

ARDUINO. CAPTEUR DE TEMPERATURE AVEC UNE THERMISTANCE CTN

Les CTN (coefficient de température négative) sont des thermistances dont la valeur de la résistance diminue lorsque la température augmente. La plage d'utilisation des ctn est comprise entre -200 et 1000 °C. La valeur nominale de la CTN est sa résistance à une température de 25 °C.

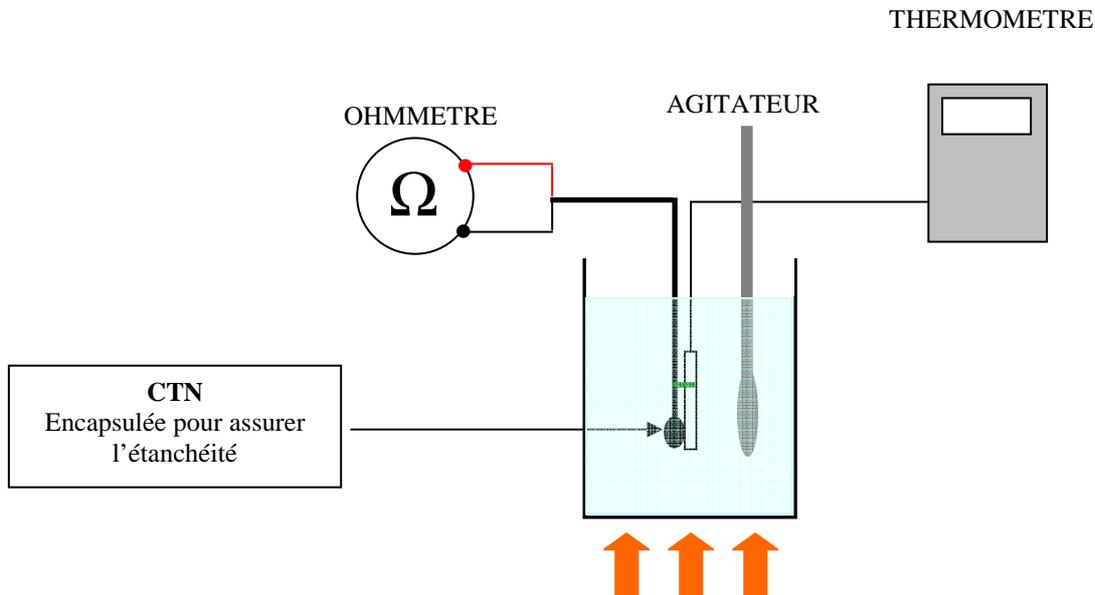
La relation de Steinhart Hart permet de déterminer la valeur de la résistance de la CTN en fonction de la température.

$$1/T = A + B \ln(R_{ctn}) + C \ln(R_{ctn})^3$$

En réalisant 3 mesures de résistance de la ctn pour 3 valeurs de température on en déduit après calcul* la valeur des 3 coefficients A, B, C.

La mesure de température doit être la plus précise possible.

Dispositif :



La CTN et la sonde du thermomètre sont réunis afin de prendre la température au même endroit du récipient. Le système de chauffage doit être muni d'un thermostat afin de stabiliser la température lors de la prise de mesure.

Mesure :

Mesure	Température (°C)	Résistance de la CTN
1	19.68	12640
2	25.92	9660
3	44.40	4640

*Source : https://fr.wikipedia.org/wiki/Relation_de_Steinhart-Hart

Calcul des trois coefficients de la relation de Steinhart Hart.

Programme Python.

```
import math

print('mesure 1')
T1 = eval(input('T1(en degres Celsius) = : '))
R1 = eval(input('R1(en ohms) = : '))

print('mesure 2')
T2 = eval(input('T2(en degres Celsius) = : '))
R2 = eval(input('R2(en ohms) = : '))

print('mesure 3')
T3 = eval(input('T3(en degres Celsius) = : '))
R3 = eval(input('R3(en ohms) = : '))

# conversion en K
T1 = T1 + 273.15
T2 = T2 + 273.15
T3 = T3 + 273.15

# 1/T
Y1 = 1/T1
Y2 = 1/T2
Y3 = 1/T3

L1 = math.log(R1)
L2 = math.log(R2)
L3 = math.log(R3)

# Calculs intermediaires
a = (L2-L3)/(L1-L2)*((L2)**3 - (L1)**3) + ((L2)**3 - (L3)**3)
b = Y2 - Y3 - ((L2-L3)/(L1-L2))*(Y1-Y2)

# Calculs des coefficients
C = b / a
B = (1/(L1-L2))*(Y1-Y2-C*((L1)**3 - (L2)**3))
A = Y1 - B*L1 - C*(L1)**3

#Affichages des coefficients
print('-----')
print('calcul des coefficients')
print('A = ', A)
print('B = ', B)
print('C = ', C)
```

Résultats:

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

```
>>>
= RESTART: C:\Users\Propriétaire\Desktop\manip en cours\ctn\ctn2020\ctn2.py =
mesure 1
T1(en degres Celsius) = : 19.68
R1(en ohms) = : 12640
mesure 2
T2(en degres Celsius) = : 25.92
```

R2(en ohms) = : 9660

mesure 3

T3(en degres Celsius) = : 44.4

R3(en ohms) = : 4640

Calcul des coefficients

A = 0.0008903731681136032

B = 0.0002685045207554958

C = -1.3467717889655259e-08

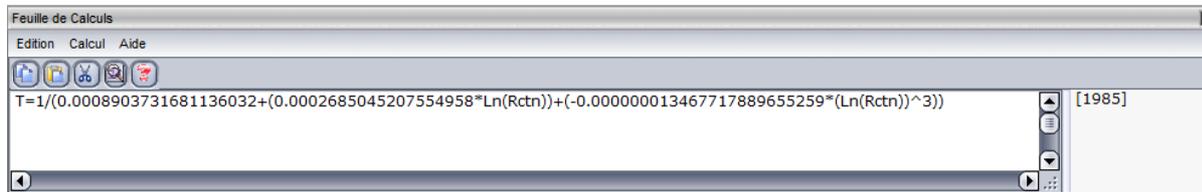
A l'aide d'un tableur (ici celui de latis-pro) on peut afficher la courbe de température en fonction de la résistance de la CTN.

Tableur.

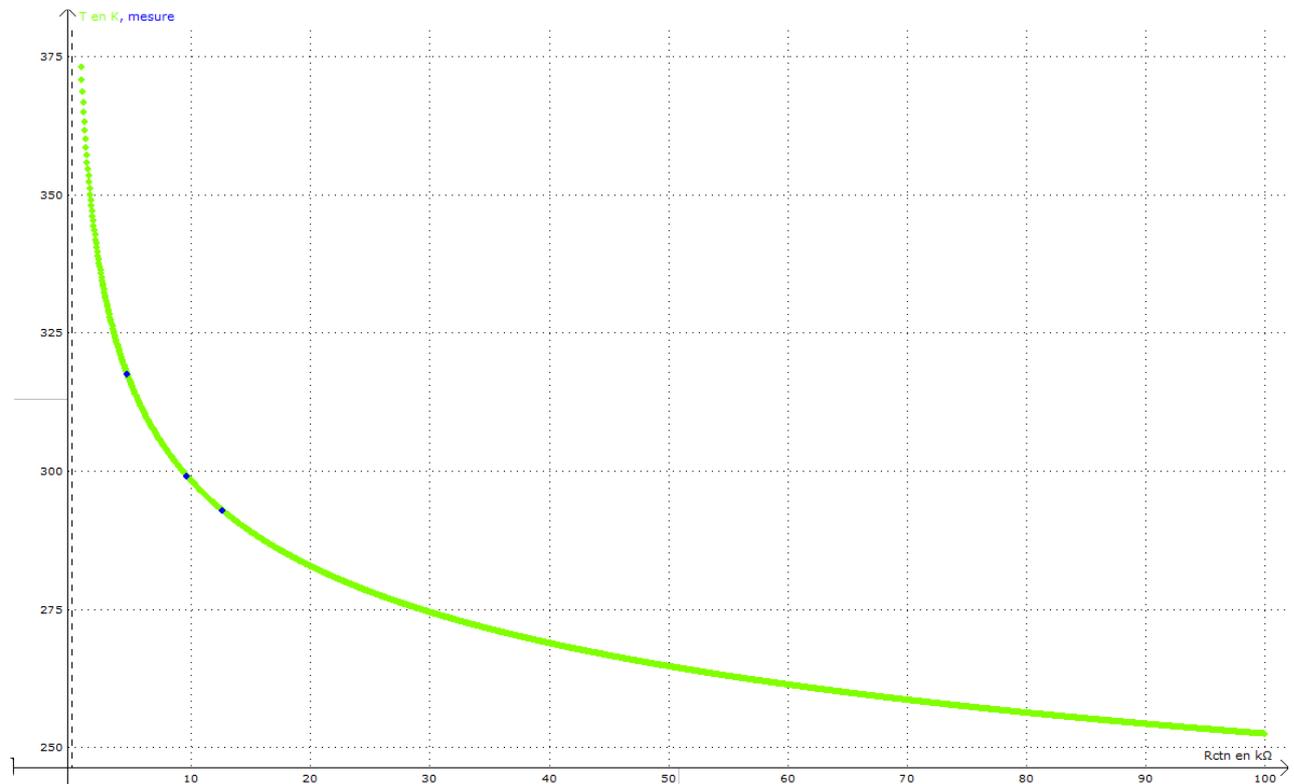
Créer une variable appelée R_{ctn} . La remplir avec une rampe de valeur initiale 800 (résistance de la ctN à environ 100°C), valeur finale 100000 (résistance de la ctN à environ -20°C) et 1985 points (incrément 50Ω).

Feuille de calcul :

On programme la feuille de calcul en remplaçant les coefficients A,B,C par les valeurs numériques précédemment calculées avec python.



Affichage de la courbe :

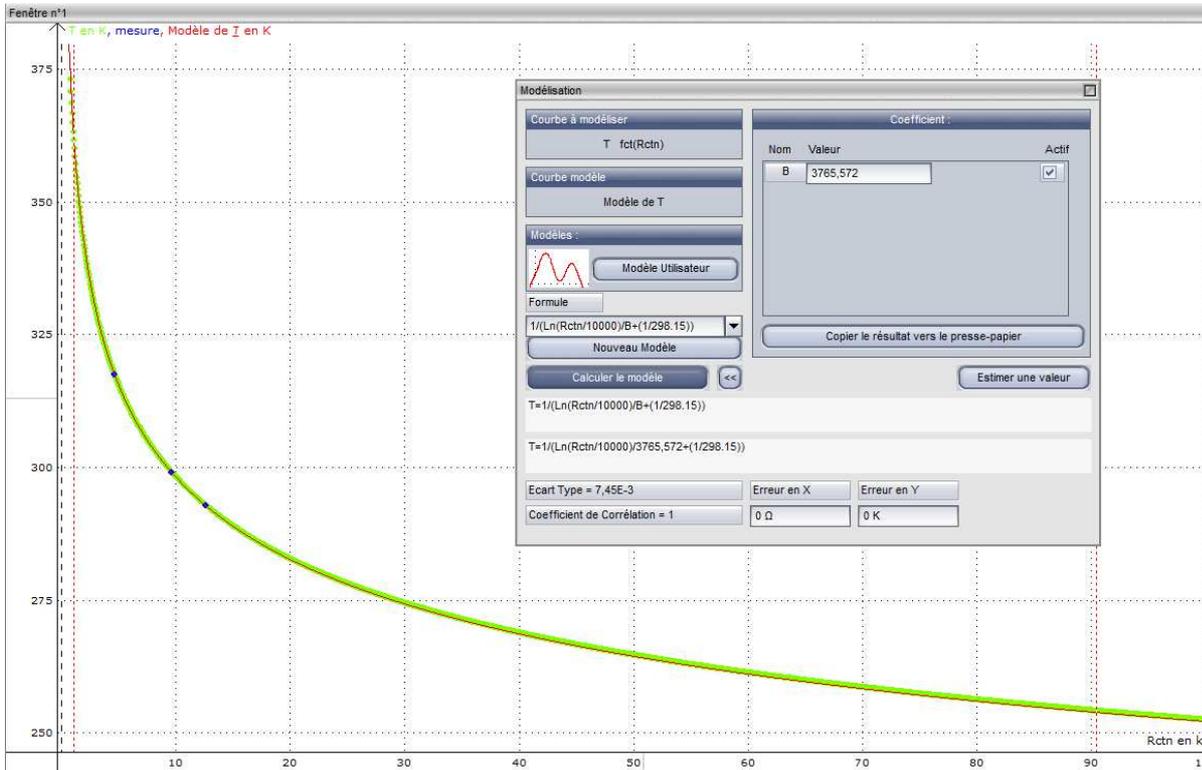


Pour une plage limitée de la température le calcul de la température est donné par la formule suivante.

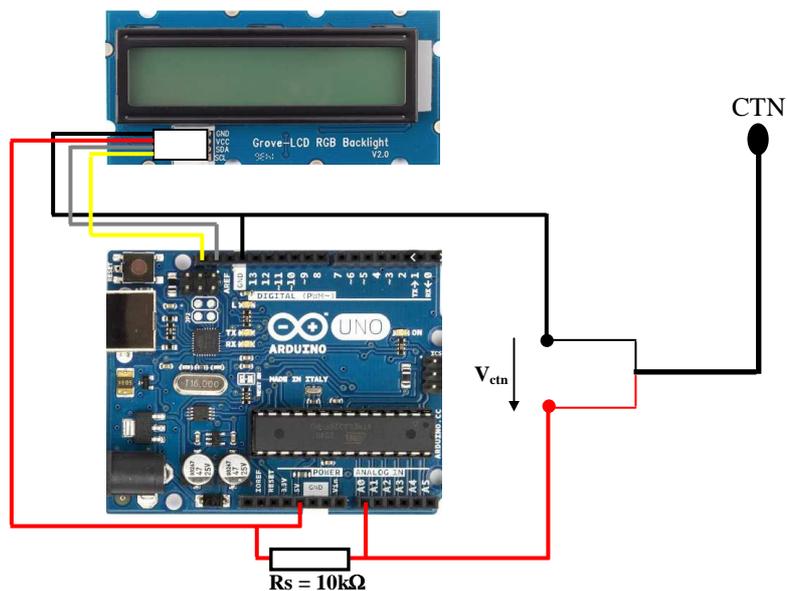
$$T = 1/((\text{Ln}(R_{\text{ctn}}/R_0)/B)+(1/T_0))$$

- R_{ctn} (Ω): valeur de la résistance de la ctn à une température donnée.
- R_0 (Ω): valeur de la résistance de la ctn à une température de 25 °C (298.15K).
- B : constante.
- T_0 (K): 298.15 K.

Modélisation de la fonction $T=f(R_{\text{ctn}})$ afin de déterminer la valeur du coefficient B.



Dispositif :



Programme arduino.

```
#include <math.h>
#include <rgb_lcd.h>

rgb_lcd lcd;

const int pinTempSensor = A0;
int Vt = 0;
double Vctn = 0 ;
double T = 0;
double B = 3765.0;           // issu de la modélisation
double R0 = 10000.0;        // résistance de la ctn à 25°C
double VIN = 5.0;
double Rs = 10000.0;        // résistance en séries avec la ctn
double Rctn =0.0;

void setup() {
  pinMode (pinTempSensor, INPUT);
  Serial.begin(9600);
  lcd.begin (16,2);
}

void loop() {
  Vt = analogRead (pinTempSensor);
  Vctn = VIN/1024*Vt;
  Rctn = Vctn/((5-Vctn)/Rs);
  T = (1/((log(Rctn/R0)/B)+(1/298.15)))-273.15;

  Serial.println (Rctn);
  Serial.println (T,1);

  lcd.clear();
  lcd.setCursor (1,0);
  lcd.print ("temperature");
  lcd.setCursor (0,1);
  lcd.print (T,1);
  lcd.print (" oC");
  delay (1000);

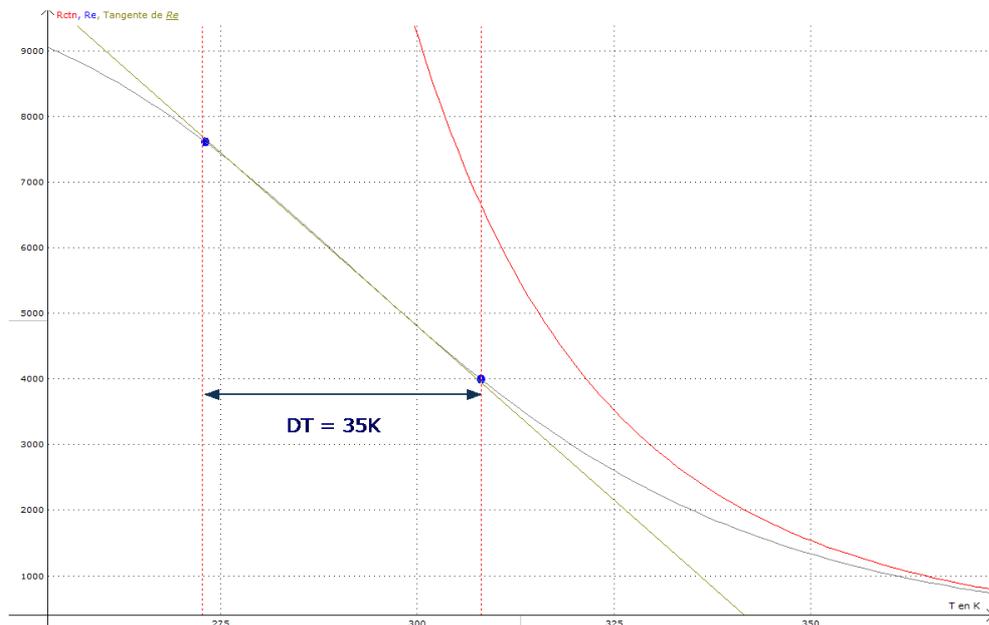
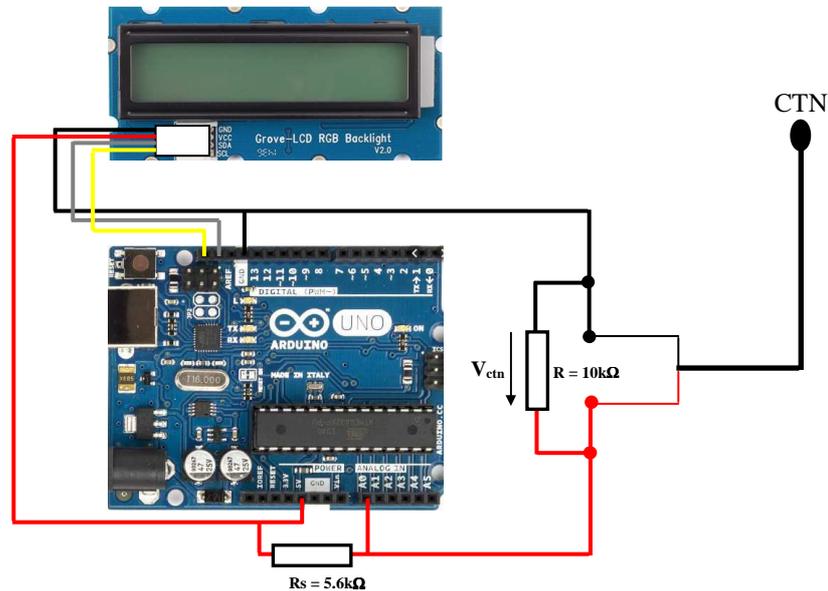
}
```

Problème de précision.

La mesure est prise en sortie d'un pont diviseur. Pour des températures élevées la valeur de la résistance de la ctn varie peu. La résolution du convertisseur A/N 10 bits étant d'environ 5mV, on observe une diminution de la précision quand la température devient trop élevée.

Linéarisation de R(T) sur une plage de mesure.

Brancher une résistance en parallèle de valeur équivalente à la valeur nominale de la ctn.



La linéarité ainsi obtenue ne concerne qu'une plage de température relativement modeste (0 à 35°C). En optimisant la valeur de la résistance R_s du pont diviseur (ici 5,6kΩ) on obtient une tension de sortie de 22,8mV/°C soit une précision d'environ 0,2°C sur cette plage.

Calcul de Rctn

Pour une température de 0°C (273.15K)

Calcul de la valeur de R_{ctn0}

$$R_{ctn} = \text{Exp}(3765 * (1/T - 1/298.15)) * 10000$$

$$R_{ctn} = 31764\Omega$$

En // avec 10k Ω

$$R_{ctn0} = 7605\Omega$$

Pour une température de 35°C (308.15K)

Calcul de la valeur de R_{ctn35}

$$R_{ctn} = 6637\Omega$$

En // avec 10k Ω

$$R_{ctn35} = 3989\Omega$$

Calcul de Vctn

Pour une température de 0°C (273.15K)

Calcul de la valeur de V_{ctn0}

$$V_{ctn0} = (5 * 7605) / (7605 + 5600)$$

$$V_{ctn0} = 2.880V$$

Pour une température de 35°C (308.15K)

Calcul de la valeur de V_{ctn35}

$$V_{ctn35} = (5 * 3989) / (3989 + 5600)$$

$$V_{ctn0} = 2.080V$$

Calcul de T

Nous obtenons une variation de 800mV pour 35°C soit 22,8mV par degrés

$$T = -(V_{ctn} - 2.88) / 0.0228$$

Le programme ci-dessous prend en compte cette nouvelle structure. Le programme est ainsi très largement simplifié. Cependant la plage de mesure reste faible.

Programme arduino

```
#include <rgb_lcd.h>
```

```
rgb_lcd lcd;
```

```
const int pinTempSensor = A0;
```

```
int Vt = 0;
```

```
double Vctn = 0 ;
```

```
double T = 0;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  lcd.begin (16,2);
```

```
}
```

```
void loop() {
```

```
  Vt = analogRead (pinTempSensor);
```

```
  Vctn = 5.000/1024*Vt;
```

```
  T = -(Vctn - 2.88)/0.0228;
```

```
  Serial.println (Vctn,3);
```

```
  lcd.clear();
```

```
    lcd.setCursor (1,0);
```

```
    lcd.print ("temperature");
```

```
    lcd.setCursor (0,1);
```

```
    lcd.print (T,1);
```

```
    lcd.print (" oC");
```

```
  delay (1000);
```

```
}
```