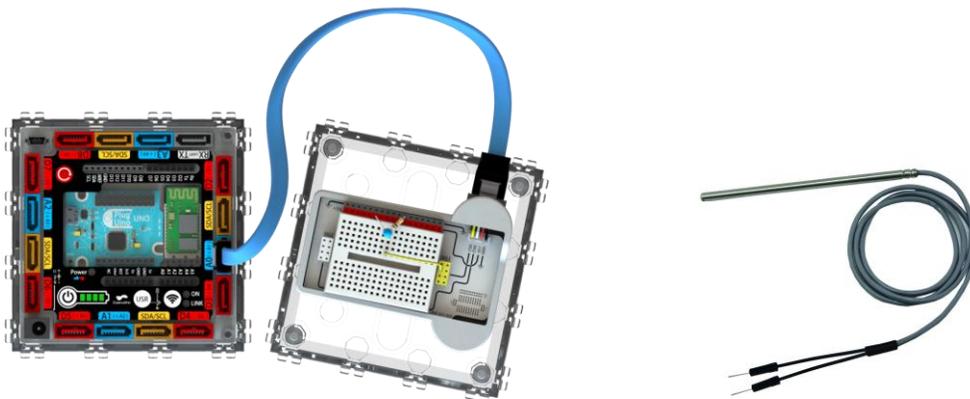


Niveau : seconde	Ondes et signaux partie 3 : Signaux et capteurs
Capteurs électriques.	<p><i>Produire et utiliser une courbe d'étalonnage reliant la résistance d'un système avec une grandeur d'intérêt (température, pression, intensité lumineuse, etc.).</i></p> <p><i>Utiliser un dispositif avec microcontrôleur et capteur.</i></p>

Dossier TP Plug'Uino[®] :
Étalonnage d'un capteur de température CTN et
réalisation d'un thermomètre électronique à partir de
l'étalonnage de la résistance
Programmation en langage Arduino



1. Résumé de l'activité

Cette activité expérimentale a pour but de réaliser la courbe d'étalonnage reliant la résistance d'une sonde de mesure de température CTN avec la température, et de réaliser ensuite un thermomètre numérique.

2. Thème du programme abordé

Niveau : seconde	Ondes et signaux partie 3 : Signaux et capteurs
Capteurs électriques.	<p><i>Produire et utiliser une courbe d'étalonnage reliant la résistance d'un système avec une grandeur d'intérêt (température, pression, intensité lumineuse, etc.).</i></p> <p><i>Utiliser un dispositif avec microcontrôleur et capteur.</i></p>

3. Matériel mis en œuvre

Une interface Plug'Uino® réf. 650 003

Un calorimètre réf. 005 025;

Un thermomètre numérique réf. 310 007

Un multimètre réf. 340 048;

Un capteur de température CTN 10 kOhms monté dans une sonde étanche réf. 651 061;



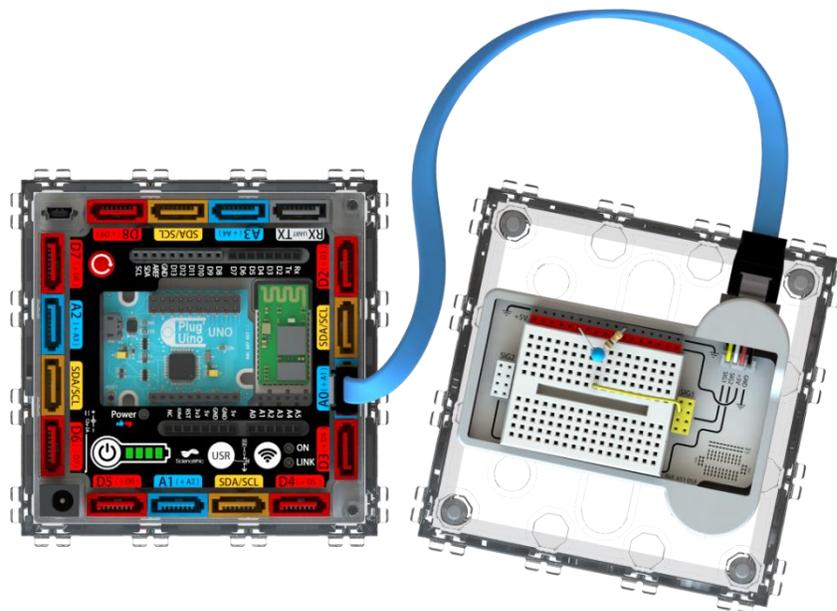
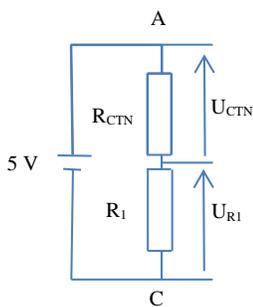
Une platine de connexion réf. 651 058;

Une résistance de 10 kΩ ;

Un tableur-grapheur.

4. Présentation de l'expérience réalisée

- On verse de l'eau chaude dans un calorimètre. On y plonge la sonde de température CTN, ainsi qu'un thermomètre de référence.
- À l'aide d'un multimètre, on relève la valeur de la résistance R_{CTN} fournie par le capteur de température CTN, en fonction de la température T . On fait varier la température de l'eau en rajoutant progressivement de la glace pilée et en homogénéisant bien le mélange.
- On entre les valeurs obtenues dans un tableur-grapheur afin de visualiser la courbe d'étalonnage $T = f(R_{CTN})$.
- On réalise alors une courbe de tendance permettant d'obtenir la relation mathématique $T = f(R_{CTN})$.
- On réalise le montage suivant, permettant de mesurer la tension U_{R1} aux bornes de la résistance R_1 de 10 kOhms, en série avec la CTN de résistance R_{CTN} , pour déterminer ensuite la résistance R_{CTN} à partir de la mesure de la tension U_{R1}



5. Réalisation de la courbe d'étalonnage de la CTN

À l'aide d'un multimètre, on relève la valeur de la résistance R_{CTN} fournie par le capteur de température CTN, en fonction de la température T . On fait varier la température de l'eau en rajoutant progressivement de la glace pilée et en homogénéisant bien le mélange.

On entre les valeurs obtenues dans un tableur-grapheur afin de visualiser la courbe d'étalonnage $T = f(R_{CTN})$.

On réalise alors une courbe de tendance permettant d'obtenir la relation mathématique $T = f(R_{CTN})$.

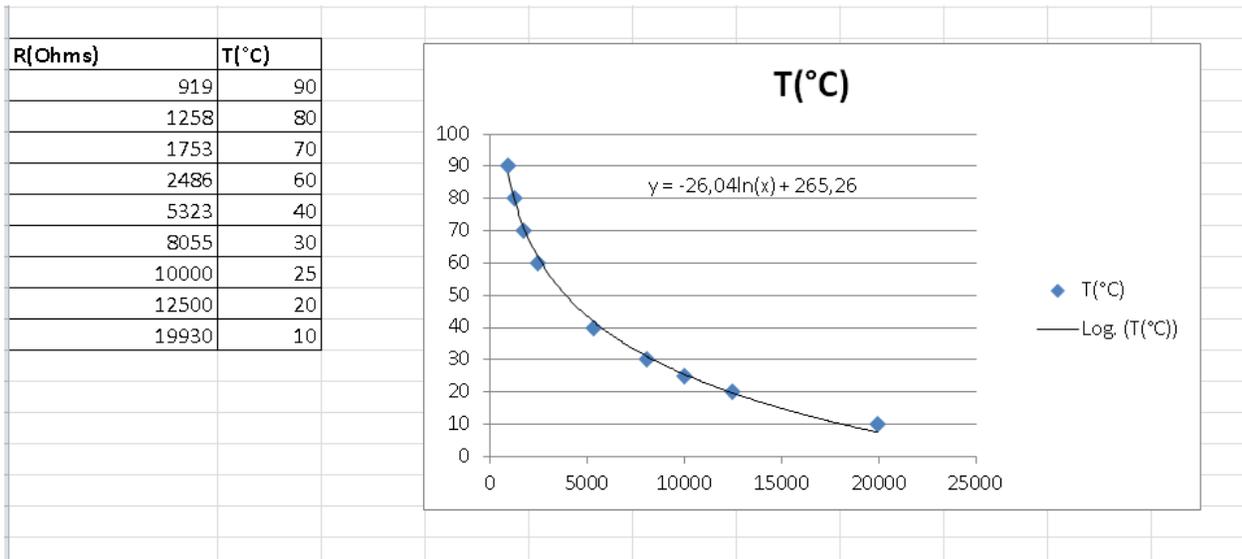
Les valeurs mesurées sont disponibles dans le fichier ScEthic_Etalonnage_CTN_MesureR.xls

L'exploitation est la suivante :

On modélise la courbe obtenue en double-cliquant sur le graphique, puis en faisant clic-droit sur la courbe de données, et « insérer une courbe de tendance » :

On sélectionne une courbe de tendance logarithmique, et on demande d'afficher l'équation

La courbe de tendance ainsi que son équation sont alors affichées :



Cette équation sera utilisée dans le programme Python pour calculer la température en fonction de la résistance R de la CTN.

- Le montage réalisé est un pont diviseur : U_{AC} est la tension de référence du microcontrôleur Arduino Uno, c'est-à-dire 5 V.

On a $U_{AC} = 5 = (R_1 + R_{CTN}) * I$

et $U_{R1} = R_1 * I$

donc $U_{R1} = \frac{5R_1}{R_{CTN} + R_1}$

On peut donc obtenir la valeur de la résistance R_{CTN} de la CTN à partir de la mesure de U_{R1} ,

$$R_{CTN} = \frac{R_1(5 - U_{R1})}{U_{R1}}$$

et en déduire la valeur de la température à partir de la courbe d'étalonnage

$T = f(R_{CTN})$ préalablement réalisée sur tableur.

On rentre donc ces fonctions dans le programme Arduino pour effectuer tous les calculs à partir de la mesure de la tension aux bornes de la résistance R_1 (U_{R1}), et afficher la température mesurée par la sonde CTN.

6. Programme Arduino

Le programme Arduino fonctionnel, permettant d'afficher la température à partir de la courbe d'étalonnage et de la tension U_{R1} mesurée aux bornes de la résistance R_1 est donné ci-après.

Le calcul de U_{R1} à partir de la valeur numérisée par l'entrée A0 du microcontrôleur est effectué à la ligne 21.

```
21 float Url = 5.000000*N/1023.000000; // Convertit le nombre N en tension Url mesurée sur BrocheConnexion
```

Le calcul de R_{CTN} à partir de U_{R1} est effectué en ligne 22 : $R_{CTN} = \frac{R_1(5-U_{R1})}{U_{R1}}$

```
22 float R = R1*(5-Url)/Url; // Calcule la résistance de la CTN à partir de la valeur de Url
```

le calcul de T à partir de R_{CTN} est effectué à la ligne 23 :

```
23 float T = -26.04*log(R)+265.26;
24 // La ligne 23 calcule la valeur de la température qui correspond à R
25 // à partir de la courbe d'étalonnage
26 // Remarque : dans Arduino, la fonction logarithme népérien se note log
27 // et le logarithme décimal log10
```

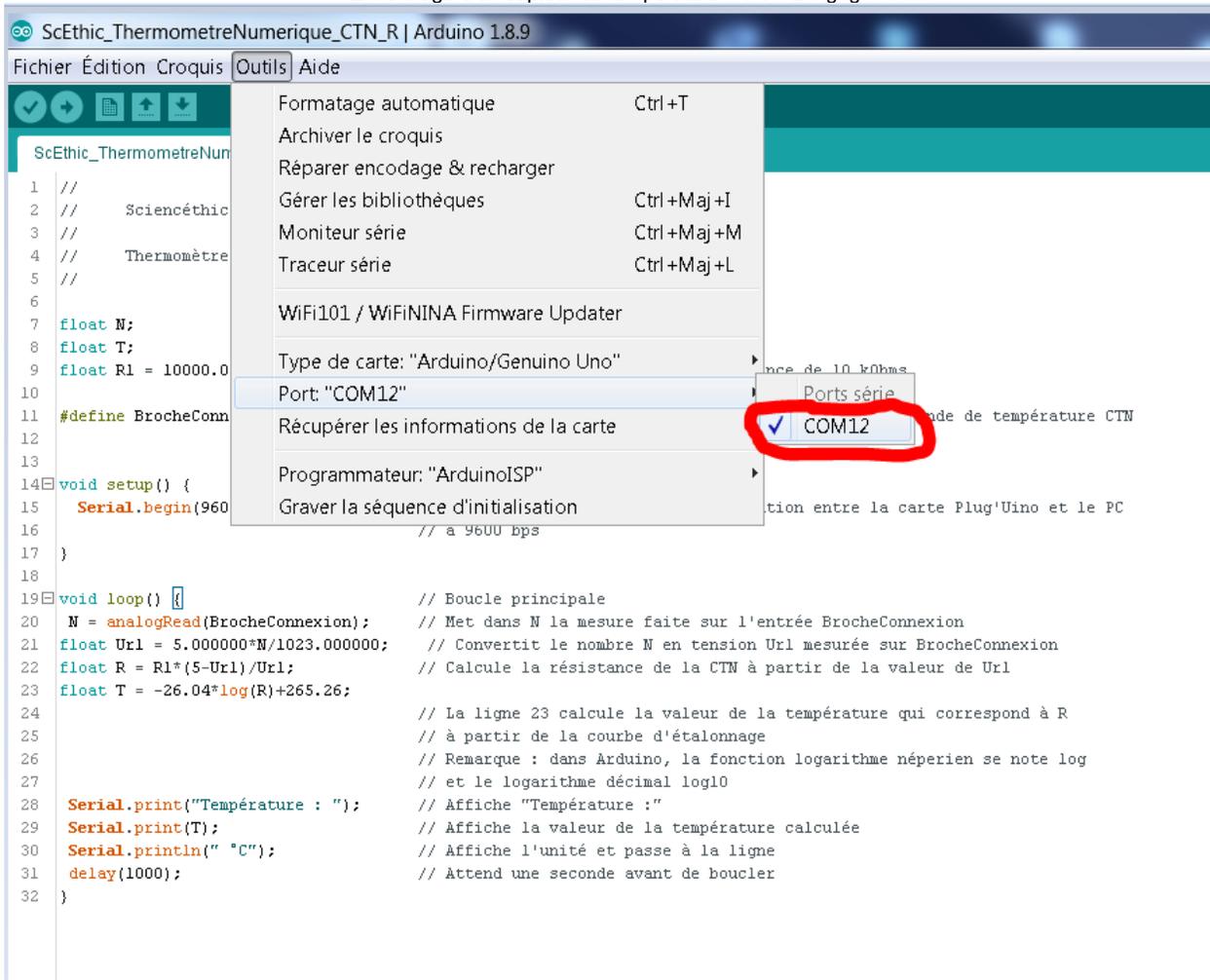
Programme complet Arduino :

```
ScEthic_ThermometreNumerique_CTN_R | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
ScEthic_ThermometreNumerique_CTN_R
1 //
2 //   Sciencéthic
3 //
4 //   Thermomètre numérique par étalonnage de résistance CTN
5 //
6
7 float N;           // Définit un nombre flottant N
8 float T;           // Définit un nombre flottant T
9 float R1 = 10000.0000; // Définit la valeur de R1 la résistance de 10 kOhms
10
11 #define BrocheConnexion A0 // Définit la broche sur laquelle sera connectée la sonde de température CTN
12 // Ici, elle sera reliée à l'entrée A0
13
14 void setup() { // Initialisation du programme
15   Serial.begin(9600); // Initialise la vitesse de communication entre la carte Plug'Uino et le PC
16                       // à 9600 bps
17 }
18
19 void loop() { // Boucle principale
20   N = analogRead(BrocheConnexion); // Met dans N la mesure faite sur l'entrée BrocheConnexion
21   float Ur1 = 5.000000*N/1023.000000; // Convertit le nombre N en tension Ur1 mesurée sur BrocheConnexion
22   float R = R1*(5-Ur1)/Ur1; // Calcule la résistance de la CTN à partir de la valeur de Ur1
23   float T = -26.04*log(R)+265.26; // La ligne 23 calcule la valeur de la température qui correspond à R
24                                   // à partir de la courbe d'étalonnage
25                                   // Remarque : dans Arduino, la fonction logarithme népérien se note log
26                                   // et le logarithme décimal log10
27
28   Serial.print("Température : "); // Affiche "Température :"
29   Serial.print(T); // Affiche la valeur de la température calculée
30   Serial.println(" °C"); // Affiche l'unité et passe à la ligne
31   delay(1000); // Attend une seconde avant de bouclier
32 }

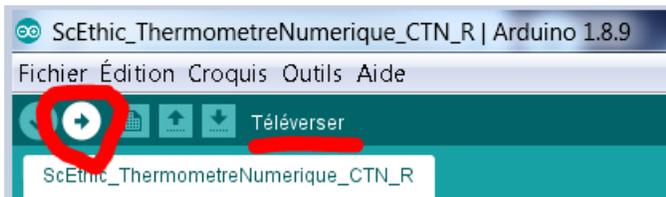
Le croquis utilise 3828 octets (11%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 220 octets (10%) de mémoire dynamique, ce qui laisse 1828 octets pour les variables locales. Le maximum est de 2048 octets.
```

Sélectionner le port série COM que vous avez utilisé pour brancher Plug'Uino®. Pour le déterminer, aller dans Outils/Port

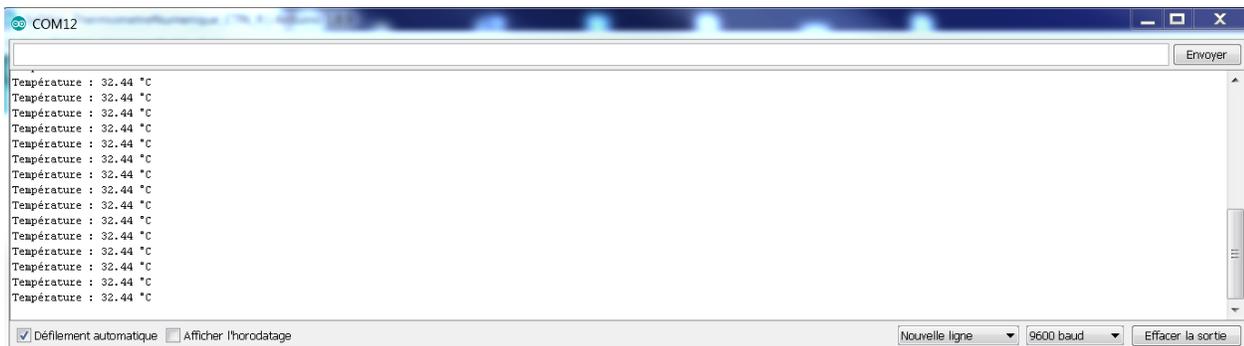
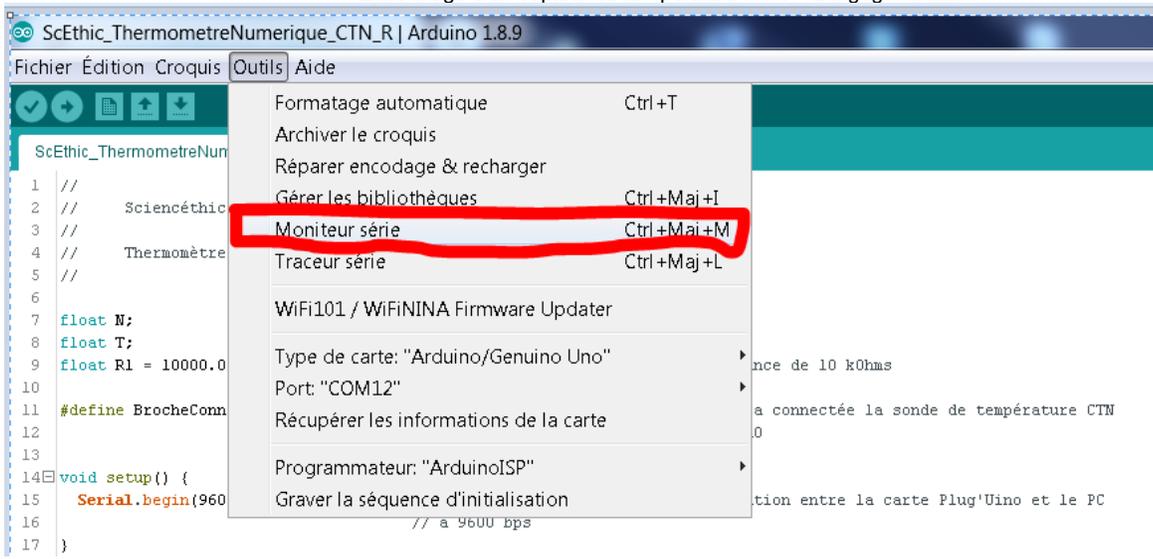
Étalonnage d'un capteur de température CTN – Langage Arduino



Téléverser ensuite le programme en cliquant sur la flèche, comme ci-dessous :



Les valeurs mesurées par le microcontrôle sont affichées dans le moniteur série :



7. Exploitation

7.1. Programmation

En fonction des capacités des différents élèves on peut leur demander diverses choses :

- à un niveau **d'initiation**, on peut enlever certains commentaires du programme et demander aux élèves à quoi correspond cette ligne ;
- à un niveau **avancé** on peut par exemple enlever les lignes 22 et 23 et demander aux élèves de rentrer les équations obtenues lors de la phase d'étalonnage.

7.2. Résultats des mesures

On peut discuter avec les élèves de la mesure de température affichée en fonction de celle mesurée par un thermomètre : l'écart est-il grand ? Varie-t-il en fonction de la plage de température sur laquelle on travaille ? Les mesures réalisées à partir du microcontrôleur sont-elles compatibles avec les grandeurs mesurées au thermomètre ?