

Capacités exigibles :

- Utiliser un dispositif avec microcontrôleur et capteur
- Suivre l'évolution d'une température pour déterminer le caractère endothermique ou exothermique d'une transformation chimique

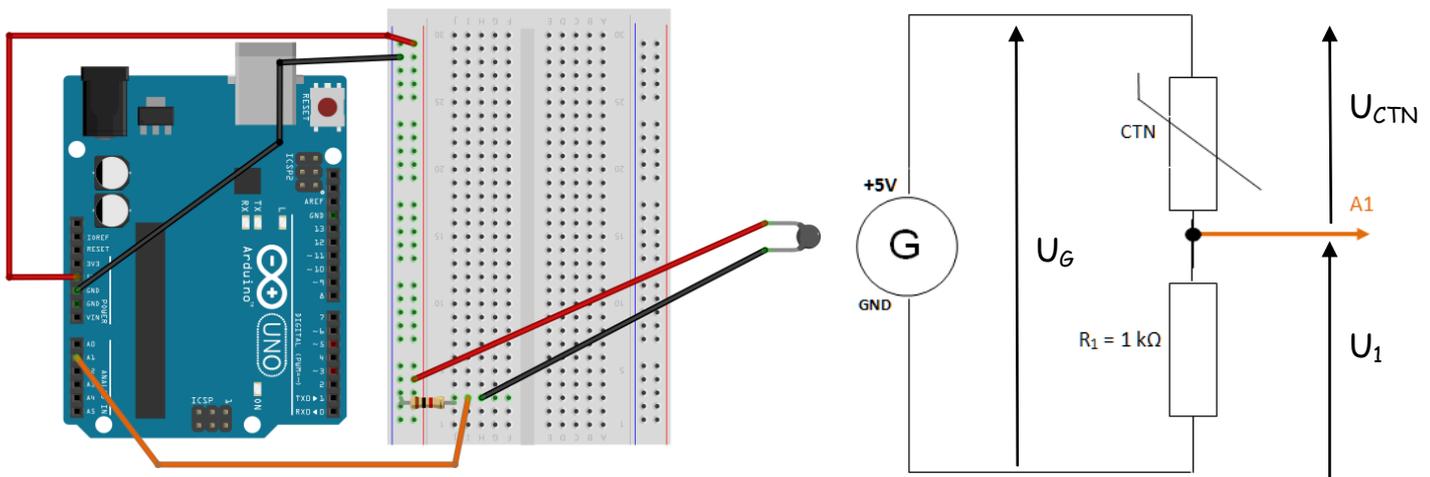
Afin de concrétiser le projet de suivre l'évolution de la température d'une transformation chimique, nous allons utiliser la thermistance étudiée au précédent TP, une résistance ainsi qu'une carte *Arduino* pour exploiter les données recueillies.

Comment suivre, avec une carte *Arduino*, la température d'une transformation chimique ?

1 Le montage avec la carte *Arduino* à réaliser

La carte *Arduino* ne peut pas relever directement une température, ce n'est pas un thermomètre ! Elle ne peut qu'envoyer ou recevoir des informations sous forme de tension électrique (en V).

On a vu au TP précédent que, grâce à la thermistance, la grandeur physique **température** θ (en °C) est « transformée » en une grandeur électrique : la **résistance** R_{CTN} (en Ω). Il faut encore pouvoir « transformer » cette grandeur en **tension électrique** U_1 (en V) pour qu'elle soit lue par une des entrées analogiques de la carte *Arduino*. C'est le rôle du montage ci-dessous, appelé « pont diviseur de tension » :



On note U_G la tension aux bornes du générateur intégré à l'*Arduino* ($U_G = 5V$), U_{CTN} la tension aux bornes de la thermistance et U_1 la tension aux bornes de la résistance $R_1 (= 1 k\Omega)$.

1. Sur le schéma, repasser au fluo l'unique maille du circuit électrique. Flécher le courant électrique I qui circule dans cette maille.
2. Donner la relation entre U_G , U_{CTN} et U_1 . Exprimer U_{CTN} en fonction de U_1 et U_G .
3. En utilisant la loi d'Ohm, exprimer R_{CTN} en fonction de I et U_{CTN} . En utilisant la même loi, exprimer I en fonction de R_1 et U_1 . En déduire l'expression de R_{CTN} en fonction de R_1 , U_{CTN} et U_1 .
4. En utilisant les résultats des questions 2 et 3, montrer que l'expression de R_{CTN} en fonction de R_1 , U_G et U_1 s'écrit :

$$R_{CTN} = \left(\frac{U_G}{U_1} - 1 \right) \times R_1 \quad \text{soit} \quad R_{CTN} = \left(\frac{5,0}{U_1} - 1 \right) \times 1000$$

5. Rappeler la relation affine trouvée au TP précédent reliant la température θ à la résistance R_{CTN} :

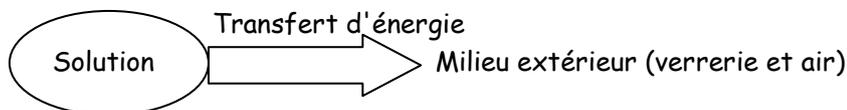
 $\theta =$

2 Le programme à téléverser dans la carte

Il ne reste plus qu'à programmer la carte Arduino afin de finaliser le projet. Puisqu'on cherche à savoir si une transformation chimique est endothermique ou exothermique, voici d'abord 2 documents :

Doc. 1 Quelques définitions

- Une transformation est **exothermique** lorsqu'elle s'accompagne d'une libération d'énergie sous forme de chaleur. La solution cède de l'énergie au milieu extérieur donc la température du milieu extérieur augmente.



- Une transformation est **endothermique** lorsqu'elle s'accompagne d'une absorption d'énergie. L'énergie provient du milieu extérieur donc la température du milieu extérieur diminue.



- Une transformation est **athermique** lorsqu'il n'y a pas d'échange d'énergie avec le milieu extérieur.
- L'énergie transférée s'exprime en Joule (symbole J). On utilise souvent le kilojoule (kJ) : 1 kJ = 1000 J

Doc. 2 Protocole de suivi d'une transformation chimique

☞ Choisir un des 3 solides ioniques à dissoudre :

Solides ioniques	Nitrate d'ammonium NH_4NO_3	Chlorure de sodium NaCl	Sulfate de magnésium MgSO_4
Sécurité	Lunettes obligatoires	X	Lunettes obligatoires

☞ Dans une coupelle en plastique, prélever 2,5 g de ce solide.

☞ Mesurer, à l'aide d'une éprouvette graduée, un volume de 20 mL d'eau distillée. Verser ce volume d'eau dans un petit bécher.

☞ Placer la thermistance (reliée au circuit) dans le bécher. Commencer le suivi automatisé de température.

☞ Noter la température initiale de l'eau : $\theta_{\text{init}} = \dots\dots\dots$

☞ Verser le solide ionique préparé dans le bécher et agiter jusqu'à dissolution complète.

Afin de finaliser le projet et d'assurer le suivi automatisé de la température, 2 programmes associés à 2 câblages sont disponibles (voir annexes) en fonction de votre avancement et de votre niveau de compétence en programmation et en électricité :

Programme n°1 : - lire la valeur de la tension U_1

- en déduire la valeur de la résistance R_{CTN} (voir formule de la question 4.)

- en déduire la valeur de la température θ (voir formule de la question 5.)

- afficher à l'écran de l'ordinateur cette valeur de température.

apprenti

Programme n°2 : programme n°1 avec en plus :

- Si la température reste autour de la température initiale ALORS allumer une DEL verte

- SI la température baisse de plus de 5°C ALORS allumer une DEL bleue

- SI la température augmente de plus de 5°C ALORS allumer une DEL rouge

confirmé

☞ Commencer par le programme n°1 : câbler la platine d'essai, compléter l'instruction manquante dans le programme fourni et le téléverser dans la carte *Arduino*.

Ne pas oublier de cliquer ensuite sur « Moniteur série »  (icône en haut à droite de l'*Arduino IDE*) pour afficher les valeurs de la température.

APPELER LE PROFESSEUR POUR QU'IL VERIFIE VOTRE MONTAGE

☞ Poursuivre avec éventuellement le programme n°2 si le professeur est d'accord.

☞ Choisir 2 des 3 solides ioniques et réaliser le suivi de température automatisé.

✎ 6. Conclure quant à la nature de la transformation chimique pour les 2 dissolutions observées.

```

float tempC;           // Variable dans laquelle sera stockée la valeur du signal
                       // analogique de la tension U1 puis transformée en température (en°C)
int tempPin = A1;     // Tension U1 à lire sur la pin A1

void setup() {

  Serial.begin(9600);  // Initialisation de la communication série à 9600 bauds

}

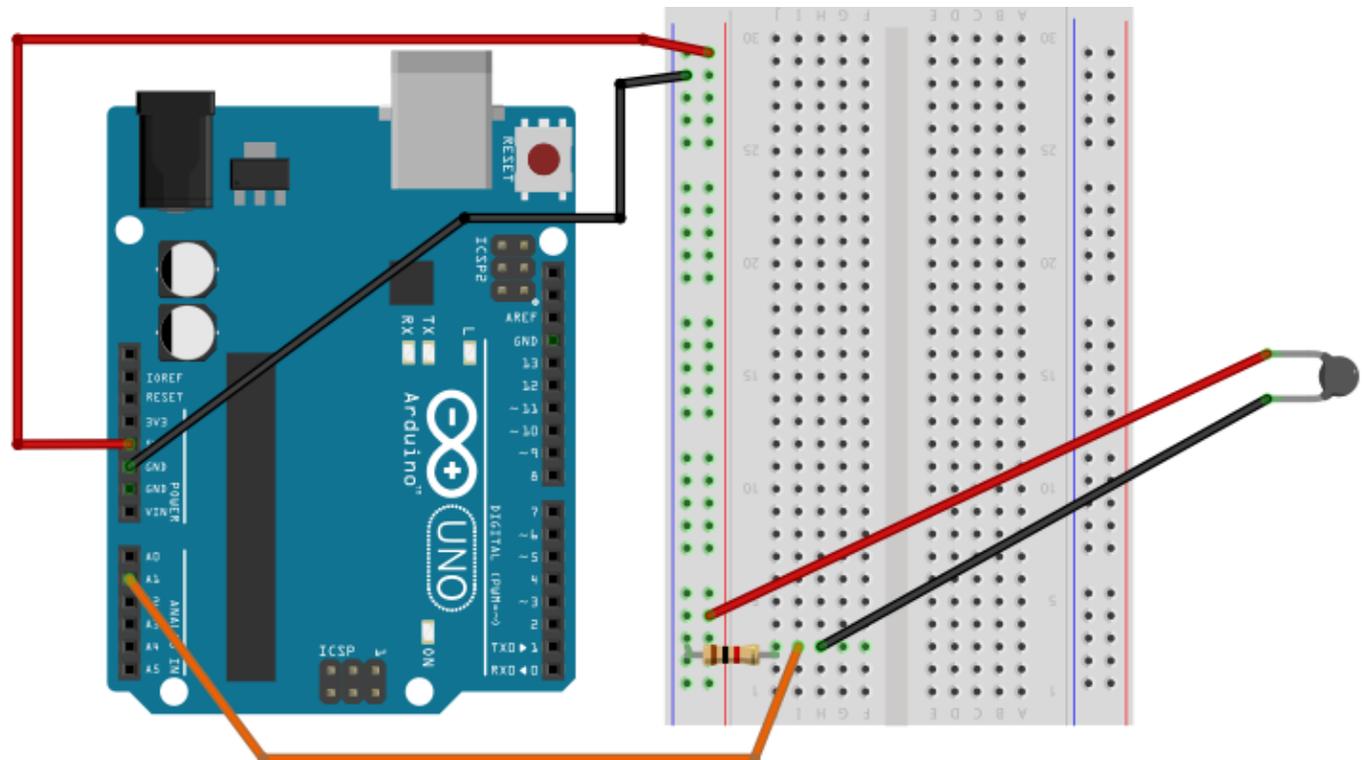
void loop() {

tempC = analogRead(tempPin);           // Lecture de U1 analogique et conversion numérique en une valeur comprise entre 0 et 1023 (10 bits)
tempC = tempC*5.0/1023;                // Conversion de la valeur comprise entre 0 et 1023 en une tension comprise entre 0,000 V et 5,000 V
tempC = (5.0/tempC - 1)*1000;          // Transformation de U1 en RCTN (question 4)
tempC =           ; // Transformation de RCTN en température (question 5)
Serial.println(tempC,1);                // Envoi de la température sur l'écran de l'ordinateur (avec 1 chiffre après la virgule)

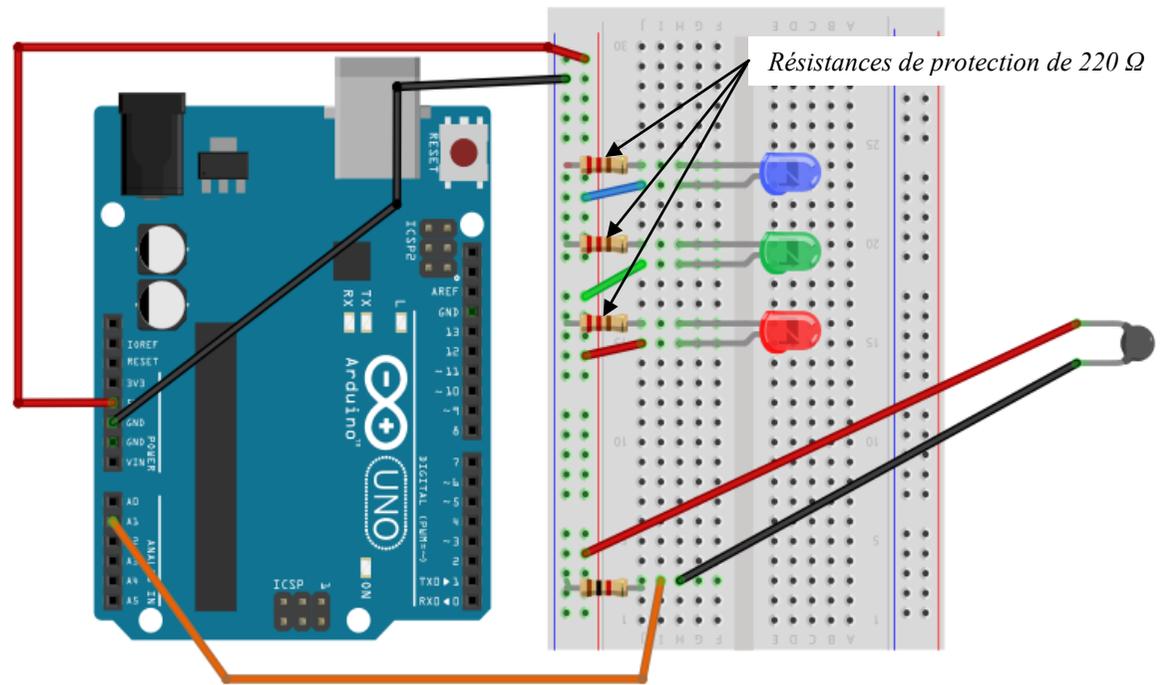
delay(1000);                            // Attendre 1000 ms avant l'envoi d'une nouvelle donnée
}

```

**Suivi automatisé de
température
Programme n°1**



**Suivi automatisé de
température
Programme n°2**



```

char tempinit= 0; // Température initiale de l'eau : 0 °C
float tempC; // Variable dans laquelle sera stockée la valeur
// analogique de la tension U1 puis transformée en température (en°C)

int tempPin = A1; // Tension U1 à lire sur la pin A1
int red1Pin= 6; // Signal pour allumer la DEL rouge à envoyer sur la pin 6
int green1Pin= 5; // Signal pour allumer la DEL verte à envoyer sur la pin 5
int blue1Pin= 4; // Signal pour allumer la DEL bleue à envoyer sur la pin 4

void setup() {
  Serial.begin(9600); // Initialisation de la communication série à 9600 bauds
  pinMode(red1Pin, OUTPUT); // Déclaration de la sortie de la DEL rouge en SORTIE
  pinMode(green1Pin, OUTPUT); // Déclaration de la sortie de la DEL verte en SORTIE
  pinMode(blue1Pin, OUTPUT); // Déclaration de la sortie de la DEL bleue en SORTIE
}

void loop() {
  tempC = analogRead(tempPin); // Lecture de U1 analogique et conversion numérique en une valeur comprise entre 0 et 1023 (10 bits)
  tempC = tempC*5.0/1023; // Conversion de la valeur comprise entre 0 et 1023 en une tension comprise entre 0,000 V et 5,000 V
  tempC = (5.0/tempC - 1)*1000; // Transformation de U1 en RCTN (question 4)
  tempC = 0; // Transformation de RCTN en temperature (question 5)
  Serial.println(tempC,1); // affichage de la température sur l'écran de l'ordinateur (avec 1 chiffre après la virgule)

  // Gestion des led avec la temperature
  digitalWrite(red1Pin,tempC>tempinit+5); // Led rouge allumee si la température augmente de +5°C
  digitalWrite(green1Pin,tempC>=tempinit-5 && tempC<=tempinit+5); // Led verte allumee si la température reste à + ou - 5°C de la température initiale
  digitalWrite(blue1Pin,tempC<tempinit-5); // Led bleue allumee si la température baisse de -5°C

  delay(1000); // Attendre 1000 ms avant l'envoi d'une nouvelle donnée
}

```