FORMATION EN

ARDUINO

Mouad LBARANI

Séances du Cours :

01 **La Carte ARDUINO**

- Introduction
- La carte Arduino
- Programme Arduino (IDE)
- Exercice

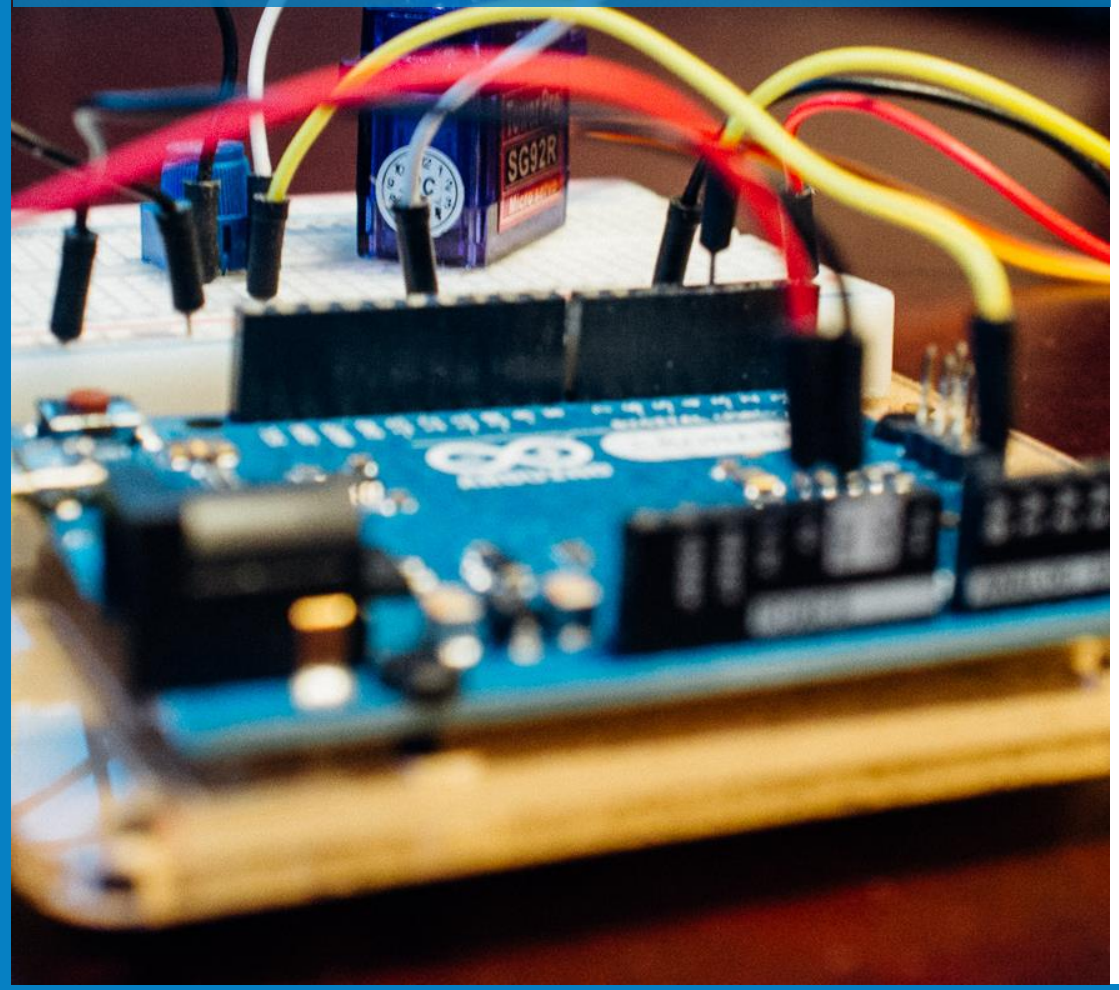
02 **Capteur / Actionneur**

- Infrarouge
- Potentiomètre
- Ultrason
- Servo Moteur

03 **Actionneur Survoltage**

- Shield Moteur
- Moteur DC

L’ARDUINO



ARDUINO

Une plate-forme de développement et de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

Open Source

Arduino est un projet en source ouverte càd l’Arduino est un programme dont le **code source** est distribué sous une licence.

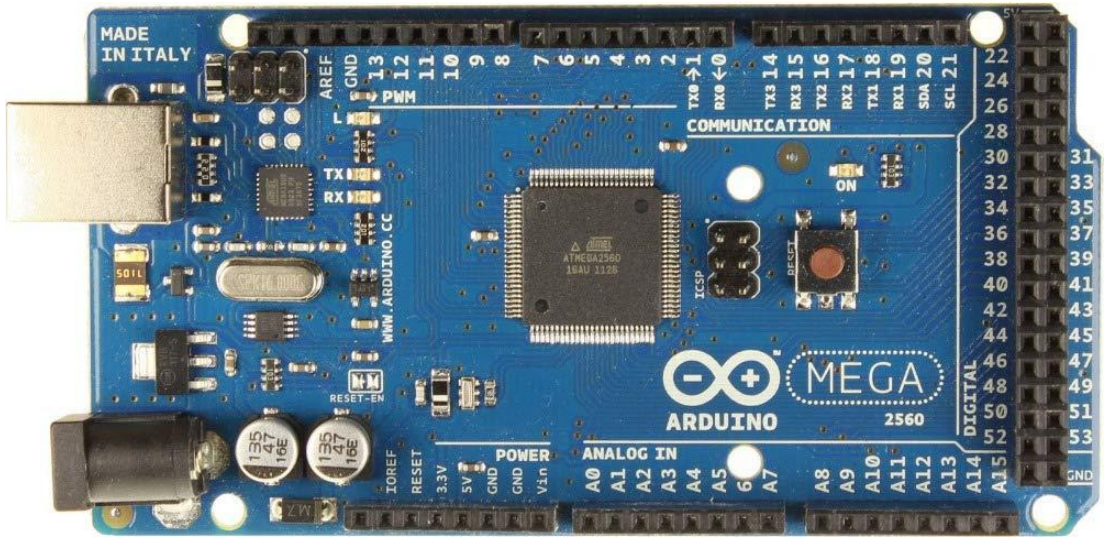
Facile (composant)

Sans tout connaître ni tout comprendre de l'électronique, vous pouvez créer votre projet à l‘aide du source online.

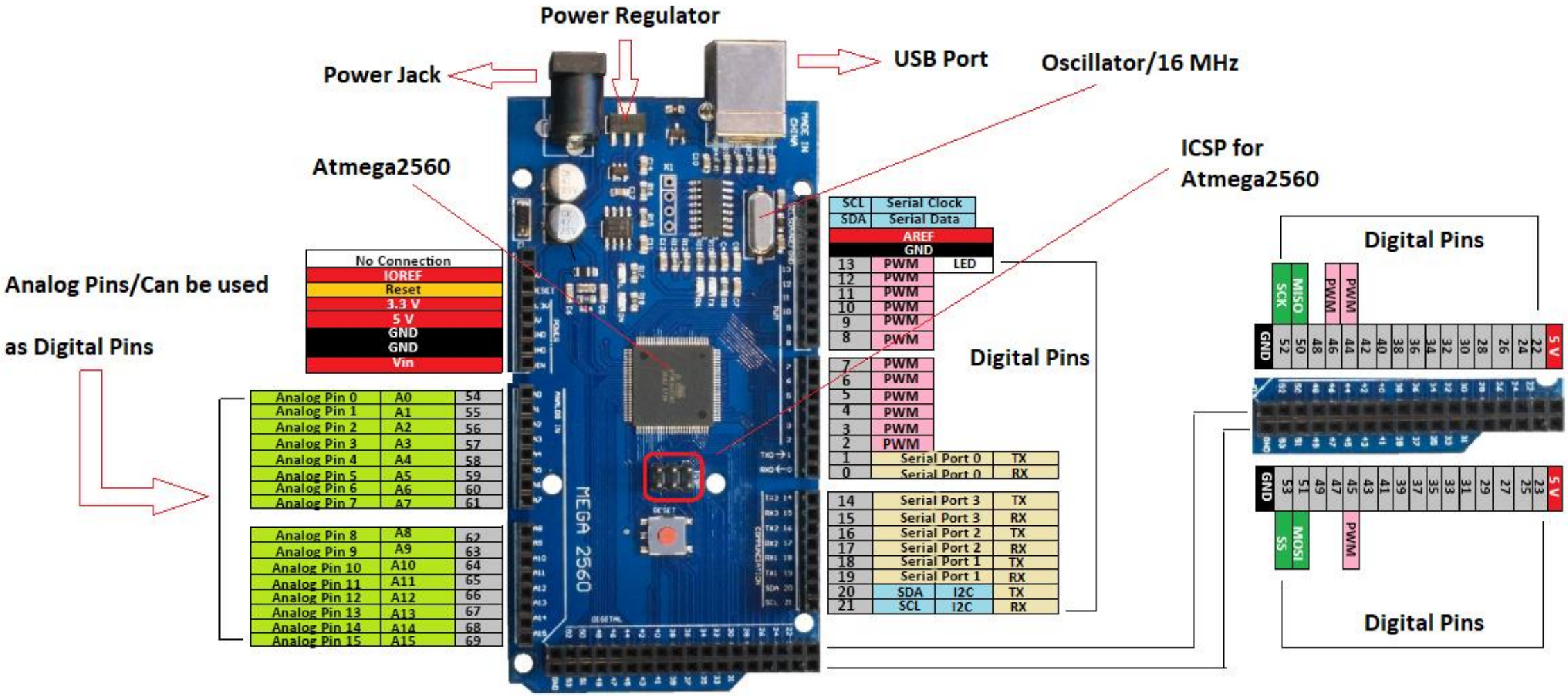
Facile (programmer)

Le langage Arduino est base sur le langage C.
La plupart des codes sont disponibles online.

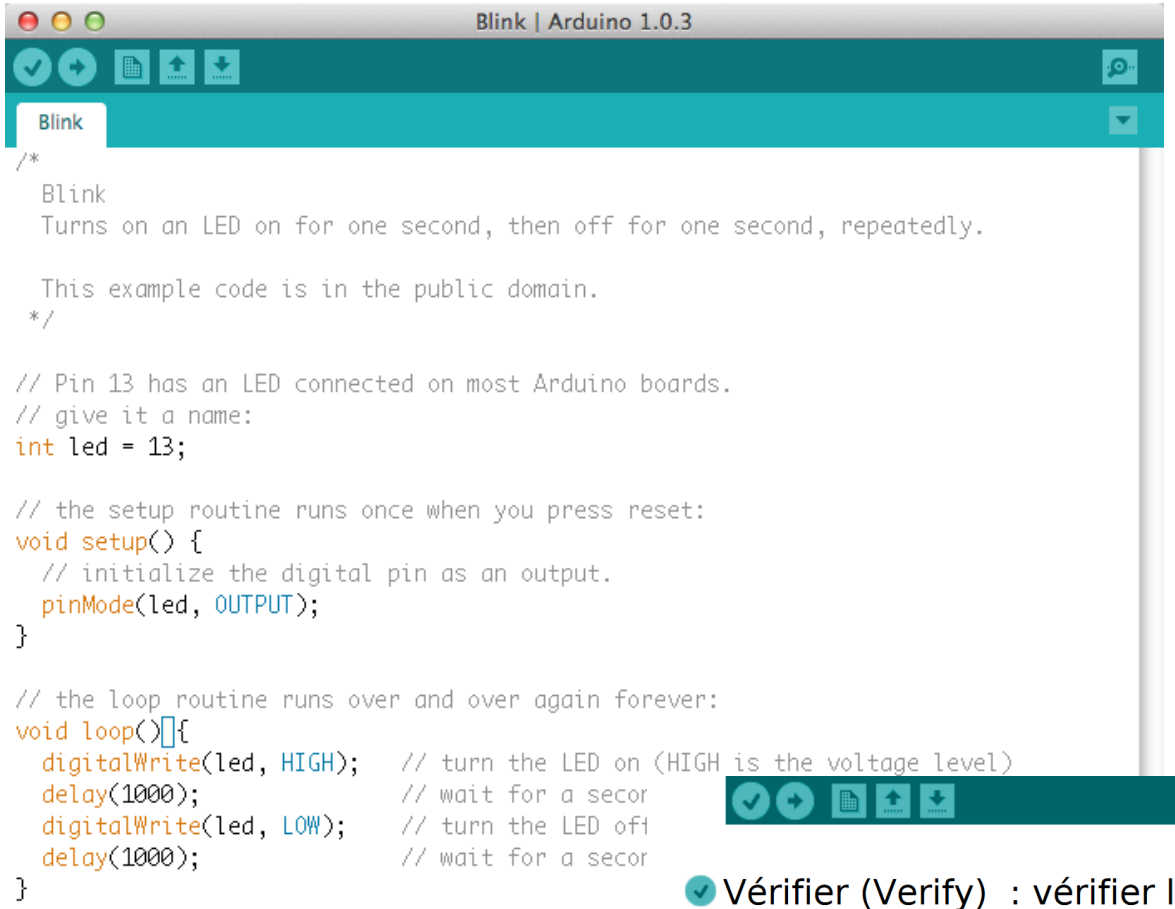
ARDUINO UNO



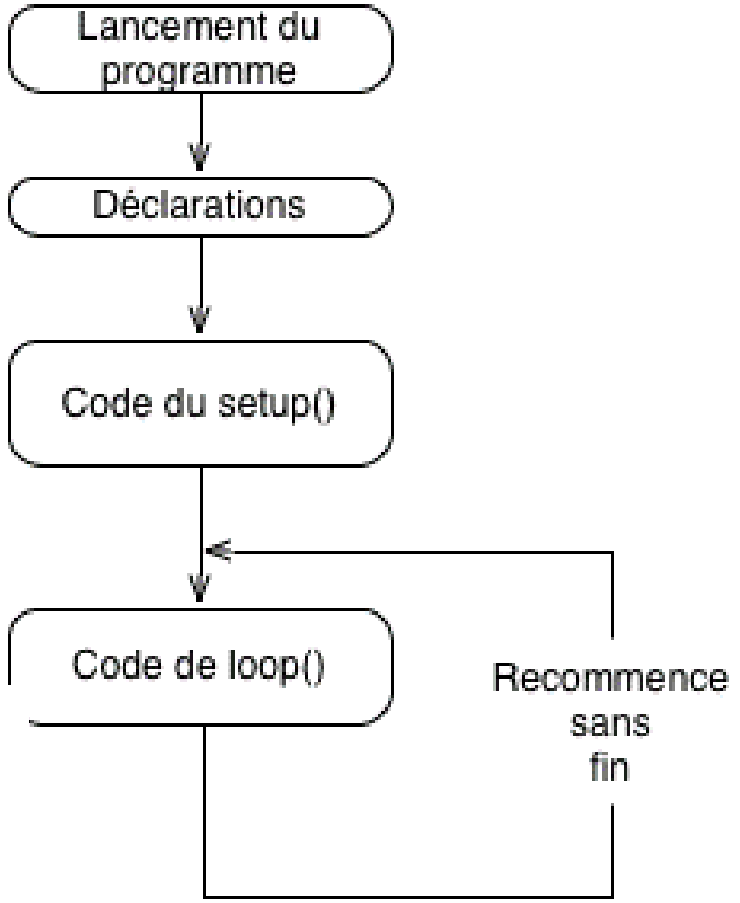
ARDUINO MEGA



Arduino Mega 2560 Pinout



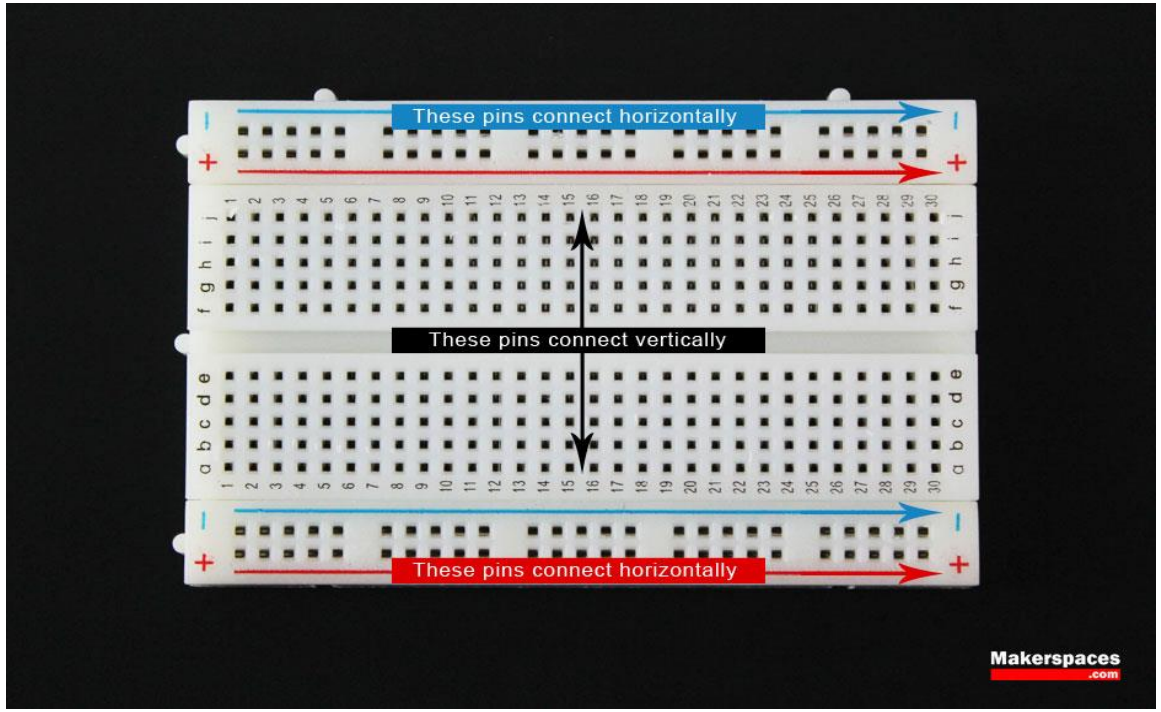
- Vérifier (Verify) : vérifier les erreurs dans le code
- Charge (Upload) : compiler le code et charge le programme sur la carte Arduino
- Nouveau (New) : créer un nouveau sketch
- Ouvrir (Open) : ouvrir un des sketchs déjà présent
- Sauvegarder (Save) : sauvegarder le sketch
- Serial Monitor : permet d'accéder au port série (en RX/TX)



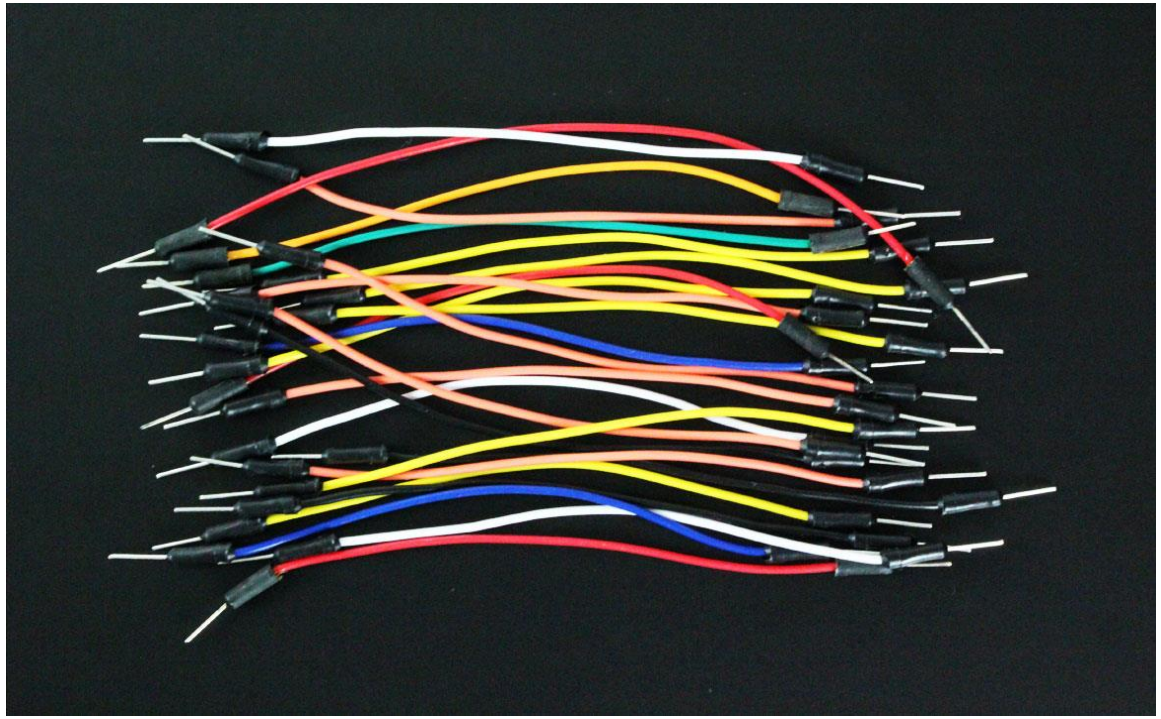
APPLICATION

Allumer et faire clignoter une LED avec Arduino :

Matériel nécessaire :



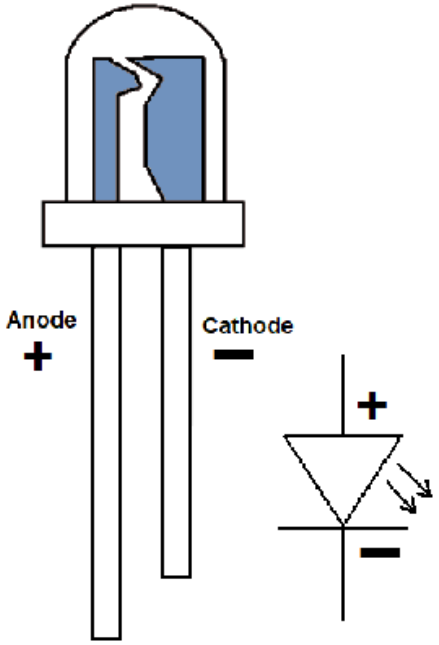
BREADBOARD



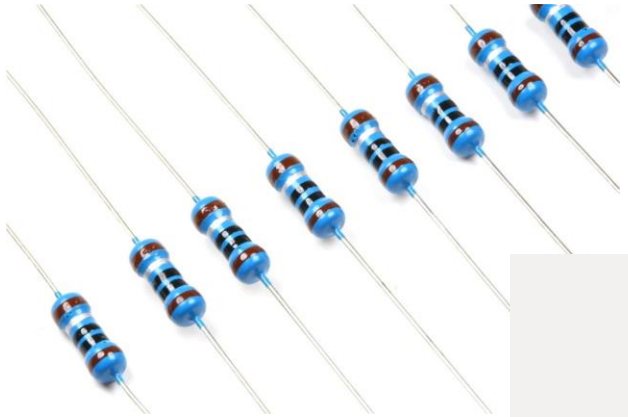
files

Allumer et faire clignoter une LED avec Arduino :

Matériel nécessaire :



LED



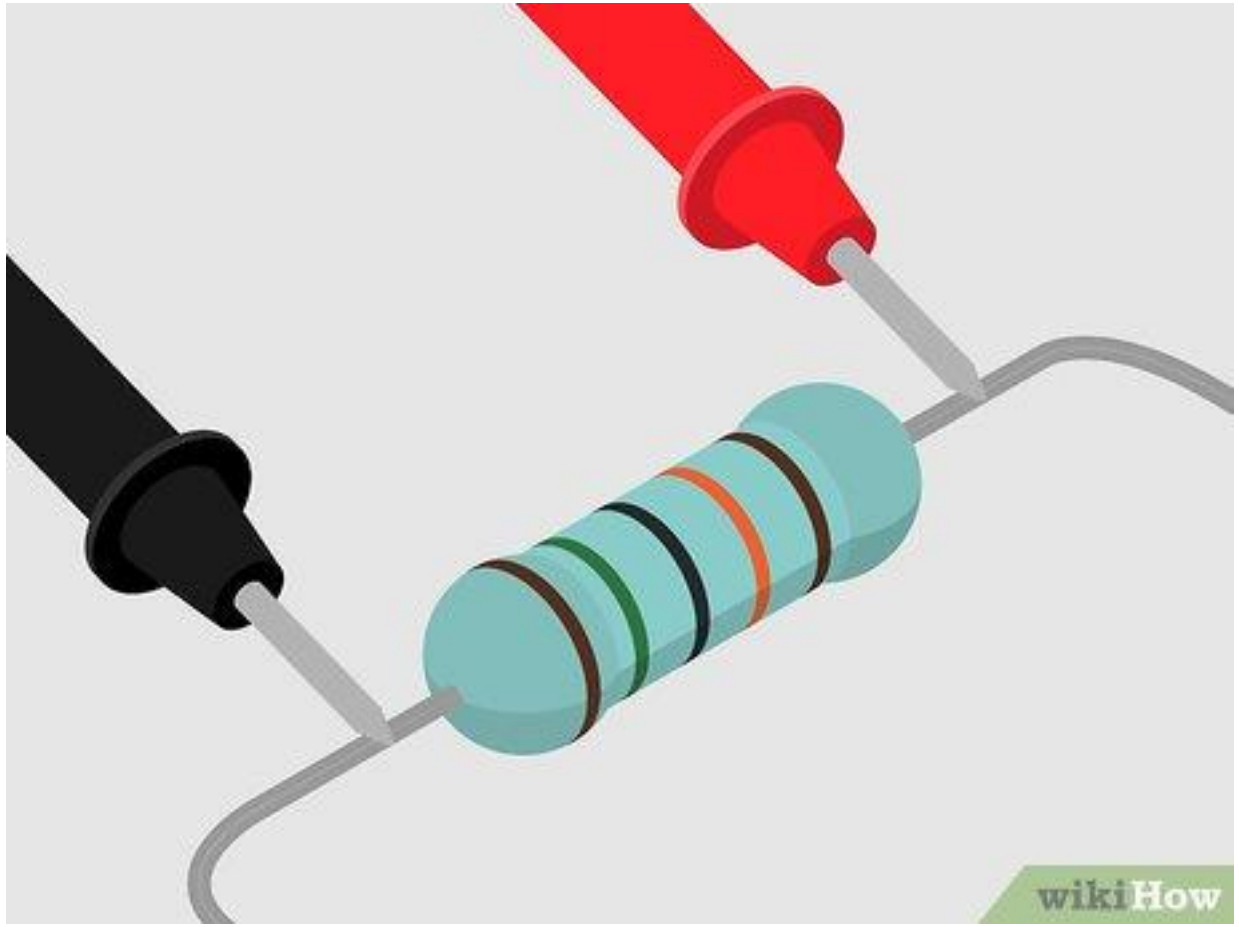
Resistance

Allumer et faire clignoter une LED avec Arduino :

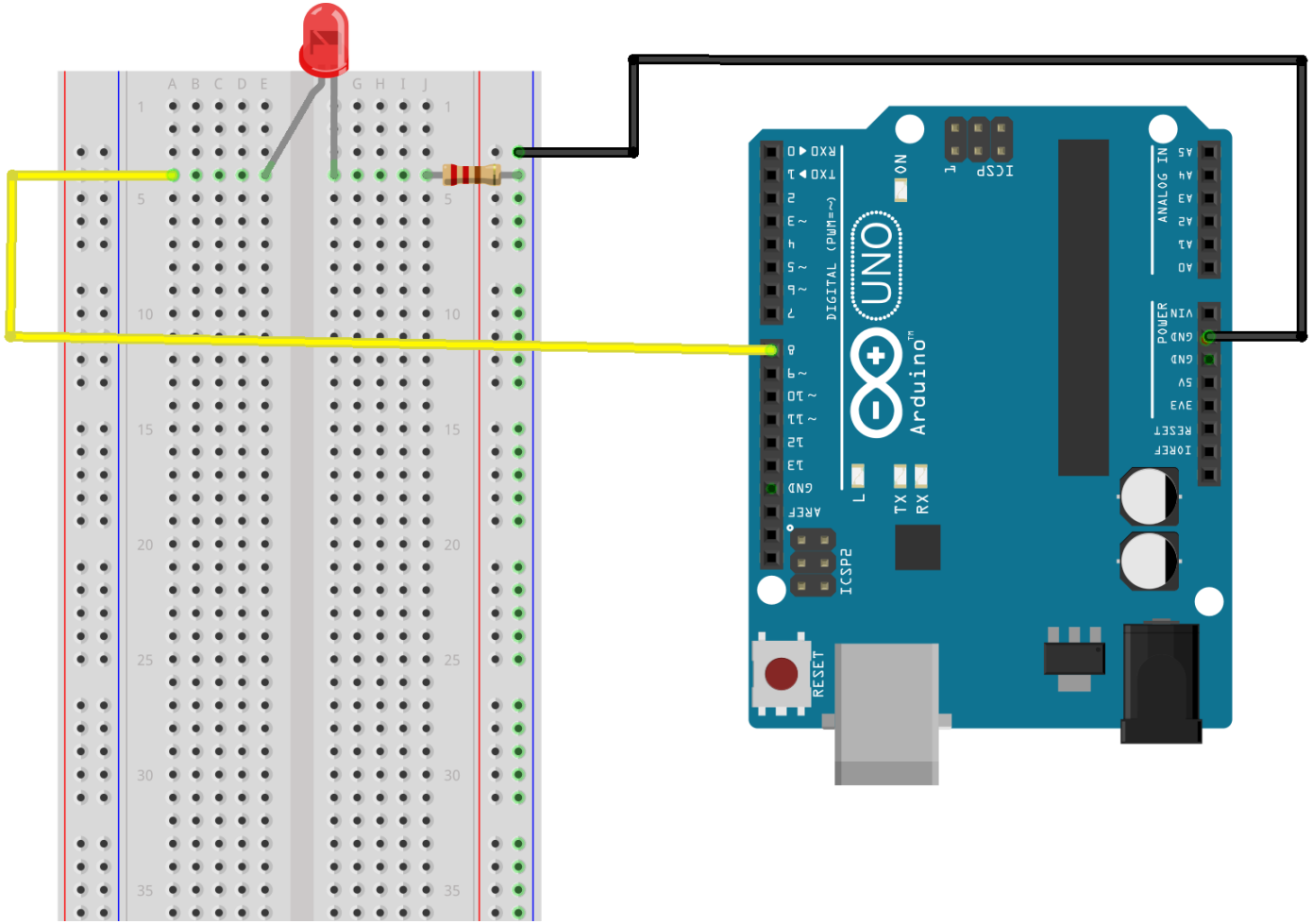
Comment utiliser un Multimètre :

Allumer et faire clignoter une LED avec Arduino :

Mesurer la valeur de la résistance :



MONTAGE :


A B C D E
G H I J
UNO
Arduino
DIGITAL (PWM~)
ANALOG IN
POWER
ICSP
ICSP2
RESET
L
TX
RX
ON

EXEMPLE CODE :

```
void setup() // boucle d'initialisation
{
  pinMode(13,OUTPUT); //connexion 13 en mode envoi de courant
}

void loop() // boucle inifie
{
  digitalWrite(13,HIGH); //envoi de courant dans la connexion 13
  delay(1000); //attente d'une seconde
  digitalWrite(13,LOW); //arret du courant dans la connexion 13
  delay(1000); //attente d'une seconde
}
```

EN RESUME :

Vous savez maintenant connecter votre Arduino, écrire la structure de base d'un programme, sauvegarder et transférer le programme dans le microcontrôleur, et débugger certaines erreurs de code.

Vous avez appris les mots-clés :

- **setup()** qui s'exécute qu'une fois ;
- **loop()** qui s'exécute à l'infini ;
- **void** (qui se met toujours devant loop et setup et dont vous comprendrez le sens plus loin dans ce cours) ;
- **pinMode()** qui permet de définir une sortie en mode envoi d'électricité ou non ;
- **digitalWrite()** qui envoie de l'électricité par une sortie ;
- **delay()** qui met le programme en pause pendant un nombre défini de millisecondes.

Je pense que vous êtes prêt(e)s pour la suite et je vous en félicite !

Capteur / Actionneur

Capteur

Un **capteur** est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable

Actionneur

Un **actionneur** est un objet qui transforme l’énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l’état d'un système

Capteur Infrarouge

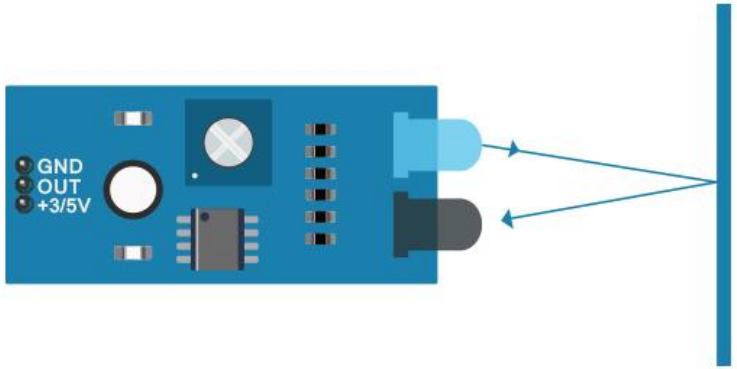
Capteur IR

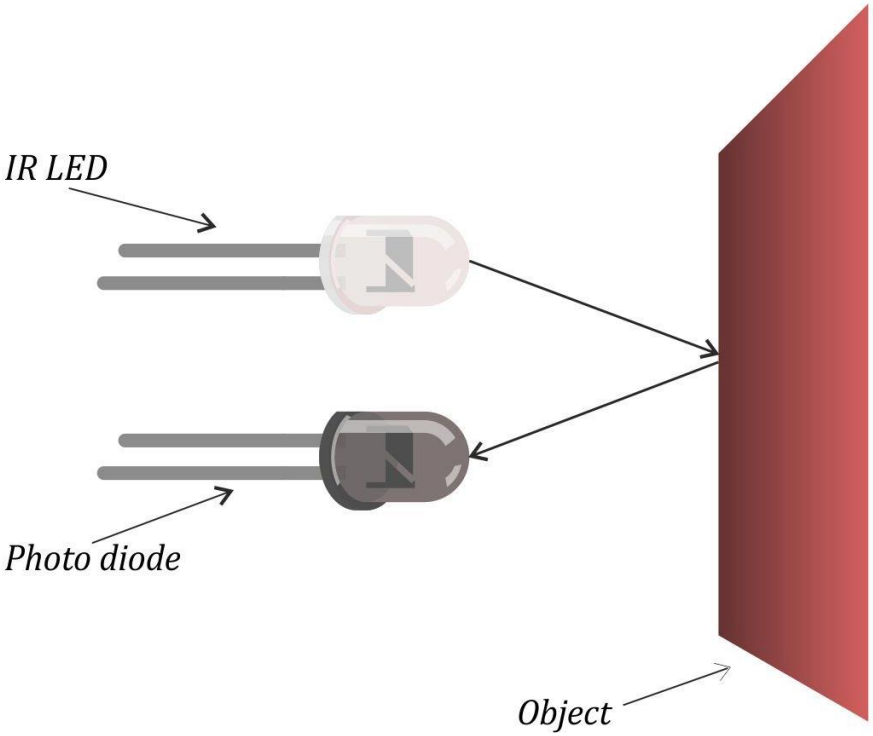
Capteur Infrarouge (de Proximité)

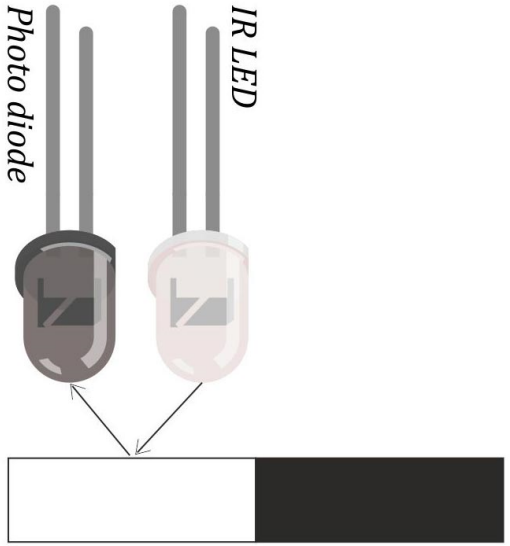
Les détecteurs de mouvements font appel à un capteur infrarouge qui leur permet de signaler la présence d'un intrus même dans l'obscurité.

Fonctionnement :

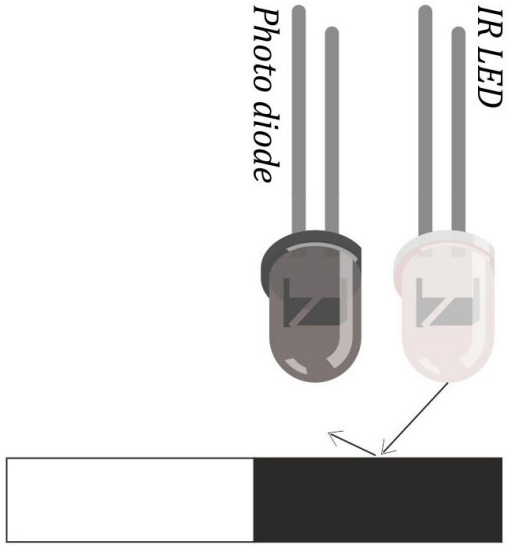
comme son nom l'indique il s'agit d'un capteur de proximité, muni de deux LEDs infrarouges. Une transparente et une plus foncée. L'une correspondant à la LED émettrice et la LED réceptrice.





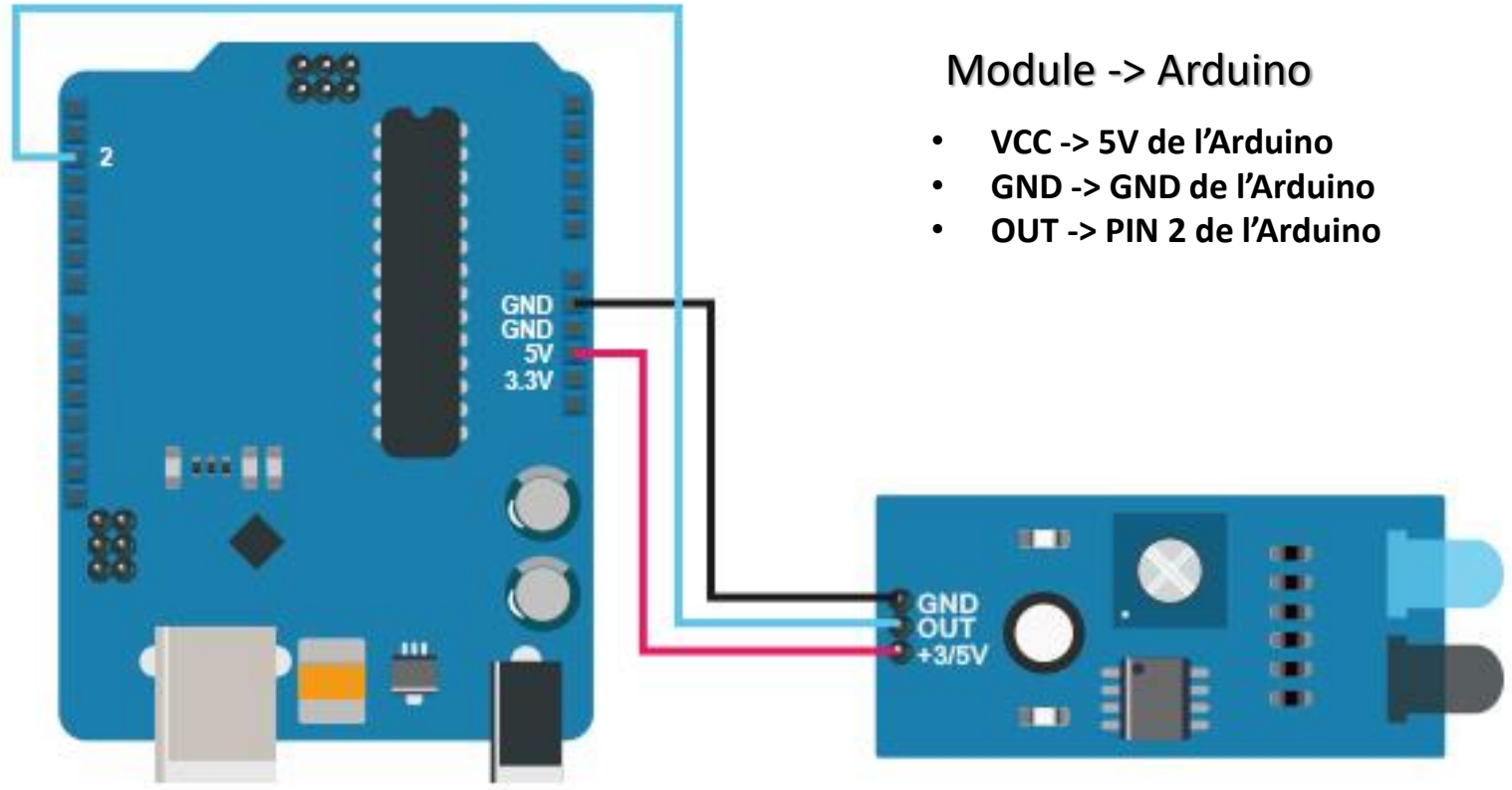


Reflection from white surface



Reflection from black surface

Branchements :



Module -> Arduino

- **VCC -> 5V de l'Arduino**
- **GND -> GND de l'Arduino**
- **OUT -> PIN 2 de l'Arduino**

code :

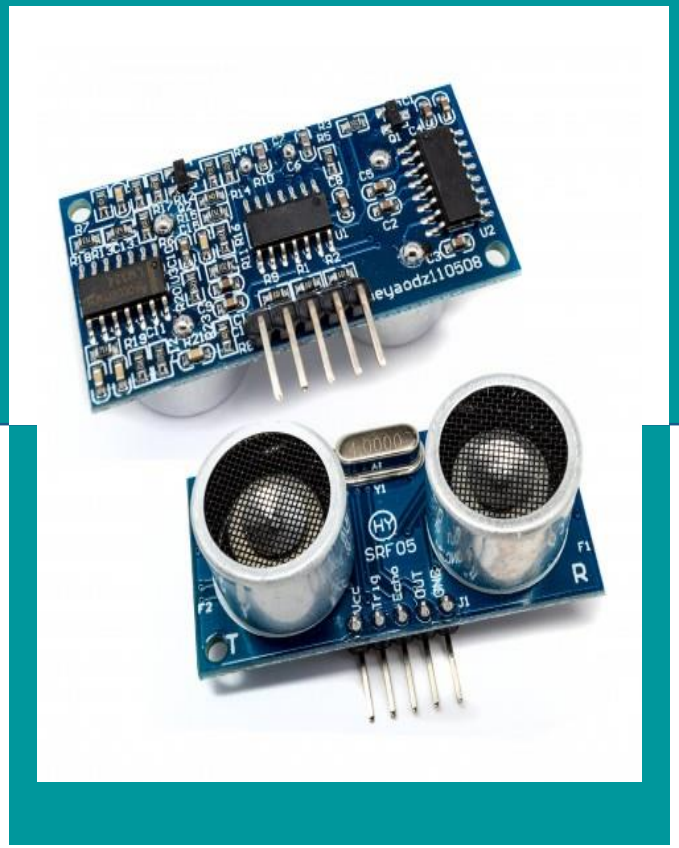
```
const int ProxSensor=2;
int inputVal = 0;

void setup()
{
  pinMode(13, OUTPUT);           // Pin 13 has an LED connected on most Arduino boards:
  pinMode(ProxSensor,INPUT);     //Pin 2 is connected to the output of proximity sensor
  Serial.begin(9600);
}

void loop()
{
  if(digitalRead(ProxSensor)==HIGH)       //Check the sensor output
  {
    digitalWrite(13, HIGH);   // set the LED on
  }
  else
  {
    digitalWrite(13, LOW);    // set the LED off
  }
inputVal = digitalRead(ProxSensor);
Serial.println(inputVal);
delay(1000);                // wait for a second
}
```

- **ProxSensor (Pin 2) : le capteur IR est branché dans la broche 2.**
- **inputVal : variable pour afficher la valeur du capteur**

Si le capteur détecte un objet la LED du Pin 13 s'allumée , sinon la LED s'éteindre. puis on affiche la valeur du capteur dans le monitor.

Ultrason

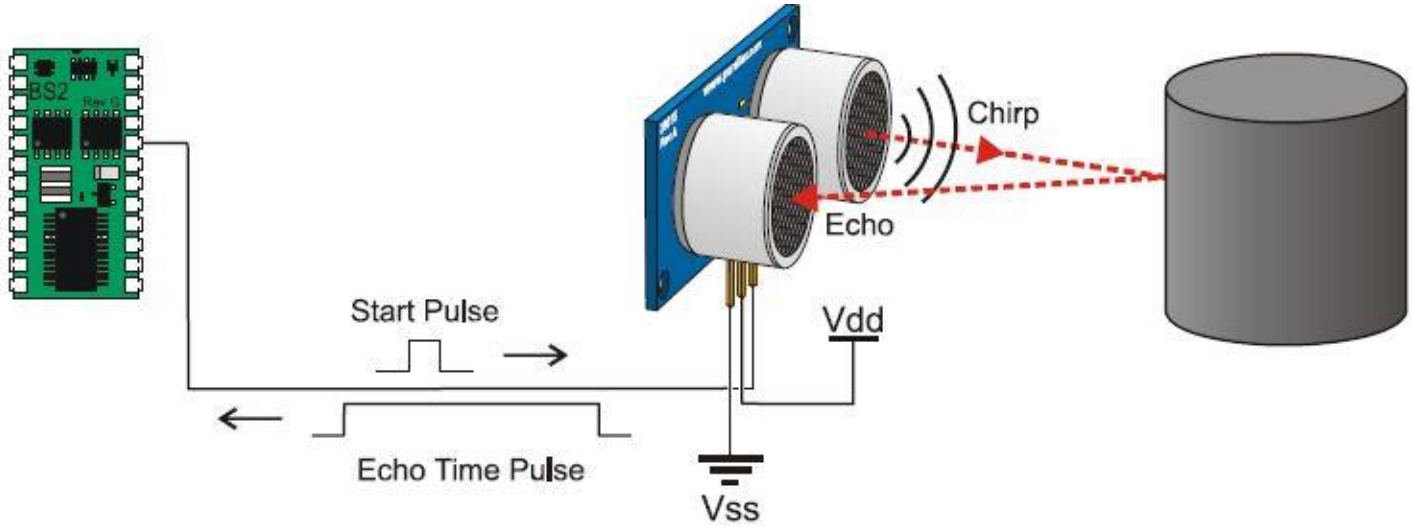
- **Comment mesurer une distance avec l'Arduino ? Ou encore… Comment détecter un obstacle avec l'Arduino ?**



Capteur IR

Fonctionnement :

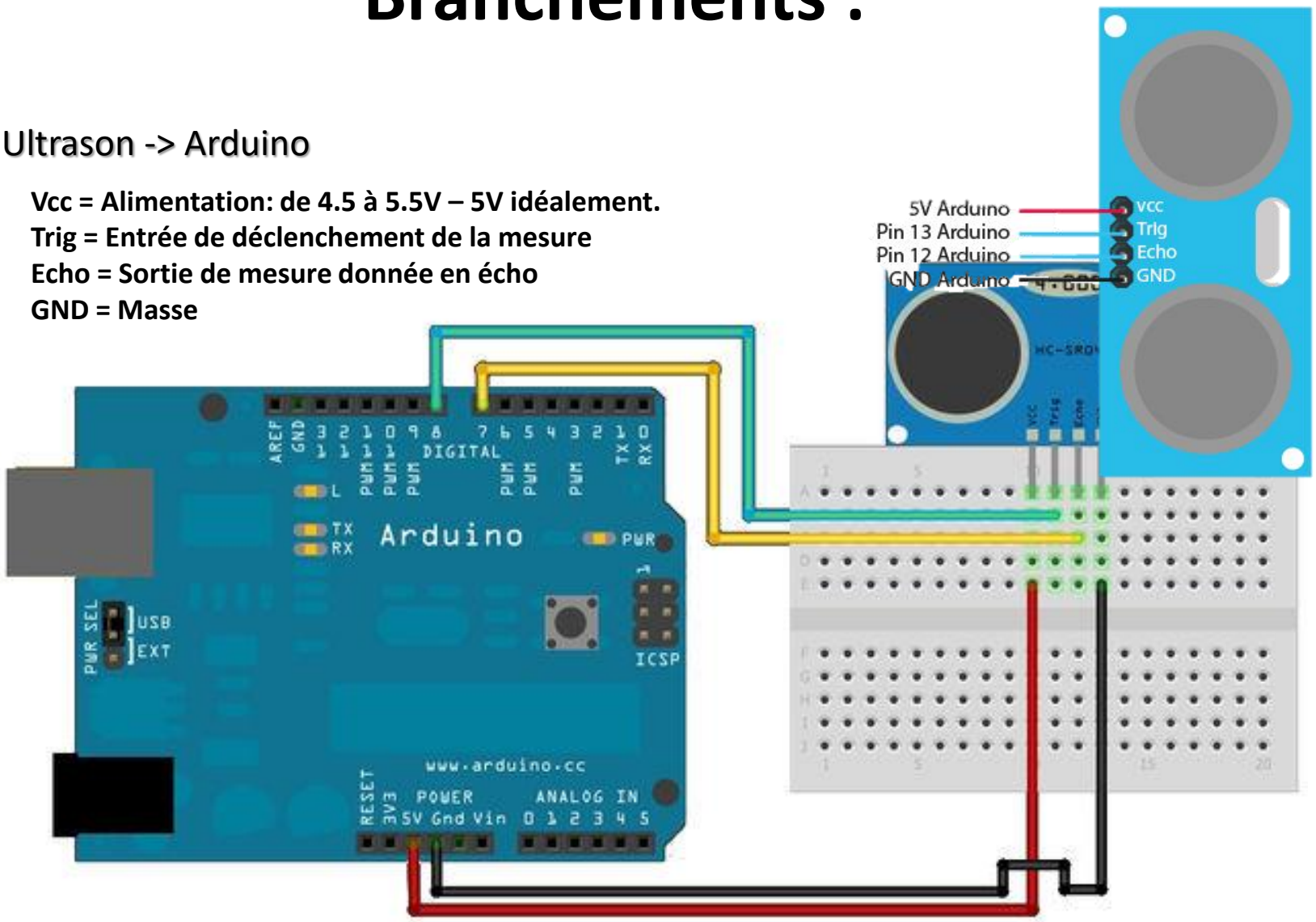
Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho.



Branchements :

Ultrason -> Arduino

- **Vcc = Alimentation: de 4.5 à 5.5V – 5V idéalement.**
- **Trig = Entrée de déclenchement de la mesure**
- **Echo = Sortie de mesure donnée en écho**
- **GND = Masse**



Code :

```
1  int trigPin = 13; // On défini la pin 13 pour le trig
2  int echoPin = 12; // On défini la pin 12 pour l'echo
3
4  void setup() {
5    Serial.begin (9600);
6    pinMode(trigPin, OUTPUT); // Le trig en sortie.
7    pinMode(echoPin, INPUT); // L'echo en entrée.
8  }
9
10 void loop() {
11   long duration;
12   float distance; // Modification de distance en type float pour les décimales.
13   digitalWrite(trigPin, LOW);
14   delayMicroseconds(2);
15   digitalWrite(trigPin, HIGH);
16   delayMicroseconds(10);
17   digitalWrite(trigPin, LOW);
18   duration = pulseIn(echoPin, HIGH);
19   distance = (duration/2) / 29.1;
20
21   if (distance >= 200 || distance <= 0){
22     Serial.println("Out of range");
23   }
24   else {
25     Serial.print(distance);
26     Serial.println(" cm");
27   }
28   delay(100);
29 }
```

Servo Moteur

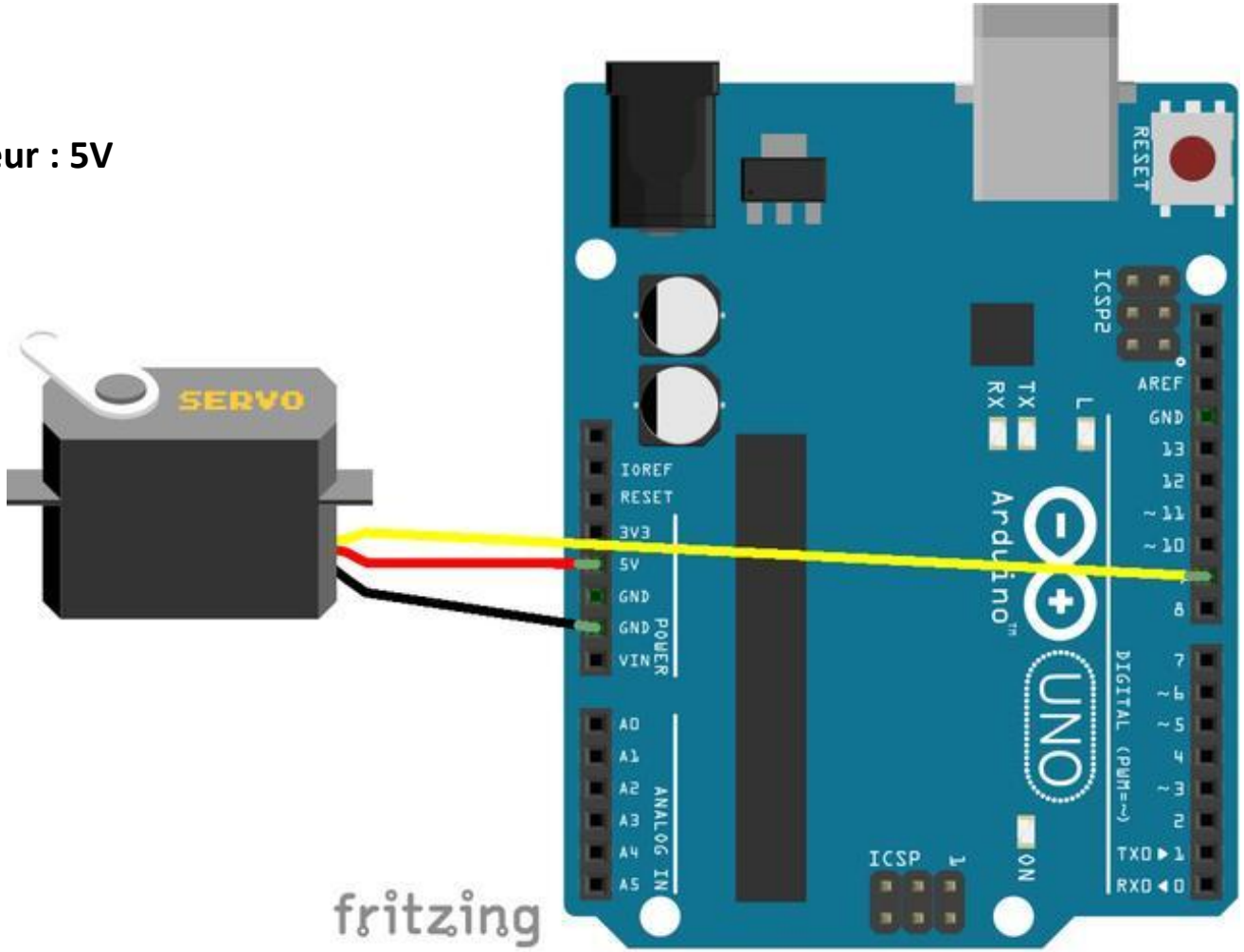
Les servomoteurs sont des moteurs un peu particuliers, qui peuvent tourner avec une liberté d'environ 180° et garder de manière relativement précise l'angle de rotation que l'on souhaite obtenir.

Capteur IR

Branchements :

ServoMoteur -> Arduino

- **Fil rouge du servomoteur : 5V**
- **Fil noir : GND**
- **Fil jaune : Pin 9**



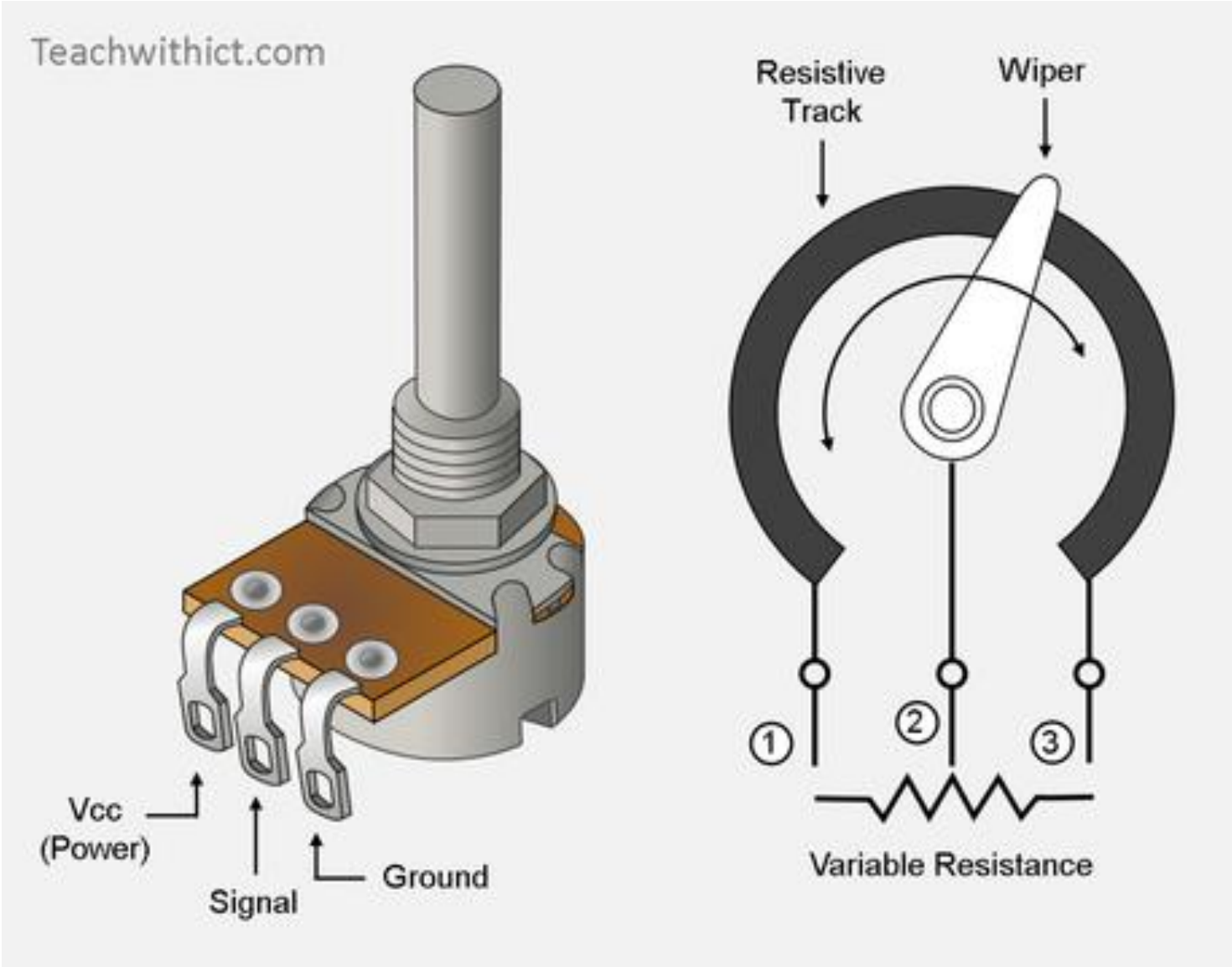
Code :

```
 /* Inclut la lib Servo pour manipuler le servomoteur */
#include <Servo.h>

/* Créer un objet Servo pour contrôler le servomoteur */
Servo monServomoteur;

void setup() {

  // Attache le servomoteur à la broche D9
  monServomoteur.attach(9);
}

void loop() {

  // Fait bouger le bras de 0° à 180°
  for (int position = 0; position <= 180; position++) {
    monServomoteur.write(position);
    delay(15);
  }

  // Fait bouger le bras de 180° à 10°
  for (int position = 180; position >= 0; position--) {
    monServomoteur.write(position);
    delay(15);
  }
}
```

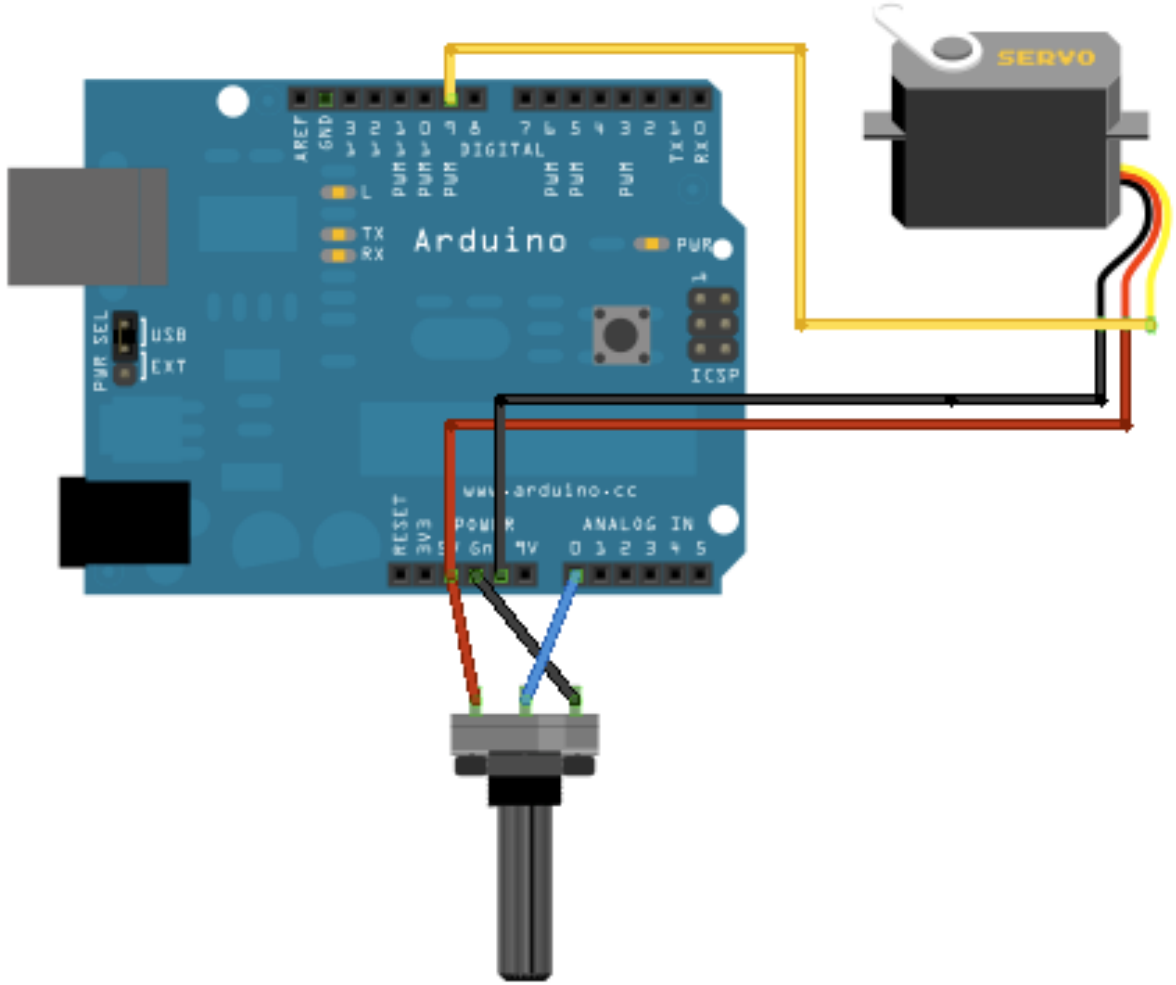
APPLICATION

Contrôler un ServoMoteur a l’aide d’un potentiomètre :

Matériel nécessaire :



MONTAGE :


SERVO
Arduino
DIGITAL
ANALOG IN
PWR SEL
USB
EXT
ICSP
RESET
www.arduino.cc

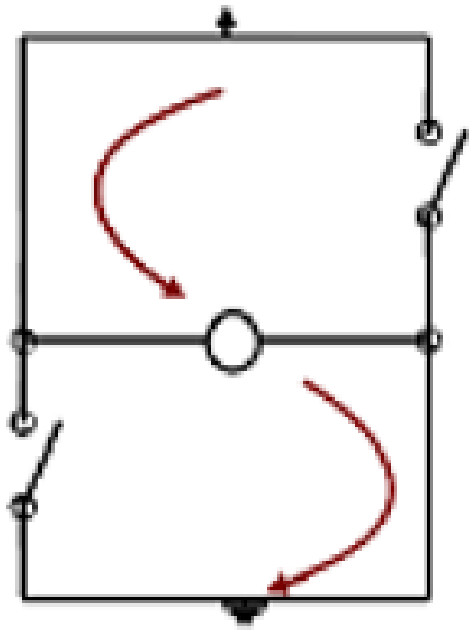
Driver Motor (Shield)

Pont-H L298N

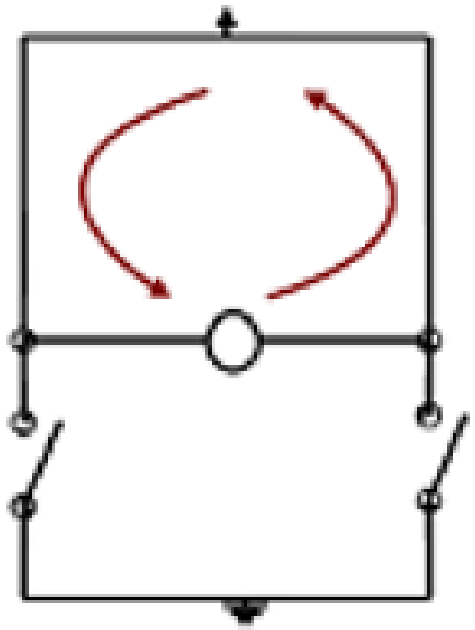
Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation.

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteur courant continu ou un moteur pas-à-pas 4 fils 2 phases. il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

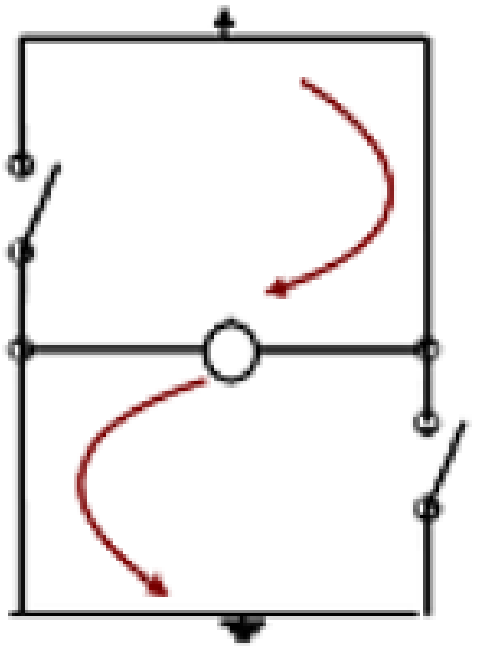
Pont-H



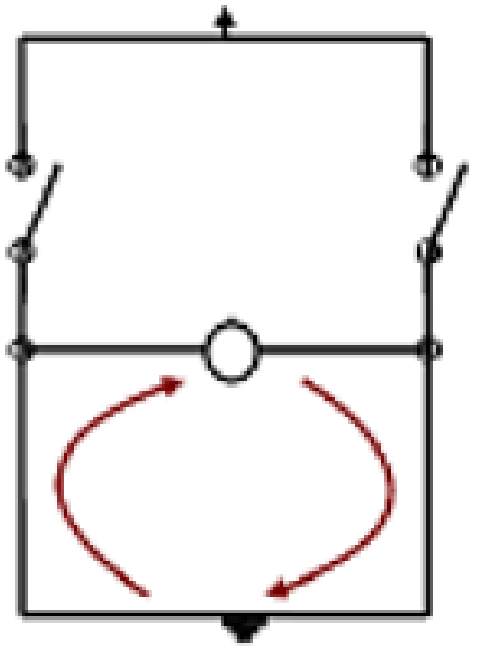
SENS1



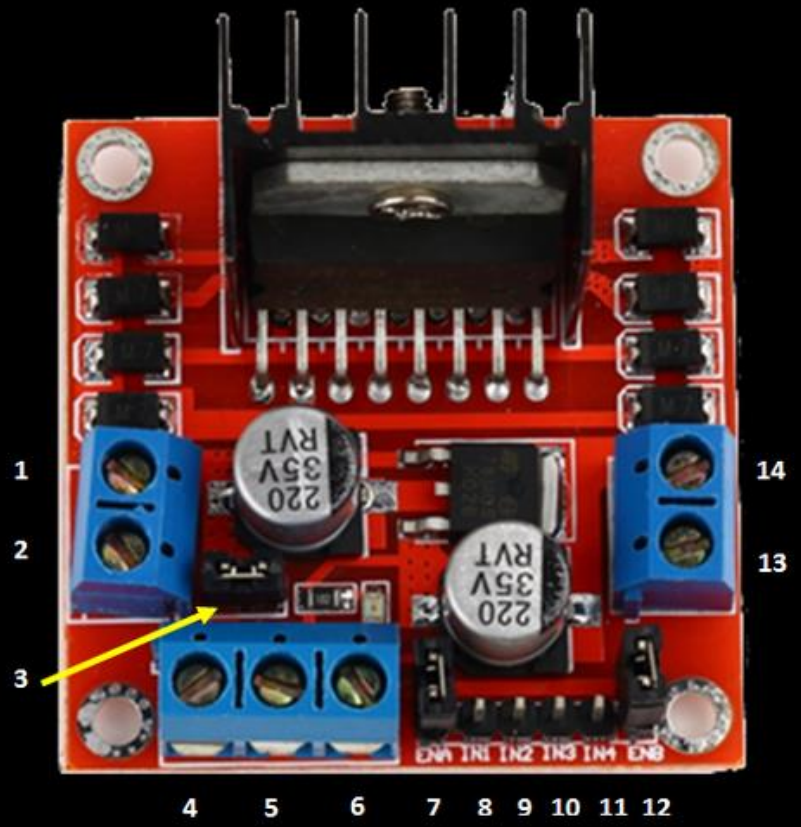
FREIN



SENS2



FREIN



1 Positif +" du Moteur 1

2 Ground -" du Moteur 1

3 cavalier 12V à désactiver si le voltage est supérieur à 12V. Permet d'alimenter le régulateur interne de 5V.

4 alimentation + jusqu'à 35V (retirer le cavalier en 3 si voltage supérieur à 12V). Le Ground de l'alimentation se branche sur l'Arduino.

5 Ground à brancher sur l'Arduino).

6 sortie **5V** pour alimenter l'Arduino si l'alimentation est supérieure à 12V et que le cavalier en 3 est mis. Il est possible d'alimenter l'Arduino et 1 moteur DC avec une pile 9V, mais cela reste juste.

7 ENA moteur 1, à brancher sur PWM de l'Arduino (ex: PIN10), elle permet de gérer la vitesse du moteur. Pour cela retirer le cavalier.

8 IN1 à brancher sur l'Arduino (ex: PIN9)

9 IN2 à brancher sur l'Arduino (ex: PIN8)

10 IN3 à brancher sur l'Arduino (ex: PIN7)

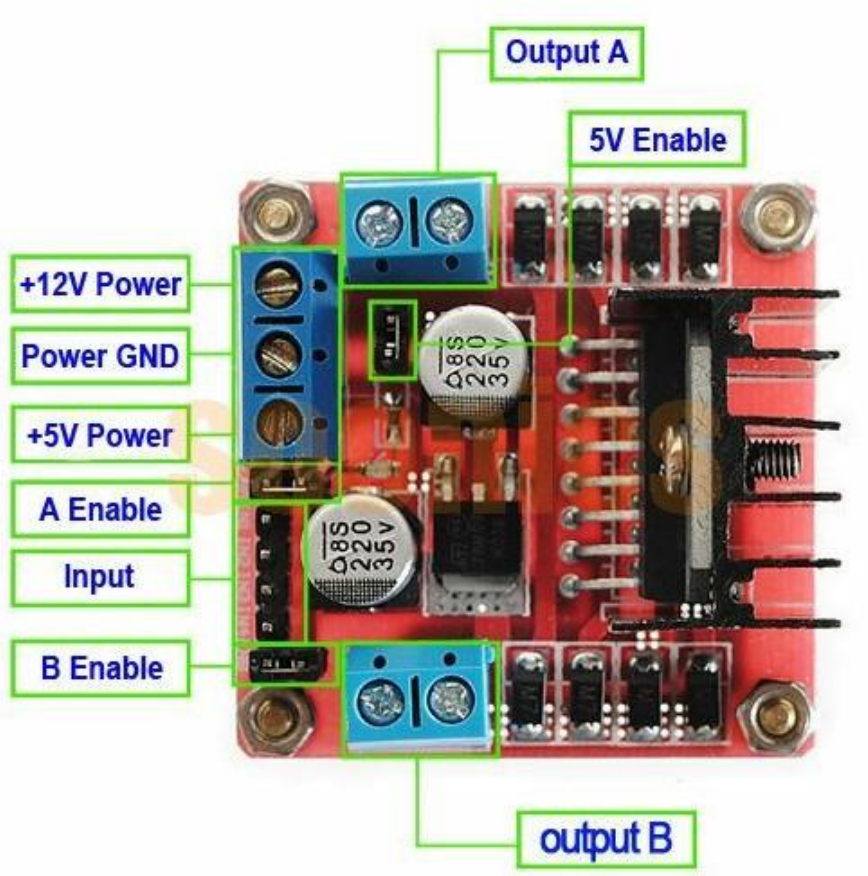
11 IN4 à brancher sur l'Arduino (ex: PIN6)

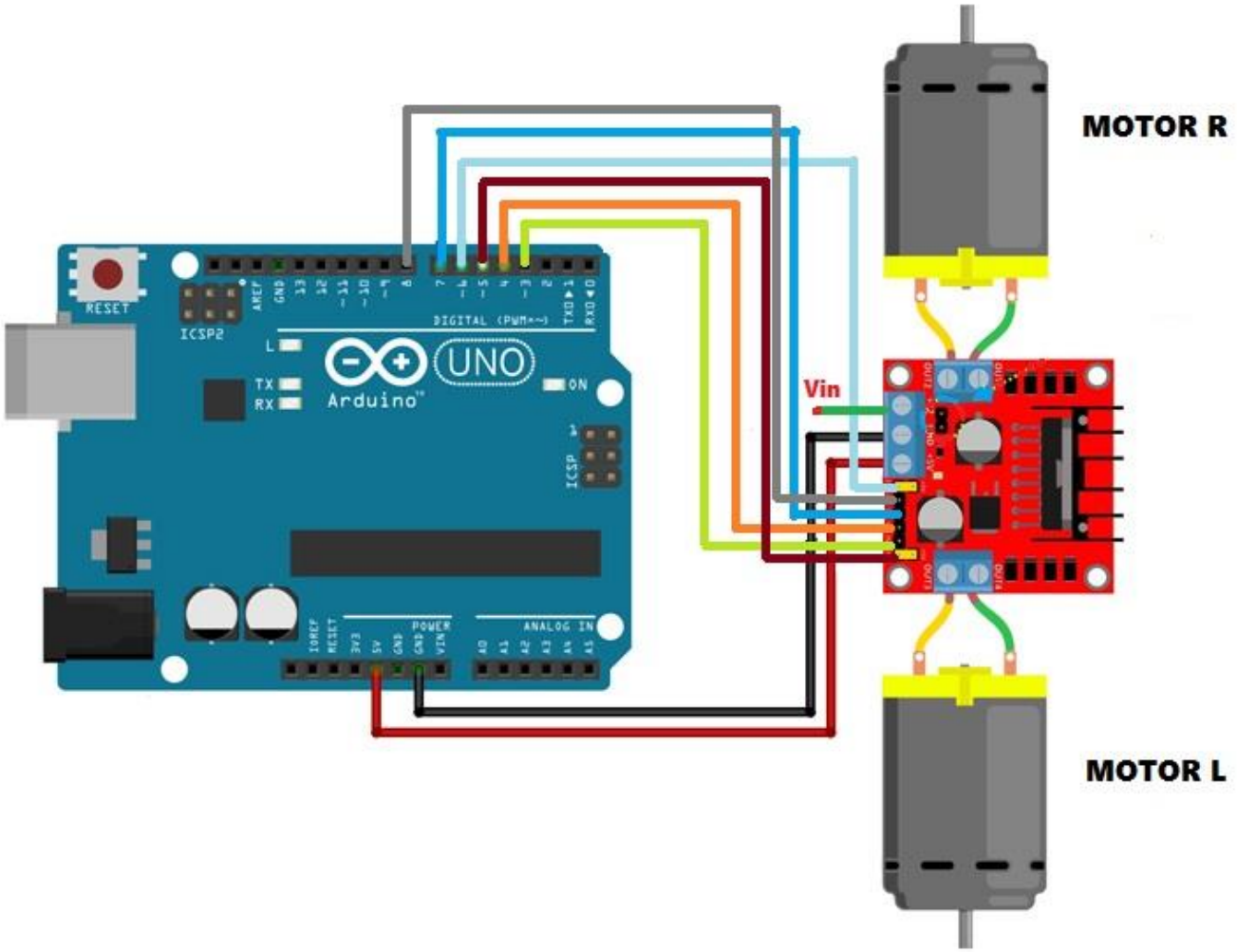
12 ENB moteur 2, à brancher sur PWM de l'Arduino (ex: PIN5), elle permet de gérer la vitesse du moteur. Pour cela retirer le cavalier.

13 Positif + du Moteur 2

14 Ground - du Moteur 2

Montage :


Output A
5V Enable
+12V Power
Power GND
+5V Power
A Enable
Input
B Enable
output B


MOTOR R
Vin
MOTOR L
UNO
Arduino

Motor DC

CODE - 1

```
//Motor Control - Motor A: motorPin1,motorpin2 & Motor B: motorpin3,motorpin4

//This code  will turn Motor A clockwise for 2 sec.
analogWrite(motorPin1, 180);
analogWrite(motorPin2, 0);
analogWrite(motorPin3, 180);
analogWrite(motorPin4, 0);
delay(5000);
//This code will turn Motor A counter-clockwise for 4 sec.
analogWrite(motorPin1, 0);
analogWrite(motorPin2, 180);
analogWrite(motorPin3, 0);
analogWrite(motorPin4, 180);
delay(5000);

//This code will turn Motor B clockwise for 3 sec.
analogWrite(motorPin1, 0);
analogWrite(motorPin2, 180);
analogWrite(motorPin3, 180);
analogWrite(motorPin4, 0);
delay(1000);
//This code will turn Motor B counter-clockwise for 1 sec.
analogWrite(motorPin1, 180);
analogWrite(motorPin2, 0);
analogWrite(motorPin3, 0);
analogWrite(motorPin4, 180);
delay(1000);
```

CODE - 2

```
#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
```

```
// Set Motor A backward
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
// Set Motor B backward
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);


analogWrite(enA, motorSpeedA); // Send PWM signal to motor A
analogWrite(enB, motorSpeedB); // Send PWM signal to motor B
```

Thank You