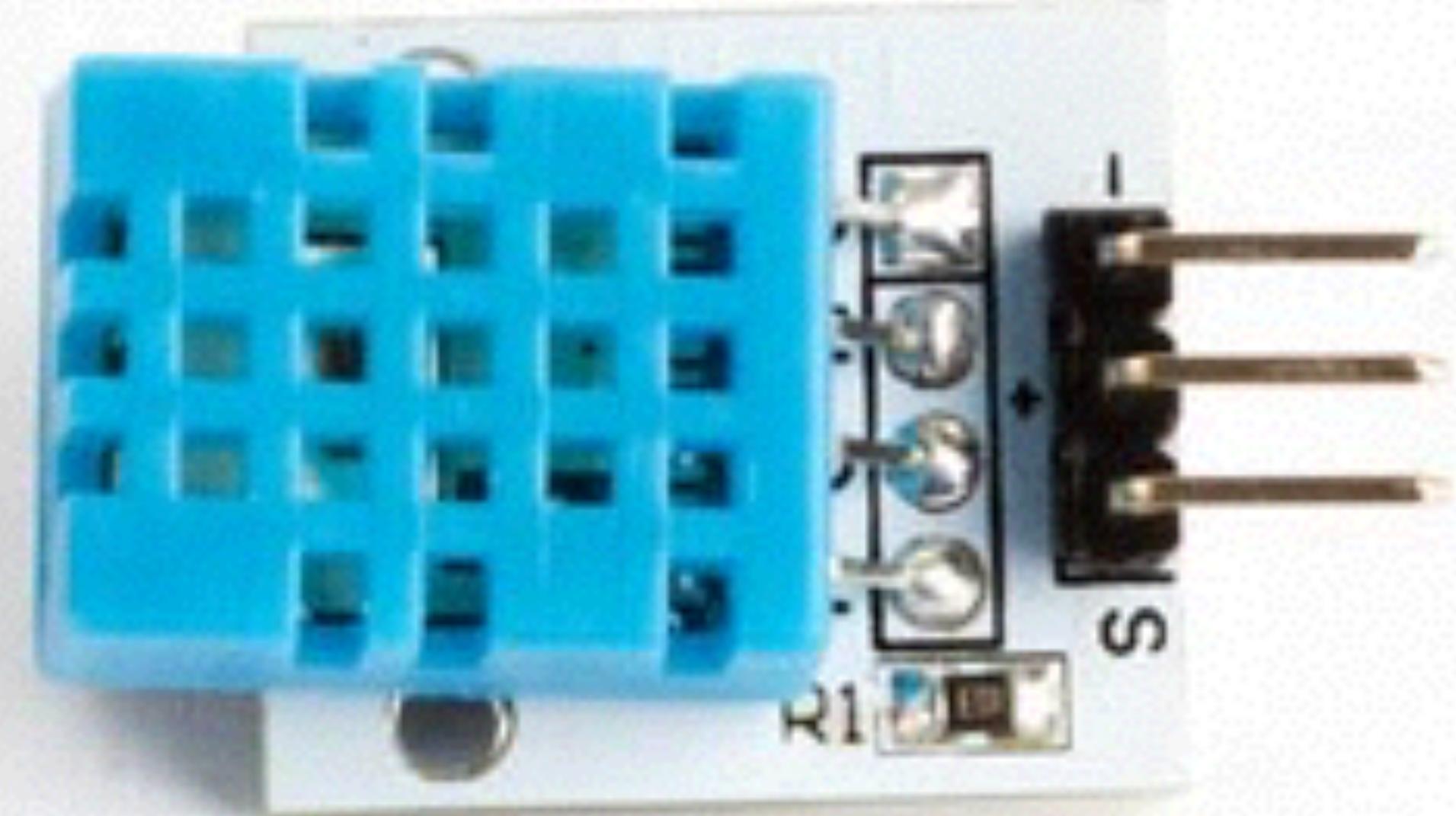
#include <NewPing.h>

#define TRIGGER_PIN 2 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN     4 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor range is approximately 400cm.
|
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance

void setup() {
    Serial.begin(9600); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
    delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be enough for most objects.
    Serial.print("Ping: ");
    Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result (0 = outside)
    Serial.println("cm");
}
```

# Leçon 20: Capteur de température et d'humidité DHT11 (Velleman VMA311)

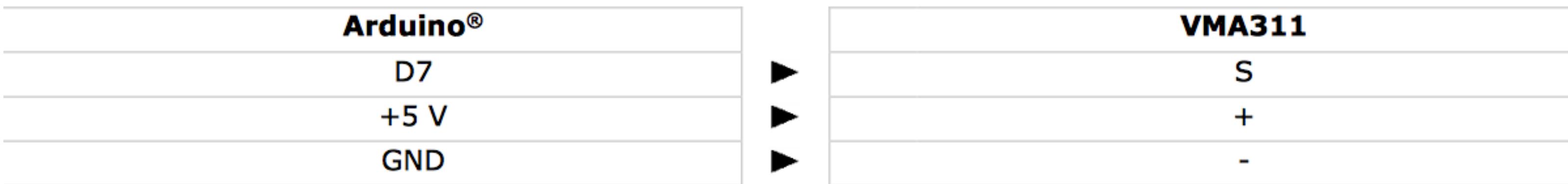


<https://www.velleman.eu/products/view/?id=435536>  
[https://www.velleman.eu/downloads/29/vma311\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma311_a4v01.pdf)

# Leçon 20: Capteur de température et d'humidité DHT11

## (Velleman VMA311)

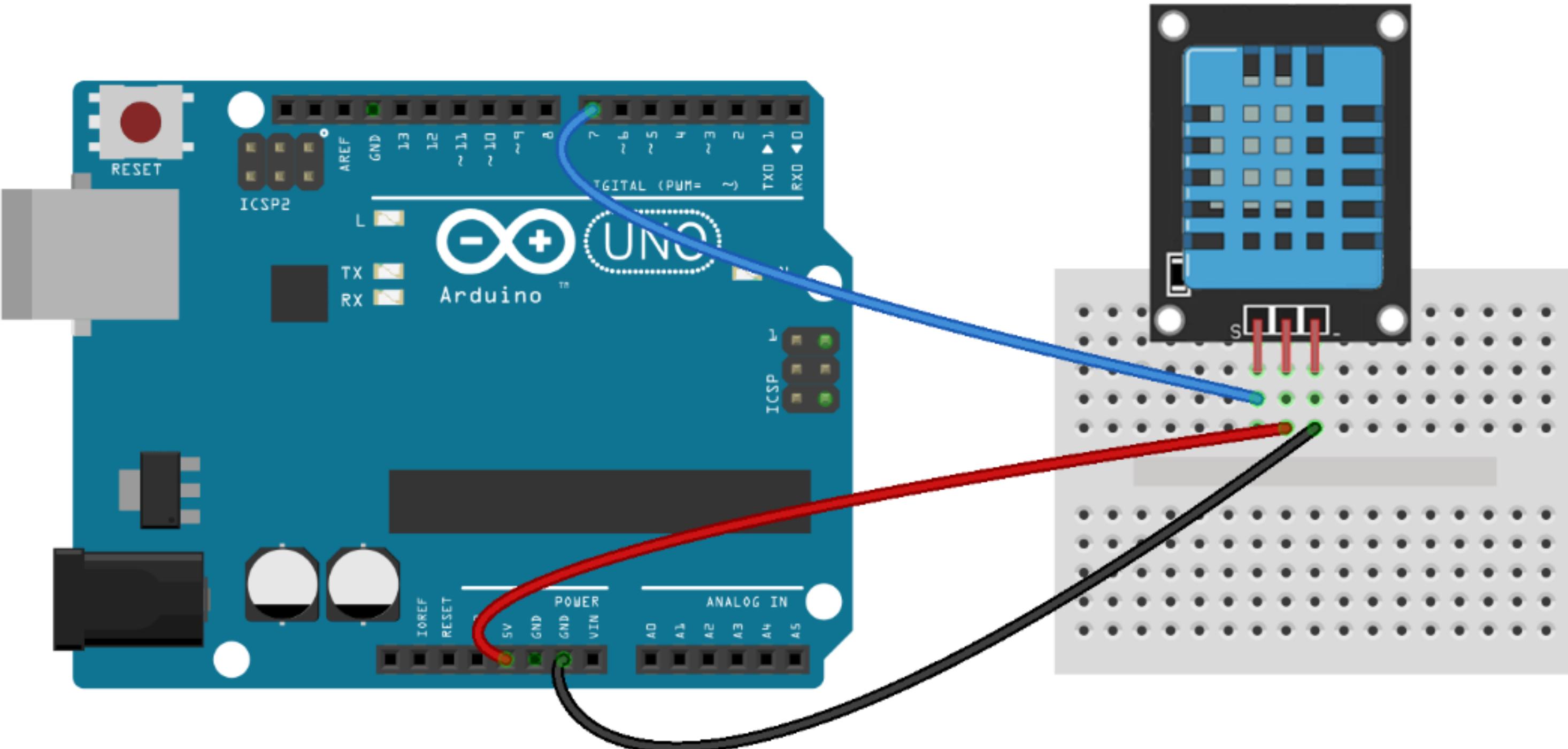
The DHT11 temperature and humidity sensor features a calibrated digital signal output with the temperature and humidity sensor complex. Its technology ensures a highly reliable and excellent long-term stability. A high-performance 8-bit micro-controller is connected. This sensor includes a resistive element and a sense of wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high cost performance advantages.



voltage.....	5 VDC
temperature range.....	0-50 °C , error of +/- 2 °C
humidity .....	20-90 % RH +/- 5 % RH error
interface.....	digital
dimensions .....	39 x 23 x 10 mm

# Leçon 20: Capteur de température et d'humidité DHT11

## (Velleman VMA311)



<https://www.velleman.eu/products/view/?id=435536>  
[https://www.velleman.eu/downloads/29/vma311\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma311_a4v01.pdf)

# Leçon 20: Capteur de température et d'humidité DHT11 (Velleman VMA311)

Screenshot of the GitHub repository page for `winlinvip / SimpleDHT`.

The repository description is: "Simple, Stable and Fast Arduino Temp & Humidity Sensors for DHT11 and DHT22. <http://learn.adafruit.com/dht>".

Tags listed under the repository: `arduino-library`, `arduino`, `dht11`, `dht22`, `dht-sensor`.

Key statistics for the repository:

- 40 commits
- 1 branch
- 7 releases
- 3 contributors
- MIT license

Branch dropdown: `master` ▾ | New pull request | Find file | Clone or download ▾

Recent commits:

File	Message	Date
<code>examples</code>	Fix #11, refine protocol, stat success rate	2 months ago
<code>.gitignore</code>	Initial commit	2 years ago
<code>LICENSE</code>	Update copyright date	3 months ago
<code>README.md</code>	Fix #11, refine protocol, stat success rate	2 months ago
<code>SimpleDHT.cpp</code>	Fix #11, refine protocol, stat success rate	2 months ago
<code>SimpleDHT.h</code>	Fix #11, refine protocol, stat success rate	2 months ago
<code>keywords.txt</code>	Update readme for DHT22	3 months ago

Latest commit: `d44f606` on 9 Jul

Repository URL: <https://github.com/winlinvip/SimpleDHT>

# Leçon 20: Capteur de température et d'humidité DHT11 (Velleman VMA311)

Fichier > Exemples > SimpleDHT > **DHT11Default**

```
#include <SimpleDHT.h>

// for DHT11,
//   VCC: 5V or 3V
//   GND: GND
//   DATA: 2
int pinDHT11 = 2;
SimpleDHT11 dht11;

void setup() {
  Serial.begin(115200);
}

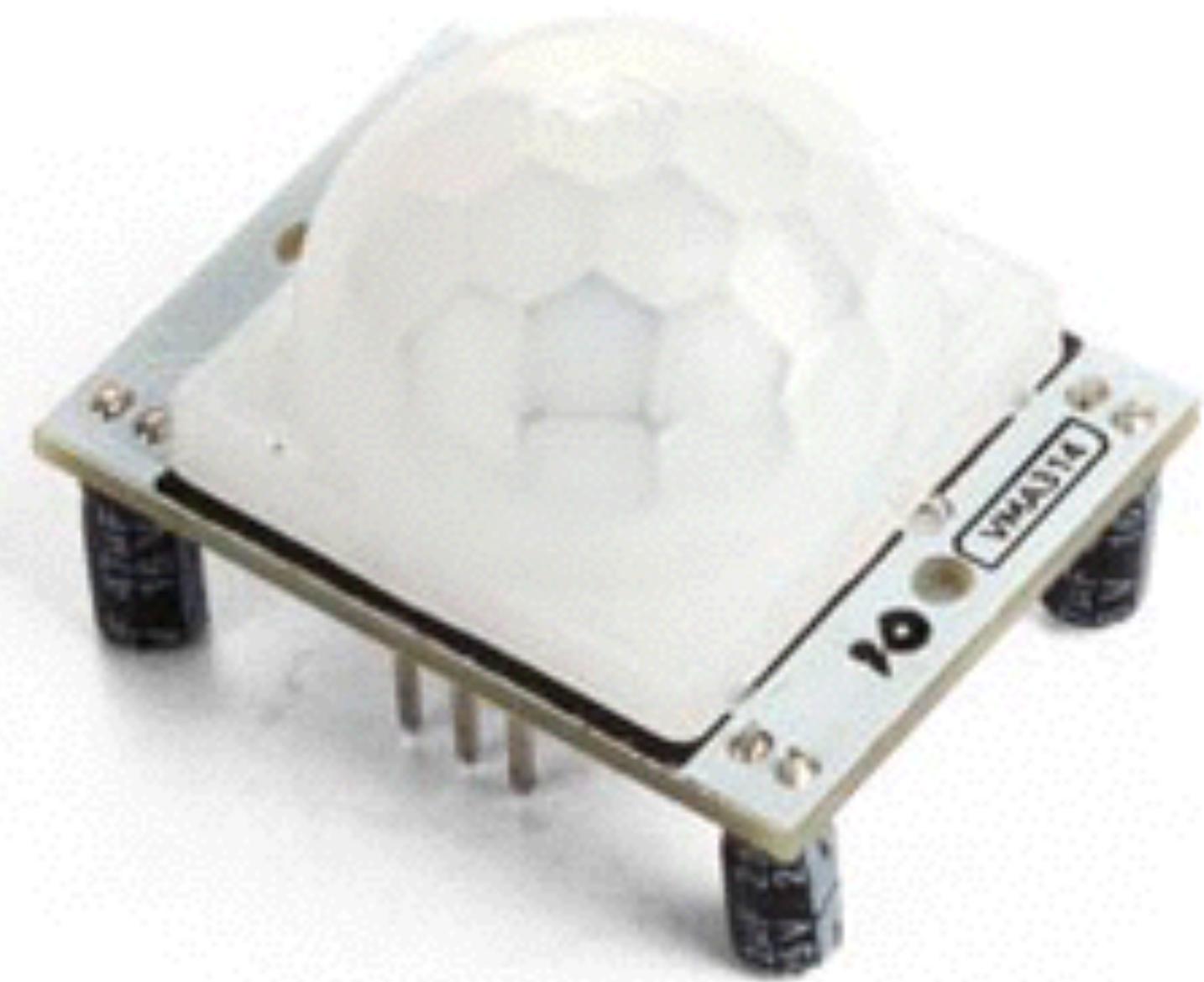
void loop() {
  // start working...
  Serial.println("=====");
  Serial.println("Sample DHT11...");

  // read without samples.
  byte temperature = 0;
  byte humidity = 0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err=");
    Serial.println(err);
    delay(1000);
    return;
  }

  Serial.print("Sample OK: ");
  Serial.print((int)temperature); Serial.print(" *C, ");
  Serial.print((int)humidity); Serial.println(" H");

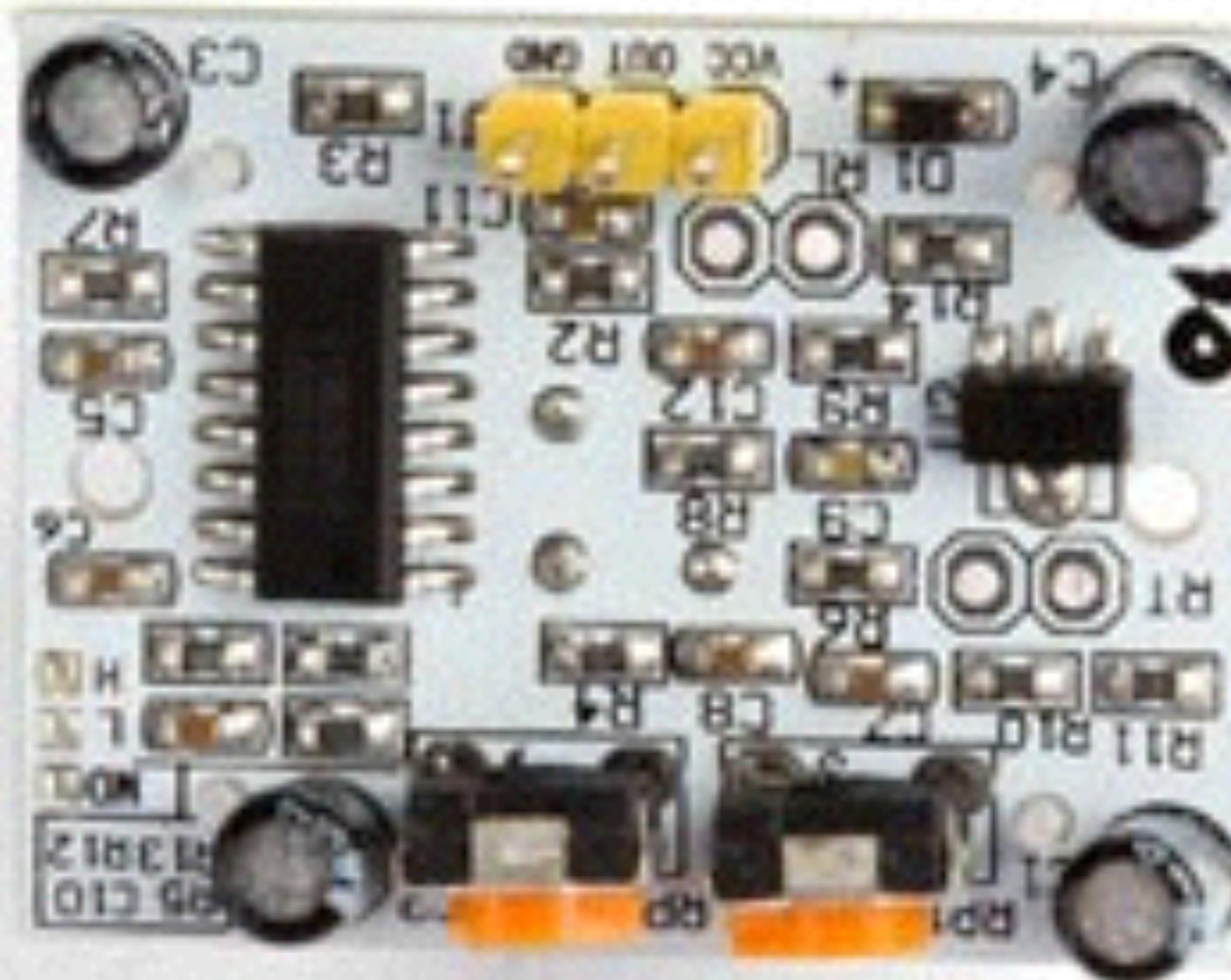
  // DHT11 sampling rate is 1HZ.
  delay(1500);
}
```

# Leçon 25: PIR Sensor (Velleman VMA 314)



<https://www.velleman.eu/products/view/?id=435542>  
[https://www.velleman.eu/downloads/29/vma314\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma314_a4v01.pdf)

# Leçon 25: PIR Sensor (Velleman VMA 314)

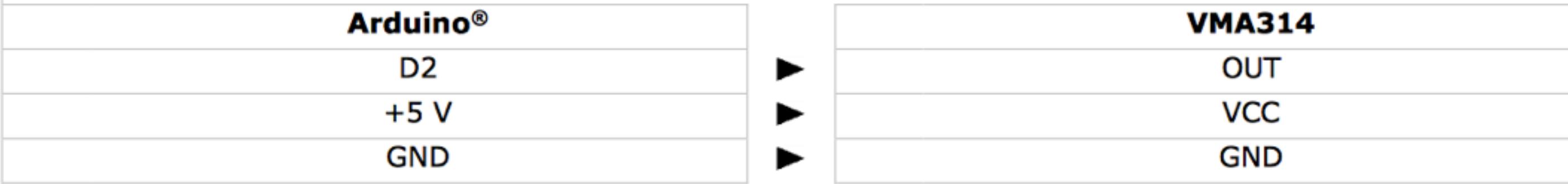


<https://www.velleman.eu/products/view/?id=435542>

[https://www.velleman.eu/downloads/29/vma314\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma314_a4v01.pdf)

# Leçon 25: PIR Sensor (Velleman VMA 314)

PIR sensors allow to sense motion, usually used to detect whether a human has moved in or out of the sensor's range.



voltage.....	5 VDC
connection .....	3 pins: GND, VCC and OUT
adjustments.....	sensitivity and delay (by trimmer)
delay time .....	0.3-18 s
output level.....	high = 3 V, low = 0 V
max. sensor distance .....	7 m
operating temperature.....	-15 °C to +70 °C
detection angle.....	120°
dimensions .....	32 x 24 x 25 mm

<https://www.velleman.eu/products/view/?id=435542>

[https://www.velleman.eu/downloads/29/vma314\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma314_a4v01.pdf)

# Leçon 25: PIR Sensor (Velleman VMA 314)

Fichier > Exemples > 01.Basics > **DigitalReadSerial**

```
// digital pin 2 has a pushbutton attached to it. Give it a name:  
int pushButton = 8;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize serial communication at 9600 bits per second:  
    Serial.begin(9600);  
    // make the pushbutton's pin an input:  
    pinMode(pushButton, INPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    // read the input pin:  
    int buttonState = digitalRead(pushButton);  
    // print out the state of the button:  
    Serial.println(buttonState);  
    delay(1);          // delay in between reads for stability  
}
```

# Leçon 25: PIR Sensor (Velleman VMA 314)

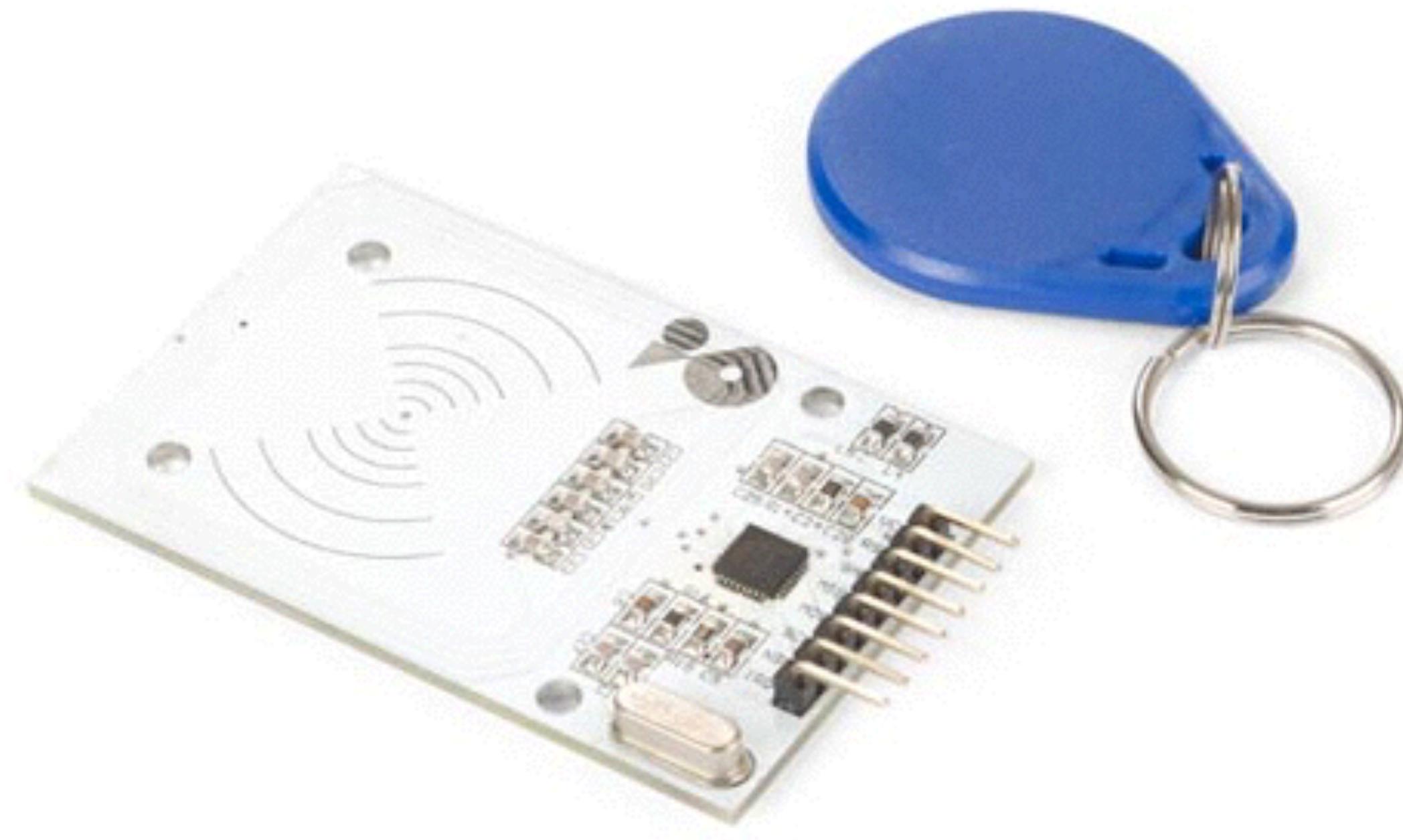
```
/*
 * PIR sensor tester
 */

int ledPin = 13;           // choose the pin for the LED
int inputPin = 8;          // choose the input pin (for PIR sensor)
int pirState = LOW;        // we start, assuming no motion detected
int val = 0;               // variable for reading the pin status
|
void setup() {
    pinMode(ledPin, OUTPUT); // declare LED as output
    pinMode(inputPin, INPUT); // declare sensor as input

    Serial.begin(9600);
}

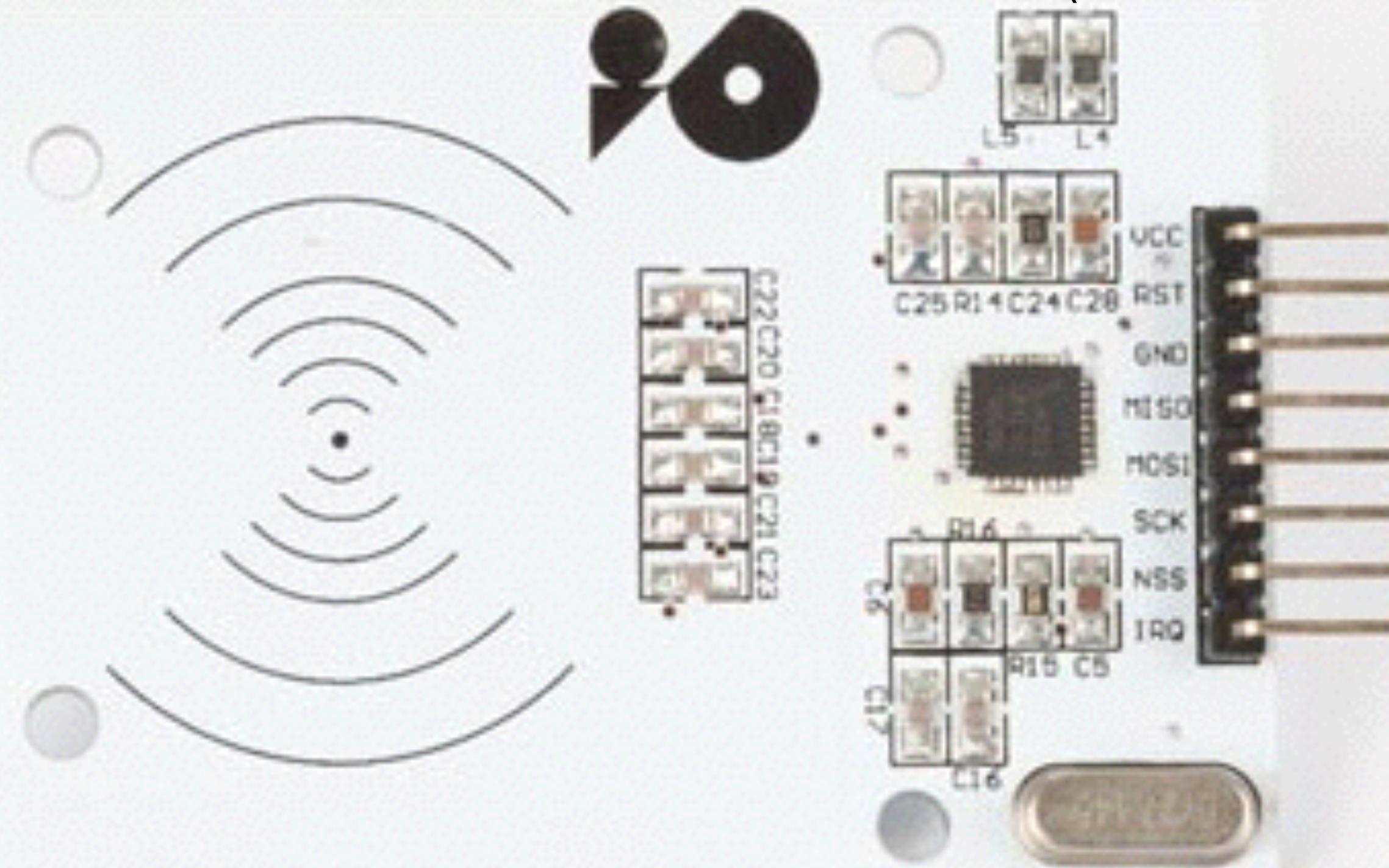
void loop(){
    val = digitalRead(inputPin); // read input value
    if (val == HIGH) {          // check if the input is HIGH
        digitalWrite(ledPin, HIGH); // turn LED ON
        if (pirState == LOW) {
            // we have just turned on
            Serial.println("Motion detected!");
            // We only want to print on the output change, not state
            pirState = HIGH;
        }
    } else {
        digitalWrite(ledPin, LOW); // turn LED OFF
        if (pirState == HIGH){
            // we have just turned off
            Serial.println("Motion ended!");
            // We only want to print on the output change, not state
            pirState = LOW;
        }
    }
}
```

# Leçon 29: Module RFID RC522 (Velleman VMA405)



<https://www.velleman.eu/products/view/?id=435568>  
<https://www.velleman.eu/support/downloads/?code=VMA405>  
[https://www.velleman.eu/downloads/29/vma405\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma405_a4v01.pdf)

# Leçon 29: Module RFID RC522 (Velleman VMA405)



<https://www.velleman.eu/products/view/?id=435568>  
<https://www.velleman.eu/support/downloads/?code=VMA405>  
[https://www.velleman.eu/downloads/29/vma405\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma405_a4v01.pdf)

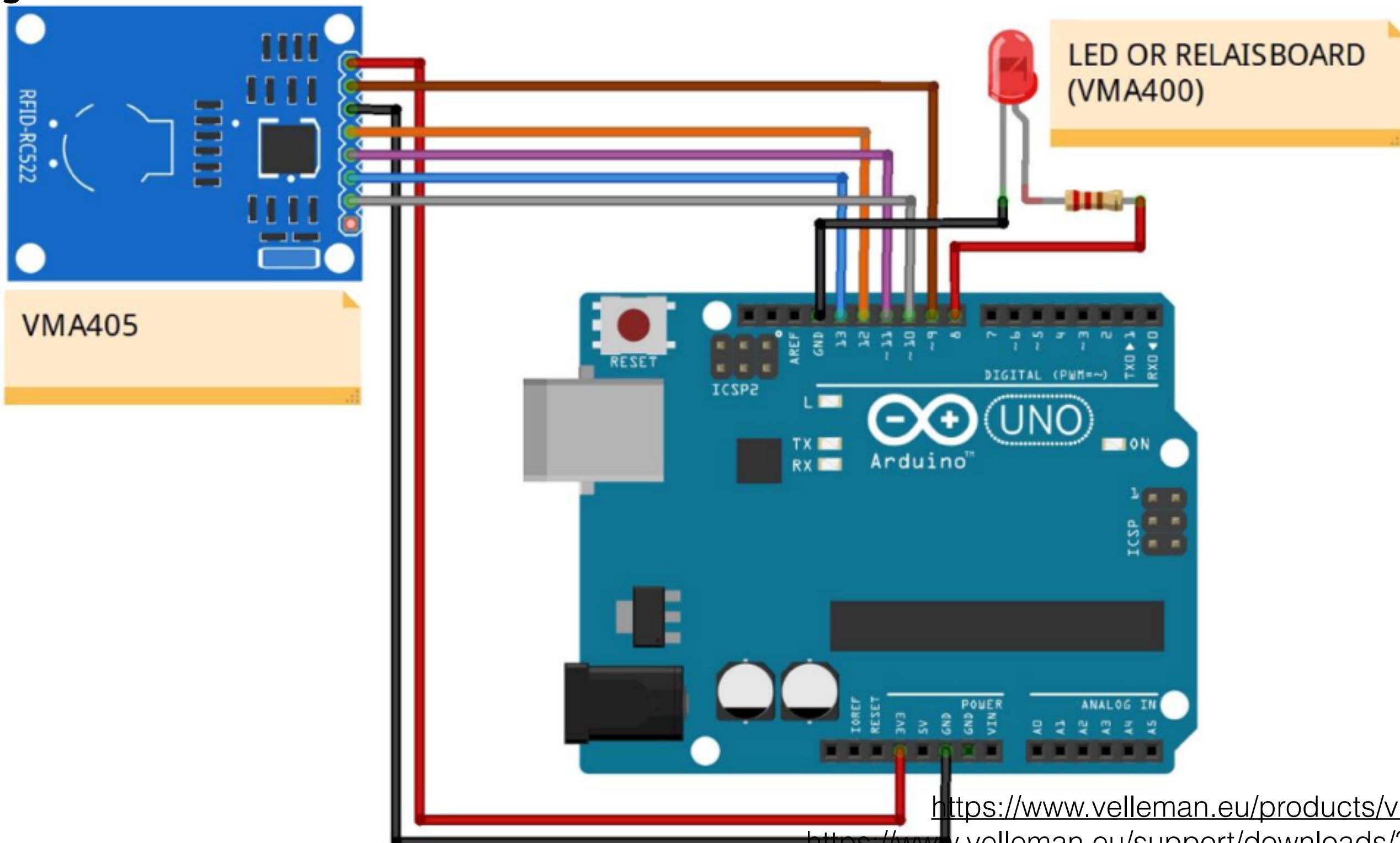
# Leçon 29: Module RFID RC522 (Velleman VMA405)



## **WARNING**

The VCC of your VMA405 must be connected to 3.3 V on your controller board.  
Do not connect to 5 V as your VMA405 will be destroyed!

# Leçon 29: Module RFID RC522 (Velleman VMA405)



<https://www.velleman.eu/products/view/?id=435568>

<https://www.velleman.eu/support/downloads/?code=VMA405>

[https://www.velleman.eu/downloads/29/vma405\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma405_a4v01.pdf)

# Leçon 29: Module RFID RC522 (Velleman VMA405)

Library

<https://www.velleman.eu/support/downloads/?code=VMA405>

<https://github.com/miguelbalboa/rfid>

<https://github.com/ljos/MFRC522>

...

