Interfaces Réseau et Objets Communicants

La carte Arduino

Salvatore M. Anzalone
sanzalone@univ-paris8.fr

La carte Arduino

- Introduction
- Rappelles d'électronique
- La carte Arduino UNO
- Entrées et sortie numériques
  - LED clignotante
  - Allumer et éteindre un LED
- Entrées et sortie analogiques
  - Émettre une séquence de tons
  - Choisir la fréquence de tons émis
- Shields
  - Serveur TCP

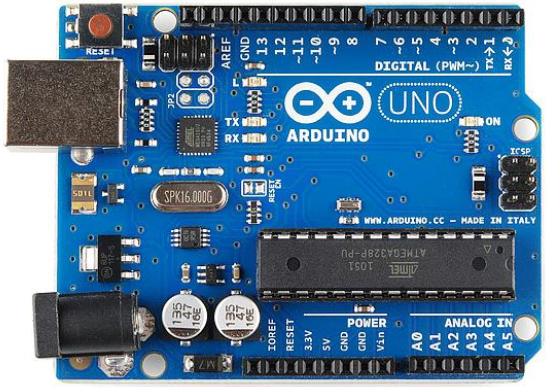
Introduction

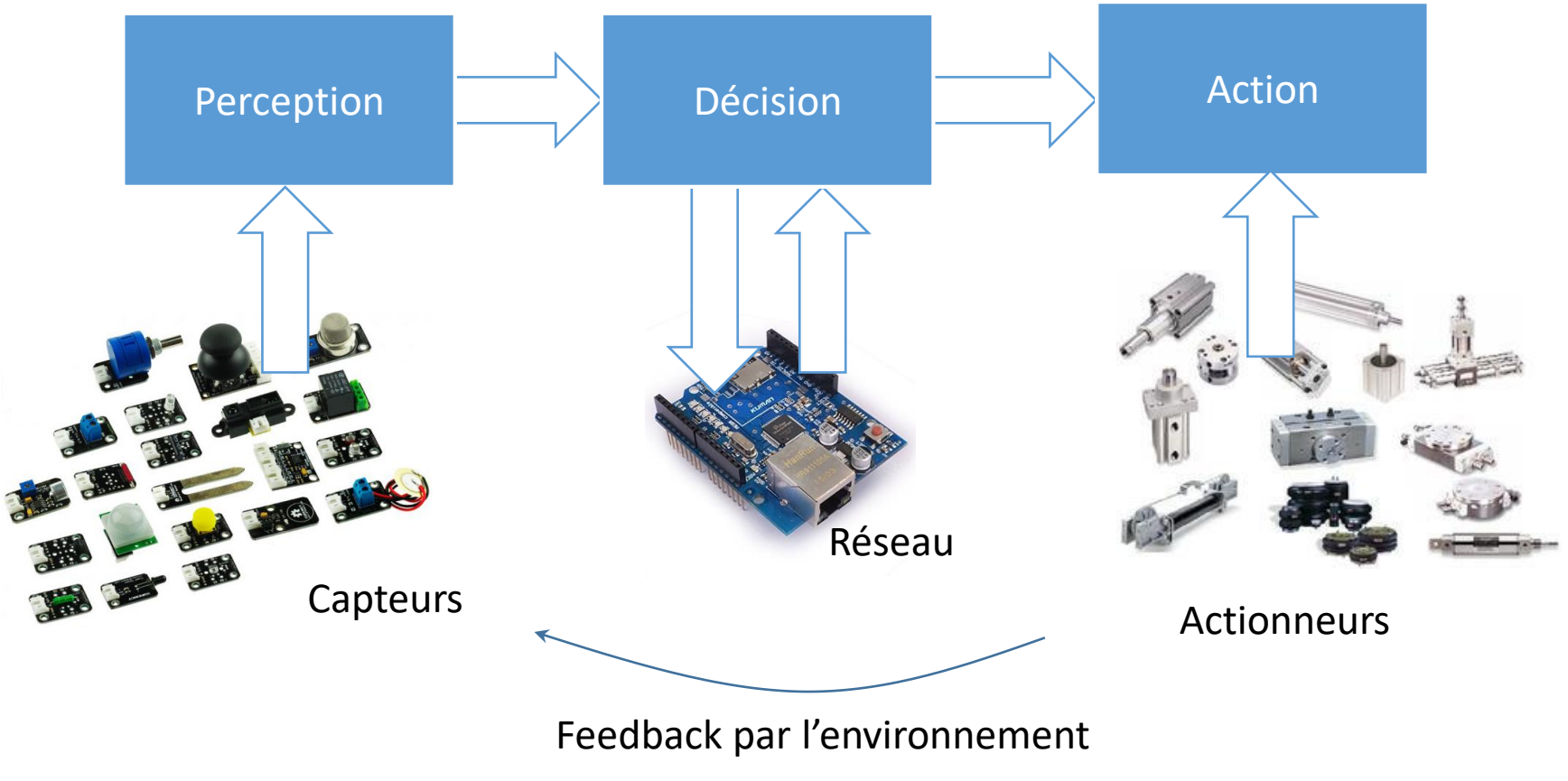
Introduction

- Internet des objets
  - Objets intelligents, « speficic purpose », communicantes

Introduction

- Boucle perception-action
  - Microcontrôleur





Introduction

- Microcontrôleur
  - Un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties.
- **Arduino**: Open Hardware et Open Software


Arduino RS232 (thru-hole parts)


Arduino Diecimila


Arduino Duemilanove (rev 2009b)


Arduino Uno R2

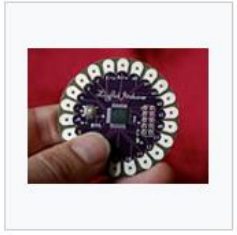

Arduino Uno SMD R3


Arduino Leonardo


Arduino Pro (No USB)


Arduino Mega


Arduino Nano (DIP-30 footprint)


Arduino LilyPad 00 (rev 2007) (No USB)


Arduino Robot


Arduino Esplora


Arduino Ethernet (AVR + W5100)


Arduino Yun (AVR + AR9331)


Arduino Due (ARM Cortex-M3 core)

Introduction

- Arduino est expansible à travers ses « shields »

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Rappelles d’électronique

Rappelles d'électronique

- Circuits électriques
  - *« ensemble simple ou complexe de conducteurs et de composants électroniques parcourus par un courant électrique »*

- Courant électrique:
  - « *un déplacement d'ensemble de porteurs de charges électriques, généralement des électrons, au sein d'un matériau conducteur* »

  - Ces déplacements sont imposés par l'existence d'une différence de potentiel entre deux points du circuit.

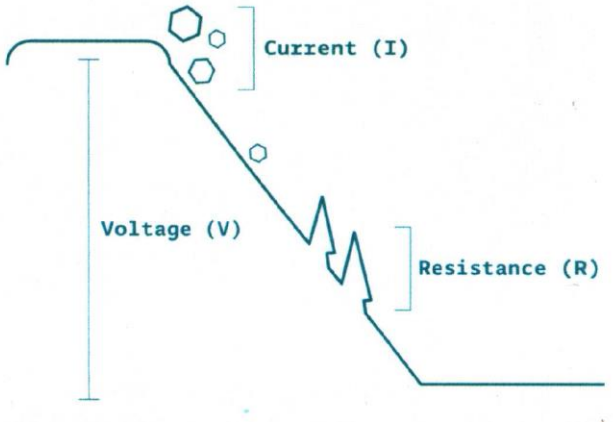
Rappelles d'électronique

- Intensité de courante
  - Charge électrique par unité de temps, en Ampères, [A].

$$I = {}^{Q}/_{t}$$

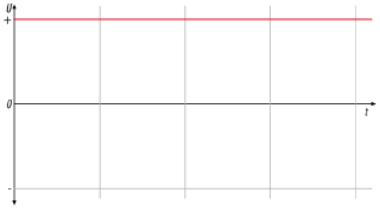
- Tension
  - Différence de potentielle entre deux points du circuit, en Volts, [V].
- Résistance
  - Propriété d'un composant à s'opposer au passage d'un courant. En Ohm, [Ω].

$$I = {}^{V}/_{R}$$

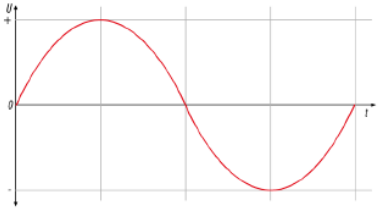


Analogie avec l'énergie cinétique

Courant continu (CC ou DC, Direct Current): tension indépendante du temps



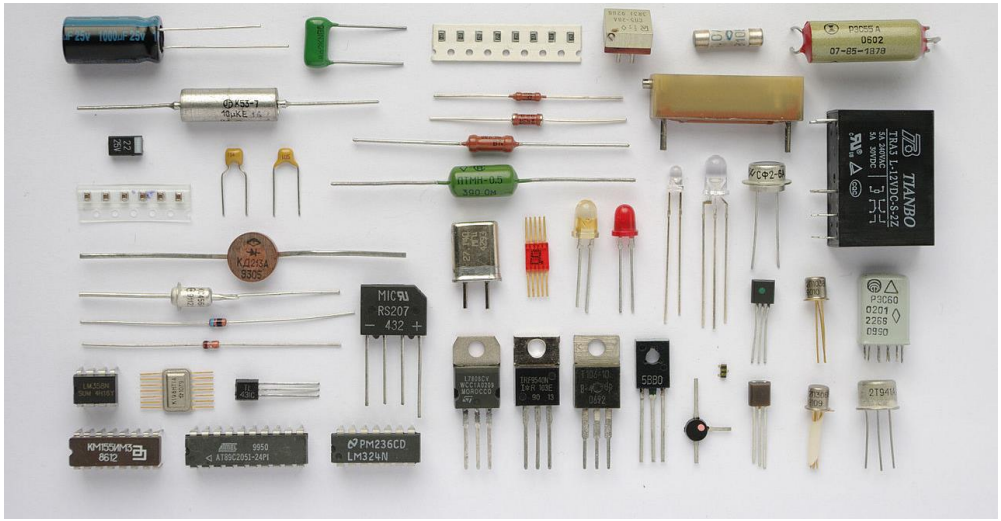
Courant alternatif (CA ou AC, Alternating Current): tension périodique qui oscille entre deux valeurs limites



Rappelles d'électronique

- Composants Electroniques
  - éléments destinés à être assemblé avec d'autres afin de réaliser une ou plusieurs fonctions électroniques

  - Résisteurs
  - Condensateurs
  - Bobines
  - Diodes
  - Transistors
  - Circuits Intégrées
  - …



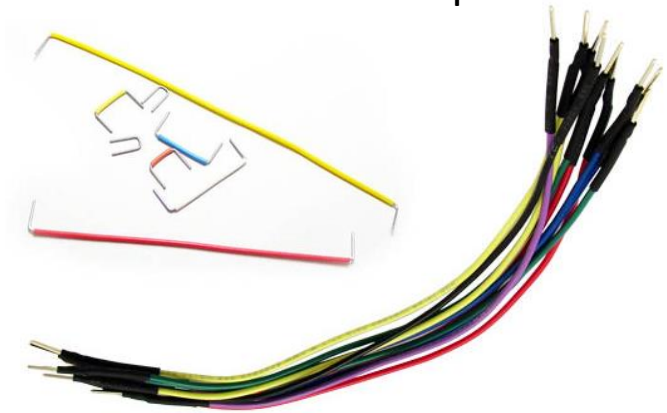
✓Chaque composant fonctionne en accord à ses caractéristiques physiques et électriques

✓Ces spécifiques de fonctionnement sont présentées en détail dans les fiches techniques des constructeurs (Datasheets)

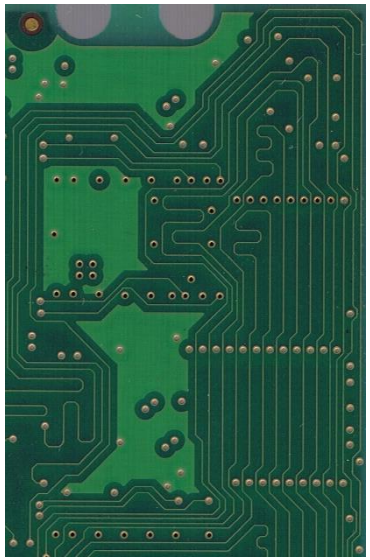
Rappelles d'électronique

- Les composantes électriques sont connectes à travers des conducteurs pour créer les circuits

Câbles électriques

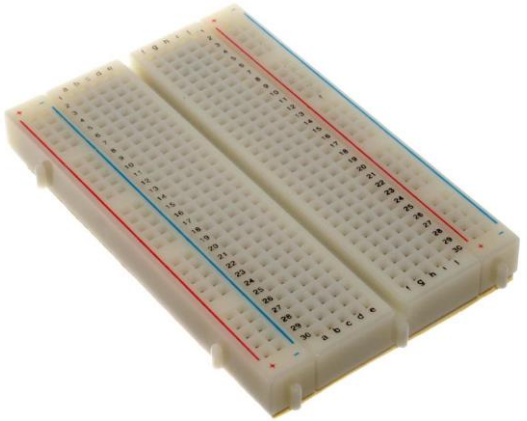


Circuits imprimés



Printed circuit board

Platine d'expérimentation

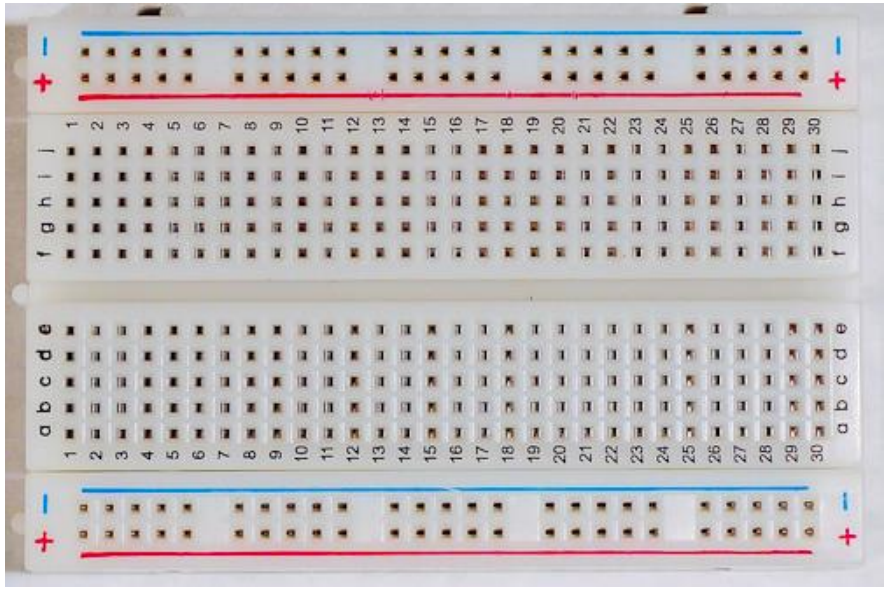


Breadboard

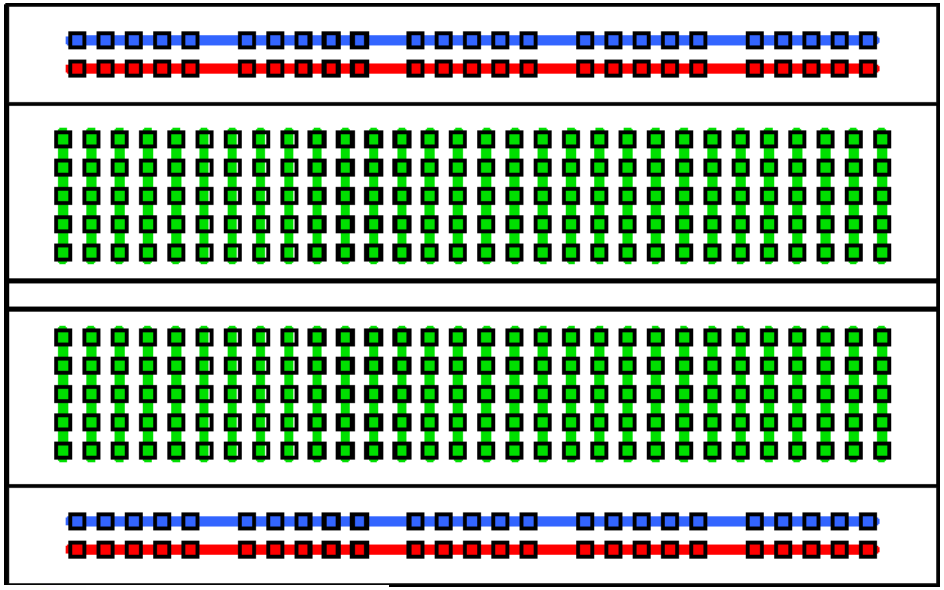
Rappelles d'électronique

- Platine d'expérimentation (*Breadboard*)

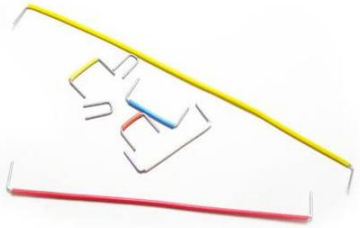
Vue de dessous



Connections

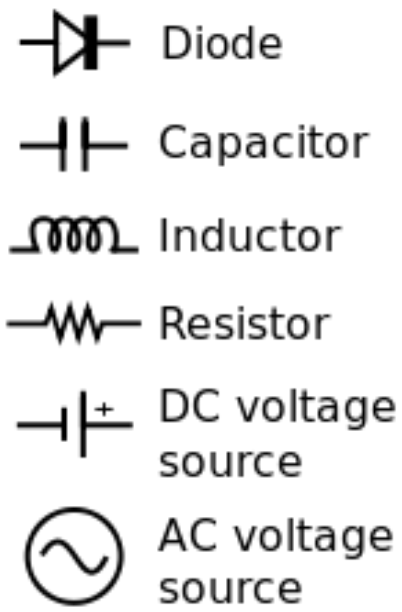


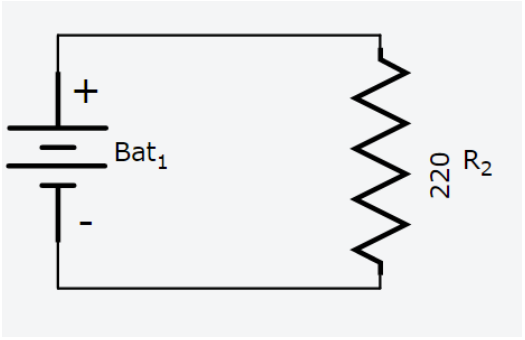
*Autres connections possibles à travers « **jumpers** » :*

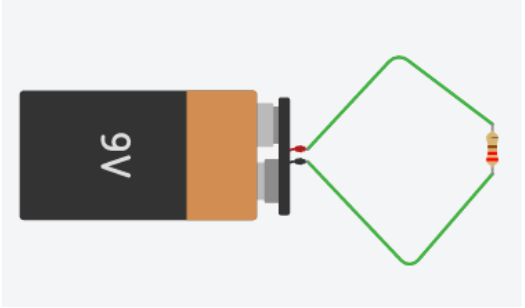


Rappelles d'électronique

- Schéma électriques
  - Représentation graphique des interconnections des composantes électroniques dans le circuit



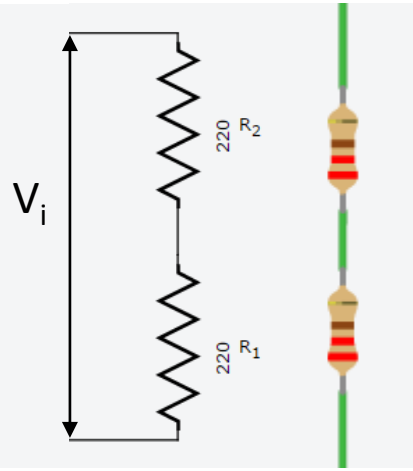




Rappelles d'électronique

- Composantes en série et en parallèle

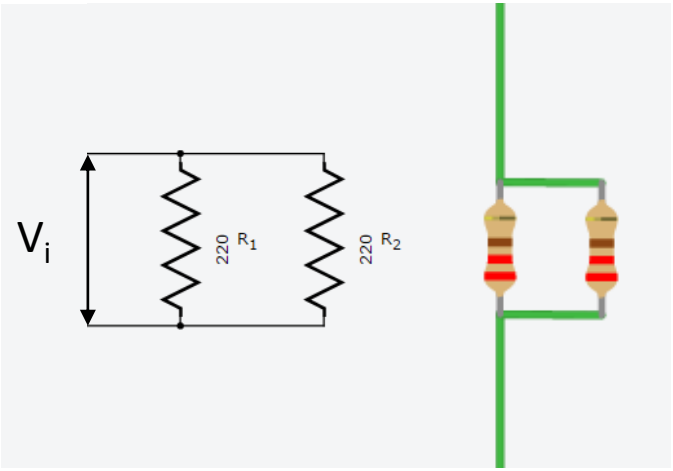
Résistances en série
La tension se partage entre les deux résistances.



La résistance équivalente est la somme des résistances.

$$R_{eq} = R_1 + R_2$$

Résistances en parallèle
La courante se partage entre les deux résistances.



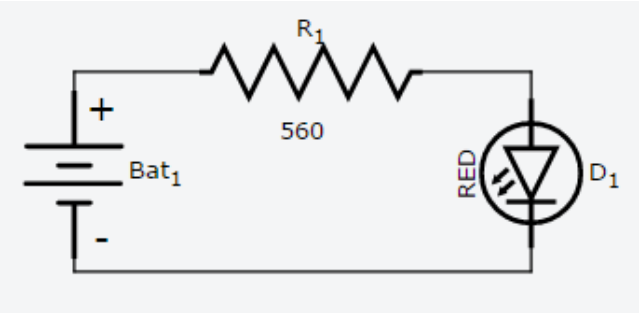
La résistance équivalente est:

$$R_{eq} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}$$

Attention aux combinaisons!

Rappelles d'électronique

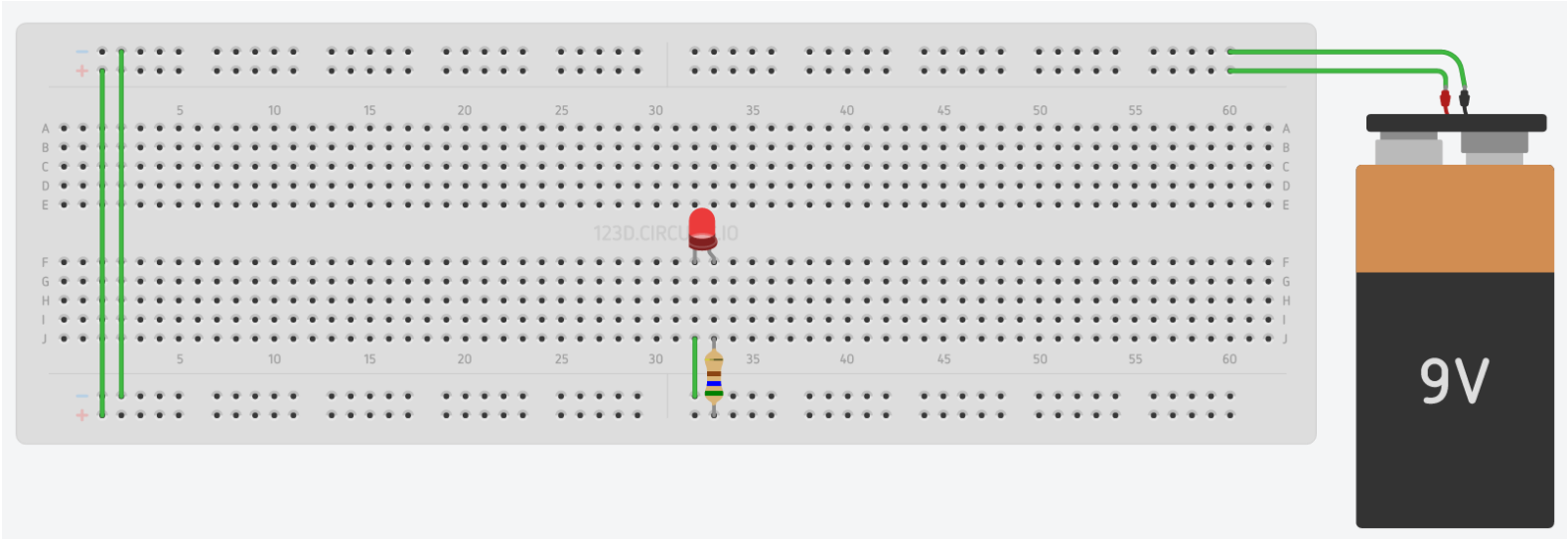
- Un petit circuit pour allumer un LED



- ✓ La batterie produit 9V
- ✓ Le diode emploie 1.8 V (Vf: forward voltage) en consommant une courante de 25mA
- ✓ On a besoin d'une résistance de:

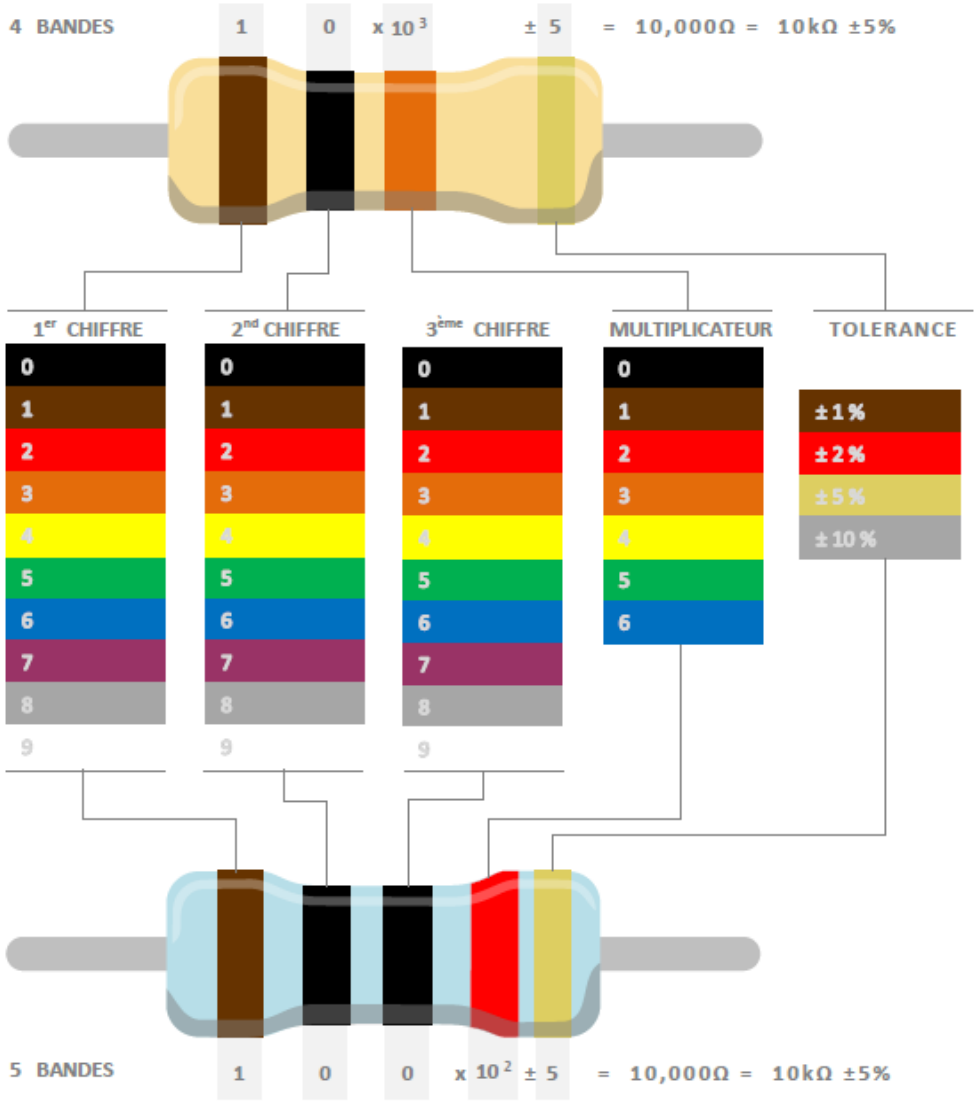
$$R_1 = \frac{V_1}{I} = \frac{V_{Bat1} - V_{D1}}{I} = \frac{9 - 1.8}{0.025} = 288\Omega$$

Resistor 560 ohm



Rappelles d'électronique

Codes résistances

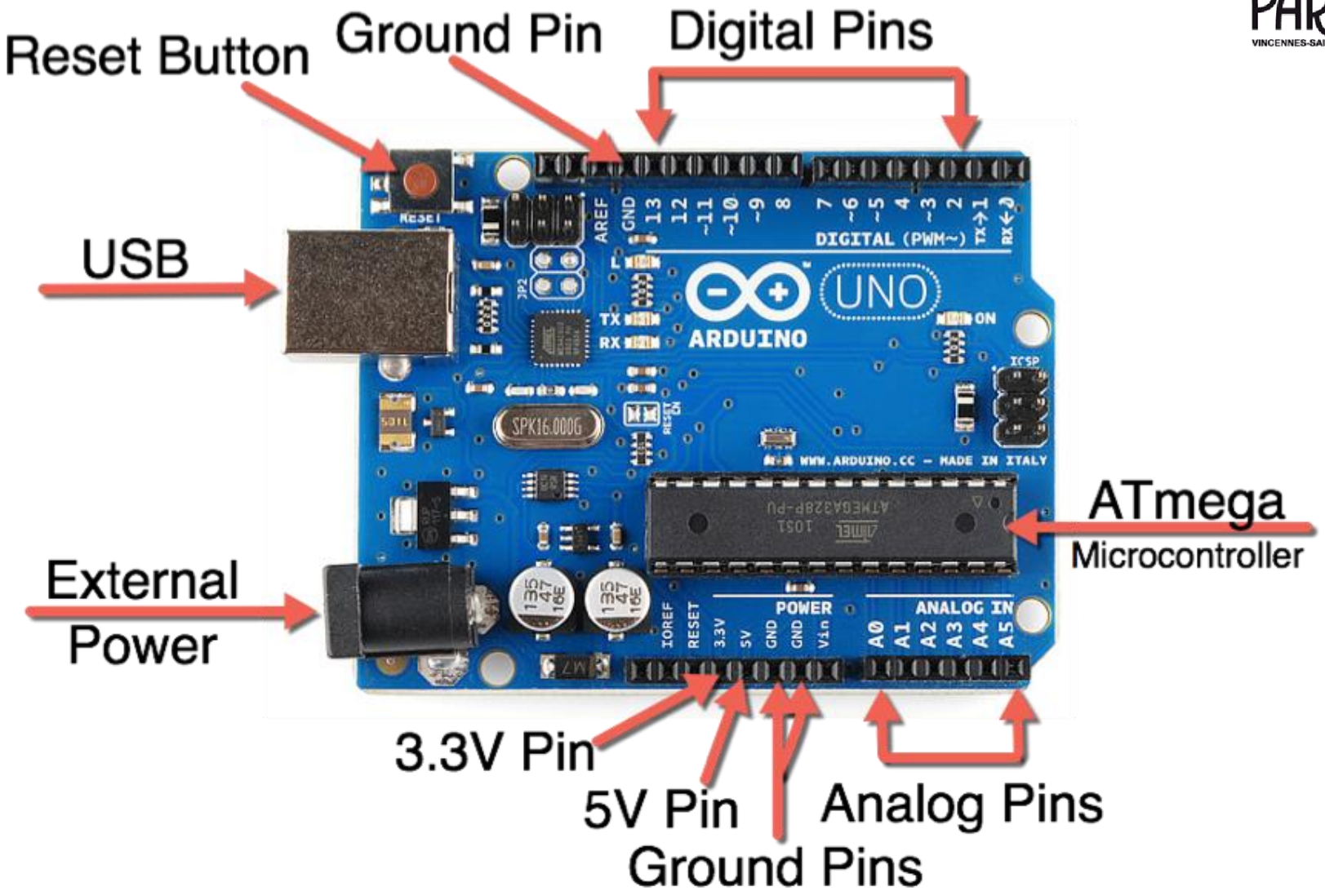


Rappelles d'électronique

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

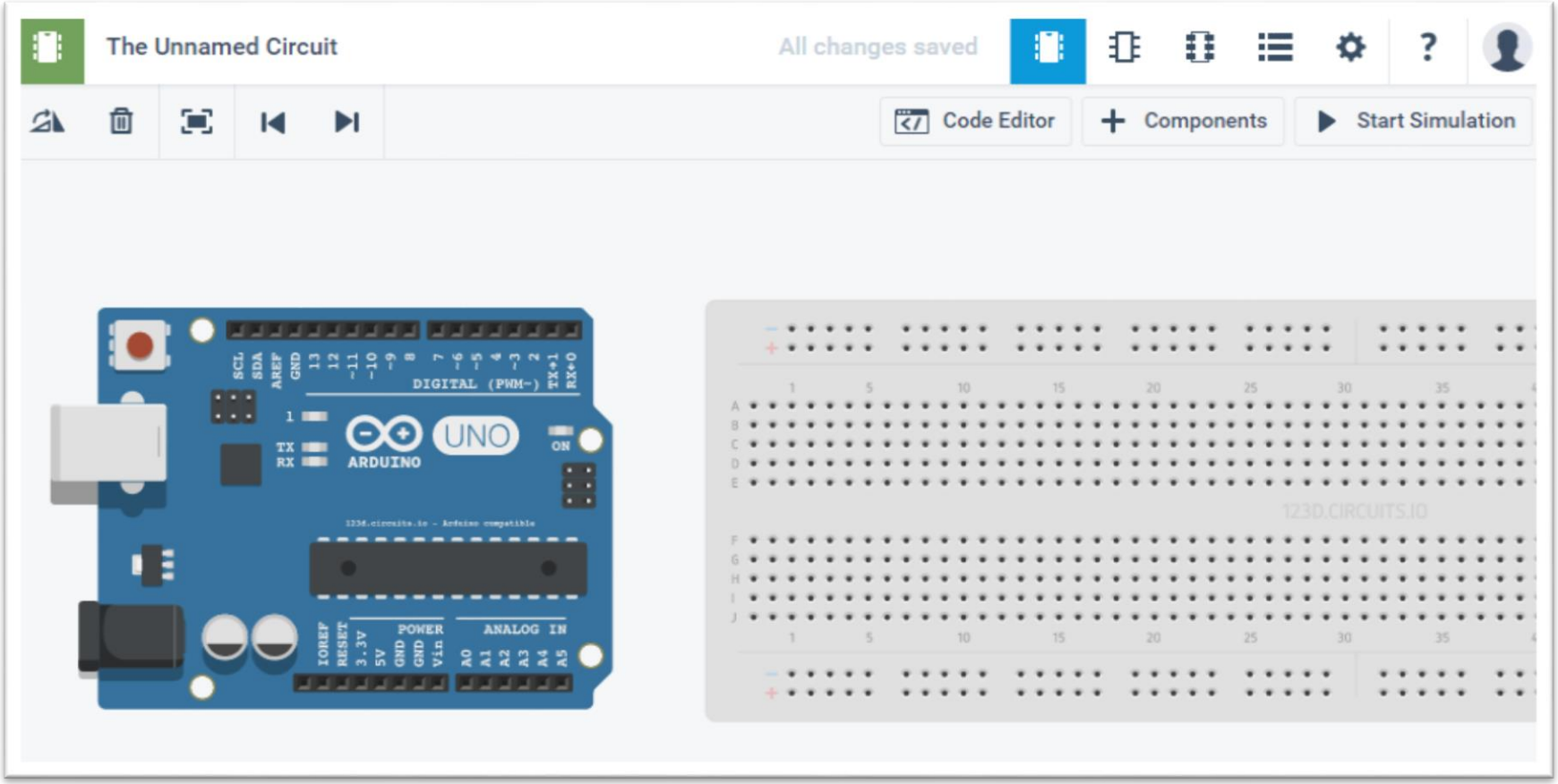
La carte Arduino UNO

La carte Arduino UNO

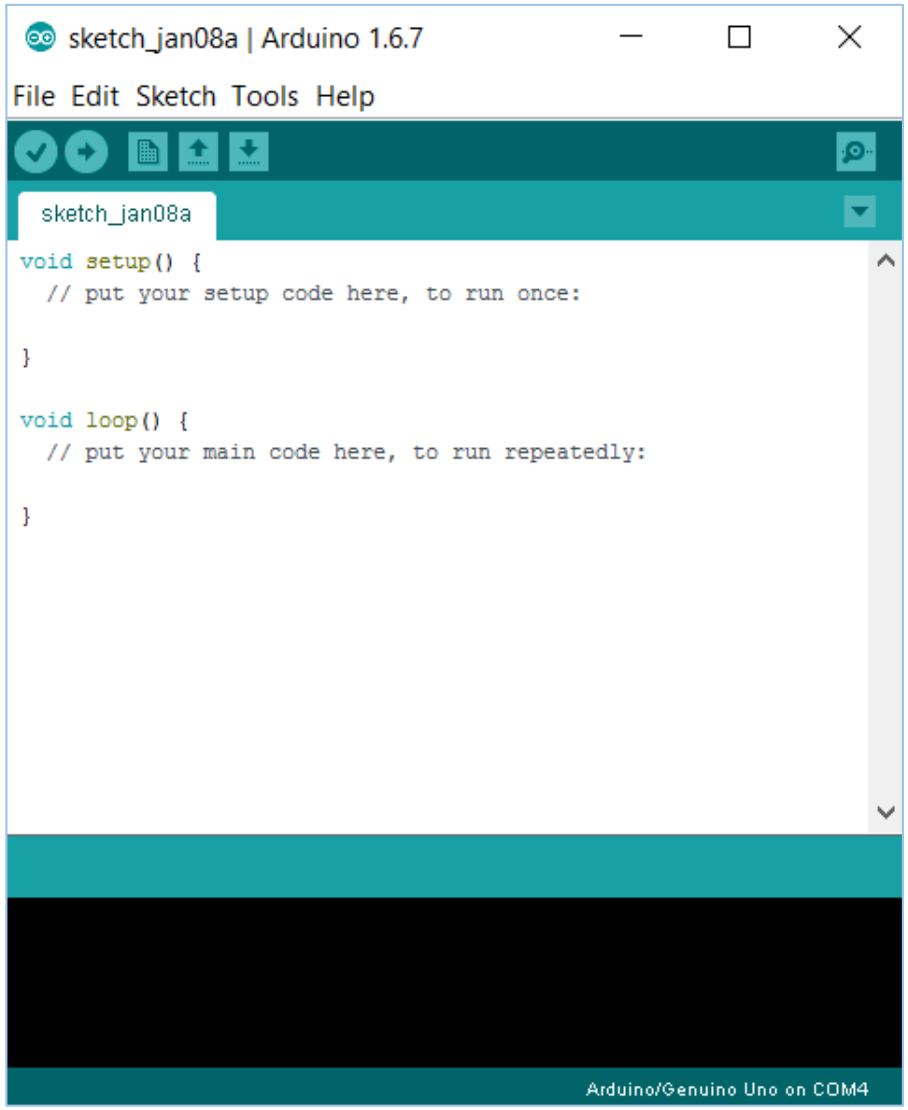


La carte Arduino UNO

- Simulateur Online: https://www.tinkercad.com



La carte Arduino UNO



- L'environnement de développement
- Langage C
- Deux events:
  - **void setup()**
    - Initialisation des variables
    - Configuration des broches
  - **void loop()**
    - Boucle principale du programme
    - Connection perception-action

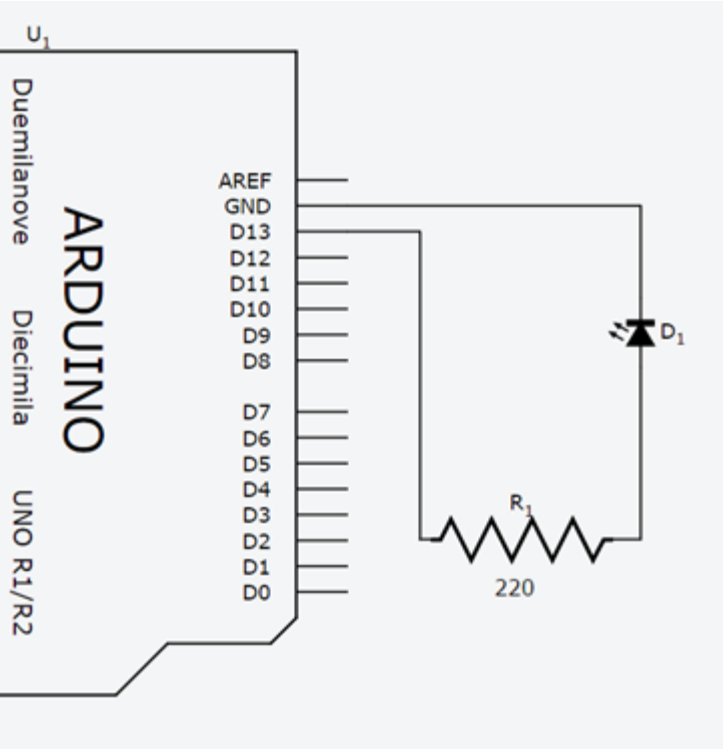
UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Entrées et sorties numériques

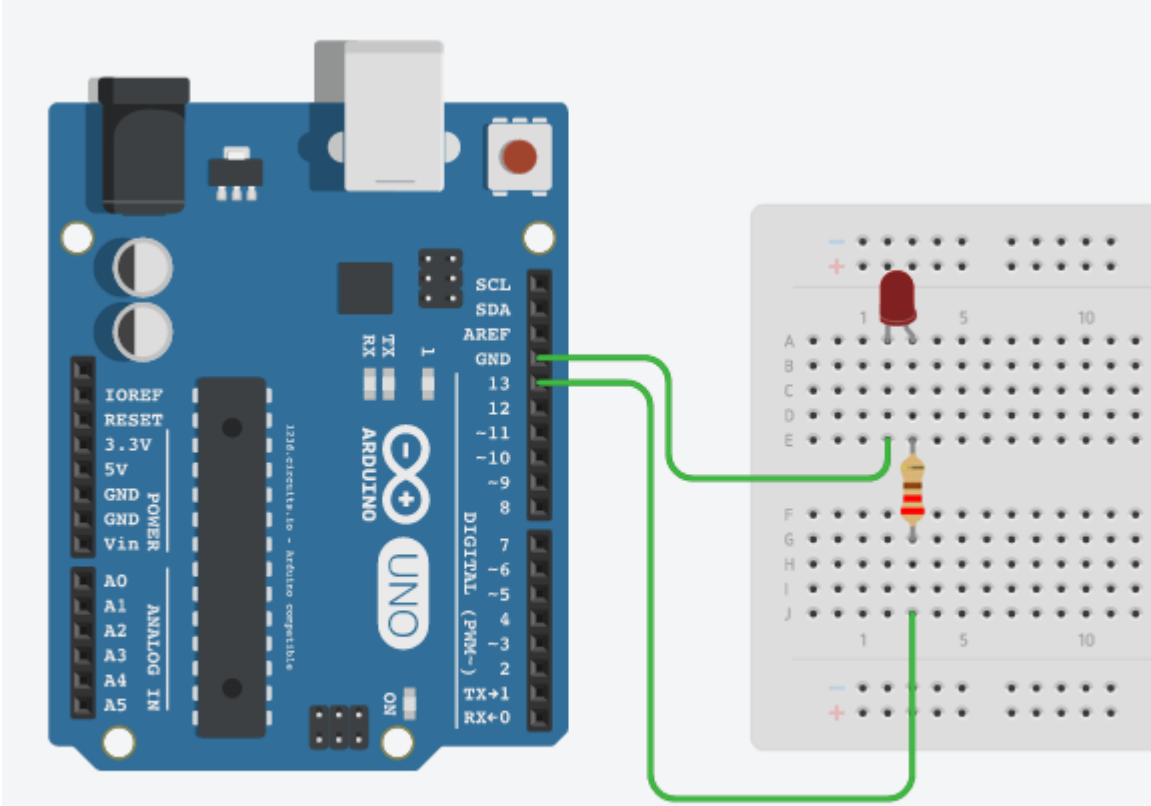
Sortie numérique

- LED clignotant

Schéma



Connections



Sortie numérique

- LED clignotant

```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

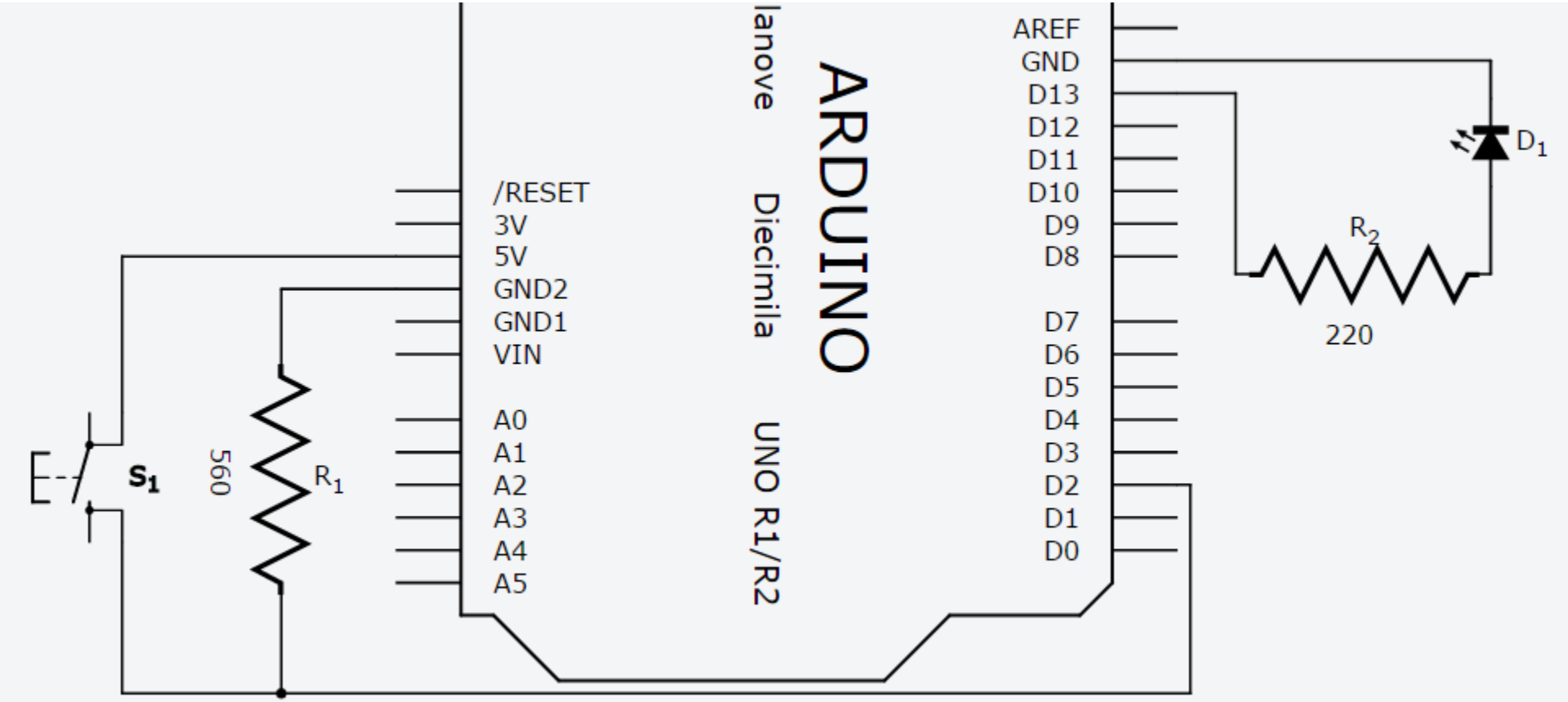
pinMode *initialize un pin digitale comme entrée ou sortie*

digitalWrite *écrive une valeur digitale sur le pin*

Entrée numérique

- Allumer et éteindre un LED

Schéma



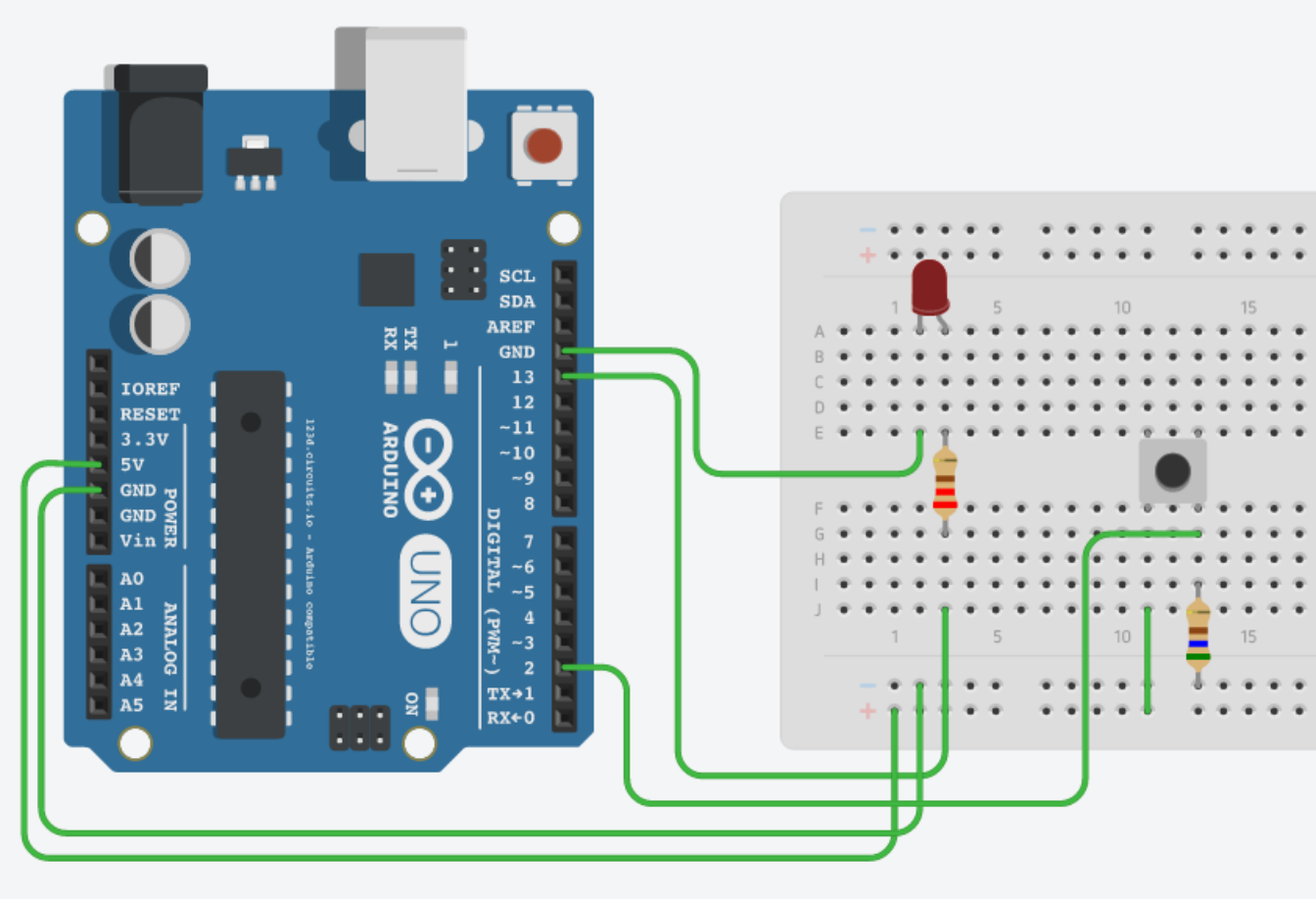
Entrée numérique

- Allumer et éteindre un LED

Connections

Attention au câble sur le pin 2 qui "sonde" la sortie de l'interrupteur.

On utilise une résistance de 560Ω pour ne créer pas un court-circuit quand l'interrupteur est fermé.



Entrée numérique

- Allumer et éteindre un LED

```
// Pin 13 has an LED connected
int led = 13;
// Pin 2 has a pushbutton
int pushbutton = 2;

// Led status
bool led_status = false;

// the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
  pinMode(led, OUTPUT);
  // initialize the digital pin as an input.
  pinMode(pushbutton, INPUT);
}

…
```

digitalRead *lis une valeur digitale*

```
…

// the loop routine runs over and over again forever:
void loop() {
  // read the pushbutton status
  int pushbutton_status = digitalRead(pushbutton);

  // if button pressed, change led status
  if(pushbutton_status == HIGH) {
    led_status = !led_status;
    delay(250);
  }

  // turn on or off the led according to its status
  if(led_status) {
    digitalWrite(led, HIGH);   // turn the LED on (HIGH is the
voltage level)
  } else {
    digitalWrite(led, LOW);    // turn the LED off by making the
voltage LOW
  }
}
```

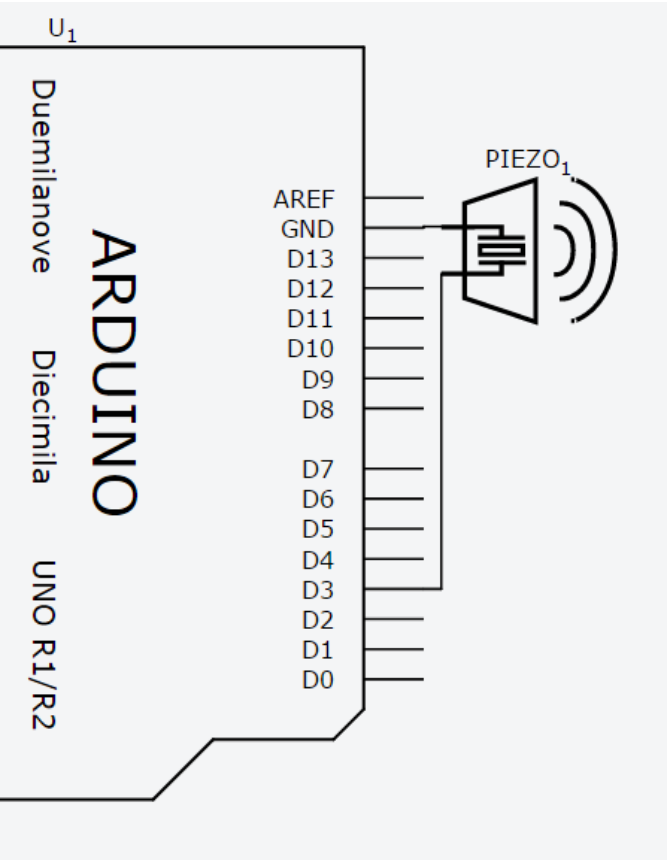
UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Entrées et sorties analogiques

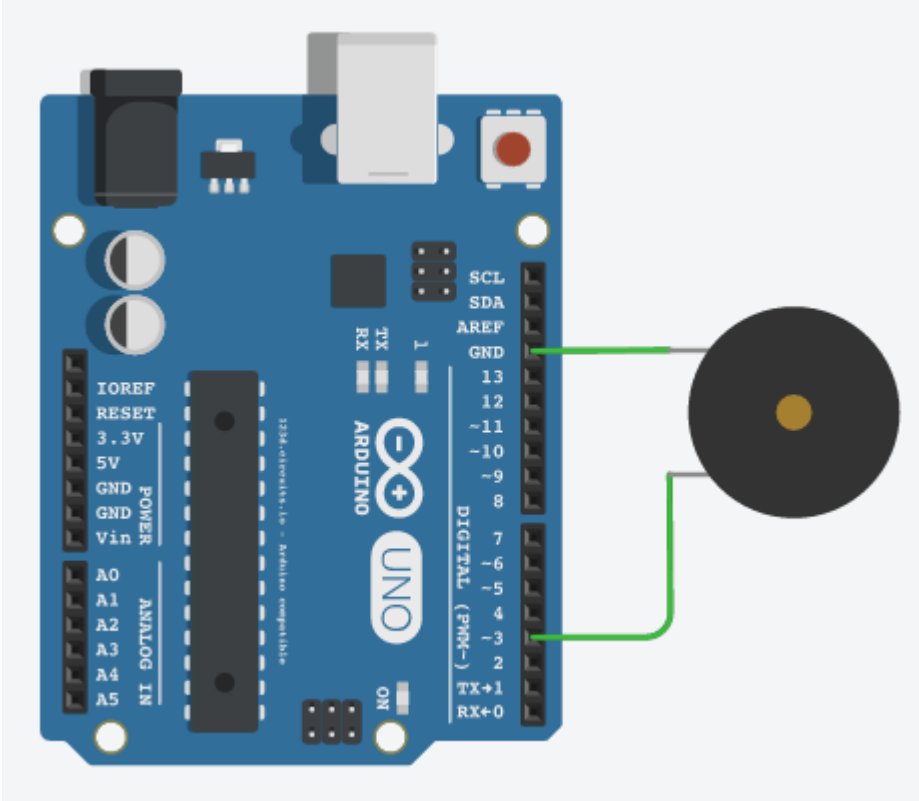
Sortie analogique

- Émettre de sons

Schéma



Connections



Attention: *pour les sorties analogiques on peut utiliser seulement les pins notés avec une tilde* ~ (3, 5, 6, 9, 10, 11).

Sortie analogique

- Émettre de sons

```
// include mathematical functions
#include <math.h>

// Pin 3 has a piezo
int piezo = 3;
// temps
float t = 0;

// the setup routine runs once when you press reset:
void setup() {
  // initalize serial for communication
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // calculate tone value
  int freq = 500 * sin(t) +500;
  t += 0.05;
  Serial.println(freq);
  tone(piezo, freq);
}
```

Attention à l'inclusion de ***math.h*** *pour utiliser la fonction* ***sin****.*

Serial *est utilisé pour la communcation*

Serial.begin *initialises la communication à 9600 bauds (bits per second)*

Serial.println *écrive des données sur la porte sérial*

On peut monitorer la serial à travers l'IDE:

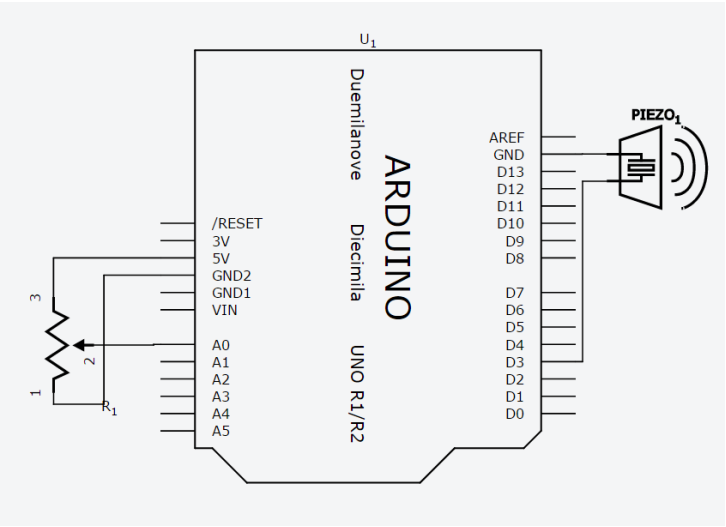
- ✓ Tools->Serial Monitor
- ✓ Tools->Serial Plotter

tone *génère un ton à une fréquence spécifié. Une durée peut être spécifié, autrement la génération continue jusqu'à un appelle à* **notone**()

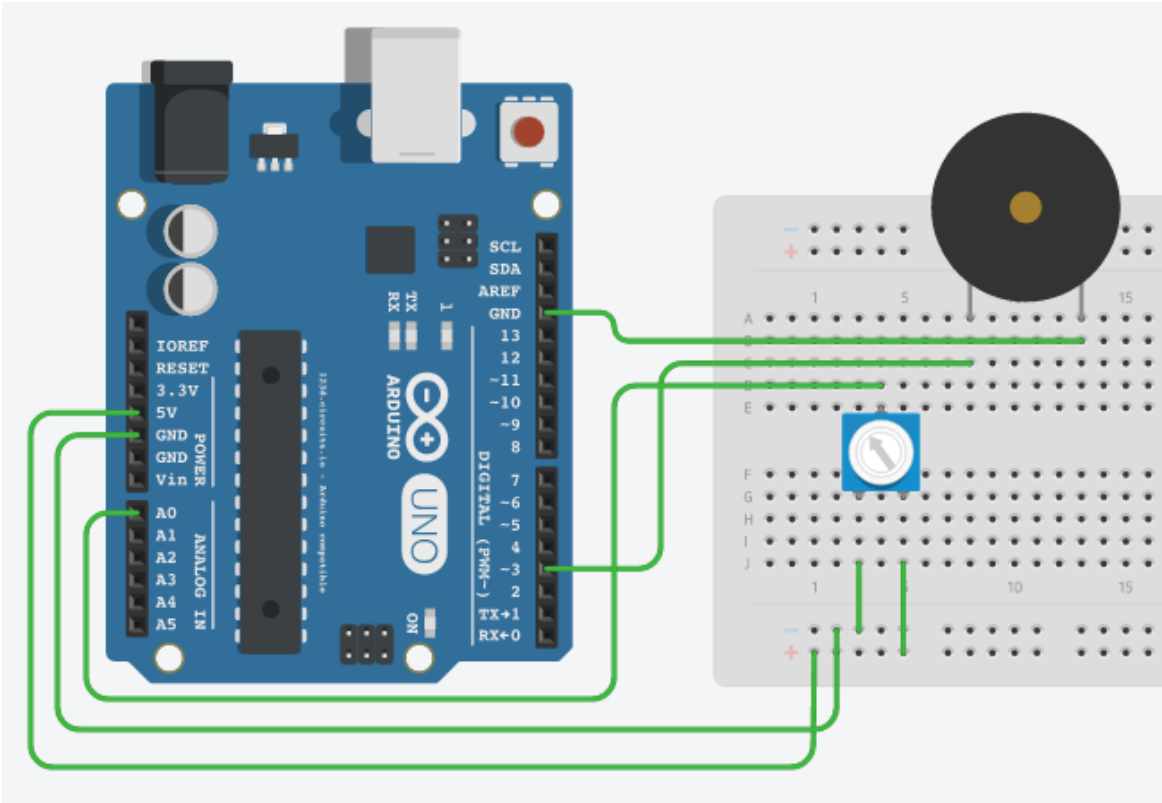
Entrée analogique

- Choisir la fréquence du tons émis

Schéma



Connections



Entrée analogique

- Choisir la fréquence du tons émis

```
// include mathematical functions
#include <math.h>

// Pin 3 has a piezo
int piezo = 3;
// Pin A0 has a potentiometer
int potentiometer = A0;

// the setup routine runs once when you press reset:
void setup() {
  // initalize serial for communication
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // calculate tone value
  int val = analogRead(potentiometer);
  int freq =  map(val, 0, 1023, 0, 1000);
  Serial.println(freq);
  tone(piezo, freq);
}
```

Attention, les entrées analogiques sont seulement les pins marqués comme ***Ax***

analogRead *lis l'état du pin*

Entrée/Sortie analogique

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

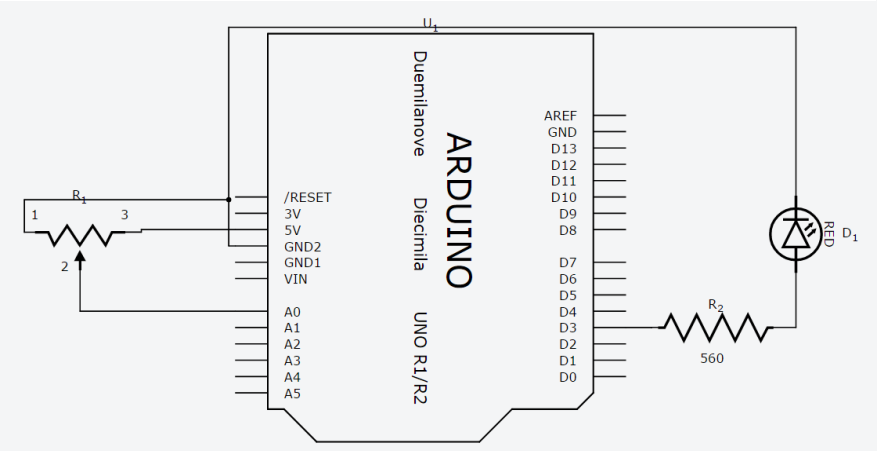
- **analogWrite**() *écrive une valeur sur un pin*

- **Exercice**: on peut développer un système similaire pour contrôler l'intensité de la luminosité d'un LED.
  - *Suggestion*: LED + résistance à la place de l'hautparleur

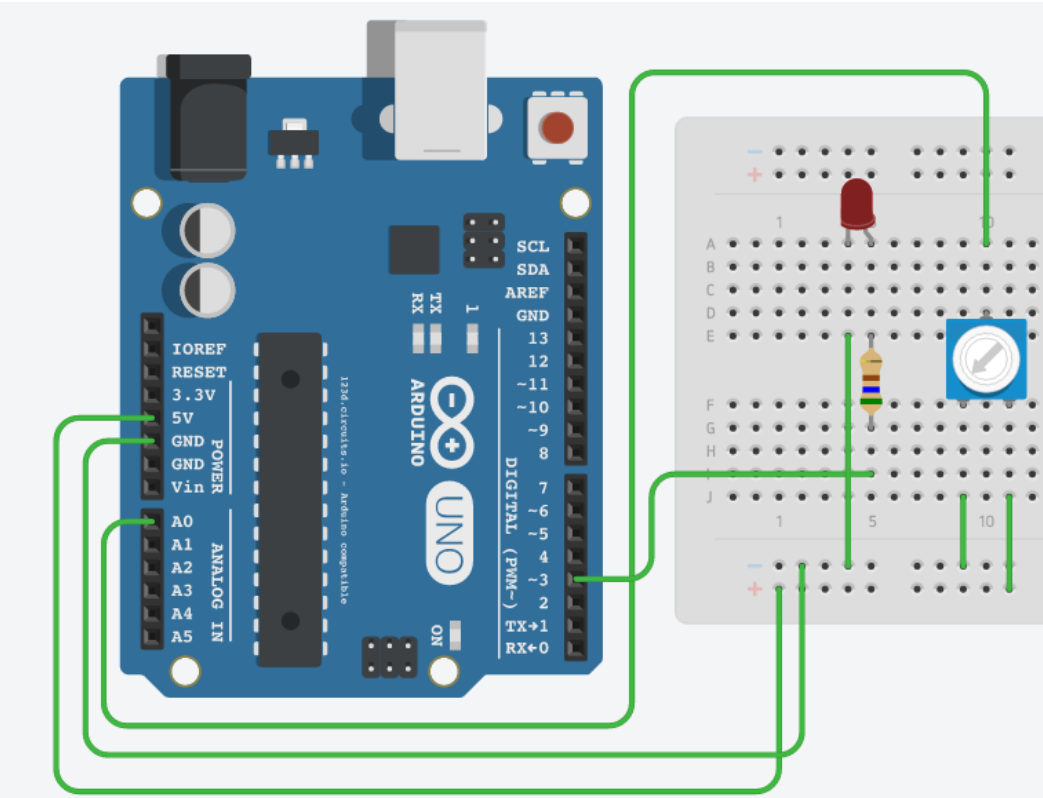
Entrée/Sortie analogique

- **Exercice**: contrôler l'intensité de la luminosité d'un LED.

Schéma



Connections



Entrée/Sortie analogique

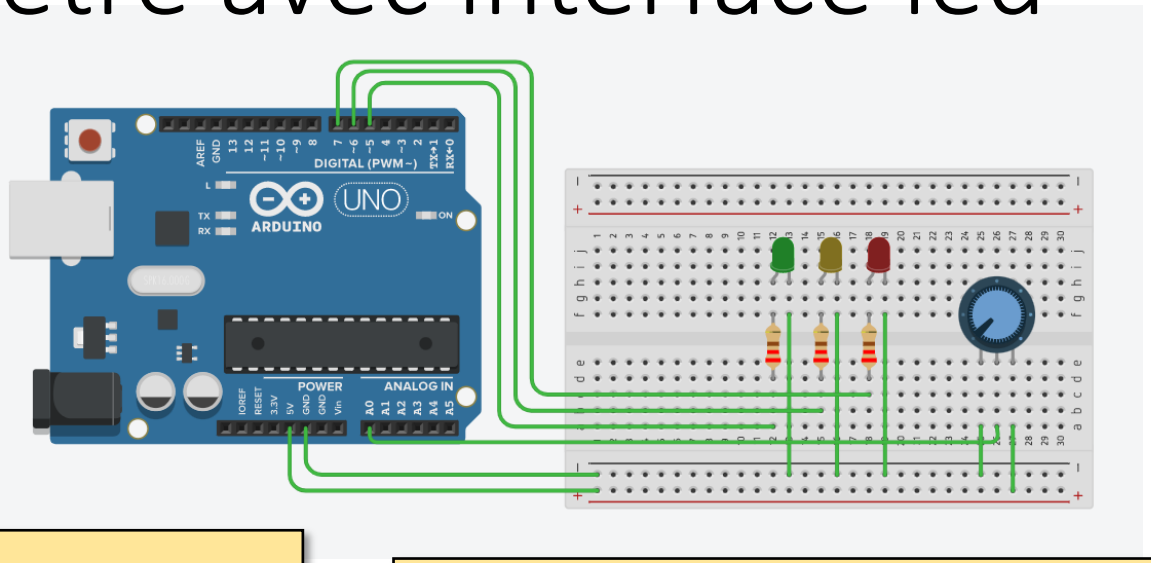
- **Exercice**: contrôler l'intensité de la luminosité d'un LED.

```
// include mathematical functions
#include <math.h>

// Pin 3 has a led
int led = 3;
// Pin A0 has a potentiometer
int potentiometer = A0;

// the setup routine runs once when you press reset:
void setup() {
  // initalize serial for communication
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // retrieve led intensity value
  int val = analogRead(potentiometer);
  val =  map(val, 0, 1023, 0, 255);
  Serial.println(val);
  analogWrite(led, val);
}
```

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Des exemples…

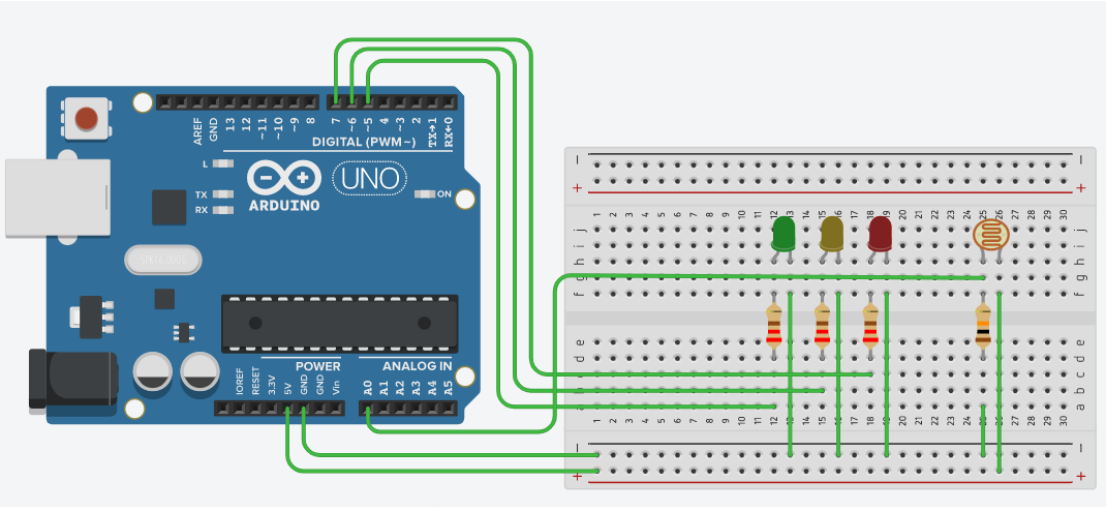
Potentiomètre avec interface led




```
int P1 = A0;
int L1 = 5;
int L2 = 6;
int L3 = 7;

void setup()
{
  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(L3, OUTPUT);

  Serial.begin(9600);

  digitalWrite(L1, HIGH);
  digitalWrite(L2, LOW);
  digitalWrite(L3, LOW);
}
```

```
void loop()
{
  int val = analogRead(P1);
  val = map(val, 0, 1023, 0, 100);
  Serial.println(val);

  if ( val > 33)
    digitalWrite(L2, HIGH);
  else
    digitalWrite(L2, LOW);

  if( val > 66)
    digitalWrite(L3, HIGH);
  else
    digitalWrite(L3, LOW);
}
```

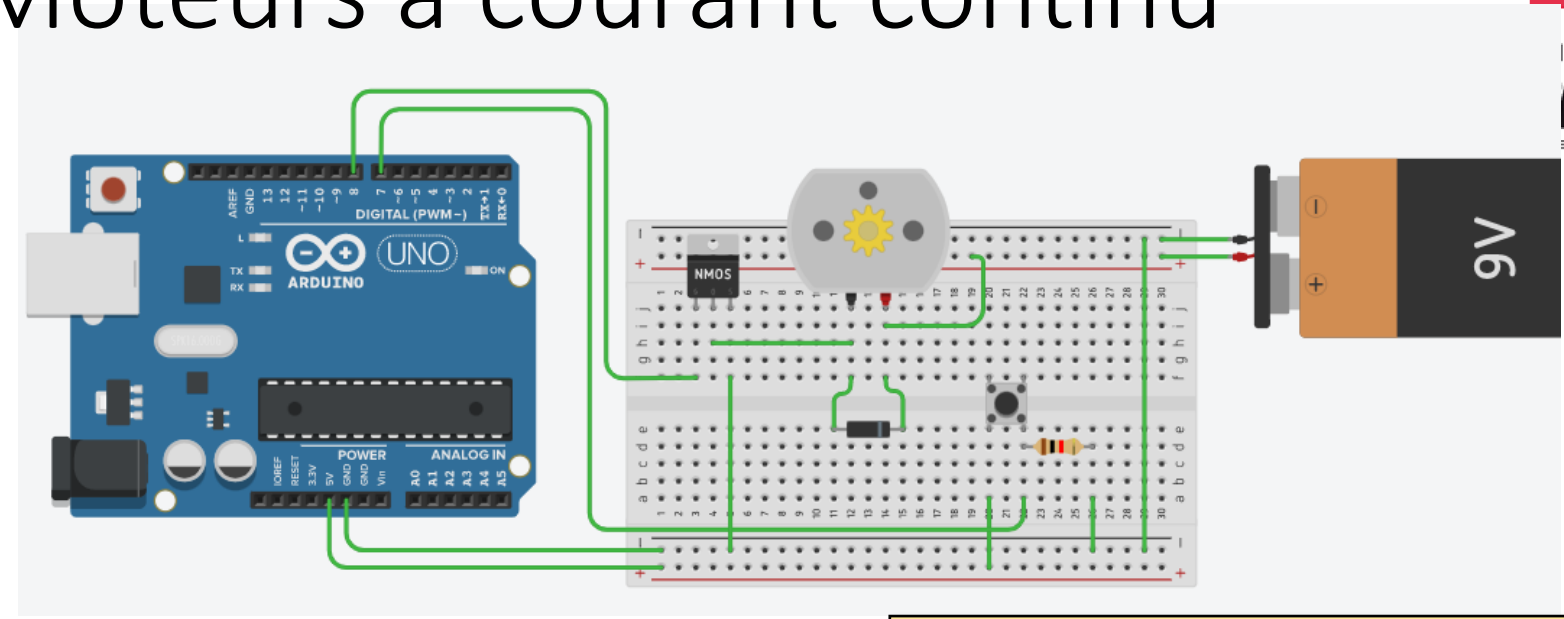
Luminosité avec interface led




```
int P1 = A0;
int L1 = 5;
int L2 = 6;
int L3 = 7;

void setup()
{
  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(L3, OUTPUT);

  Serial.begin(9600);

  digitalWrite(L1, HIGH);
  digitalWrite(L2, LOW);
  digitalWrite(L3, LOW);
}
```

```
void loop()
{
  int val = analogRead(P1);
  val = map(val, 50, 1000, 0, 100);

  Serial.println(val);

  if ( val > 33)
    digitalWrite(L2, HIGH);
  else
    digitalWrite(L2, LOW);

  if( val > 66)
    digitalWrite(L3, HIGH);
  else
    digitalWrite(L3, LOW);
}
```

Moteurs à courant continu

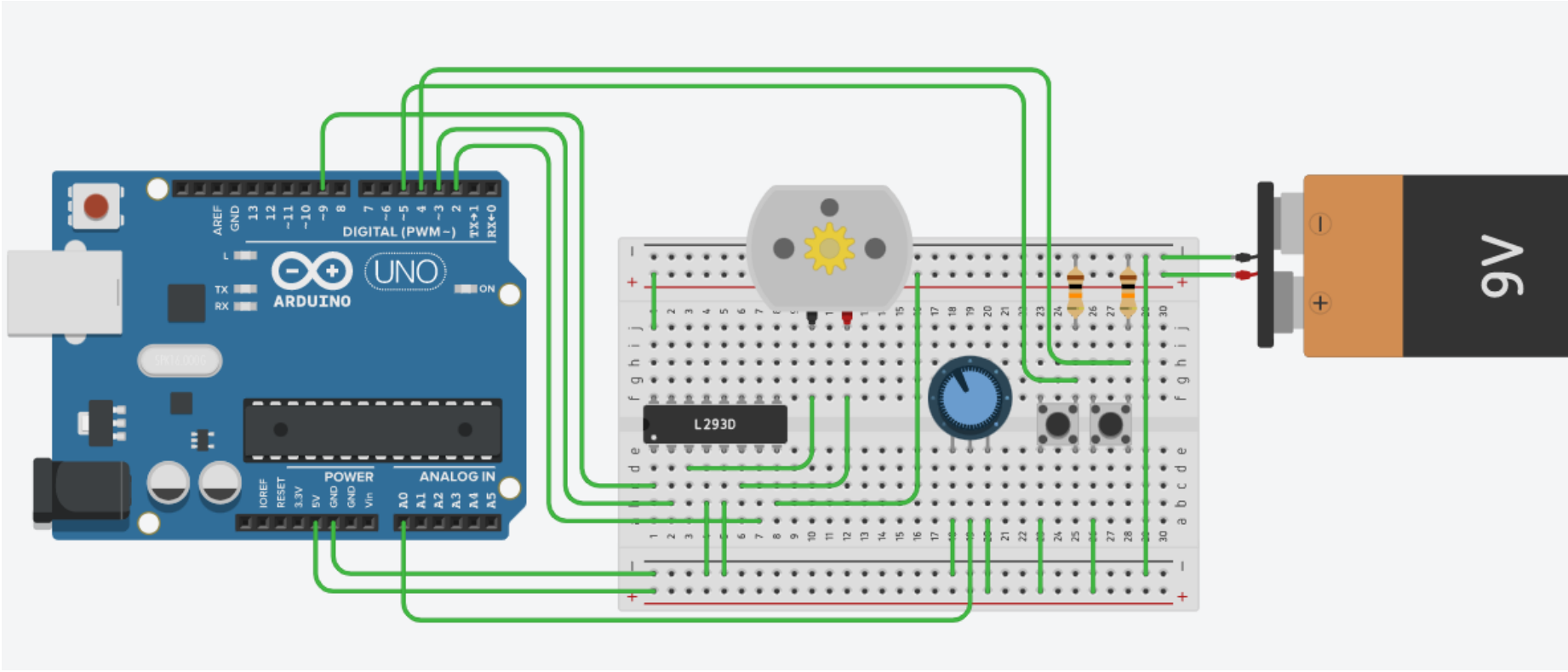


Transistor Mosfet *N : tripôle (Gate-Drain-Source)*
Pour connecter la batterie au moteur

```
int Motor1 = 8;
int Button1 = 7;

bool motorState = false;

void setup()
{
  pinMode( Button1, INPUT);
  pinMode( Motor1, OUTPUT);
}
```

```
void loop()
{
  int val = digitalRead(Button1);

  if(val == HIGH) {
    motorState = !motorState;

    if (motorState)
      digitalWrite(Motor1, HIGH);
    else
      digitalWrite(Motor1, LOW);
    delay(1000);
  }
}
```

Moteurs à courant continu



Circuit intégré L293D *: Pont en H qui permet de contrôler la polarité d'un dipôle:*

- Inverser le sens de rotation du moteur en inversant le courant aux bornes du moteur;

- Varier la vitesse du moteur en modulant la tension à ses bornes.

Moteurs à courant continu

```
int button1pin = 5;
int button2pin = 4;
int potentiometer1pin = A0;
int enable1pin = 9;
int ctrl1pin = 3;
int ctrl2pin = 2;

bool motorState = false;
bool motorClockwise = true;
int motorSpeed = 0;

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting..");

  pinMode( button1pin, INPUT);
  pinMode( button2pin, INPUT);
  pinMode( enable1pin, OUTPUT);
  pinMode( ctrl1pin, OUTPUT);
  pinMode( ctrl2pin, OUTPUT);

  analogWrite(enable1pin, 0);
}
```

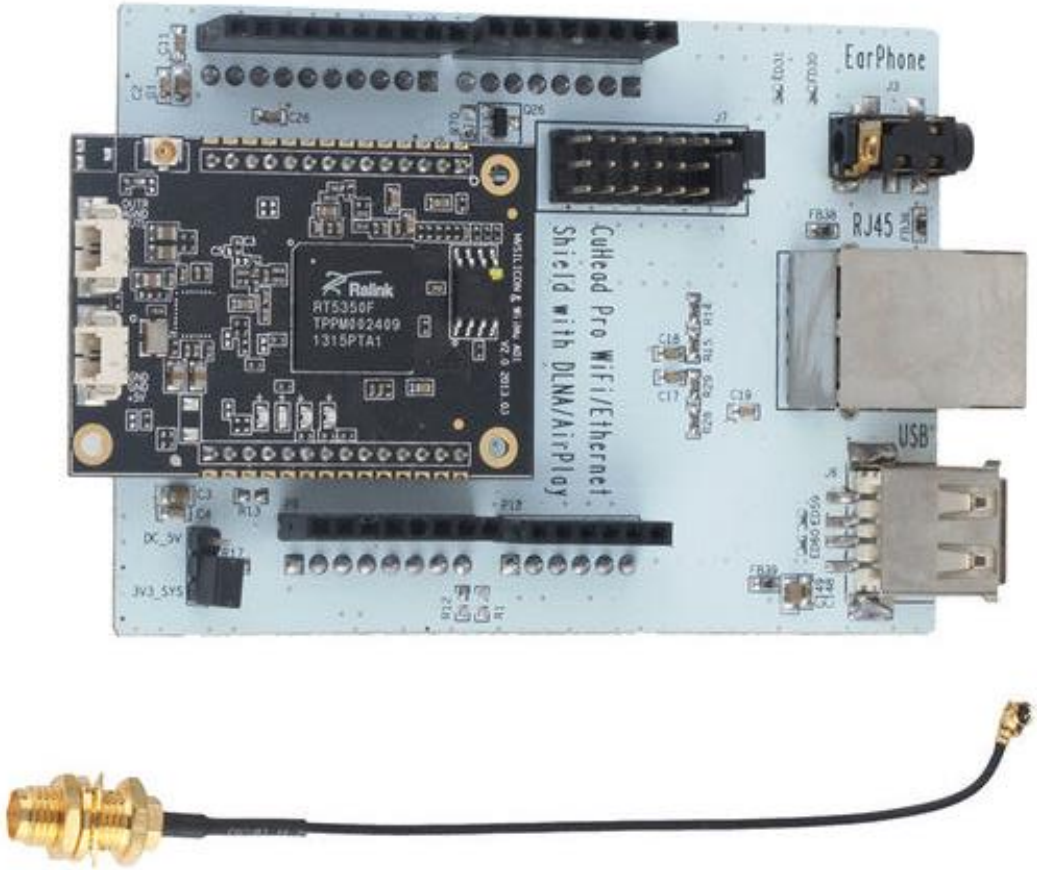
```
void loop() {
  int button1val = digitalRead(button1pin);
  if(button1val == HIGH) {
    motorState = !motorState;
    delay(100);
  }
  Serial.print(motorState);
  Serial.print("\t");

  int button2val = digitalRead(button2pin);
  if(button2val == HIGH) {
    motorClockwise = !motorClockwise;
    delay(100);
  }
  Serial.print(motorClockwise);
  Serial.print("\t");

  motorSpeed = analogRead(potentiometer1pin);
  Serial.print(motorSpeed);
  Serial.println();

  if(motorState) {
    analogWrite(enable1pin, motorSpeed);

    if(motorClockwise) {
      digitalWrite(ctrl1pin, LOW);
      digitalWrite(ctrl2pin, HIGH);
    } else {
      digitalWrite(ctrl1pin, HIGH);
      digitalWrite(ctrl2pin, LOW);
    }
  } else
    analogWrite(enable1pin, 0);
}
```

Shields

- Cartes d’expansion de Arduino

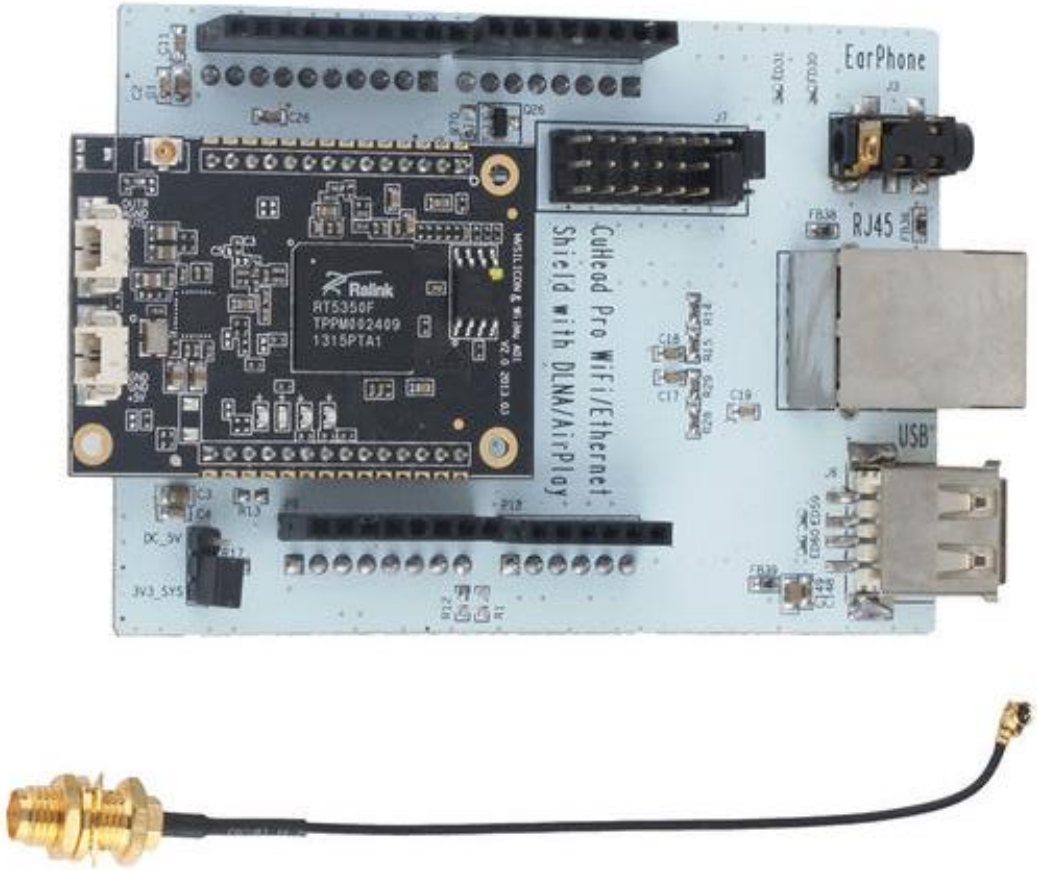
Serveur TCP

- On utilisera le shield CuHead Pro WiFi/Ethernet



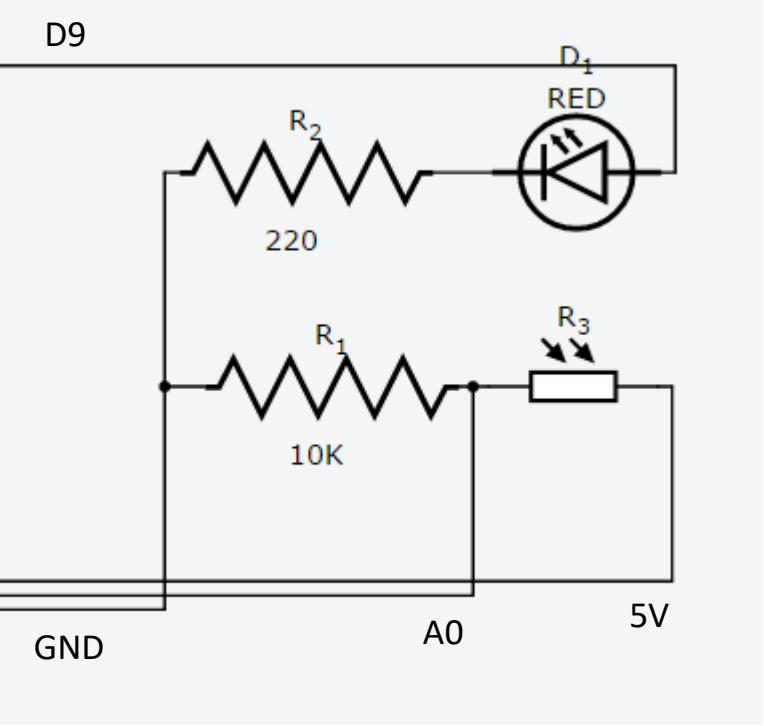
Serveur TCP

- On utilisera le shield CuHead Pro WiFi/Ethernet



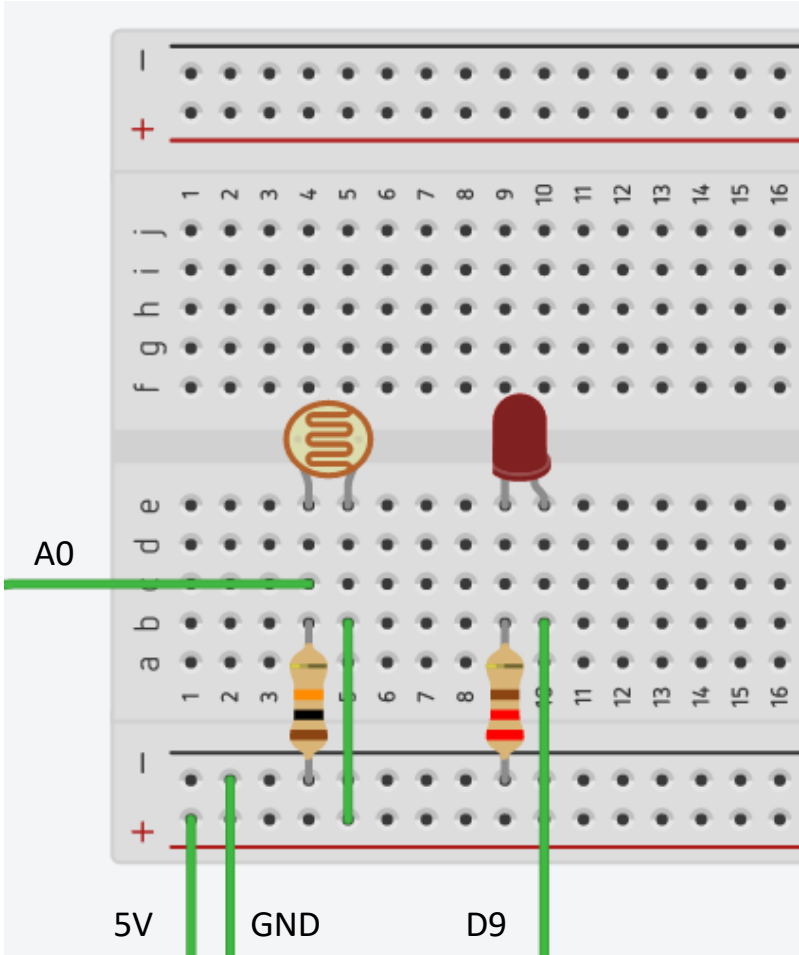
Serveur TCP

- Le shield est connectée à la carte Arduino
- Le shield a besoin d'une configuration de la réseau: http://*10.10.10.254*



Schéma

Connections



Serveur TCP

- Transmission et réception à travers les pins 6 et 7

```
#include <SoftwareSerial.h>

// RX, TX from and to TCP-IP (10.10.10.254:8899)
SoftwareSerial tcpipSerial(6, 7);

int in_pin = A0;
int out_pin = 9;

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

  tcpipSerial.begin(9600);
  tcpipSerial.setTimeout(5000);

  Serial.println("Server Ready");
}

void loop() {
  String str = tcpipSerial.readStringUntil('\n');
  Serial.println(str);
```

```
 switch(str[0])
  {
    case 'g':
    {
      int val = analogRead(A0);
      String response = "Ok " + String(val) + "\n";
      Serial.println(response);
      tcpipSerial.write(response.c_str());
      break;
    }
    case 's':
    {
      int val = str.substring(1).toInt();
      analogWrite(9, val);
      String response = "Ok " + String(val) + "\n";
      Serial.println(response);
      tcpipSerial.write(response.c_str());
      break;
    }
    default: ;
  }
}
```

Serveur TCP

```
import java.io.*;
import java.net.*;

public class MyTCPClient
{
  public static void main(String[] args) throws Exception
  {
    String request, response;

    Socket clientSocket = new Socket("192.168.1.98", 8899);
    DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
    BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

    request = "g\n";
    System.out.println(request);
    outToServer.writeBytes(request);
    response = inFromServer.readLine();
    System.out.println(response);

    request = "s255\n";
    System.out.println(request);
    outToServer.writeBytes(request);
    response = inFromServer.readLine();
    System.out.println(response);

    clientSocket.close();
  }
}
```

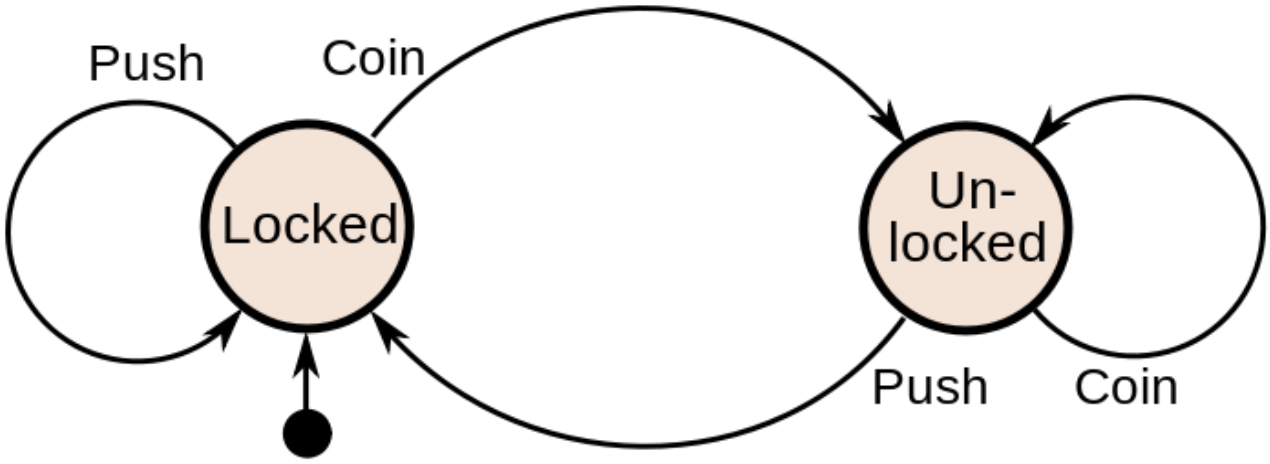
- Client Java de test

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Perception-Action

Perception-Action

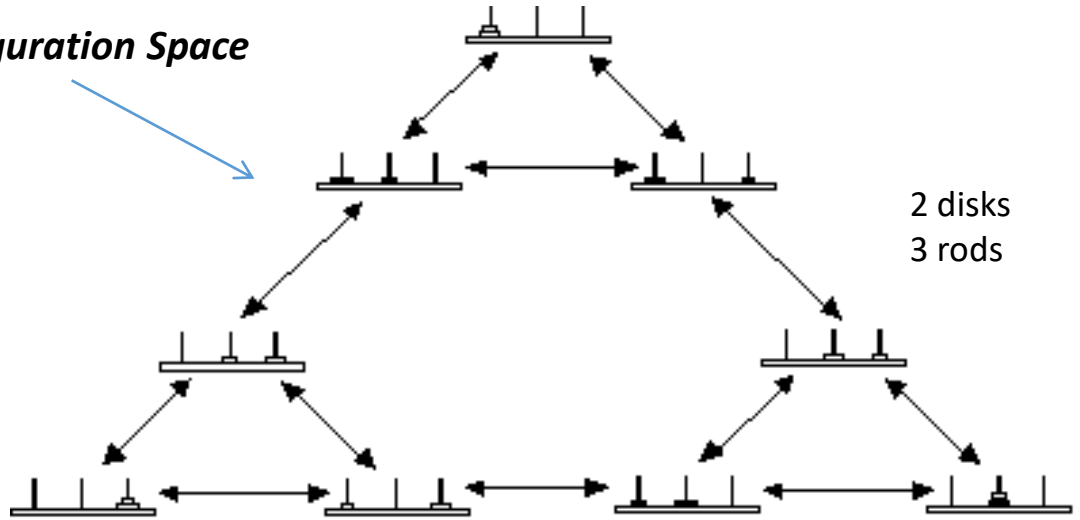
- Automate fini (Finite State Machine)
  - Description des comportements du système à travers un nombre finit d'états
    - *État* : l'état du système en attente de l'exécution d'une transition (nœud)
    - *Transition* : l'ensemble d'actions à exécuter quand une condition est vérifiée (flèche)



Perception-Action

- Espace des configurations
  - Tous les possibles configurations de l'univers

- Exemple : Tour de Hanoi
  - Planification du mouvement de la tour d'une barre à une autre
    - Un disque à la fois
    - Le disque peu bouger seulement si au but d'une pile
    - Le disque ne peut pas être placé sur un disque plus petit



UNIVERSITÉ PARIS 8
VINCENNES-SAINT-DENIS

Merci!