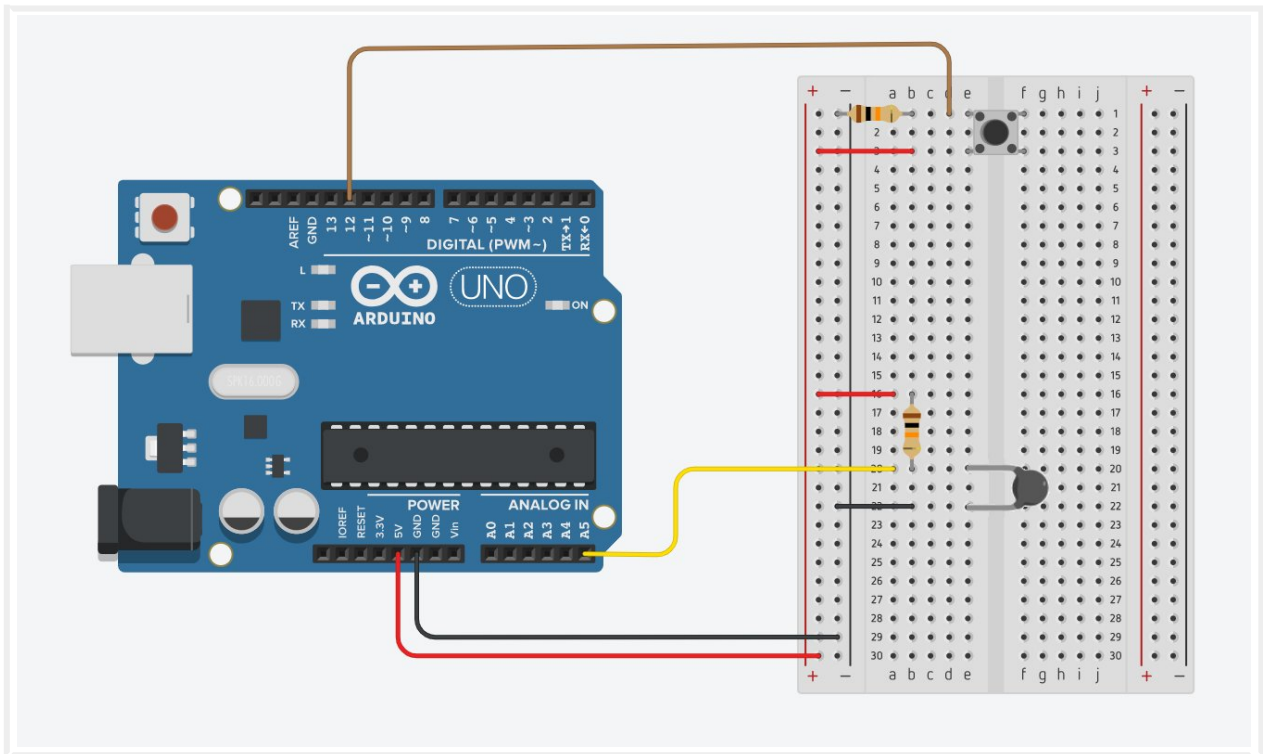


Thermistances CTN

(Mesure de températures avec une thermistance CTN)



. Liste des composants

- . 1 thermistance CTN
- . 2 résistances de 10 k Ω (résistance du bouton poussoir et de la CTN)
- . 1 bouton poussoir
- . 1 plaque d'essais
- . Fils de connexion

. Objectif

Pour mesurer une température avec une thermistance CTN, il faut connaître ses grandeurs caractéristiques. Le plus souvent, le constructeur fournit les valeurs suivantes :

- La valeur de sa résistance R_0 (résistance nominale en Ω) à la température de référence $T_0 = 25$ °C (298,15 K)
- La valeur de β (en K)
- La plage de température pour laquelle la relation entre la température T (en K) et R_T , la résistance (en ohms) de la CTN à cette température, est vérifiée :

$$\frac{1}{T} = \frac{1}{\beta} \ln\left(\frac{R_T}{R_0}\right) + \frac{1}{T_0}$$

On en déduit :

$$T \text{ (en } ^\circ\text{C)} = \frac{1}{\frac{1}{\beta} \ln\left(\frac{R_T}{R_0}\right) + \frac{1}{T_0}} - 273,15$$

Par exemple, le capteur de température Groove est basé sur une thermistance CTN :

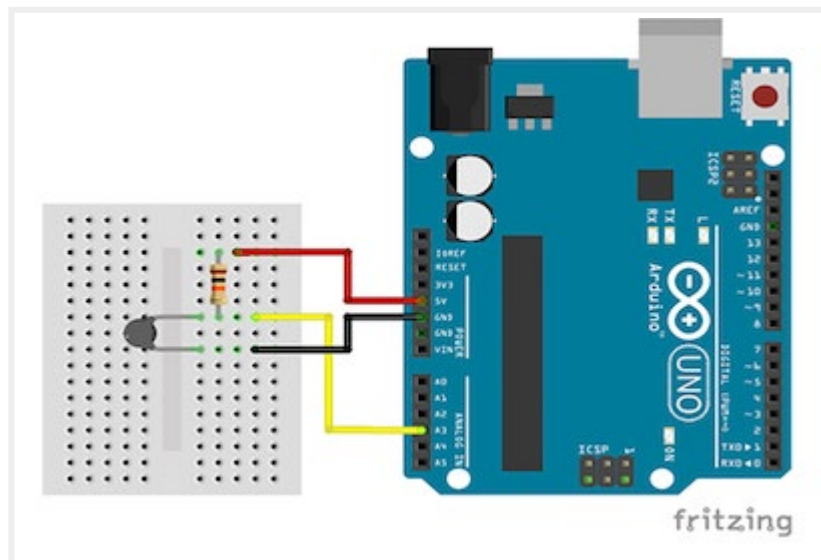


Le constructeur donne les informations suivantes :

- . Voltage : 3.3 ~ 5V
- . Zero power resistance : 100 K Ω
- . Resistance Tolerance : $\pm 1\%$
- . Operating temperature range : -40 ~ +125 $^\circ\text{C}$
- . Nominal B-Constant: 4250 ~ 4299K

En l'absence de ces grandeurs caractéristiques, il faut procéder à un étalonnage, afin de les déterminer expérimentalement.

L'objectif de l'activité est d'écrire un programme généraliste permettant de mesurer une température avec une thermistance CTN quelconque, avec ce montage :



Le code demande, à l'initialisation du programme, de renseigner les valeurs de T_0 , R_0 et β afin de pouvoir calculer la température à partir de la mesure de la résistance de la CTN.

. Le programme

Voici le code de l'activité :

```
Thermistances_CTN
// Inclusion des librairies

#include <math.h>

// Déclaration des constantes et variables

const int PinCTN = 5;
const int PinButton = 12;

int ValCTN = 0;
float Temp = 0.0;
float OldTemp = 0.0;
float Rt = 0.0;
float Vctn = 0.0;
int TempRef = 0;
long Ro = 0;
int B = 0;

int ValButton = 0;
int OldValButton = 0;
int State = 0;
int OldState = 0;

// Initialisation des entrées et sorties

void setup() {
    Serial.begin(9600);
    pinMode(PinButton, INPUT);
}
```

```

Serial.print("Veuillez saisir la temperature de reference ");
Serial.println("(valeur entre 1 et 100 degre Celsius).");
while(TempRef<1 || TempRef>100)
{
    TempRef=Serial.parseInt();
}
Serial.print("Temperature de reference (degre Celsius) : ");
Serial.println(TempRef);
Serial.println("");

Serial.print("Veuillez saisir la valeur de la resistance nominale ");
Serial.println("(valeur entre 1 et 200000 ohms).");
while(Ro<1 || Ro>200000)
{
    Ro=Serial.parseInt();
}
Serial.print("Resistance nominale (ohms) : "); Serial.println(Ro);
Serial.println("");

Serial.print("Veuillez saisir la constante B ");
Serial.println("(valeur entre 1 et 10000 K).");
while(B<1 || B>10000)
{
    B=Serial.parseInt();
}
Serial.print("Constante B (K) : "); Serial.println(B);
Serial.println("");

Serial.println("Appuyez sur le bouton poussoir pour commencer les mesures.");
}

// Fonction principale en boucle

void loop() {
    ValButton = digitalRead(PinButton);
    delay(10);

    if ((ValButton == HIGH) && (OldValButton == LOW))
    {
        State=1-State;
    }
    OldValButton = ValButton;
}

```

```

if (State==1)
{
  if (OldState == 0)
  {
    Serial.println("Mesure de la temperature en cours.");
    Serial.println("");
    Serial.println ("Resistance CTN en Ohms; Temperature en degre Celsius:");
    OldState=1;
  }

  ValCTN = analogRead(PinCTN);
  Vctn = (ValCTN/1023.0)*5.0;
  Rt = 10000*Vctn/(5-Vctn);
  Temp = 1.0/(log(Rt/Ro)/B+1/(TempRef+273.15))-273.15;

  if (OldTemp != Temp)
  {
    Serial.print(Rt,1);
    Serial.print(" ; ");
    Serial.println(Temp,1);
    OldTemp = Temp;
  }

  delay(100);
}
else
{
  if (OldState == 1){
    Serial.println("Fin des mesures.");
    OldState = 0;}
}
}

```

Déroulement du programme :

– 1. Inclusion des librairies

Pour effectuer les calculs de température, le code nécessite l’importation de la librairie ”**math.h**”.

– 2. Déclaration des constantes et variables :

- . **const int PinCTN = 5** (constante nombre entier correspondant à la broche de la CTN)
- . **const int PinButton = 12** (constante nombre entier correspondant à la broche du bouton poussoir)
- . **int ValCTN = 0** (variable nombre entier pour stocker la valeur de la broche de la CTN)
- . **float Temp = 0.0** (variable nombre décimal pour stocker la valeur de la température)
- . **float OldTemp = 0.0** (variable nombre décimal pour stocker la valeur précédente de la température)
- . **float Rt = 0.0** (variable nombre décimal pour stocker la valeur de la résistance de la CTN)
- . **float Vctn = 0.0** (variable nombre décimal pour stocker la valeur de la tension de la broche de la CTN)

- . **int TempRef = 0** (variable nombre entier pour stocker la valeur de la température de référence)
- . **long Ro = 0** (variable nombre entier long pour stocker la valeur de la résistance de la CTN à la température de référence)
- . **int B = 0** (variable nombre entier pour stocker la valeur de la constante β)
- . **int ValButton = 0** (variable nombre entier pour stocker la valeur de la broche bouton poussoir)
- . **int OldValButton = 0** (variable nombre entier pour stocker la valeur précédente de la broche du bouton poussoir)
- . **int State = 0** (variable nombre entier correspondant à l'action à effectuer)
- . **int OldState = 0** (variable nombre entier correspondant à l'action effectuée précédemment)

– 2. Initialisation des entrées et sorties :

- . **Initialisation de la liaison série à un débit de 9600 bauds,**
- . **Initialisation de la broche du bouton poussoir en entrée,**
- . **Saisie des valeurs de T_0 , R_0 et β afin de pouvoir calculer la température à partir de la mesure de la résistance de la CTN.**

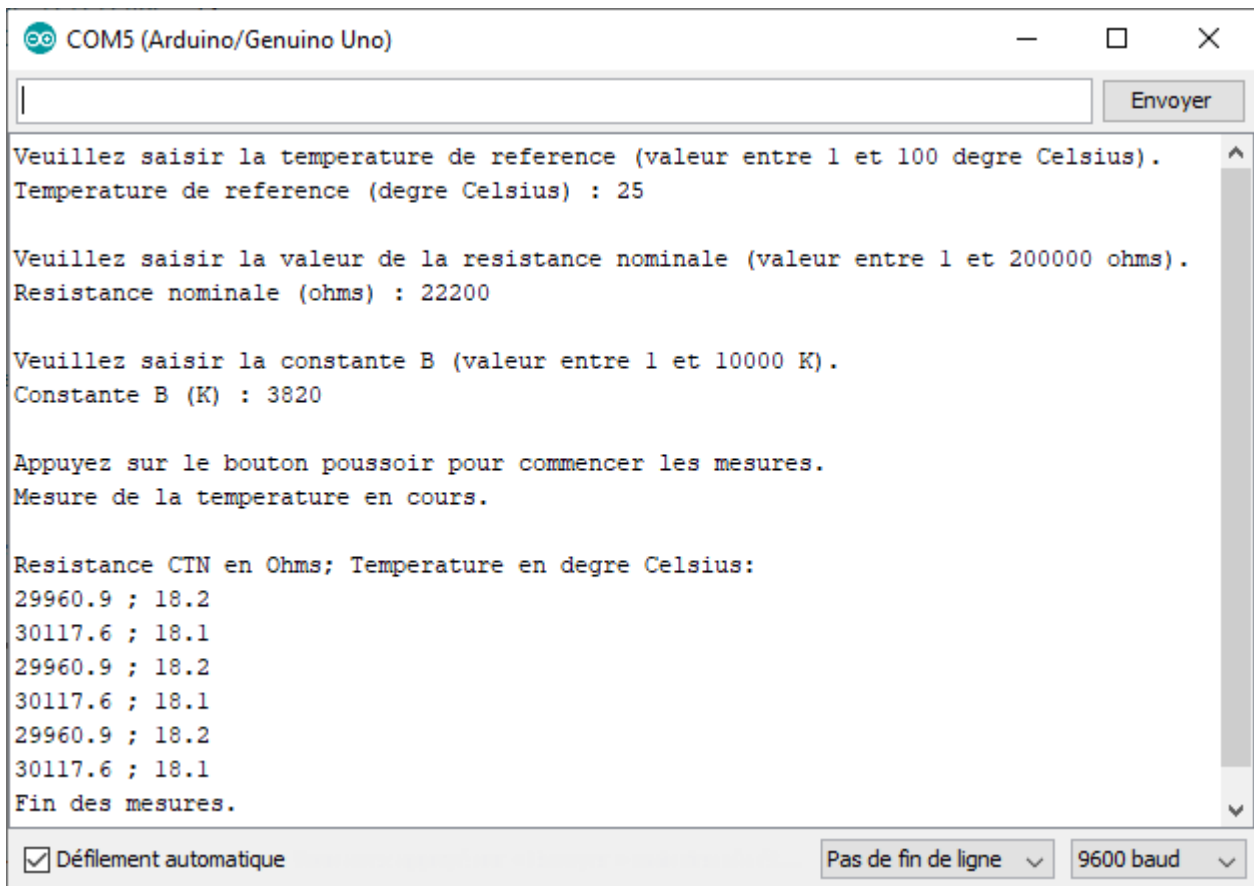
Remarque :

La fonction **"parseInt()"** de la classe **"Serial"** retourne le premier entier long du tampon de la liaison série. Les caractères lettres ou le signe **"-"** sont ignorés. Au-delà, d'un certain temps (par défaut, 1 s), la fonction se termine et retourne **"0"** si le tampon est vide ou ne contient pas de nombre.

– 3. Fonction principale en boucle :

- . **Début des mesures en appuyant sur le bouton poussoir :**
 - > **Lecture de la valeur de la broche de la CTN**
 - > **Calcul de la résistance de la CTN**
 - > **Calcul de la température**
 - > **Affichage de la valeur de la résistance et de la température dans le moniteur série si la valeur de la température est différente de celle mesurée précédemment**
- . **Fin des mesures en appuyant de nouveau sur le bouton poussoir**

Résultats dans le moniteur série :



The screenshot shows the 'COM5 (Arduino/Genuino Uno)' serial monitor window. It contains a text input field at the top with an 'Envoyer' button. The main area displays the program's output, which includes prompts for reference temperature, nominal resistance, and constant B, followed by a series of measurements and a final status message. At the bottom, there are checkboxes for 'Défilement automatique' and dropdown menus for 'Pas de fin de ligne' and '9600 baud'.

```
COM5 (Arduino/Genuino Uno)

| Envoyer

Veuillez saisir la temperature de reference (valeur entre 1 et 100 degre Celsius).
Temperature de reference (degre Celsius) : 25

Veuillez saisir la valeur de la resistance nominale (valeur entre 1 et 200000 ohms).
Resistance nominale (ohms) : 22200

Veuillez saisir la constante B (valeur entre 1 et 10000 K).
Constante B (K) : 3820

Appuyez sur le bouton poussoir pour commencer les mesures.
Mesure de la temperature en cours.

Resistance CTN en Ohms; Temperature en degre Celsius:
29960.9 ; 18.2
30117.6 ; 18.1
29960.9 ; 18.2
30117.6 ; 18.1
29960.9 ; 18.2
30117.6 ; 18.1
Fin des mesures.

☒ Défilement automatique
Pas de fin de ligne
9600 baud
```