

TP Chauffeurette chimique

Capacités exigibles du programme :

Bilans d'énergie

- Mettre en œuvre une technique de calorimétrie.
- Déterminer la valeur en eau d'un calorimètre.
- Estimer les fuites thermiques lors d'expériences réalisées avec un calorimètre.

Liste du matériel :

- | | |
|---|--|
| — Acétate de sodium anhydre (~ 40 g par groupe) | — Plaque chauffante avec agitation magnétique |
| — Eau distillée | — Balance (précision $\sim 0,1$ g) |
| — Éprouvette graduée de 50 mL | — Eau chaude (bouilloire) |
| — Erlenmeyer de 500 mL | — Module Arduino, câble USB |
| — Calorimètre | — Capteur de température étanche Grove DS18B20 |
| — Agitateur de verre | |

1 Réalisation d'une chauffeurette chimique

Déterminer et mettre en œuvre un protocole expérimental permettant de réaliser le mélange contenu dans une chauffeurette chimique à partir de 200 mL d'eau.

2 Détermination de l'enthalpie standard de la réaction de cristallisation

Déterminer et mettre en œuvre un protocole expérimental permettant de mesurer l'évolution de la température au cours de la cristallisation. Déterminer alors l'enthalpie standard de la réaction de cristallisation du mélange et comparer la valeur obtenue à la valeur théorique.

Données

Contenu de la chauffeurette : solution d'éthanoate de sodium $\text{CH}_3\text{COO}^-(\text{aq}) + \text{Na}^+(\text{aq})$

Réaction de cristallisation : $\text{CH}_3\text{COO}^-(\text{aq}) + \text{Na}^+(\text{aq}) + 3\text{H}_2\text{O}(\ell) = \text{CH}_3\text{COONa}, 3\text{H}_2\text{O}(\text{s})$

$$M(\text{C}) = 12 \text{ g}\cdot\text{mol}^{-1}$$

$$M(\text{O}) = 16 \text{ g}\cdot\text{mol}^{-1}$$

$$M(\text{Na}) = 23 \text{ g}\cdot\text{mol}^{-1}$$

$$\Delta_f H^\circ(\text{CH}_3\text{COONa}, 3\text{H}_2\text{O}(\text{s})) = -1602 \text{ kJ}\cdot\text{mol}^{-1}$$

$$\Delta_f H^\circ(\text{CH}_3\text{COO}^-(\text{aq})) = -486 \text{ kJ}\cdot\text{mol}^{-1}$$

$$\Delta_f H^\circ(\text{Na}^+(\text{aq})) = -240 \text{ kJ}\cdot\text{mol}^{-1}$$

$$\Delta_f H^\circ(\text{H}_2\text{O}(\ell)) = 286 \text{ kJ}\cdot\text{mol}^{-1}$$

$$c_p^\circ(\text{H}_2\text{O}(\ell)) = 4,18 \text{ kJ}\cdot\text{kg}^{-1}$$

$$c_p^\circ(\text{CH}_3\text{COONa}, 3\text{H}_2\text{O}(\text{s})) = 3,31 \text{ kJ}\cdot\text{kg}^{-1}$$

A Annexe 1 - Suivi de température à l'aide d'un module Arduino

A.1 Importation des bibliothèques nécessaires au capteur

Le capteur de température utilisé ici possède un circuit intégré DS18B20. Il nécessite des bibliothèques qui ne sont pas incluses par défaut dans l'IDE Arduino. Il faut donc les importer. Ces bibliothèques ont été récupérées sur le site du constructeur, ce sont les fichiers `OneWire.zip` et `DallasTemperature.zip`.

Pour les importer, utiliser le menu *Croquis* → *Inclure une bibliothèque* → *Ajouter la bibliothèque .ZIP* et choisir chaque fichier.

A.2 Programme utilisé

De la même façon un programme type a été téléchargé sur le site du constructeur, `suivi_temperature.ino`.

Ouvrir ce programme, et vérifier son bon fonctionnement en le téléversant puis en ouvrant le moniteur série (attention **sélectionner la même vitesse de transfert des données dans le moniteur série**, ici : 115200 bauds).

A.3 Mesure d'une grandeur au cours du temps et récupération des données

Si l'on doit récupérer les données pour en faire par exemple une représentation graphique en Python, le plus simple est de réaliser un copier-coller des données directement depuis le moniteur série.

Il faut donc les afficher d'une façon qui permette de les incorporer facilement dans un script Python. Une possibilité est de les afficher dans le moniteur série sous la forme `(mesure_x, mesure_y)`, à chaque ligne, par exemple :

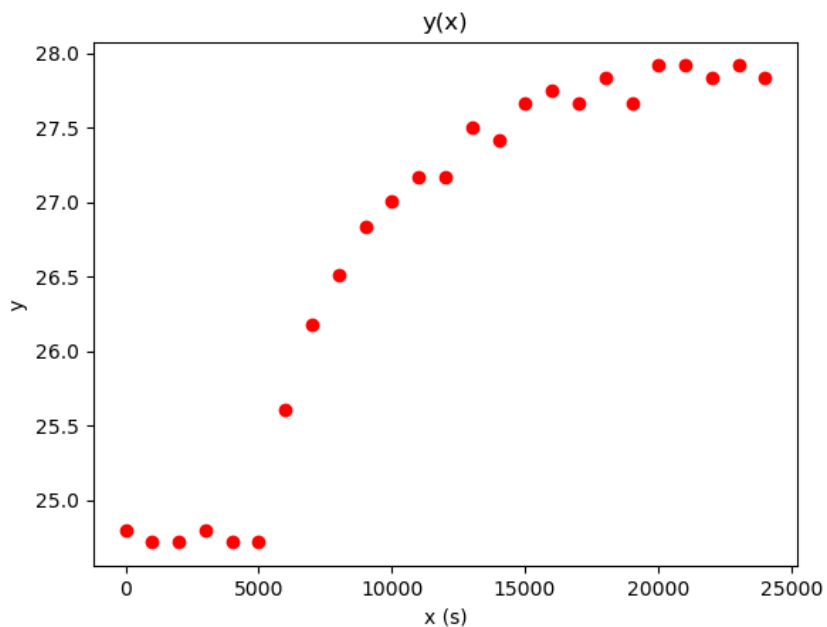


Puis d'utiliser un script Python où il suffit de coller les données sélectionnées pour réaliser leur représentation graphique, par exemple :

```
1 import matplotlib.pyplot as plt
3 donnees=[
4 (0,24.80),
5 (1000,24.72),
6 (2001,24.72),
7 (3003,24.80),
8 (4003,24.72),
9 (5005,24.72),
10 (6006,25.61),
11 (7007,26.18),
12 (8008,26.51),
13 (9010,26.84),
14 (10010,27.01),
15 (11012,27.17),
16 (12013,27.17),
17 (13014,27.50),
18 (14015,27.42),
19 (15016,27.67),
```

```
(16018, 27.75),  
21 (17018, 27.67),  
(18020, 27.84),  
23 (19021, 27.67),  
(20022, 27.92),  
25 (21023, 27.92),  
(22025, 27.84),  
27 (23026, 27.92),  
(24027, 27.84),  
29 ]  
  
31 liste_x=[mesure[0] for mesure in donnees]  
liste_y=[mesure[1] for mesure in donnees]  
33  
plt.figure(1)  
35 plt.plot(liste_x, liste_y, 'ro')  
plt.xlabel('x (s)')  
37 plt.ylabel('y')  
plt.title('y(x)')  
39 plt.show()
```

recuperer_donnees_arduino_moniteur_serie_copier-coller.py



A.4 Modification du programme Arduino

Modifier le programme Arduino fourni pour qu'il affiche dans le moniteur série le couple de données (t, T) toutes les 2 secondes environ et tester son bon fonctionnement en plaçant le capteur dans votre main. Représentez graphiquement le résultat pour vérifier le bon fonctionnement de l'ensemble.