

1

Utilisation des informations de la datasheet

On utilise par exemple la CTN marquée TD320 dont voici [la datasheet](#).

Voici l'extrait important à relever dans la datasheet:

■ TDCSeries-5Φ

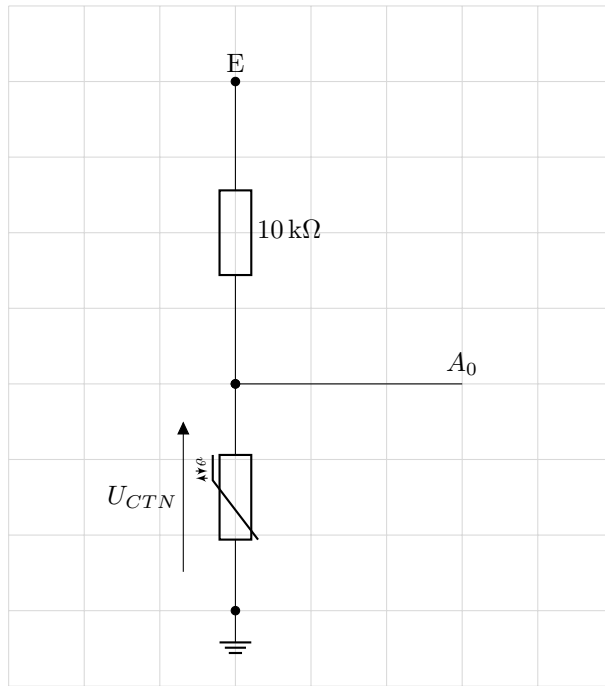
Disc Diameter	Part No.	Zero Power Resistance at 25°C	β Constant 25-50	Thermal Dissipation Constant Approx.	Thermal Time Constant Approx.	Max. Power Rating at 25°C	Operating Temperature Range	UL E482628
		R25 (Ω)	(K)	δ (mW/°C)	τ (Sec.)	(mW)	(°C)	
5Φ	TDC05C315□-5	15K	4150	6	20	450	-20 ~ +125	V
5Φ	TDC05C320□-5	20K	4150	6	20	450	-20 ~ +125	V
5Φ	TDC05C322□-5	22K	4200	6	20	450	-20 ~ +125	V

Figure 1.1: Extrait datasheet

Pour cette CTN on a $\beta = 4150$ correspondant à une résistance $R_{25} = 20\,000\,\Omega$ pour une température $T_{25} = 298\text{ K}$

On intègre ensuite le composant dans le montage suivant qui permet de transformer la résistance de sortie du capteur en une tension d'entrée dans U_{A0} dans le microcontrôleur.

1.1 Intégration du composant dans le montage



Questions sur le pont diviseur

1. Exprimer la loi d'Ohm aux bornes de R et de R_{CTN} , puis en déduire l'expression de i .

2. Exprimer la loi d'Ohm aux bornes R_{CTN} et en déduire U_{A0} la tension délivrée dans le micro-contrôleur.

3. En déduire l'expression de R_{CTN} .

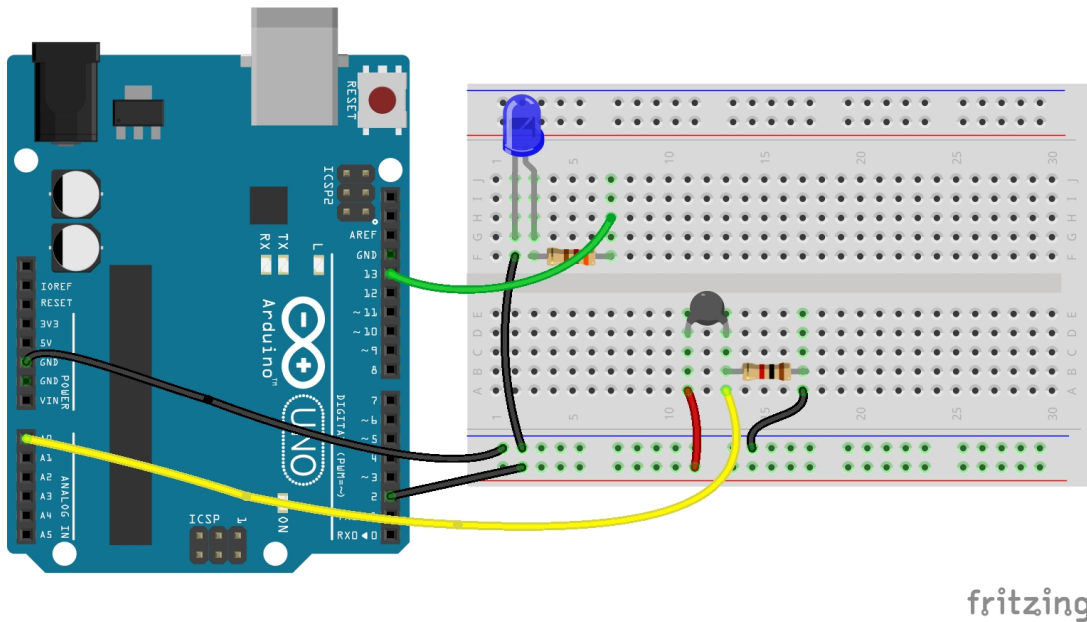


Figure 1.2: Schéma complet du montage

1.2 Formule Steinhartd simplifiée

La température est donnée par la relation:

$$T = \frac{1}{\frac{1}{\beta} \times \ln \frac{R_T}{R_{25}} + \frac{1}{T_{25}}}$$

Question

1. Remplacer les paramètres de la formule par leurs valeurs d'après les données du tableau de la figure 1.1.

.....

2. Modifier les lignes où figurent des points d'interrogation dans les parties du programme "A compléter".

```

1  //// Librairies
2  #include <math.h> // pour le calcul de la fonction ln(), ici notée
3
    
```

```
4   log()
5   /***** A compléter *****/
6   /***** Paramètres du montage *****/
7   // Broche de l'entrée analogique utilisée
8   int capteurPin = A0;
9   // Broche qui alimente le montage
10  int alimPin = 2;
11  // Broche de la led
12  int ledPin = 13;
13  // valeur de la résistance du pont en Ohms
14  float R = ??????;
15  // tension d'alimentation en V
16  int E = 5;
17
18  /*****
19  /**** Variables associées aux mesures *****/
20  //valeur du capteur de 0 à 1023
21  int valeurCapteur;
22  // tension calculée en V (de 0 à 5V)
23  float Uctn;
24  //résistance de la CTN calculée en Ohm
25  float Rctn;
26  //température TC calculée en °C et TK en K
27  float TK, TC;
28
29
30
31  /***** A compléter *****/
32  /** Variables associées aux données de la datasheet*/
33  // Coefficient beta de Steinhart{Hart
34  float B = ??????;
35  // Température de référence
36  float T0 = ?????;
37  // Résistance pour la température de référence
38  float R0 = ?????;
39
40  /*****
41  /**** Initialisation de la carte *****/
42
43  void setup(){
44    // Démarrer la communication série à 9600 bauds (valeur par défaut)
45    Serial.begin(9600);
46    // broches d'alimentation et de la LED à configurer en sortie
47    pinMode(ledPin, OUTPUT);
48    pinMode(alimPin, OUTPUT);
49  }
```

```
50
51 /***** A compléter *****/
52 /***** Programme *****/
53
54 void loop() {
55 // allumer la led 13 -> montrer qu'on fait une mesure
56 digitalWrite (ledPin, HIGH);
57 // alimenter le montage
58 digitalWrite (alimPin, HIGH);
59 // attendre un peu (pour stabiliser la mesure)
60 delay (100);
61 // lecture capteur
62 valeurCapteur = analogRead (capteurPin);
63 // conversion en tension
64 Uctn = valeurCapteur / 1023.0 * 5.0; //valeur de 0 à 1023, tension de 0 à 5V
65 //valeur vers résistance
66 Rctn = R * Uctn / (E - Uctn);
67 // résistance vers température
68
69
70 /***** Modele Steinhart { Hart *****/
71 TK = ?????? ; // température en K
72 TC = TK - 273.2; // température en °C
73
74 /*****/
75
76
77 // arrêter d'alimenter le montage
78 digitalWrite (alimPin, LOW);
79 // eteindre la led 13 -> mesure finie
80 digitalWrite (ledPin, LOW);
81 //envoi de la température par lien série avec l'unité (°C)
82 Serial.print (TC);
83 Serial.println (" °C");
84 Serial.println (valeurCapteur);
85
86 //attendre 2s avant de refaire une mesure
87 delay (2000);
88 }
89
```

Utilisation d'un écran LED

Modifier le programme disponible sur le site de la classe pour afficher la température. On effectuera le montage suivant:

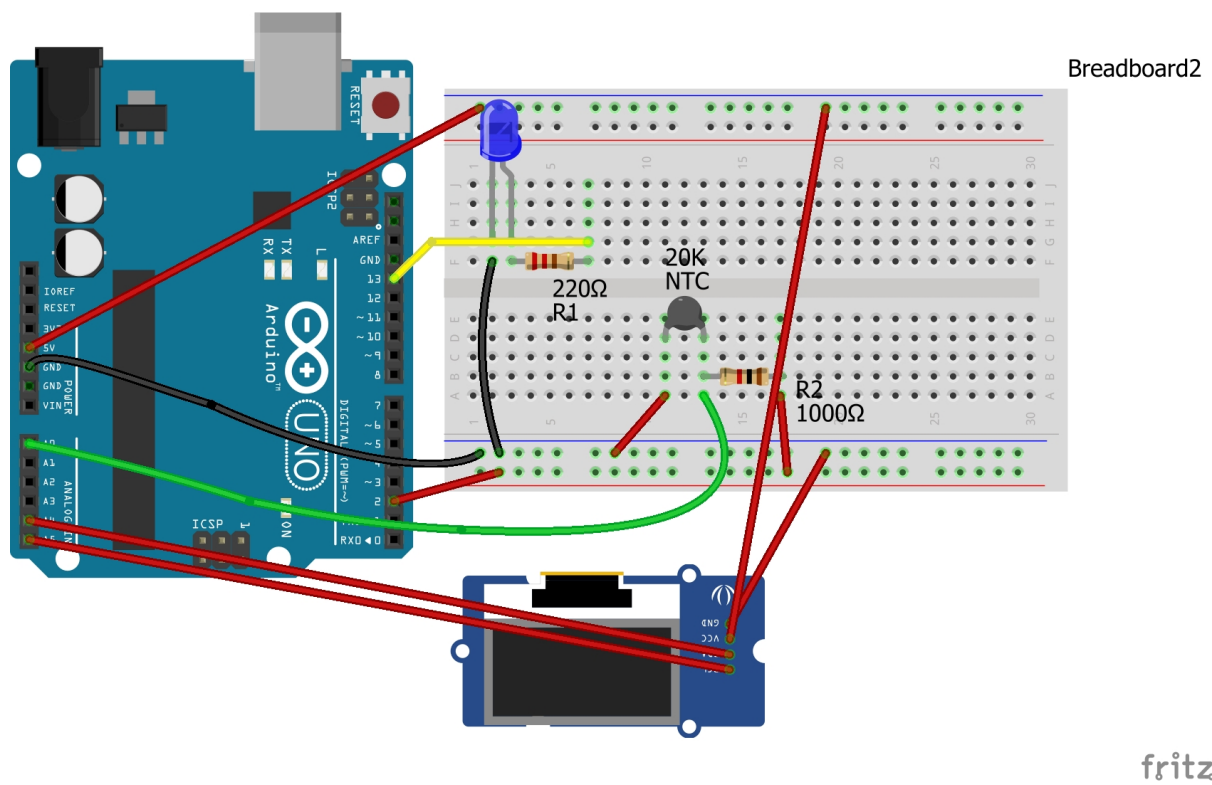


Figure 2.1: Schéma complet du montage

3

Etalonnage capteur sans les données de la datasheet pour des mesures plus précises

- Consulter le cours d'IUT Génie Chimique et génie des procédés sur la thermistance dont voici le lien: <https://www.youtube.com/watch?v=wMcYkilShfw>

3.1 Caractéristiques du capteur

Capteur	Modèle	R	Plage de mesures
CTN	QAP1030.200 NTC	10 kΩ	-25 °C à 95 °C

3.1.1 Calcul des coefficients de Steinhart

Formules qui établissent les coefficients:

$$\begin{cases} A + (\ln(R_1) \cdot B) + (\ln(R_1)^3 \cdot C) = \frac{1}{T_1} \\ A + (\ln(R_2) \cdot B) + (\ln(R_2)^3 \cdot C) = \frac{1}{T_2} \\ A + (\ln(R_3) \cdot B) + (\ln(R_3)^3 \cdot C) = \frac{1}{T_3} \end{cases} \quad (3.1)$$

On pose $Y_1 = \frac{1}{T_1}$ $Y_2 = \frac{1}{T_2}$ $Y_3 = \frac{1}{T_3}$ ainsi que $L_1 = \ln(R_1)$ $L_2 = \ln(R_2)$ $L_3 = \ln(R_3)$

$$\text{On définit } \begin{cases} a = \frac{L_2 - L_3}{L_1 - L_2} \times (L_2^3 - L_1^3) + (L_2^3 - L_3^3) \\ b = (Y_2 - Y_3) - \frac{L_2 - L_3}{L_1 - L_2} \times (Y_1 - Y_2) \end{cases} \quad (3.2)$$

On obtient:

$$\begin{cases} C & = \frac{b}{a} \\ B & = \left(\frac{1}{L_1 - L_2} \right) \times [Y_1 - Y_2 - (L_1^3 - L_2^3) \times C] \\ A & = Y_1 - L_1 \times B - L_1^3 \times C \end{cases} \quad (3.3)$$

Valeurs pour calculer les coefficients

R_1	31 600 Ω	R_2	11 500 Ω	R_3	1200 Ω
T_1	0,4 °C	T_2	23,2 °C	T_3	79 °C

Valeur des coefficients

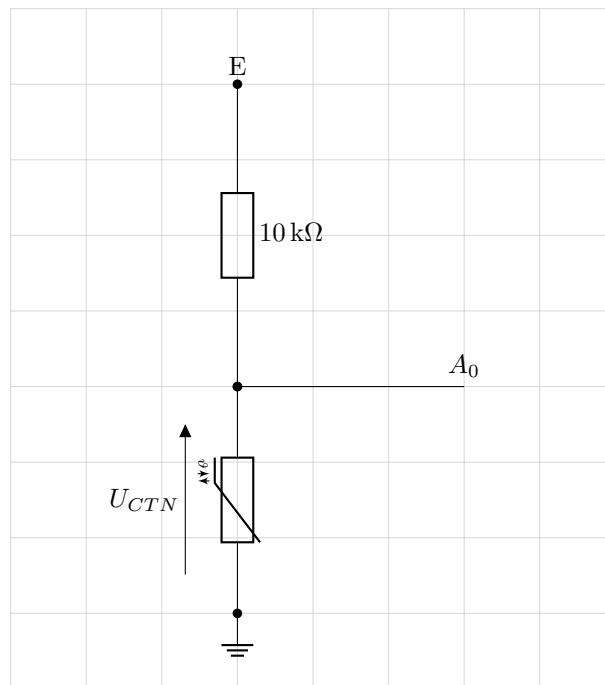
$$A = 0.0016801953368564453$$

$$B = 0.00013965097792430537$$

$$C = 4.7520087301765837e - 07$$

3.1.2 Montages

Pont diviseur de tension



$$U_{CTN} = E \times \frac{R_{CTN}}{R + R_{CTN}}$$

On en déduit que la résistance de la CTN est donnée par la relation:

$$R_{CTN} = \frac{R \times U_{CTN}}{E - U_{CTN}}$$

Programme Arduino

```
1  /// Librairies
2  #include <math.h> // pour le calcul de la fonction ln(), ici noté log()
3
4  /******* Paramètre à adapter au besoin du professeur *****/
5  // Broche de l'entrée analogique utilisée
6  int capteurPin = A0;
7  // Broche qui alimente le montage
8  int alimPin = 2;
9  // Broche de la led
10 int ledPin = 13;
11 // valeur de la résistance du pont en Ohms
12 float R = 10000.0;
13 // tension d'alimentation en V
14 int E = 5;
15 /*******
16 /**** Variables associées aux mesures *****/
17 //valeur du capteur de 0 à 1023
18   int valeurCapteur;
19   // tension calculée en V (de 0 à 5V)
20   float Uctn;
21   //résistance de la CTN calculée en Ohm
22   float Rctn;
23   //température TC calculée en °C et TK en K
24   float TK, TC;
25   // Coefficients A,B et C de Steinhart
26
27   float A = 0.0012439041467480335;
28   float B = 0.0002077621039497996;
29   float C = 2.2977854892403423e-07;
30
31
32
33 void setup(){
34   // Démarrer la communication série à 9600 bauds (valeur par défaut)
35   Serial.begin(9600);
36   // broches d'alimentation et de la LED à configurer en sortie
37   pinMode(ledPin, OUTPUT);
38   pinMode(alimPin, OUTPUT);
39 }
40
```

```
41 void loop() {
42 // allumer la led 13 -> montrer qu'on fait une mesure
43 digitalWrite (ledPin, HIGH);
44 // alimenter le montage
45 digitalWrite (alimPin, HIGH);
46 // attendre un peu (pour stabiliser la mesure)
47 delay (100);
48 // lecture capteur
49 valeurCapteur = analogRead (capteurPin);
50 // conversion en tension
51 Uctn = valeurCapteur / 1023.0 * 5.0; //valeur de 0 à 1023, tension de 0 à 5V
52 //valeur vers résistance
53 Rctn = R * Uctn / (E - Uctn);
54
55 // résistance vers température
56 /***** Modele Steinhart { Hart *****/
57 TK = 1 / (A + B * log(Rctn) + C*(pow(log(Rctn),3))) ; // température en K
58 TC = TK - 273.2; // température en °C
59
60 /*****/
61 // arrêter d'alimenter le montage
62 digitalWrite (alimPin, LOW);
63 // eteindre la led 13 -> mesure finie
64 digitalWrite (ledPin, LOW);
65 //envoi de la température par lien série avec l'unité (°C)
66 Serial.print (TC);
67 Serial.println (" °C");
68 Serial.println (valeurCapteur);
69
70 //attendre 2s avant de refaire une mesure
71 delay (2000);
72 }
73
```