Compléments à l'article « Construire un colorimètre et évaluer l'incertitude des méthodes de dosage par étalonnage », J. Randon (*L'Act. Chim.*, 2020, *452*, p. 29)

Annexe 1

Colorimètre Arduino : programme structuré autour d'une étape de mesure, d'une étape de traitement du signal et d'une étape d'affichage de l'absorbance

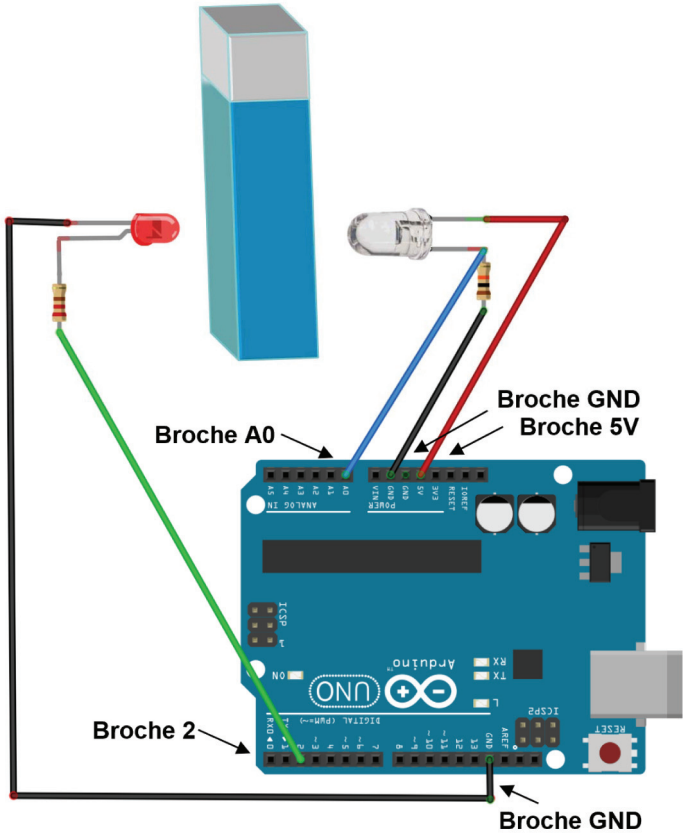
```
int PinLED = 2;                 // Numéro broche Alimentation la LED
int PinPhotoDiode = A0;         // Numéro broche signal photodiode

float Ion;                      // I on
float Ioff;                     // I off
float Imesure;                  // Ion-Ioff pour s'affranchir du bruit de fond
float I0;                       // I blanc
float alpha = 0.05;             // pour calculer la moyenne exponentielle

float A;                        // Absorbance=-log10(I/I0)

int TempsEntreAffichage = 1000;
float TempsAffichageSuivant;

//-----------------------------------------------------------------------------------------
void setup() {
  Serial.begin(9600);                       // Communication avec le moniteur série
  pinMode(PinLED, OUTPUT);                  // Définition broche connectée à la LED
  pinMode(PinPhotoDiode, INPUT);            // Définition broche connectée à la photodiode

  Serial.println("Initialisation : Mesure de I0 pendant 10 secondes");
  Imesure = MesureONOFF();
  do {
    Imesure = alpha * MesureONOFF() + (1 - alpha) * Imesure;
  } while (millis()  < 10000);
  I0 = Imesure;                                              // Stocke la valeur de I0

  TempsAffichageSuivant = millis()+ TempsEntreAffichage;
}

//-----------------------------------------------------------------------------------------
void loop() {
  Imesure= alpha * MesureONOFF() + (1 - alpha) * Imesure;

  if (millis() > TempsAffichageSuivant) {                  // Test pour affichage
    Serial.println(log10(I0/Imesure), 3);                     // Affiche l'absorbance

    if (Serial.available()) {                                 // Teste si un caractère arrive du terminal
       if (String(Serial.readString()) == "0") {                 // Stocke les I0
          Serial.println("Mémorisation de la valeur de I0");
          I0 = Imesure;
          }                                                         // fin de stockageI0
       }                                                         // fin du test arrivée d'un caractère
    TempsAffichageSuivant = TempsAffichageSuivant + TempsEntreAffichage;
    }                                                       // fin test Affichage
}

//-----------------------------------------------------------------------------------------
float MesureONOFF() {
  int Tstab = 10;                           // temps d'attente pour stabilisation de la LED entre état allumé et éteint

  digitalWrite(PinLED, HIGH);   delay(Tstab);    //---------- LED allumée  -------------
  Ion = analogRead(PinPhotoDiode);

  digitalWrite(PinLED, LOW);    delay(Tstab);    //---------- LED éteinte  -------------
  Ioff = analogRead(PinPhotoDiode);

  return Ion - Ioff;
}
```

Le programme proposé utilise une première astuce pour s'affranchir de la lumière parasite arrivant sur le capteur : I est obtenu par la différence entre le signal du capteur lorsque la LED est allumée (Ion) et lorsque la LED est éteinte (Ioff). Cette opération est exécutée en quelques millisecondes via la fonction « MesureONOFF() ». Une telle méthode permet ainsi au colorimètre de fonctionner sans capot recouvrant le système de mesure, et donc dans la lumière ambiante.

D'autre part, un calcul de moyenne peut être mis en œuvre qui permet de diminuer considérablement le bruit associé au signal (le programme réalise un calcul de moyenne mobile exponentielle sur I). Un autre moyennage arithmétique pourrait être introduit lors de la mesure de Ion et Ioff, par exemple en faisant la moyenne du signal sur 50 mesures réalisées successivement après le changement d'état de la LED et le temps de stabilisation.

Ces astuces, non obligatoires pour un colorimètre, peuvent être proposées aux étudiants en fonction de leur avancement dans l'élaboration du colorimètre, permettant une adaptation des activités à la progression des membres du groupe.

Câblage : la résistance reliée à la photodiode (10 kΩ) sera adaptée pour avoir une valeur de Ion dans la gamme de mesure du microcontrôleur (au voisinage de 850 sur la plage 0-1023 pour la mesure de I_0).



Annexe 2

Fichier Excel de l'exemple de la feuille de calcul présentée

Nom du fichier : **Regression linéaire - Eurachem - affichage simple.xls**.