

# Stage laboratoire

*Laboratoire*



Les EEMCP2 de physique-chimie et SVT

Gilles Claudel, Laurent Micoud, Sébastien Steiner

Lycée Lyautey, Casablanca

# Table des matières



<b>I - Les microcontrôleurs</b>	<b>3</b>
1. Capteur de lumière .....	3
1.1. Découverte d'Arduino .....	3
1.2. Capteur de lumière - partie 1 .....	5
1.3. Capteur de lumière - partie 2 .....	6
1.4. Capteur de lumière - partie 3 .....	8
1.5. Simulation avec Tinkercad .....	9
1.6. Exercice : Ce que je dois retenir .....	9
2. Capteur de température .....	11
2.1. Étalonnage d'une thermistance .....	12
2.2. Réalisation d'un dispositif de mesure de température .....	13
<b>II - La sécurité au laboratoire de chimie</b>	<b>15</b>
1. La sécurité au laboratoire - Auto-évaluation .....	15
<b>III - Les incertitudes liées à la mesure</b>	<b>22</b>
1. Déterminer une incertitude de mesure avec GUM MC .....	22
<b>IV - Ateliers à la carte</b>	<b>23</b>
1. Utilisation des tableurs en sciences .....	23
1.1. Créer un graphique avec Excel .....	23
1.2. Modéliser un ensemble de points de mesure .....	23
1.3. Déterminer l'incertitude-type d'une grandeur modélisée à partir d'une régression linéaire .....	23
1.4. Calculer des grandeurs à partir de listes .....	23
<b>Conclusion</b>	<b>24</b>
<b>Contenus annexes</b>	<b>25</b>

# Les microcontrôleurs

I

## 1. Capteur de lumière

### Objectifs

- Utiliser un microcontrôleur Arduino™ en respectant les règles de sécurité
- Réaliser un circuit électrique sur une platine de montage et tester son fonctionnement
- Programmer un microcontrôleur pour faire des mesures et commander des actionneurs

L'utilisation des microcontrôleurs est inscrite aux programmes de lycée

- de physique-chimie : au moins deux TP en seconde, trois TP en spécialité de 1re et autant en terminale
- de sciences de la vie et de la Terre
- d'enseignement scientifique : pour le projet numérique

### 1.1. Découverte d'Arduino

#### Définition : Microcontrôleur

---

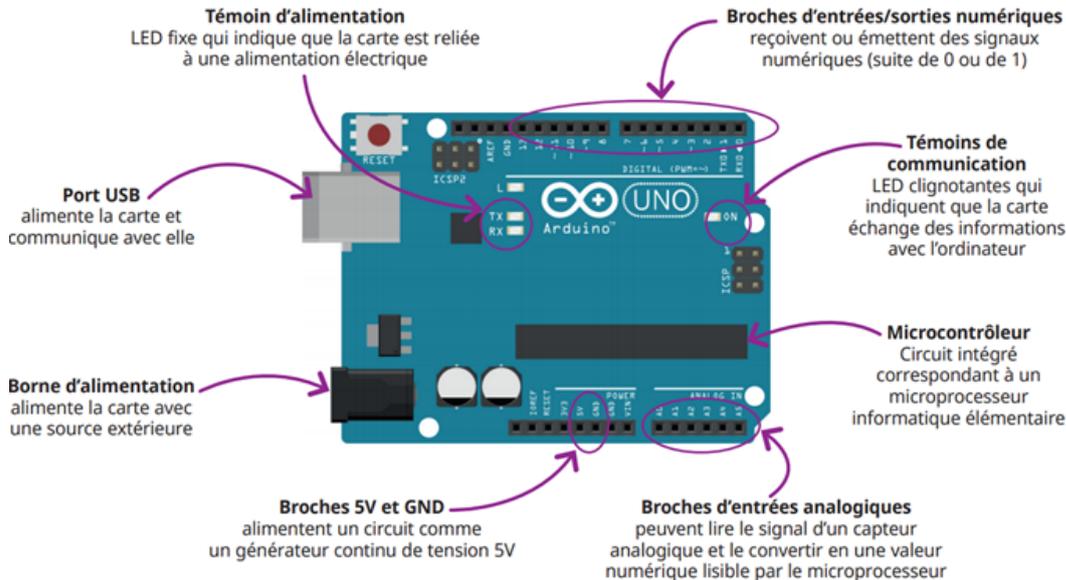
Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties, convertisseurs.

Un microcontrôleur est une carte programmable simple sur laquelle on branche des composants électroniques.

Il possède un grand nombre d'entrées/sorties ce qui permet de brancher beaucoup de composants.

Il est principalement utilisé pour réaliser des interfaces électroniques avec des capteurs et des actionneurs (qui nécessitent parfois un module d'adaptation de puissance).

Une fois que le programme est chargé, le microcontrôleur l'exécute en boucle de manière autonome.



Description de la carte Arduino

**Complément : Caractéristiques techniques de l'Arduino™ Uno**

- Microcontrôleur ATmega328 cadencé à 16 MHz. (Ne pas espérer pour autant réaliser des mesures à cette fréquence).
- 14 entrées/sorties digitales (0 à 13) dont 6 peuvent être utilisées en PWM (Pulse With Modulation avec un rapport cyclique réglable par pas de 1/255). Les broches 0 et 1 servent à la communication série.
- Impédances d'entrées : 100 MΩ.
- CAN (conv. analogique-numérique) de 5 V sur 10 bits donc résolution proche de 5 mV.
- Fréquence d'échantillonnage liée à la vitesse de communication série, soit une fréquence d'échantillonnage de l'ordre de 1,2 kHz pour une vitesse de communication de 115200 baud. En utilisant la vitesse de communication standard de 9600 baud, la fréquence de travail descend à 90 Hz.
- 6 entrées analogiques (peuvent être utilisées en numérique).
- Deux sorties régulées en tension 5 V (500 mA max si alim USB) et 3,3 V (50 mA max).
- Convertisseur série-USB permettant d'alimenter et de communiquer par le port USB.
- Alimentation autre que par de l'USB.
- Possibilité d'utiliser des modules wifi, GPS, ...
- Système de protection auto-réarmable contre les surintensités (> 500 mA) sur le port USB.

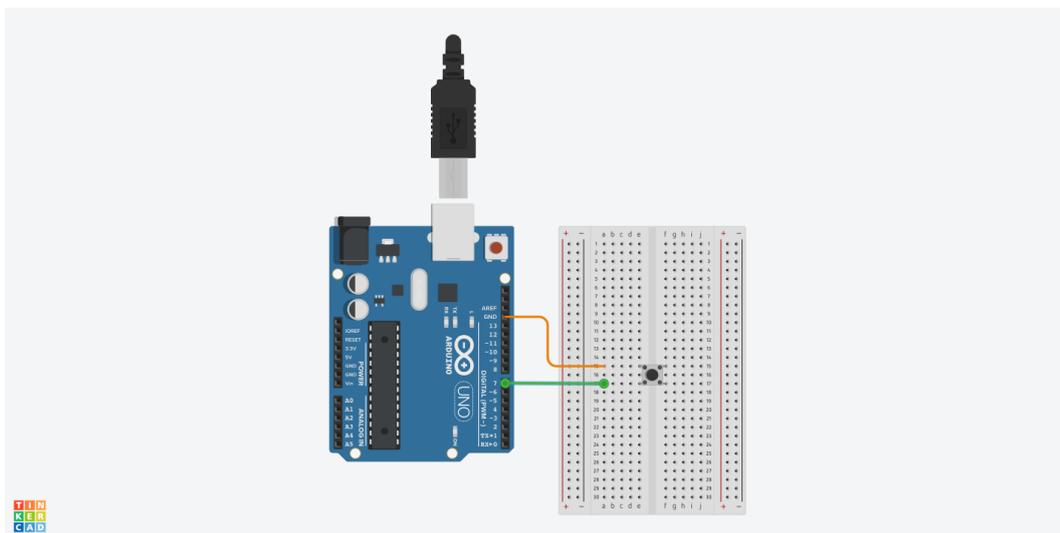
**Complément : Précautions d'usage**

- Alimentation externe possible de 6 V à 20 V mais fortement conseillé de se limiter à l'intervalle 7 V -12 V
- 5 V maximum sur les entrées analogiques.
- Maximum 40 mA par E/S analogique et 200 mA sur la totalité des E/S analogiques.
- Pas de tension négative sur les entrées analogiques, ce qui impose de créer un décalage (offset) pour traiter une tension alternative.

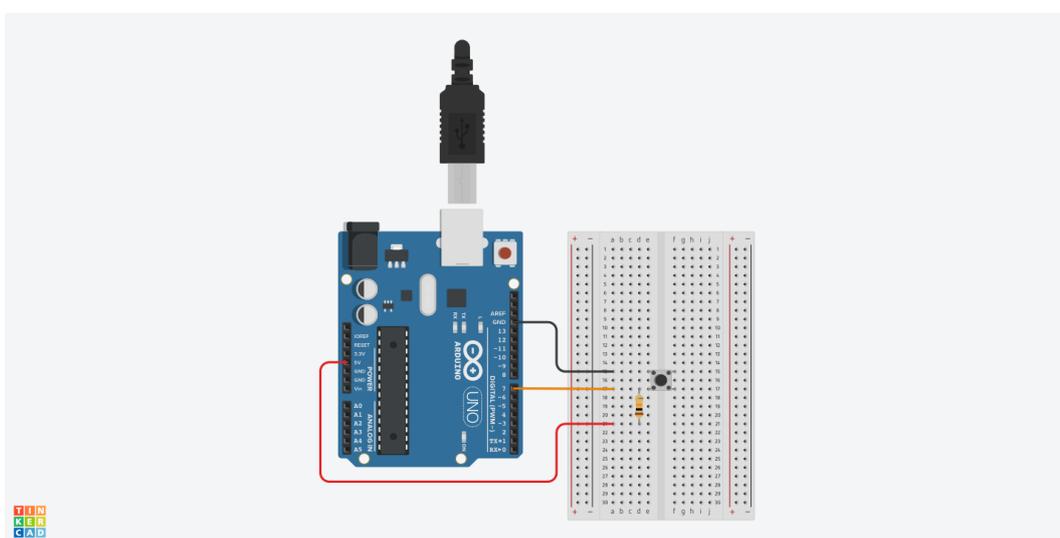
**Complément : Précaution de câblage**

Pour tester la fermeture d'un interrupteur, on utilise une entrée numérique, broches (ou pin) de 2 à 13.

Il est nécessaire de ne pas laisser l'entrée flottante et d'utiliser une résistance pour forcer le potentiel, soit à 0, soit à 5V.



*Setup d'une entrée numérique*



*Résistance de PullUp*

## 1.2. Capteur de lumière - partie 1

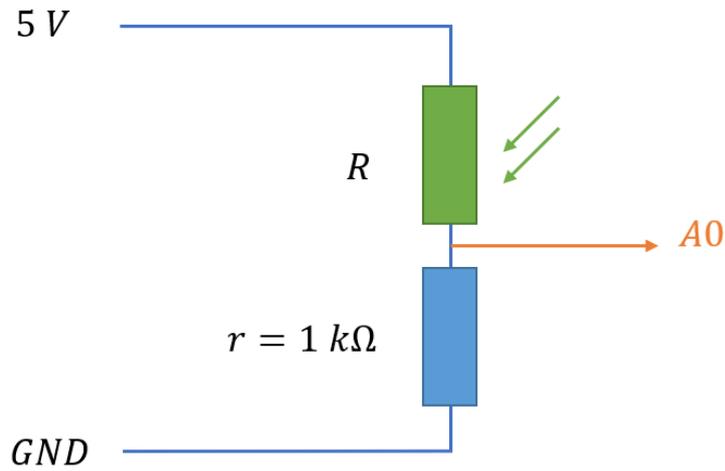
### *Cahier des charges*

On souhaite réaliser un dispositif qui commande l'éclairage d'une DEL lorsqu'il n'y a pas assez de lumière.

On utilise pour cela une photorésistance (capteur passif résistif).

### *Méthode : Premières acquisitions d'une grandeur analogique*

Réaliser sur la platine de câblage le circuit suivant.



Une photorésistance en série avec une résistance de 1 kΩ

**X** Méthode : Présentation de la platine de câblage

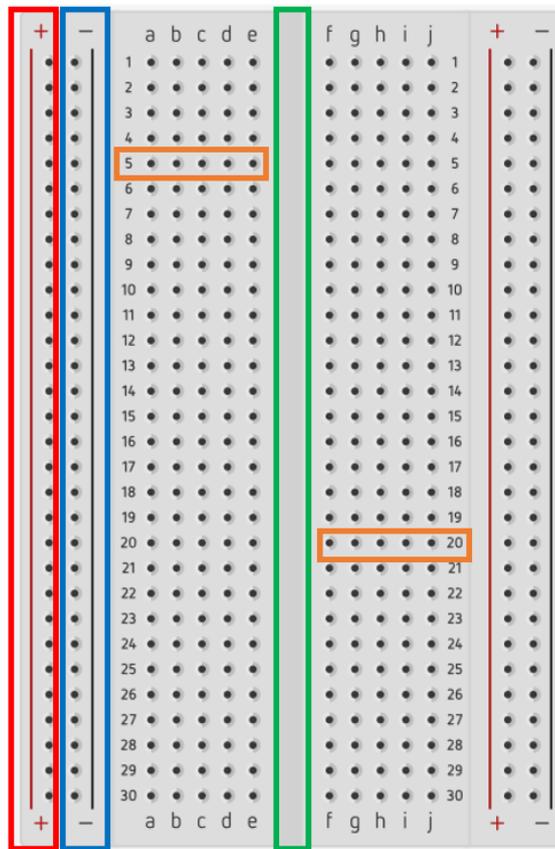
Tous les points d'une zone encadrée sont au même potentiel.

   On relie l'un de ces points au pin 5V de l'Arduino.

   On relie l'un de ces points à la masse de l'Arduino, notée GND (ground).

   Les points 5a, 5b, 5c, 5d et 5e sont donc au même potentiel.

   C'est à ce cheval sur cette zone séparatrice que l'on connecte les circuits intégrés.

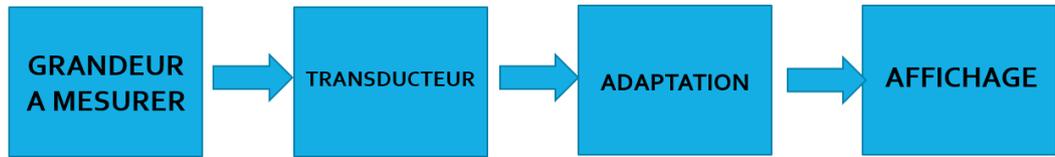


Platine de câblage ou breadboard ou plaque de tests

### 1.3. Capteur de lumière - partie 2

*Objectif*

On souhaite écrire un programme faisant la lecture de l'entrée analogique A0 et affichant la valeur de la résistance de la photorésistance.



Chaîne de mesure

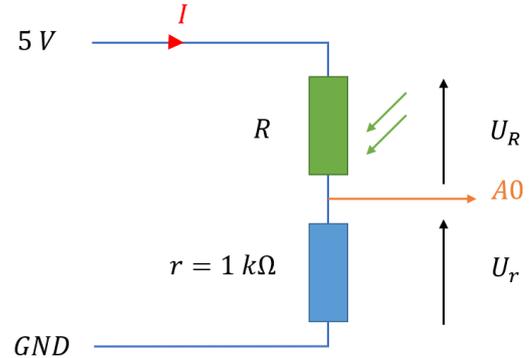
Loi des mailles (loi d'additivité des tensions) :

$$5 V = U_R + U_r$$

Loi d'Ohm :  $I = \frac{U_r}{r}$  et  $U_R = R \times I$

$$5 V - U_r = \frac{R}{r} \times U_r$$

$$R = r \times \left( \frac{5 V}{U_r} - 1 \right)$$



### Syntaxe : Environnement de programmation

Le logiciel Arduino permet deux choses :

1. Rédiger les instructions que le microcontrôleur devra exécuter et les lui téléverser
2. Accéder par le moniteur série aux mesures effectuées (print)

```

1 const int sensorPin = A0;
2 int sensorValue = 0;
3 float U_r, r, R;
4
5 void setup() {
6   r = 1000.0;
7   Serial.begin(9600);
8 }
9
10 void loop() {
11   sensorValue = analogRead(sensorPin);
12   U_r = sensorValue*5.0/1023;
13   R = r*(5.0/U_r - 1);
14   Serial.println(R);
15 }
    
```

1. Entête déclarative  
Déclaration des constantes et des variables
  2. Fonction setup  
Initialisation des variables  
Initialisation de la liaison série avec l'ordinateur
  3. Fonction loop  
Instructions exécutées en boucle
- Compilation du programme
  - Téléversement du programme compilé vers le microcontrôleur
  - Moniteur série

Environnement de programmation et structure d'un programme

### Méthode : Travail à réaliser

- Lancer le logiciel Arduino
- Copier/coller le programme ci-dessous
- Le compiler pour vérifier l'absence d'erreur
- Connecter le microcontrôleur à l'ordinateur et sélectionner le port COM dans le menu /Outils
- Téléverser le programme au microcontrôleur
- Ouvrir le moniteur série et vérifier la cohérence du fonctionnement du programme.

```

1 const int sensorPin = A0;
    
```



## Méthode : Utilisation d'une sortie numérique pour commander la diode

---

1. Déclarer la constante entière ledPin avec la valeur 8. Dans quel bloc ?
2. Définir la broche 8 comme une sortie. Dans quel bloc ?

```
1 const int ledPin = 8;
```

```
1 pinMode(ledPin, OUTPUT);
```

## Méthode : Structure conditionnelle

---

Utiliser une structure conditionnelle Si... Alors... Sinon... pour remplir le cahier des charges. Dans quel bloc ?

```
1 if (condition) {
2 }
3 else {
4 }
```

Une nouvelle commande est utilisée dans la structure conditionnelle. Voir **ici** (cf. p.25) les commandes usuelles.

## 1.5. Simulation avec Tinkercad

### Remarque : Utiliser la simulation

---

Les professeurs peuvent demander à leurs élèves, en amont d'une séance expérimentale, de tester un montage en utilisant un outil de simulation.

Deux avantages à cela :

- se familiariser avec la platine de câblage
- tester le code Arduino et simuler le dispositif.



### Simulation

---

[cf. Capteur de lumière]

## 1.6. Exercice : Ce que je dois retenir

**Lois de l'électricité** : je dois connaître la loi des mailles et la loi d'Ohm et savoir les appliquer à des circuits électriques simples.

**Câblage sur platine** : je dois avoir compris la structure d'une platine (notion de broches reliées) et savoir construire un circuit fermé en utilisant l'alimentation de la carte Arduino™.

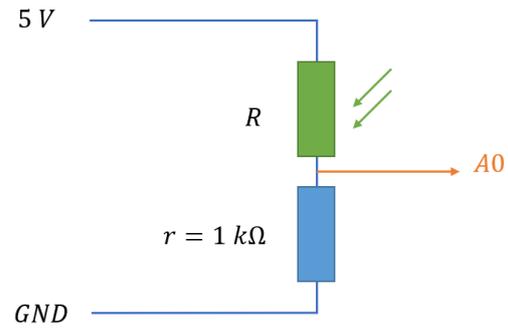
**Programmation du microcontrôleur** : je dois connaître la structure d'un programme (les trois blocs) et utiliser les commandes pour lire une entrée ou écrire sur une sortie du microcontrôleur.

Exercice : Câblage de la platine

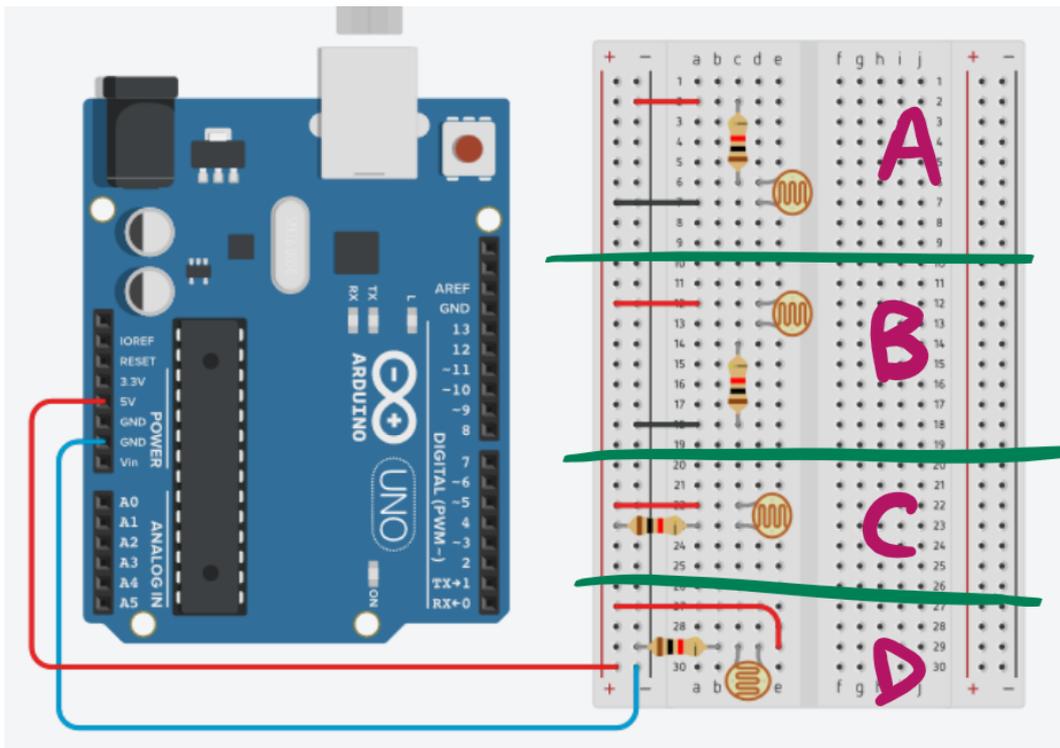
---

Quel est le câblage correct du circuit correspondant au schéma ci-contre ?

On n'a pas souhaité représenter la liaison vers l'entrée analogique A0 pour ne pas alourdir le montage.



Une photorésistance en série avec une résistance de 1 kΩ



Cliquer sur l'image pour l'agrandir.

- Circuit A
- Circuit B
- Circuit C
- Circuit D

### Exercice : Programmer une sortie numérique

Dans le script ci-dessous, compléter les instructions manquantes telles que la sortie numérique 12 soit à l'état haut lorsque la tension  $U_r$  aux bornes de la résistance est inférieure à 1 V.

```
const int _____ ;
```

```
int valeur = 0;
```

```

float U_r;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  valeur = analogRead(0);
  U_r = valeur * 5.0 / 1023;
  delay(100);
  if (valeur > 500) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}

```

## 2. Capteur de température

### Objectifs

- Produire une courbe d'étalonnage : représenter un nuage de points associé à la caractéristique d'un dipôle et modéliser la caractéristique de ce dipôle à l'aide d'un langage de programmation.
- Utiliser une courbe d'étalonnage reliant la résistance d'un système avec une grandeur d'intérêt (température).
- Utiliser un dispositif avec microcontrôleur et capteur électrique résistif (thermistance) pour mesurer une température.

### Intérêt de cette partie

- Modéliser par une équation le comportement d'un dipôle « non linéaire » : la CTN.
- Utiliser ce modèle mathématique pour remonter à la grandeur à mesurer : la température.
- Applications : thermomètre, thermostat, ...
- Technique transposable à d'autres capteurs.

## 2.1. Étalonnage d'une thermistance

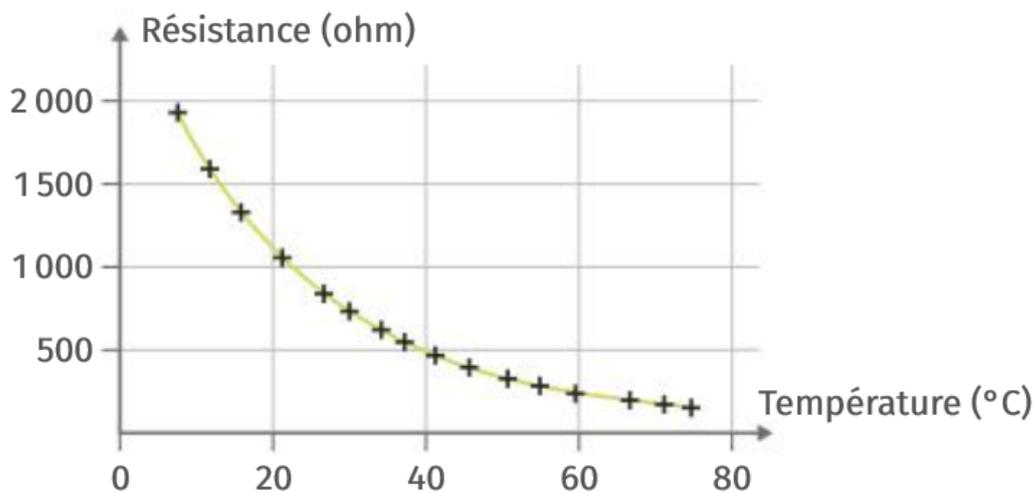
### 🔗 Définition : Généralités

L'étalonnage d'un capteur est une opération qui, dans des conditions spécifiées, établit une relation entre la grandeur physique mesurée et une indication issue du transducteur.

La relation entre les deux grandeurs ne se suffit pas à elle-même. Il convient de préciser les limites du modèle retenu. Ces limites permettront de donner le résultat de la mesure avec les incertitudes-type associées.

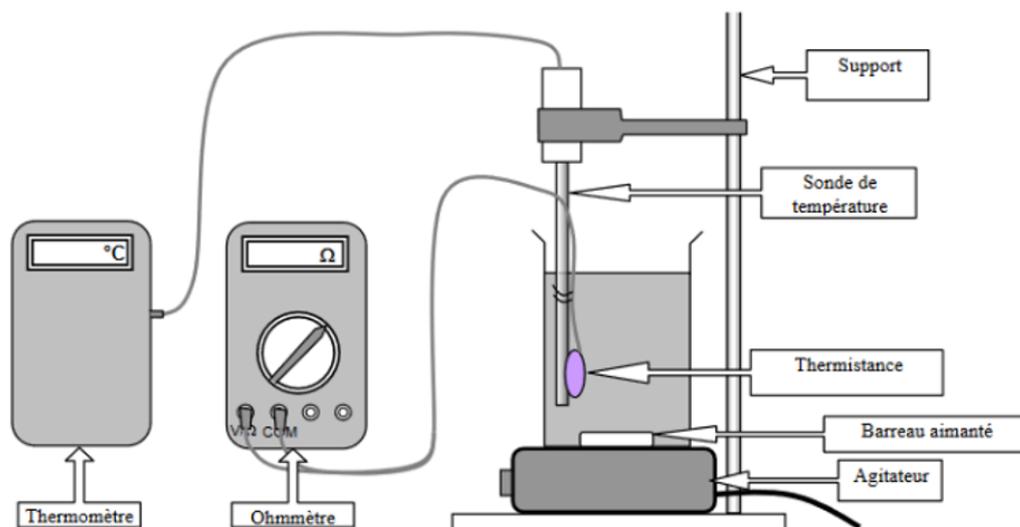
### 👉 Exemple : La thermistance

La résistance d'une thermistance dépend de la température. S'il s'agit d'une CTN, le relevé de sa résistance en fonction de la température du milieu doit donner une représentation ayant l'allure ci-dessous. On cherche donc le modèle  $R = f(\theta)$  ou  $\theta = g(R)$ .



### 🔧 Méthode : Série de mesures

Réaliser une série de mesures de la température et de la résistance de la thermistance. On pourra, par exemple, laisser refroidir un liquide (de l'eau) et prendre des mesures toutes les 30 s.



*Montage pour l'étalonnage d'une thermistance*

Reporter ces mesures dans le tableur-grapheur ci-dessous.

[cf. Classeur Excel pour la saisie des mesures d'étalonnage]

### *Méthode : Modèle mathématique*

Chercher un modèle mathématique, c'est déterminer la relation mathématique qui convient **le mieux** entre deux grandeurs physiques.

On détermine cette relation à partir de la représentation graphique, en général un nuage de points entre les deux grandeurs.

Dans le cas d'une thermistance, en première approximation,  $R$  et  $T = \theta + 273,15$  (K) sont liées par cette relation :

$$R(T) = R(T_0) \times \exp \left[ B \left( \frac{1}{T} - \frac{1}{T_0} \right) \right]$$

avec  $T_0$  la température normalisée 298 K (NF C93-271) et  $B$  l'indice de sensibilité thermique caractéristique du matériau qui constitue la CTN.

### *Fondamental : Relation $\theta$ en fonction de $R$*

Le programme Arduino devra être capable de déterminer la température mesurée à partir de la résistance de la thermistance. On utilisera donc la relation :

$$\frac{1}{T} = C_1 + C_2 \times \ln R$$

Et puisque  $\theta = T - 273,15$  (°C), alors :

$$\theta = \frac{1}{C_1 + C_2 \times \ln R} - 273,15$$

## 2.2. Réalisation d'un dispositif de mesure de température

### *Cahier des charges*

On souhaite contrôler la température d'un milieu réactionnel dont le contenant est placé sur un agitateur magnétique chauffant.

- La température doit être comprise entre 50 et 60 °C. Dans ce cas, une diode verte doit être allumée.
- Si la température est inférieure à 50 °C, une diode jaune doit s'allumer.
- Si la température est supérieure à 60°C, un buzzer doit retentir.

### *Syntaxe : Création d'une fonction basée sur la courbe d'étalonnage*

Avant la fonction setup(), rentrer le modèle mathématique comme suit.

```
float C_1 = ???;
```

```
2 float C_2 = ???;
3
4 float temp(float R){
5     float T = 1/(C_1 + C_2*log(R))-273.15;
6     return T;
7 }
```

Dans la fonction `loop()` (la boucle), appeler le modèle avec cette instruction :

```
1 if (temp(R)<50) {
2     ...
3 }
```

# La sécurité au laboratoire de chimie

II

## 1. La sécurité au laboratoire - Auto-évaluation

Exercice : Question 1

---

Sur une étiquette d'un flacon de butanone, on peut lire la phrase suivante : H336, peut provoquer somnolences et vertiges. Le sigle H correspond à :

- une mention de danger
- un conseil de prudence

Exercice : Question 2

---

Vous préparez une solution aqueuse d'hydroxyde de sodium à  $0,15 \text{ mol} \cdot \text{L}^{-1}$  pour un TP. Quel matériel de protection faut-il donner aux élèves (en plus de leur blouse) ?

- une paire de lunettes
- des gants
- une paire de lunettes et des gants
- aucun

Exercice : Question 3

---

Vous préparez une solution aqueuse d'hydroxyde de sodium à  $2 \text{ mol} \cdot \text{L}^{-1}$  pour un TP. Quel matériel de protection faut-il donner aux élèves (en plus de leur blouse) ?

- une paire de lunettes
- des gants
- une paire de lunettes et des gants
- aucun

Exercice : Question 4

---

Pourquoi depuis 2012 doit-on remplacer la phénolphtaléine par du bleu de thymol ?



après dilution à l'eau

Exercice : Question 9 : Pictogrammes de sécurité

Identifiez les pictogrammes de sécurité suivants.



Mutagène, cancérigène, toxique	Substance corrosive	Dangereux, toxicité aiguë	Gaz comprimé	Toxique pour l'environnement	Substance explosive	Liquide inflammable	Su i:
--------------------------------------	------------------------	---------------------------------	-----------------	---------------------------------	------------------------	------------------------	----------

Exercice : Question 10

Où devez-vous jeter ces différents produits ? Cochez dans la colonne correspondante.

	Bidon solvants	Bidon alcools	Poubelle biologique	Container spécifique pour déchets solides	Evier	Emballage pour déchets de soin perforants
<b>Chloroforme</b>	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6
<b>KCl</b>	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10	<input type="checkbox"/> 11	<input type="checkbox"/> 12
<b>Acétone</b>	<input type="checkbox"/> 13	<input type="checkbox"/> 14	<input type="checkbox"/> 15	<input type="checkbox"/> 16	<input type="checkbox"/> 17	<input type="checkbox"/> 18
<b>Scalpel</b>	<input type="checkbox"/> 19	<input type="checkbox"/> 20	<input type="checkbox"/> 21	<input type="checkbox"/> 22	<input type="checkbox"/> 23	<input type="checkbox"/> 24
<b>Seringue contaminée</b>	<input type="checkbox"/> 25	<input type="checkbox"/> 26	<input type="checkbox"/> 27	<input type="checkbox"/> 28	<input type="checkbox"/> 29	<input type="checkbox"/> 30
<b>Suspension bactérienne</b>	<input type="checkbox"/> 31	<input type="checkbox"/> 32	<input type="checkbox"/> 33	<input type="checkbox"/> 34	<input type="checkbox"/> 35	<input type="checkbox"/> 36
<b>Ethanol</b>	<input type="checkbox"/> 37	<input type="checkbox"/> 38	<input type="checkbox"/> 39	<input type="checkbox"/> 40	<input type="checkbox"/> 41	<input type="checkbox"/> 42
<b>Boîtes Pétri non ensemencées</b>	<input type="checkbox"/> 43	<input type="checkbox"/> 44	<input type="checkbox"/> 45	<input type="checkbox"/> 46	<input type="checkbox"/> 47	<input type="checkbox"/> 48
<b>Aiguille</b>	<input type="checkbox"/> 49	<input type="checkbox"/> 50	<input type="checkbox"/> 51	<input type="checkbox"/> 52	<input type="checkbox"/> 53	<input type="checkbox"/> 54

- 1.
- 2. Bidon alcools
- 3. Poubelle biologique
- 4. Container spécifique pour déchets solides
- 5. Évier
- 6. Emballage pour déchets de soin perforants
- 7.
- 8. Bidon alcools
- 9. Poubelle biologique
- 10. Container spécifique pour déchets solides
- 11. Évier
- 12. Emballage pour déchets de soin perforants
- 13. Bidon solvants
- 14. Bidon alcools
- 15. Poubelle biologique

- 16. Container spécifique pour déchets solides
- 17. Évier
- 18. Emballage pour déchets de soin perforants
- 19.
- 20. Bidon alcools
- 21. Poubelle biologique
- 22. Container spécifique pour déchets solides
- 23. Évier
- 24. Emballage pour déchets de soin perforants
- 25. Bidon solvants
- 26. Bidon alcools
- 27. Poubelle biologique
- 28. Container spécifique pour déchets solides
- 29. Évier
- 30. Emballage pour déchets de soin perforants
- 31. Bidon solvants
- 32. Bidon alcools
- 33. Poubelle biologique
- 34. Container spécifique pour déchets solides
- 35. Évier
- 36. Emballage pour déchets de soin perforants
- 37. Bidon solvants
- 38. Bidon alcools
- 39. Poubelle biologique
- 40. Container spécifique pour déchets solides
- 41. Évier

- 42. Emballage pour déchets de soin perforants
- 43. Bidon solvants
- 44. Bidon alcools
- 45. Poubelle biologique
- 46. Container spécifique pour déchets solides
- 47. Évier
- 48. Emballage pour déchets de soin perforants
- 49. Bidon solvants
- 50. Bidon alcools
- 51. Poubelle biologique
- 52. Container spécifique pour déchets solides
- 53. Évier
- 54. Emballage pour déchets de soin perforants

Exercice : Question 11 : Les déchets infectieux

---

Qu'est-ce qu'un déchet à caractère infectieux ?

- Un résidu de produit chimique
- Un produit anatomique (animaux morts, sang, tissus)
- Des gants souillés
- Des résidus radioactifs
- Un surnageant de culture bactérienne

Exercice : Question 12 : Le matériel adapté

---

En fonction de la manipulation à réaliser, choisissez 1 appareil qui convient.

Centrifugeuse

Chromatographe

PSM

Une étuve à CO<sub>2</sub>

Cuve à électrophorèse

Agitateur magnétique

Autoclave vapeur

Spectrophotomètre

Bain thermostaté sous agitation

Mesurer une absorbance	Réaliser un profil d'ADN	Séparer les différents composants d'un échantillon	Changer le milieu d'une culture cellulaire	Obtenir un culot bactérien à partir d'une solution	Mettre en culture un échantillon de tissu	Réaliser une culture bactérienne avec agitation	Préparer 1 litre de gélose
------------------------	--------------------------	--	--	--	---	---	----------------------------

# Les incertitudes liées à la mesure



III

## 1. Déterminer une incertitude de mesure avec GUM MC

*Le logiciel GUM MC*

Deux versions :

1. une **version prof** (cf. Gum\_MC\_2) pour construire le fichier avec les grandeurs dont dépend la grandeur de sortie. Exemple :

$$t = \frac{P \times m_s}{V_{solution}}$$

- le pourcentage massique  $P$ , la masse de solide  $m_s$  et le volume de la solution  $V_{solution}$  sont les trois grandeurs d'entrée.
- La concentration en masse  $t$  est la grandeur de sortie.

2. une **version élève** (cf. Gum\_MC\_Eleve\_Lycee\_2), prête à l'emploi, où l'élève devra rentrer l'estimateur de chaque grandeur et la valeur des erreurs.

# Ateliers à la carte

## IV

### 1. Utilisation des tableurs en sciences

Le classeur Excel en téléchargement ci-dessous permet de travailler quatre aspects du tableur le plus souvent mobilisés en sciences :

- construction de graphique de type nuage de points
- modélisation d'une caractéristique ou d'une partie de caractéristique
- détermination d'une incertitude-type sur un coefficient de régression
- calcul de grandeur physique à partir d'un pointage vidéo

[cf. Différentes activités avec Excel]

Pour chaque activité, un tutoriel vidéo est proposé. Vous pouvez dans un premier temps chercher par vous-même, puis vous aider du tutoriel.

#### 1.1. Créer un graphique avec Excel

Ouvrir l'onglet "graphique" du classeur Excel (cf. Différentes activités avec Excel) et construire un graphique de type nuage de points à partir des couples de mesure  $(U, I)$ .

#### 1.2. Modéliser un ensemble de points de mesure

Ouvrir l'onglet "modélisation" du classeur Excel (cf. Différentes activités avec Excel) et déterminer le modèle  $I = a \times U + b$ .

#### 1.3. Déterminer l'incertitude-type d'une grandeur modélisée à partir d'une régression linéaire

Ouvrir l'onglet "stats" du classeur Excel (cf. Différentes activités avec Excel) et déterminer l'incertitude-type sur le coefficient  $R_d$  dans le modèle  $U = R_d \times I + U_0$

#### 1.4. Calculer des grandeurs à partir de listes

Ouvrir l'onglet "graphique" du classeur Excel (cf. Différentes activités avec Excel) et construire les grandeurs vitesse, énergie potentielle élastique, énergie cinétique et énergie mécanique.

# Conclusion



Télécharger les documents (cf. Les documents du stage en un seul pdf) du stage.

# Contenus annexes

## > Quelques commandes Arduino

### *Syntaxe : Commandes basiques sur les entrées/sorties*

Lors de l'utilisation d'un transducteur, la première commande qu'on vient à utiliser est la lecture d'une entrée analogique.

Concernant les entrées/sorties numériques, il est nécessaire de spécifier si elles sont utilisées en entrée (lecture) ou en sortie (écriture). Cela se fait avec la commande `pinMode()`.

```
1 pinMode(2, INPUT);
2 pinMode(3, OUTPUT);
```

Résumé des quatre commandes de lecture/écriture sur les entrées/sorties.

```
1 // lire la valeur de l'entrée analogique A2 (10 bits donc valeurs de 0 à
  1023)
2 analogRead(A2);
3
4 // appliquer 5.0 V sur la sortie 4
5 digitalWrite(4, HIGH);
6
7 // lire l'état (haut ou bas, 5.0 V ou 0) de l'entrée numérique 5
8 digitalRead(5);
9
10 // envoyer la valeur 138 sur la sortie 9~ (8 bits seulement donc valeurs
    de 0 à 255)
11 analogWrite(9, 138);
```

### *Complément : Pour aller plus loin*

La commande `tone()` permet de produire un signal numérique (5.0 V ou 0) ayant une fréquence donnée pour appliquer sur un buzzer ou un petit haut-parleur.

```
1 // note La à 440 Hz
2 tone(8, 440);
```

La commande `map()` permet de réaliser un changement d'échelle.

Les instructions suivantes permettent de convertir une valeur lue sur une entrée analogique pouvant aller jusqu'à 1023 en une valeur limitée à l'étendue 0 -> 255. Cette dernière valeur peut être appliquée à une sortie numérique avec le symbole `~`.

```
1 sensorValue = analogRead(A0);
2 int val = map(sensorValue, 0, 1023, 0, 255);
3 analogWrite(10, val);
```