

On se propose de tester le modèle théorique établi lors de notre première activité.

Notions et contenus	Capacités exigibles
Loi phénoménologique de Newton, modélisation de l'évolution de la température d'un système au contact d'un thermostat.	Effectuer un bilan d'énergie pour un système incompressible échangeant de l'énergie par un transfert thermique modélisé à l'aide de la loi de Newton fournie. Établir l'expression de la température du système en fonction du temps. <i>Suivre et modéliser l'évolution de la température d'un système incompressible.</i>

On a vu dans la première activité que l'on peut exprimer la variation de température d'un corps par la relation :  $\frac{dT}{dt} = -k(T - T_{env})$

La résolution de cette équation du premier ordre donne :

$$T(t) = (T_0 - T_{env})e^{-kt} + T_{env} \quad \text{ou} \quad T_{env}(1 - e^{-kt}) + T_0e^{-kt}$$

On peut étudier le temps de réponse à 5 %, c'est-à-dire le temps que met la température pour atteindre 95% de sa valeur finale.  $T_{5\%} = 3 \tau$

On peut aussi étudier le temps de montée qui est le temps pour que la température passe de 10% à 90% de sa variation.  $T_m = 2,2 \tau$

$$\tau = \frac{m \cdot c}{S \cdot h}$$

## Matériel :

Calorimètre contenant de l'eau à température constante (thermostat)

Eau chaude – eau froide

On peut mesurer la température avec un :

Thermocouple type K

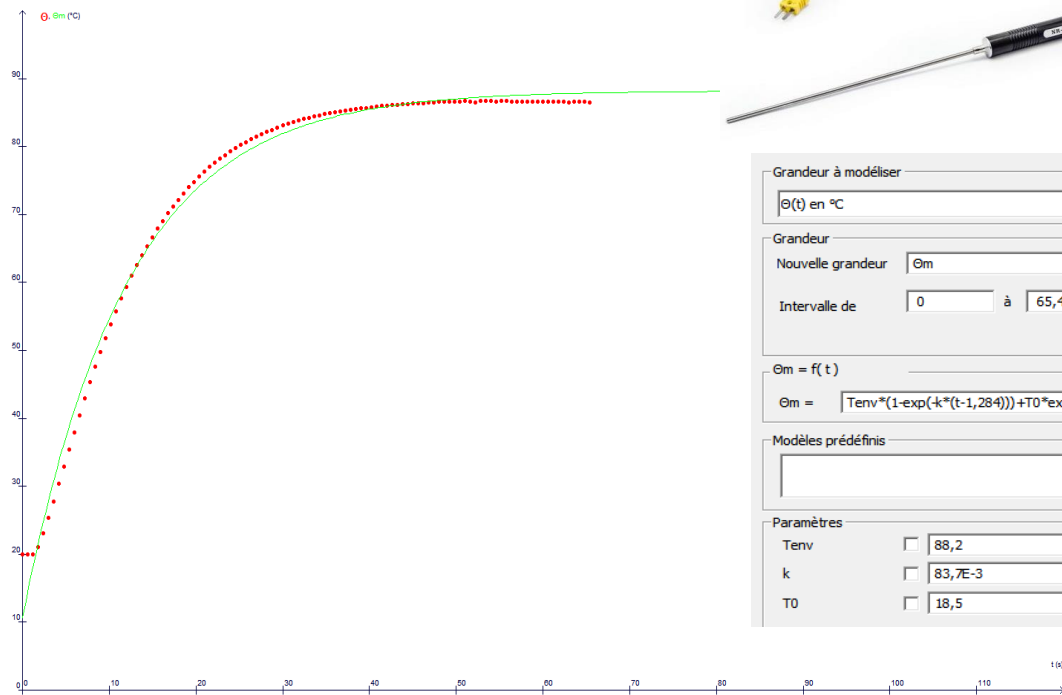
Thermistance CTN

Circuit intégré LM 335 (mesure en K) ou LM 35 (mesure en °C)

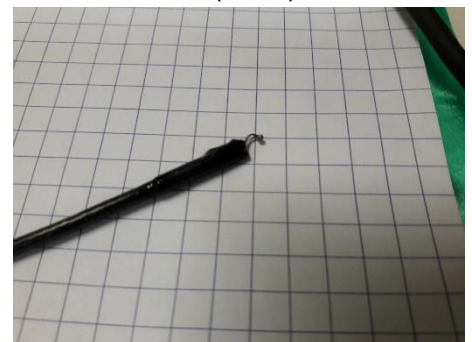
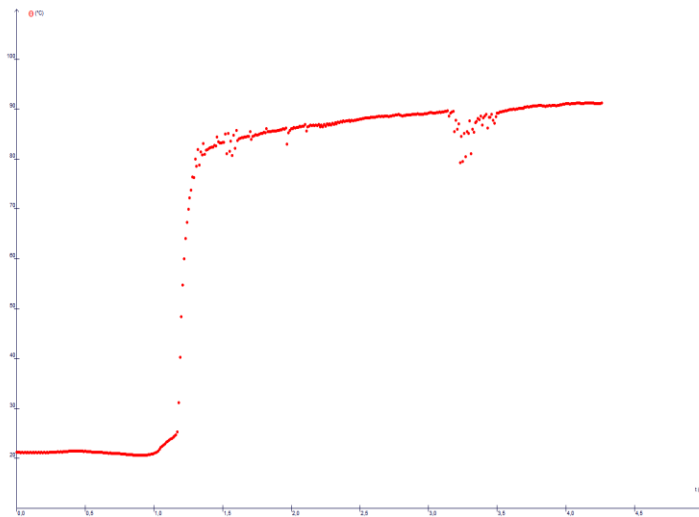
Il faudra s'affranchir du temps de réponse du capteur devant le temps de réponse du système  
La grande difficulté de cette relation (qui semble) simple est la détermination du coefficient k.

## Exemples de mesures :

Sonde thermocouple type K gaine inox : passe de l'air ambiant (19,9 °C) à l'eau chaude (86°C)



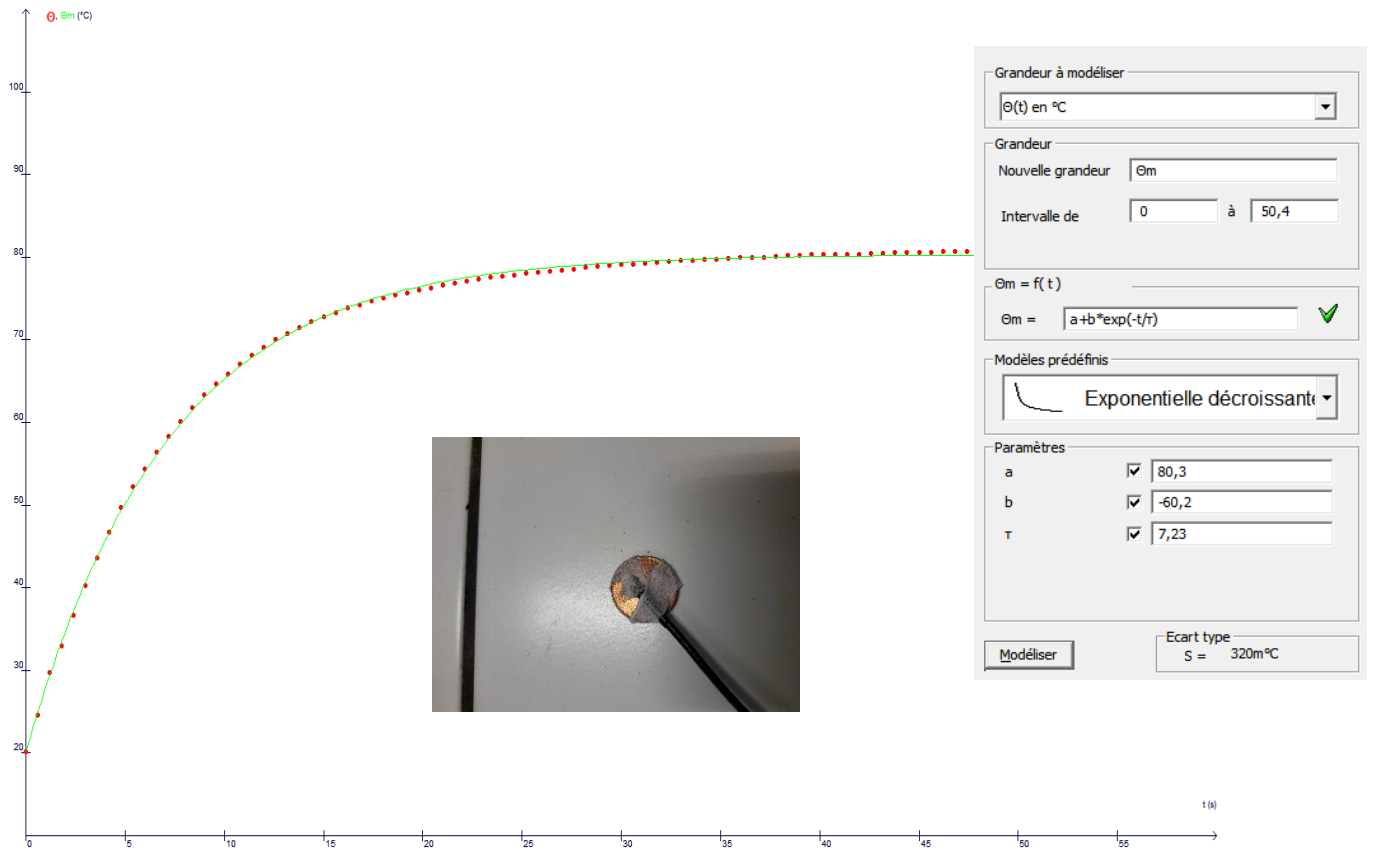
Sonde thermocouple type K non gainée : passe de l'air ambiant (21 °C) à l'eau chaude (90°C)



Le logiciel refuse de modéliser mais on voit que le temps de réponse est très court

sonde non étanche, j'ai dû mettre de la gaine thermorétractable

Sonde thermocouple type K non gainée fixée sur une pièce de 5 centimes pour reprendre la manipulation de Luc Angrand : passe de l'air ambiant (21 °C) à l'eau chaude (80°C)

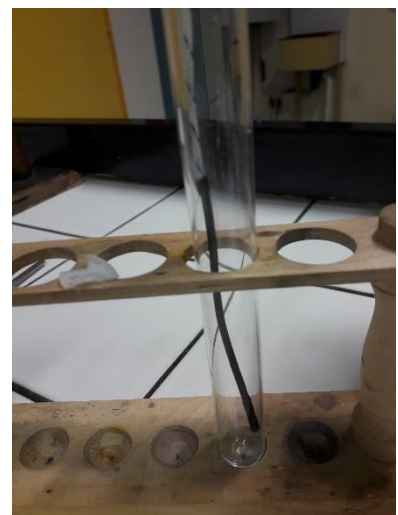
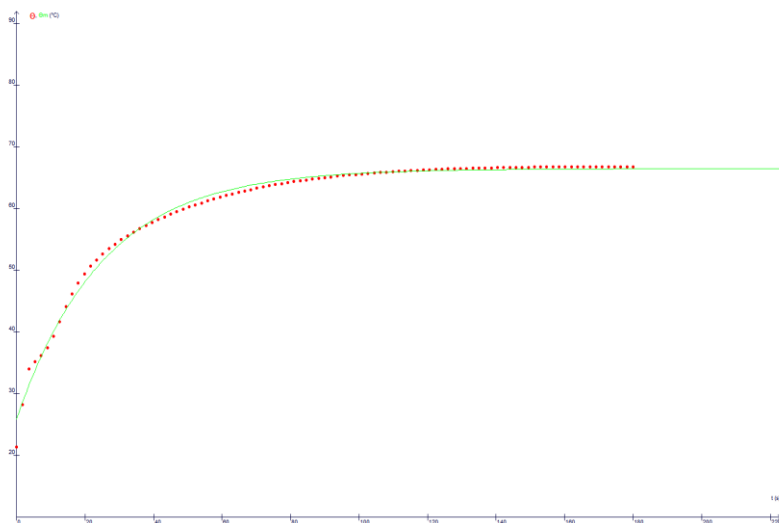


Difficile de modéliser avec le logiciel ... mais cela donne  $T(t) = (T_0 - T_{env})e^{-kt} + T_{env}$

Avec  $T_0 - T_{env} = b = -60,2$  °C       $T_{env} = a = 80,3$  °C    et  $k = \frac{1}{\tau}$

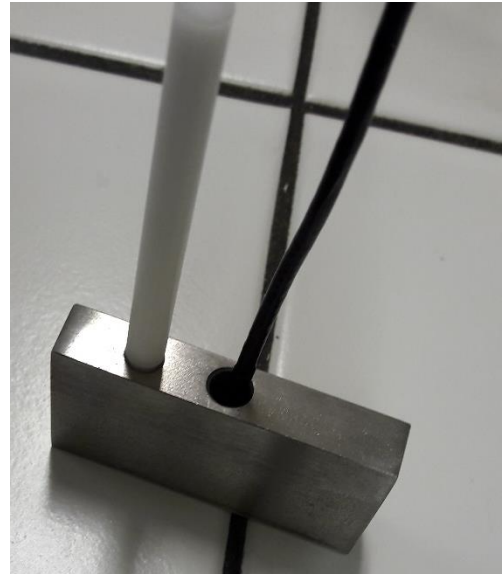
Le temps de réponse est beaucoup plus grand avec la pièce de monnaie que sans. On peut donc négliger le temps de réponse de la sonde.

Sonde K dans un tube à essai avec bouchon (rempli d'air) plongé dans l'eau chaude (90°C)



Résultat peu exploitable, la température de l'air ne s'équilibre pas avec celle de l'eau (ou durée trop longue). On reste sur un système du premier ordre.

Bloc d'acier chauffé puis se refroidit à l'air. Il reste de l'eau dans la fente où l'on place le thermomètre, faussant la lecture. Il faudrait à nouveau fixer la thermistance sur la face du bloc et le chauffer sur plaque chauffante. J'ai manqué de temps pour tester à nouveau.



### Avec Arduino et Python

Programme Arduino :

```
#include <LiquidCrystal.h>
// initialisation, on définit les ports pour RS, E et D4 à D7
LiquidCrystal lcd(12, 11, 4, 5, 6, 7);
float A = 1.009249522e-03, B = 2.378405444e-04, C = 2.019202697e-07;
float T,logRt,Tf,Tc;
float Thermistor(int Vo) {
  logRt = log(10000.0*((1024.0/Vo-1)));
  T = (1.0 / (A + B*logRt + C*logRt*logRt*logRt)); // We get the temperature value in Kelvin from this Stein-Hart
equation
  Tc = T - 273.15;          // Convert Kelvin to Celcius
  return T;
}
void setup(){
  lcd.begin(16,2);
  lcd.clear();
  Serial.begin(9600);
}
void loop()
{
  lcd.setCursor(0,0);
  lcd.print("Temp:");
  lcd.print((Thermistor(analogRead(0))));
  lcd.print("k ");
  lcd.setCursor(0,1);
  lcd.print((Tc));
  lcd.print(" C;");
  // print the temperature in celsius to the serial port
  Serial.println(Tc);
  delay(800);
}
```

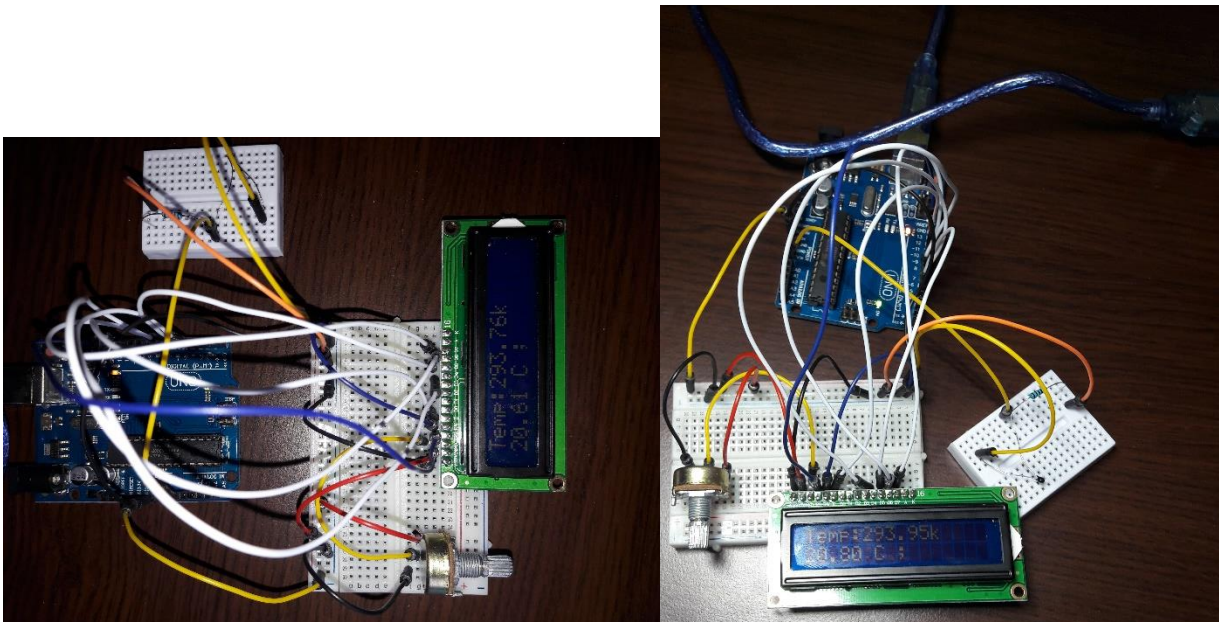
Programme Python

```

import serial
import matplotlib.pyplot as plt
import time
valeurs=[]
temps=[]
serie=serial.Serial('COM3',9600)      # ouvre une liaison serie en 9600bps
# adapter /dev/ttyACM0 au systeme et port serie
plt.style.use('bmh')
plt.ylabel("valeur")
plt.xlabel("temps en s")
plt.ion()                             # on entre en mode interactif
start=time.time()                    # mesure de l'instant initial
i=0
while (i<100): # pour 100 mesures
    mesure= float(serie.readline()) # lit la donnee sur la liaison serie
    valeurs.append(mesure)          # ajout de mesure a la liste des valeurs
    instant=time.time()-start      # calcul du temps ecoule depuis l'instant initial
    temps.append(instant)          # ajout de instant a la liste des temps
    print(mesure,instant)         # affiche dans la console les coordonnees du point
    plt.plot(temps,valeurs,marker='o') # trace la courbe
    plt.draw()                     # affiche la courbe en mode interactif
    i=i+1
plt.ioff()                          # on quitte le mode interactif pour rendre la main a l'utilisateur sur la courbe
plt.show()                          # afficher la courbe

```

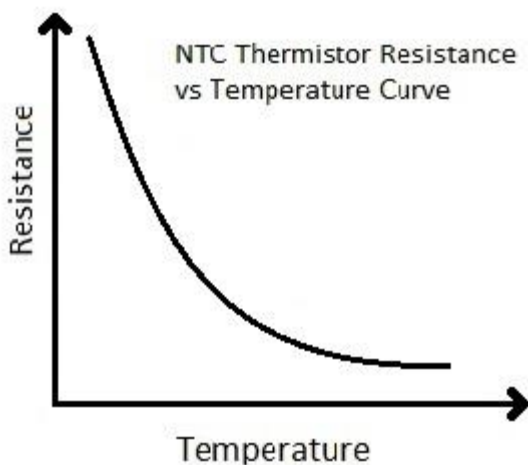
la thermistance est chauffée avec un sèche-cheveu, lorsque le sèche-cheveu est coupé, l'acquisition commence. Le délai entre 2 mesures est de 800ms (dans le code arduino) et il y a 100 points de mesures (dans python)



**Remarque, avec un LM35 (ou LM335), la conversion en °C est plus simple ainsi que l'étalonnage. Modéliser la CTN devient inutile.**

## Thermistor

The key component in this circuit is Thermistor, which has been used to detect the rise in temperature. **Thermistor is temperature sensitive resistor**, whose resistance changes according to the temperature. There are two types of thermistor NTC (Negative Temperature Co-efficient) and PTC (Positive Temperature Co-efficient), we are using a NTC type thermistor. NTC thermistor is a resistor whose resistance decreases as rise in temperature while in PTC it will increase the resistance as rise in temperature.



### Calculating Temperature using Thermistor: We

know from the Voltage divider circuit that:

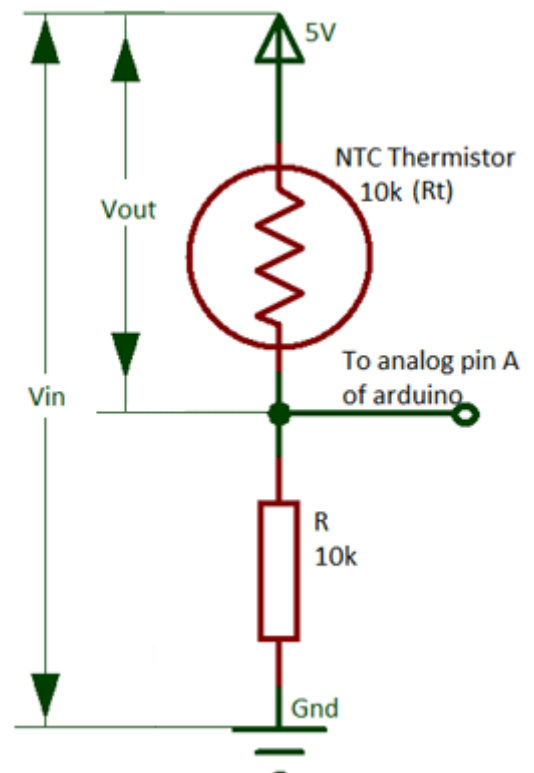
$$V_{out} = (V_{in} * R_t) / (R + R_t)$$

So the value of  $R_t$  will be:

$$R_t = R (V_{in}/V_{out}) - 1$$

Here,  $R_t$  will be the resistance of thermistor and  $R$  will be 10k ohm resistor. You can also calculate the values from this [voltage divider calculator](#).

This equation is used for the calculation of thermistor resistance from the measured value of output voltage  $V_o$ . We can get the value of Voltage  $V_{out}$  from the ADC value at pin A0 of Arduino as shown in the Arduino Code given below.





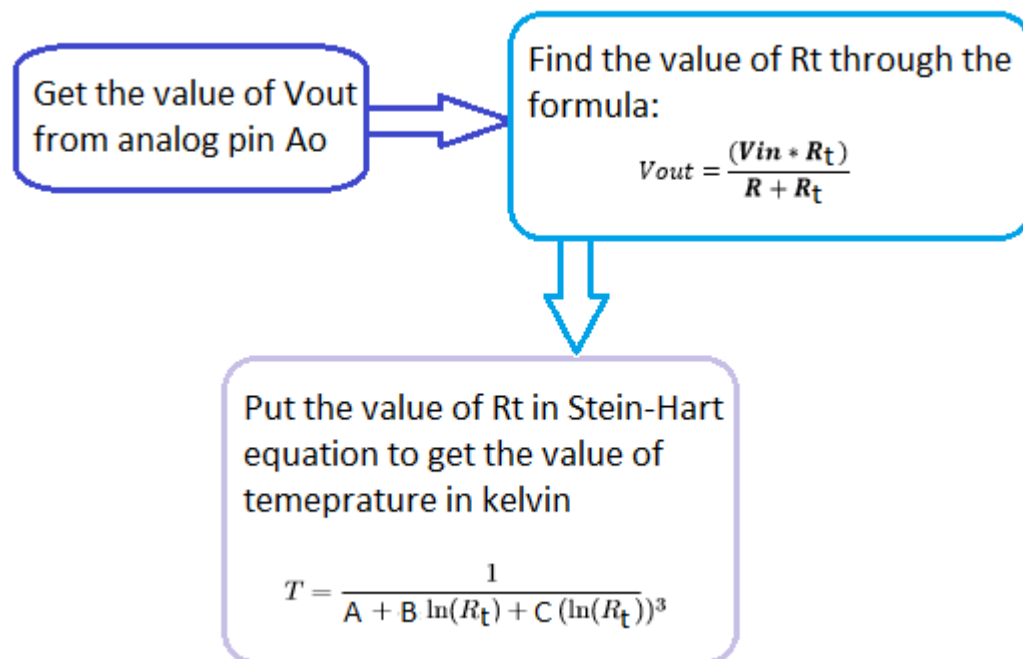
## Calculation of Temperature from the thermistor resistance:

Mathematically the thermistor resistance can only be compute with the help of the [Stein-Hart equation](#).

$$T = 1 / (A + B \ln(R_t) + C \ln(R_t)^3)$$

Where, A, B and C are the constants,  $R_t$  is the thermistor resistance and  $\ln$  represents log.

**The constant value for the thermistor used in the project is**  $A = 1.009249522 \times 10^{-3}$ ,  $B = 2.378405444 \times 10^{-4}$ ,  $C = 2.019202697 \times 10^{-7}$ . These constant values can be obtained from [the calculator here](#) by entering the three resistance values of thermistor at three different temperatures. You can either get these constant values directly from the datasheet of the Thermistor or you can get three resistance values at different temperature and get the Constants values using the given calculator.



So, for calculating the temperature we need the value of thermistor resistance only. After getting the value of  $R_t$  from the calculation given above put the values in the Stein-hart equation and we will get the value of temperature in the unit kelvin. As there is minor change in the output voltage cause change in the temperature.

