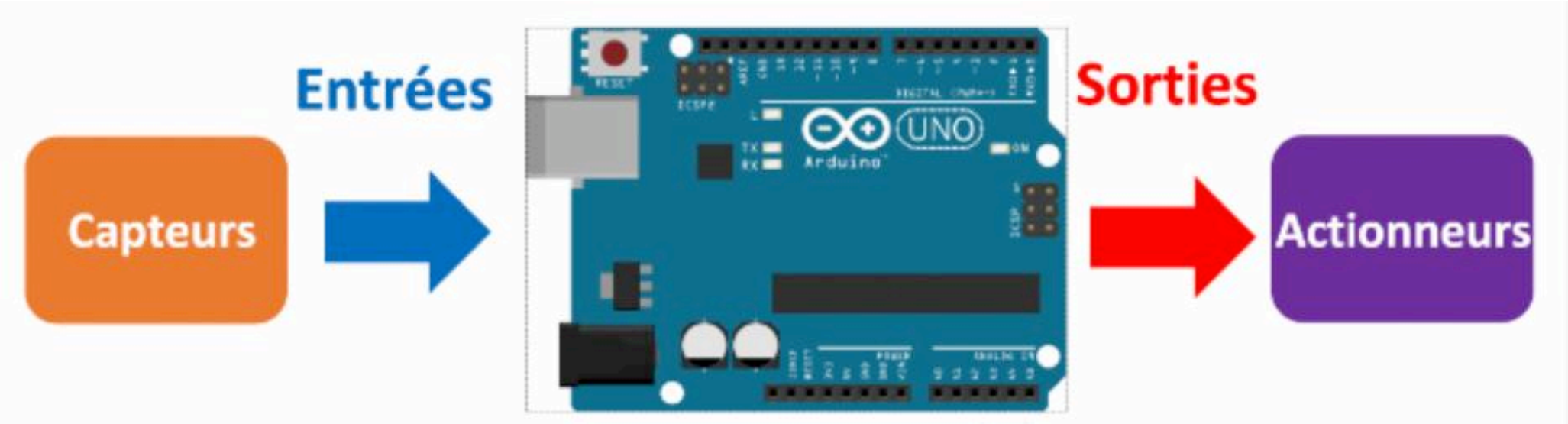
# Arduino UNO


UNO
DIGITAL PWM~
POWER
ANALOG IN
ICSP
RX TX 5V GND
SCL SDA 5V GND
3.3V3.3VGNDGND

- 32 ko de mémoire pour stocker les programmes
- 2 ko de RAM, une EEPROM 1 ko pour stocker… 2 ou 3 trucs, typiquement des paramètres
- Et tout ce petit monde est cadencé à la fréquence totalement dingo de 16 MHz !

Un micro-contrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs. Une carte Arduino est donc une interface programmable.



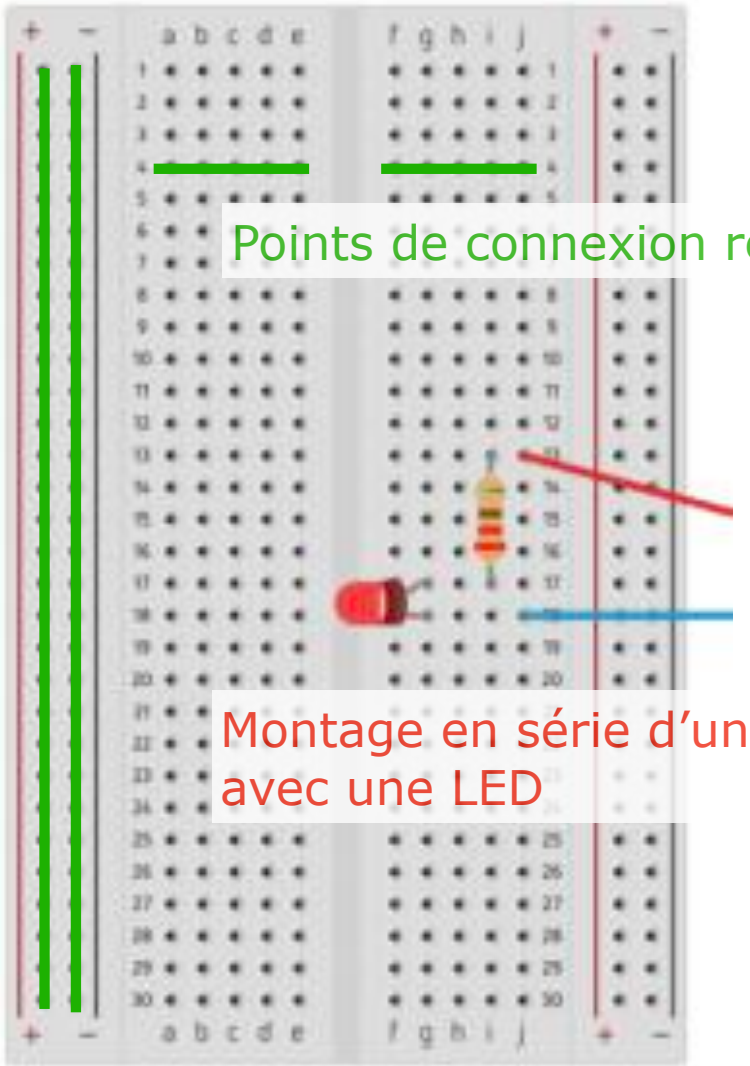
# Ca sert à quoi ?

Une carte comme celle-ci peut gérer

- des accéléromètres,
- des capteurs de température,
- des capteurs de pression atmosphérique,
- Des capteurs de luminosité, etc….
- des moteurs,
- des électroaimants,
- des LED,
- des éléments chauffants,
- des haut-parleurs,
- des valves hydrauliques et pneumatiques,
- des ventilateurs,
- des pompes,
- des systèmes d'affichage
- des grille-pains ! etc…

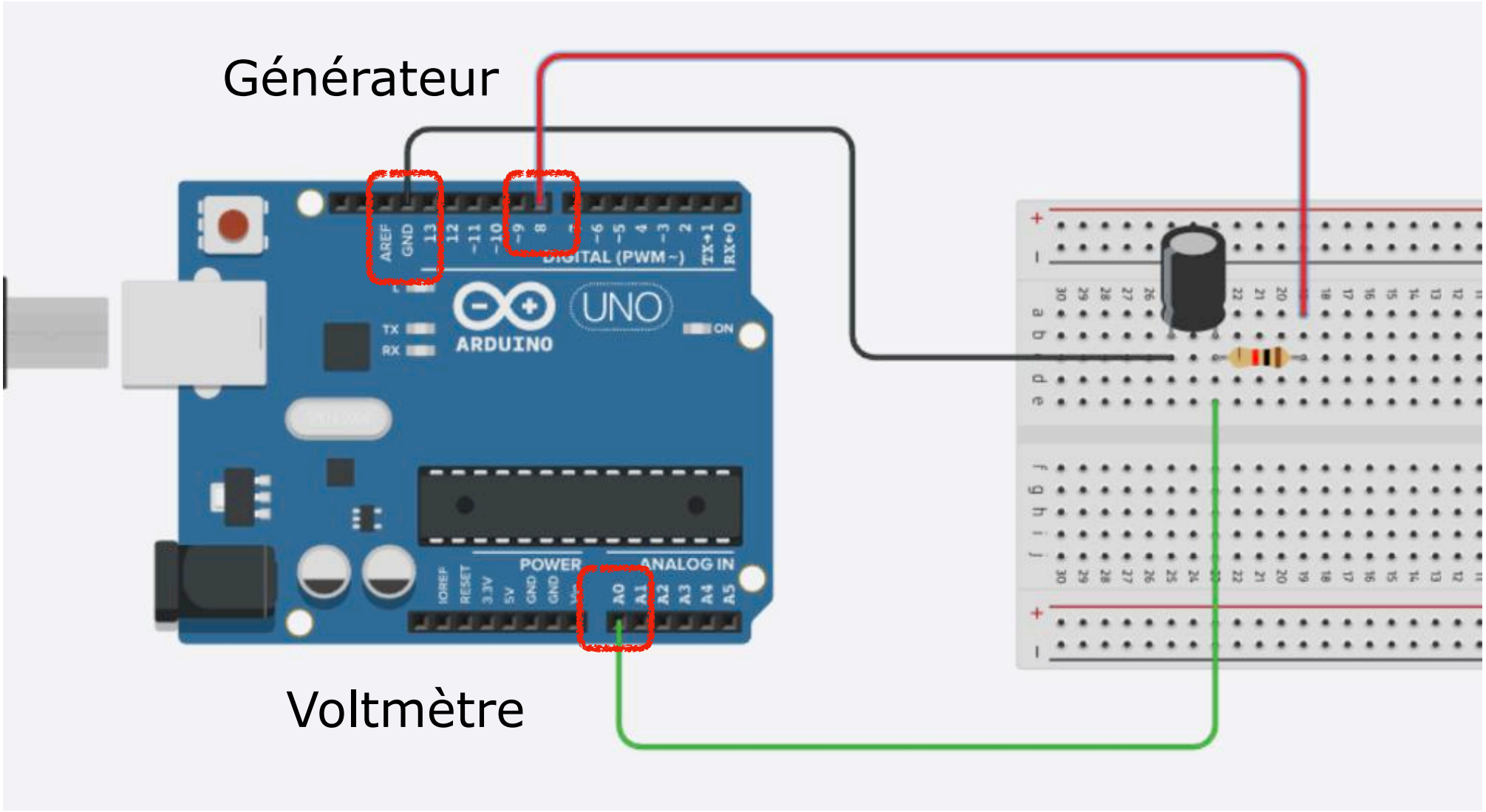
Au final, ça sert à fabriquer des robots qui interagissent avec leur environnement

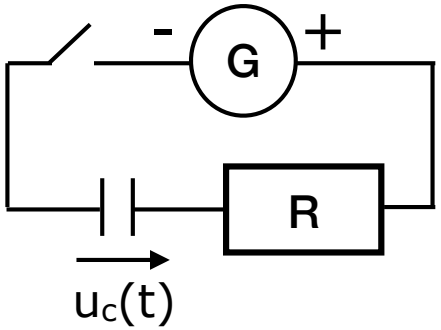
# Carte électronique de prototypage (breadboard)

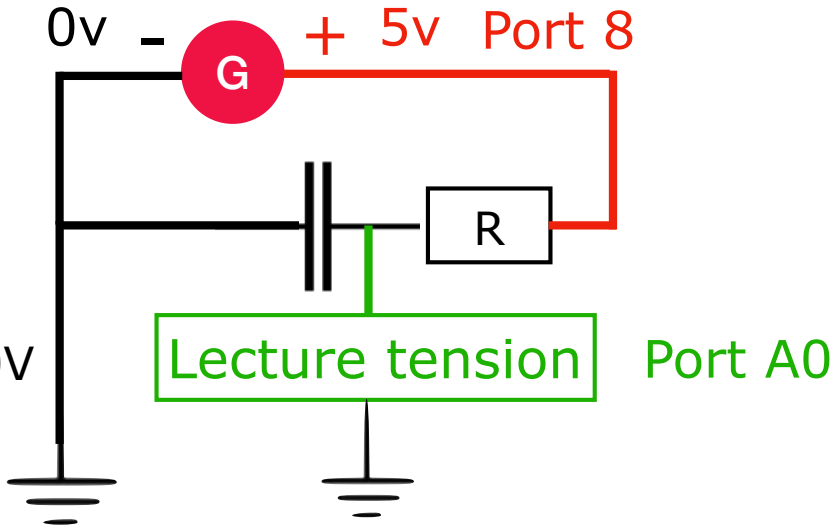


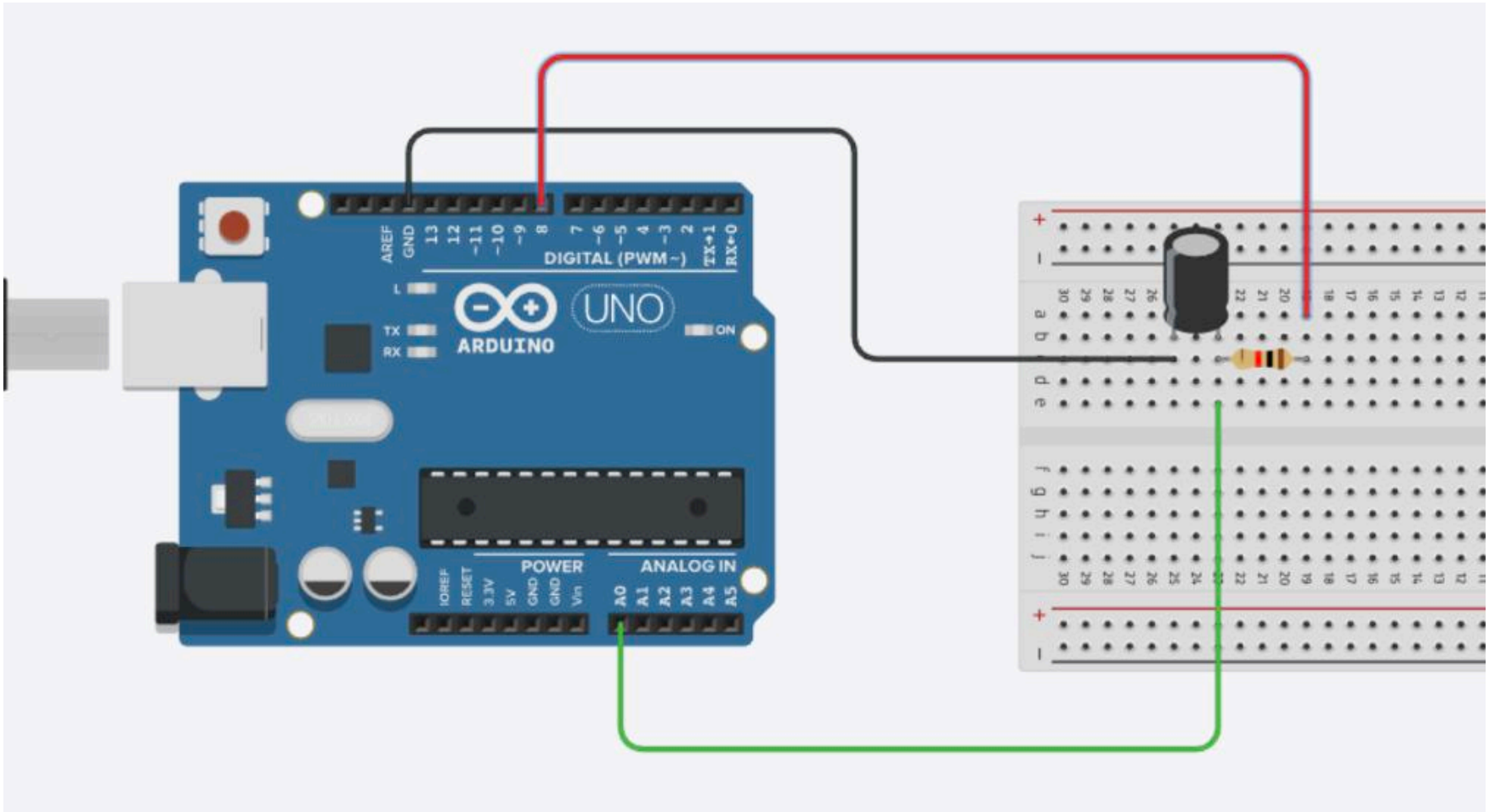
Points de connexion reliés entre eux

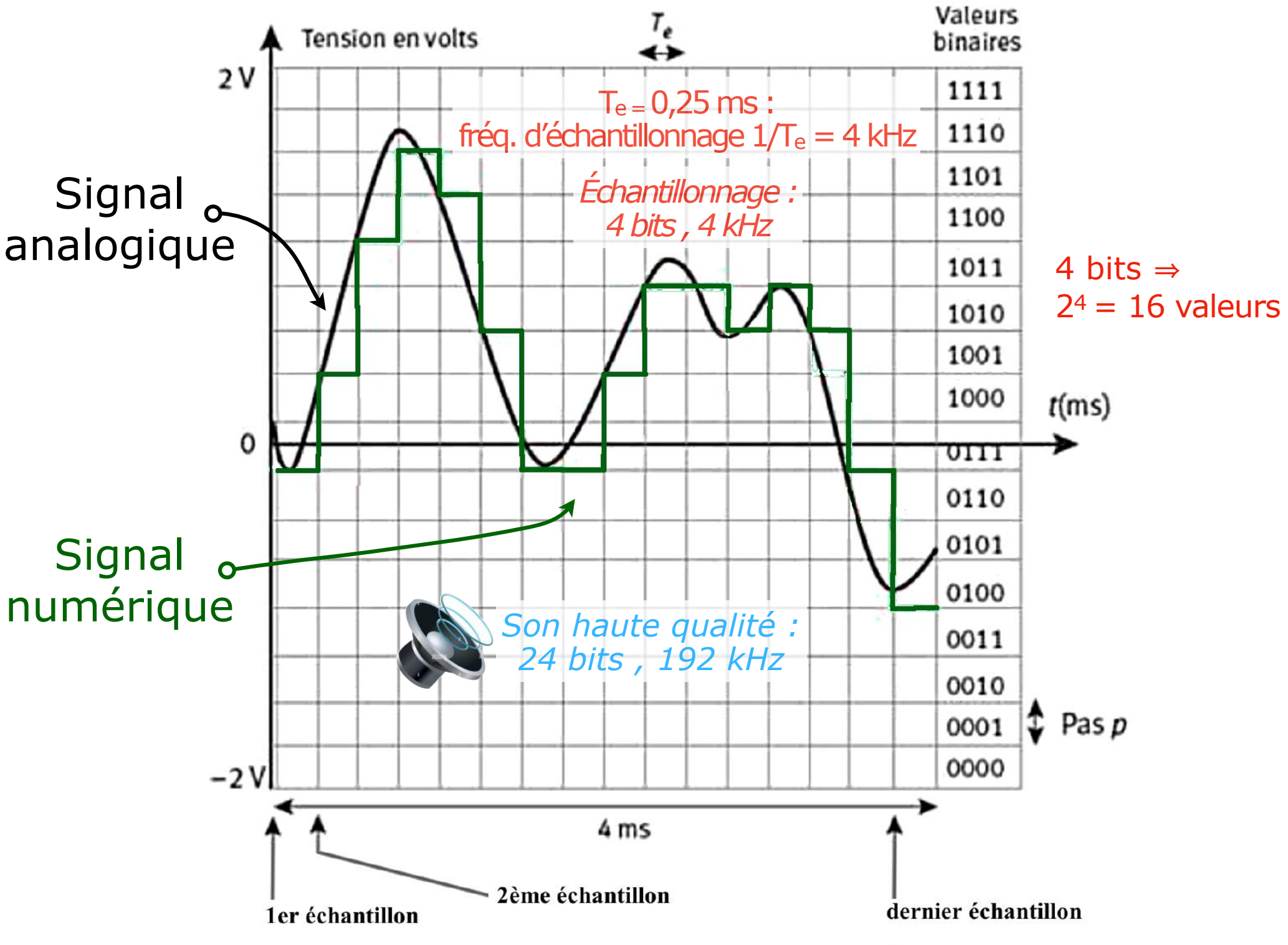
Montage en série d'une résistance avec une LED

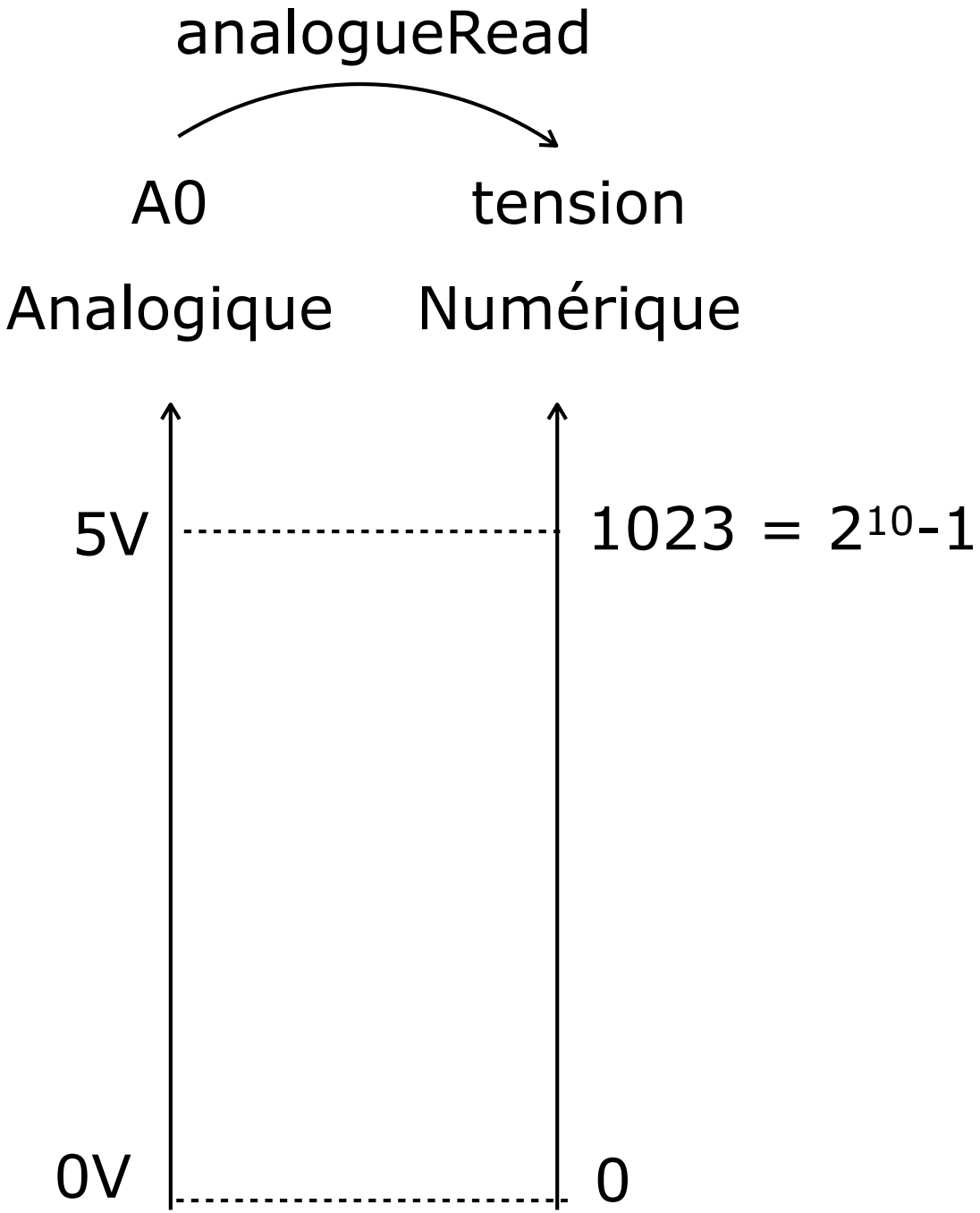









Arduino : codage sur 10 bits

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

tension = analogRead(A0)

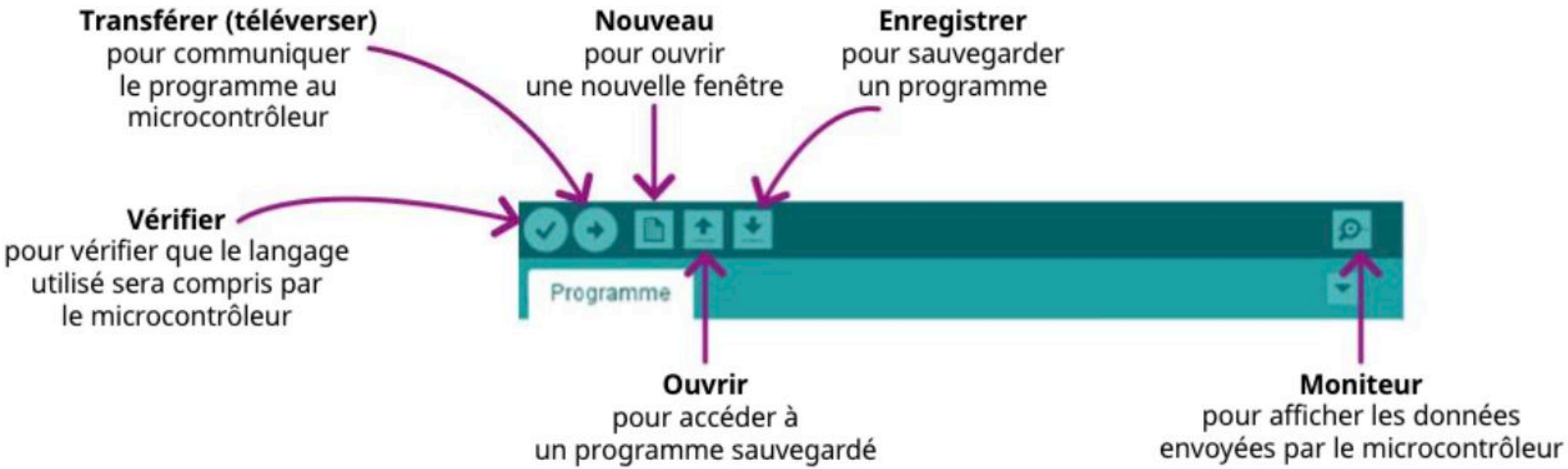
valeur numérique — valeur analogique

Quelle est la tension analogique lue sur A0 (fraction de E=5V) ?

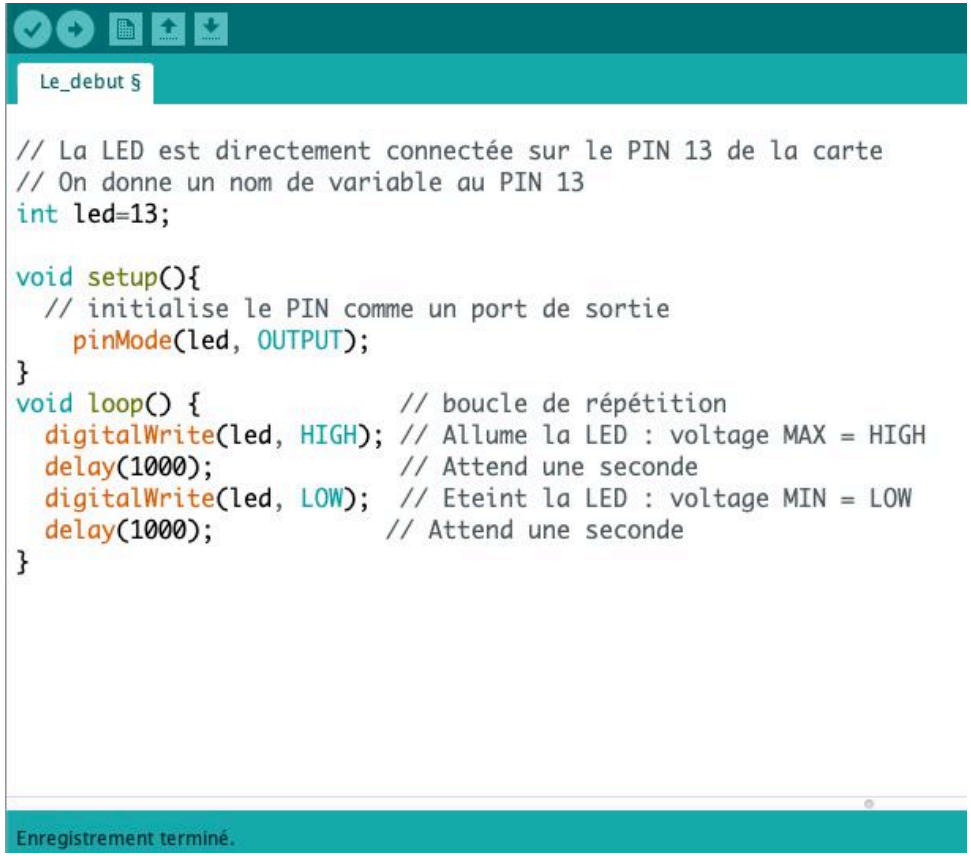
| Arduino | ⟷ | Tension lue |
|---|---|---|
| 1023 | ..... | $5\ V$ |
| 1 | ..... | $\frac{5\ V}{1023}$ |
| x *tension* | ..... | $tension \times \frac{5\ V}{1023}$ |

La tension analogique lue sur A0 est : $tension \times \frac{5\ V}{1023}$

# Environnement de développement intégré Arduino



# Programme Arduino : setup() + loop()




un IDE
(**I**ntegrated **D**evelopment **E**nvironment)
Environnement de développement intégré

# Charge

```
  Ce sketch permet de réaliser la charge d'un condensateur.
  On décharge dans un premier temps de le condensateur pour que la tension à ses bornes soit presque nulle
  On récupère les données depuis le port série pour les afficher dans un fichier csv.
*/

long previousMillis = 0;
long interval = 10;                                          // durée de millisecondes entre chaque mesure
unsigned long topdepart = 0;                                 //on initialise un variable topdepart à 0 ;
float tension;                                               //variable tension qui sera lue


void setup()
{
  pinMode(8, OUTPUT);                                        // alimentation du condensateur
  Serial.begin(9600);                                        //vitesse de communication avec arduino
  tension = analogRead(A0);                                  // on lit la tension sur la sortie analogique A0

  if (tension > 3) {
    Serial.println("décharge préalable du condensateur... patienter");    // on indique qu'on décharge au préalable le condensateur.
    while (tension > 3) {
      digitalWrite(8, LOW);
      delay(500);
      tension = analogRead(A0);                              // on lit la tension sur la sortie analogique A0 toutes les 500 ms
    }
  }
  Serial.println();
  Serial.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
  Serial.println();
  Serial.println("condensateur déchargé");                    // on indique que le condensateur est déchargé
  Serial.println("la charge va démarrer, l'arret sera automatique quand le condensateur sera chargé");
  Serial.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
  delay(2000);
  topdepart = millis() ;                                     // le top départ et initialié sur la durée millis() du timer interne de l'arduino
  digitalWrite(8, HIGH); // on met en marche l'alimentation  // on alimente le circuit pour charger le condensateur
}

void loop()
{
  unsigned long currentMillis = millis();                    // on crée une variable currentMillis qui correspond à millis() dans cette boucle

  tension = analogRead(A0);
  if (currentMillis - previousMillis >= interval) {          // lorsque la durée courante - la durée précédente est supérieure à 10 ms (interval)
    previousMillis = currentMillis;                          // l'instant previousMillis devient CurrentMillis (réinitialisation du timer)

    if (tension < 1019) {                                    // si cette tension est inférieur à 1018 sur 1023 (non complètement chargé
      float tension_lue = tension * 5.0 / 1023.0;            // tension analogique calculée
      //Serial.print("\t");
      Serial.print(millis() - topdepart);                    // on écrit la durée millis-topdepart (première valeur à 0)
      Serial.print("\t");                                    // on sépare la donnée avec une tabulation (permettra de faire un csv par copier coller)
      Serial.println(tension_lue);                           // on sépare la donnée avec une tabulation (permettra de faire un csv par copier coller)

    }
```

# Décharge

```
  Ce sketch permet de réaliser la décharge d'un condensateur.
  On charge dans un premier temps de le condensateur pour que la tension à ses bornes soit presque nulle
  On récupère les données depuis le port série pour les afficher dans un fichier csv.
*/

long previousMillis = 0;
long interval = 10;                                          // durée de millisecondes entre chaque mesure
unsigned long topdepart = 0;                                 //on initialise un variable topdepart à 0 ;
float tension;                                               //variable tension qui sera lue


void setup()
{
  pinMode(8, OUTPUT);                                        // alimentation du condensateur
  Serial.begin(9600);                                        //vitesse de communication avec arduino
  tension = analogRead(A0);                                  // on lit la tension sur la sortie analogique A0

  if (tension <1019) {
    Serial.println("charge préalable du condensateur... patienter");    // on indique qu'on charge au préalable le condensateur.
    while (tension < 1019) {
      digitalWrite(8, HIGH);
      delay(500);
      tension = analogRead(A0);                              // on lit la tension sur la sortie analogique A0 toutes les 500 ms
    }
  }
  Serial.println();
  Serial.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
  Serial.println();
  Serial.println("condensateur chargé");                       // on indique que le condensateur est chargé
  Serial.println("la décharge va démarrer, l'arret sera automatique quand le condensateur sera entièrement");
  Serial.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");
  delay(2000);
  topdepart = millis() ;                                     // le top départ et initialié sur la durée millis() du timer interne de l'arduino
  digitalWrite(8, LOW);                                      // on coupe l'alimentation du circuit pour décharger le condensateur
}

void loop()
{
  unsigned long currentMillis = millis();                    // on crée une variable currentMillis qui correspond à millis() dans cette boucle

  tension = analogRead(A0);
  if (currentMillis - previousMillis >= interval) {          // lorsque la durée courante - la durée précédente est supérieure à 10 ms (interval)
    previousMillis = currentMillis;                          // l'instant previousMillis devient CurrentMillis (réinitialisation du timer)

    if (tension > 3) {                                      // si cette tension est inférieur à 1018 sur 1023 (non complètement chargé
      float tension_lue = tension * 5.0 / 1023.0;            // tension analogique calculée
      //Serial.print("\t");                                  // on sépare la donnée avec une tabulation (permettra de faire un csv par copier coller)
      Serial.print(millis() - topdepart);                    // on écrit la durée millis-topdepart (première valeur à 0)
      Serial.print("\t");                                    // on sépare la donnée avec une tabulation (permettra de faire un csv par copier coller)
      Serial.println(tension_lue);
    }
```