



Capteurs - Température

[Mise à jour le 22/10/2019]

Généralités

La température est une grandeur physique mesurée à l'aide d'un thermomètre et étudiée en thermométrie. Dans la vie courante, elle est reliée aux sensations de froid et de chaud, provenant du transfert thermique entre le corps humain et son environnement.



En physique, elle se définit de plusieurs manières : comme fonction croissante du degré d'agitation thermique des particules (en théorie cinétique des gaz), par l'équilibre des transferts thermiques entre plusieurs systèmes ou à partir de l'entropie (en thermodynamique et en physique statistique).

La température est une variable importante dans d'autres disciplines : météorologie et climatologie, médecine, et chimie.



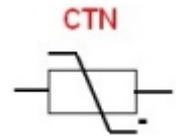
- Ressource à consulter sur **Wikiversité** : [Capteur de température](#)
- *Tableau comparatif des capteurs Groove*
 - Fichier **Excel** à télécharger [ici](#)

Capteurs analogiques

Thermistance

Généralités

Résistance électrique dont la valeur varie rapidement en fonction de la température.

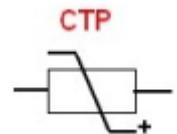


- **CTN**

Les CTN (Coefficient de Température Négatif, en anglais NTC, Negative Temperature Coefficient) sont des thermistances dont la résistance diminue de façon uniforme quand la température augmente. Leur modèle est donné ci-dessous.

$$R_{(T)} = R_0 \cdot e^{B \left[\frac{1}{T} - \frac{1}{T_0} \right]}$$

R: Resistance in ambient temperature T (K)
(K: absolute temperature)
R₀: Resistance in ambient temperature T₀ (K)
B: B-Constant of Thermistor



- **CTP**

Les CTP (Coefficient de Température Positif, en anglais PTC, Positive Temperature Coefficient) sont des thermistances dont la résistance augmente avec la température. On distingue les thermorésistances (augmentation continue et régulière de la résistance avec la température, voir ci-dessus) des CTP dont la valeur augmente fortement avec la température dans une plage de température limitée (typiquement entre 0 °C et 100 °C).

Module SEN23292P



- Source : [wiki](#)

Ce capteur de température compatible Grove à CTN **NCP18WF104F03RC** délivre un signal analogique de 0 à 5 Vcc en fonction de la température mesurée.

- Distributeur : [Gotronic](#)

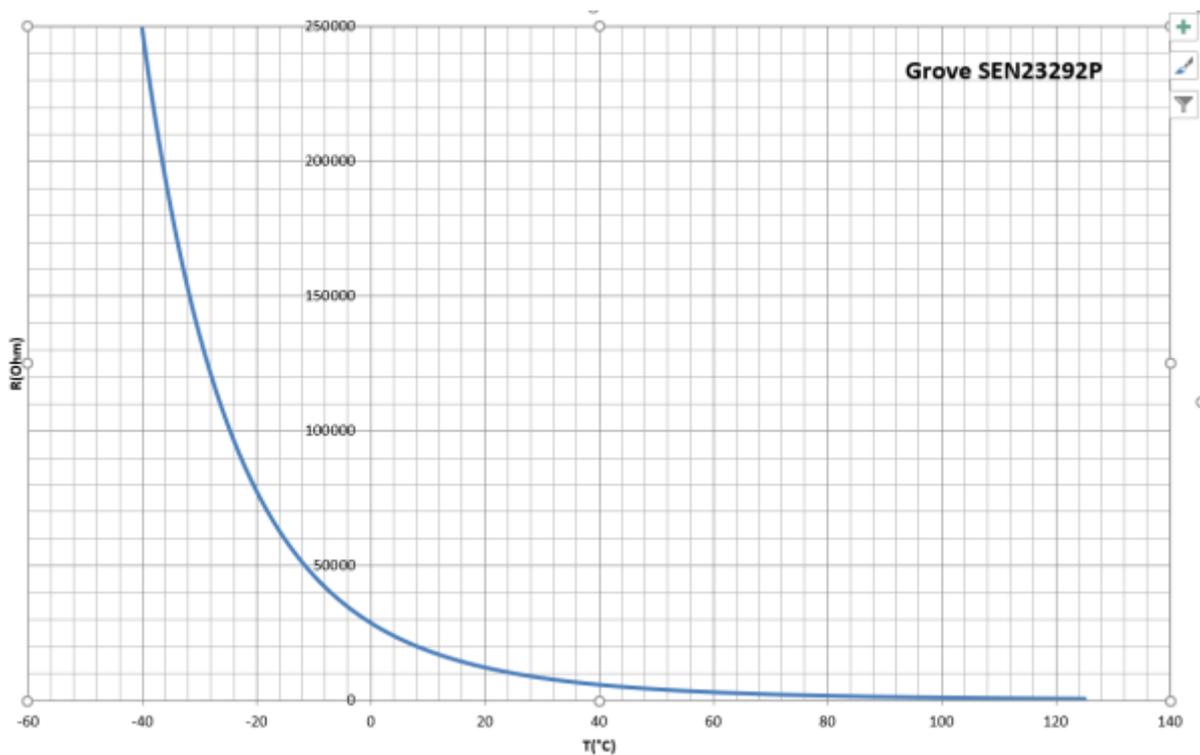
- *Caractéristiques*

- Alimentation: 5 Vcc
- Plage de mesure: -40 à +125 °C
- Précision: 1,5 °C
- Dimensions: 20 x 20 x 13 mm



- *Modèle*

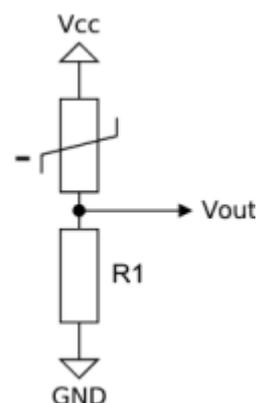
- Fichier Acrobat Reader à télécharger [ici](#)



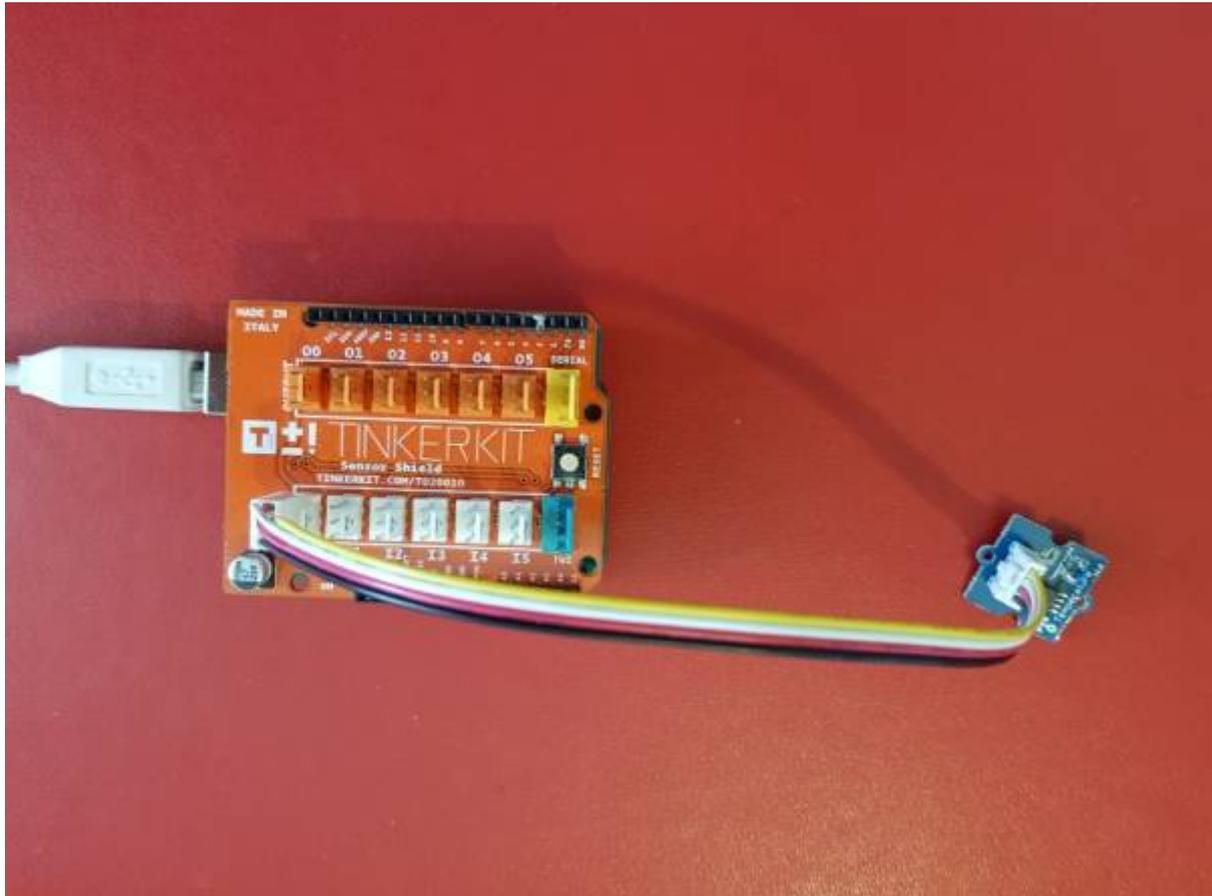
- *Aide pour la **simulation de la chaîne de mesure***

- Les équations de la chaîne de mesure sont téléchargeables [ici](#)
- Le modèle à simuler est téléchargeable [ici](#)

- *Bibliothèques à installer dans l'IDE : aucune*



- Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno.



- Un premier exemple pour tester le capteur
 - Traitement à réaliser : $T=f(N)$



Le projet pour l'IDE **VSCode** de l'exemple ci-dessus est téléchargeable : [A venir](#)

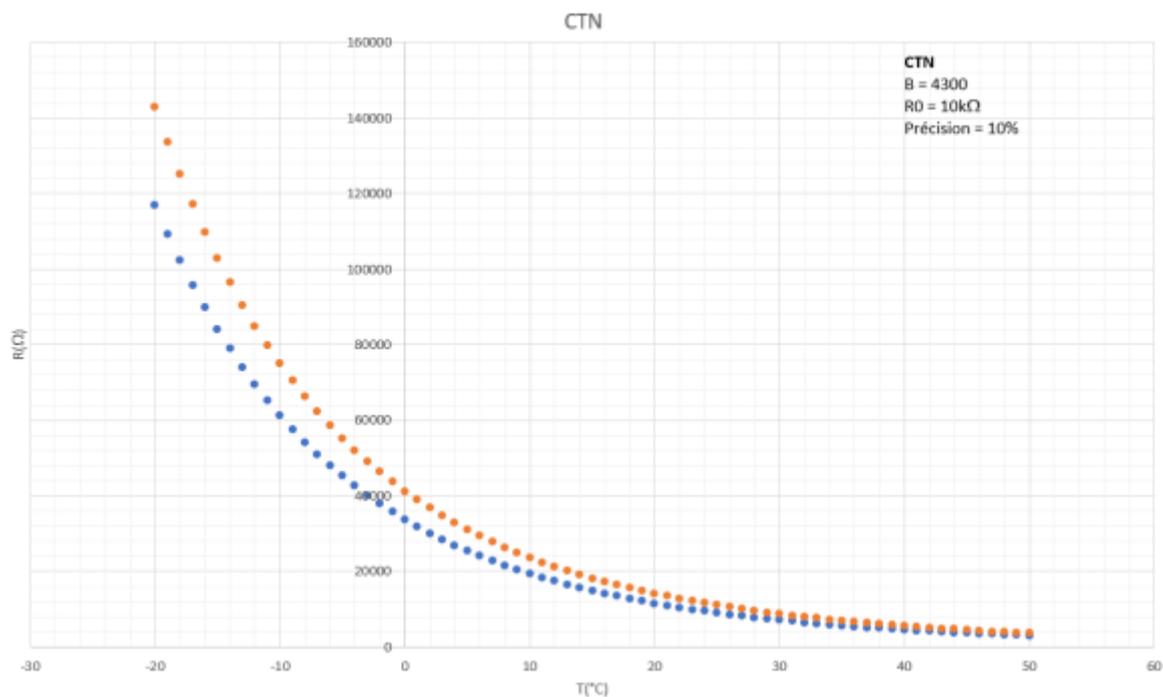
CTN 10k



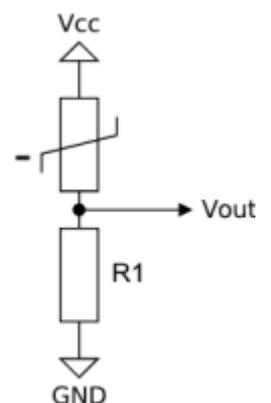
- Distributeur : [Gotronic](#)
- Caractéristiques
 - Résistance à 25°C : 10 k Ω
 - Puissance: 0.25 W.
 - Tolérance: $\pm 10\%$.
 - B=4300.



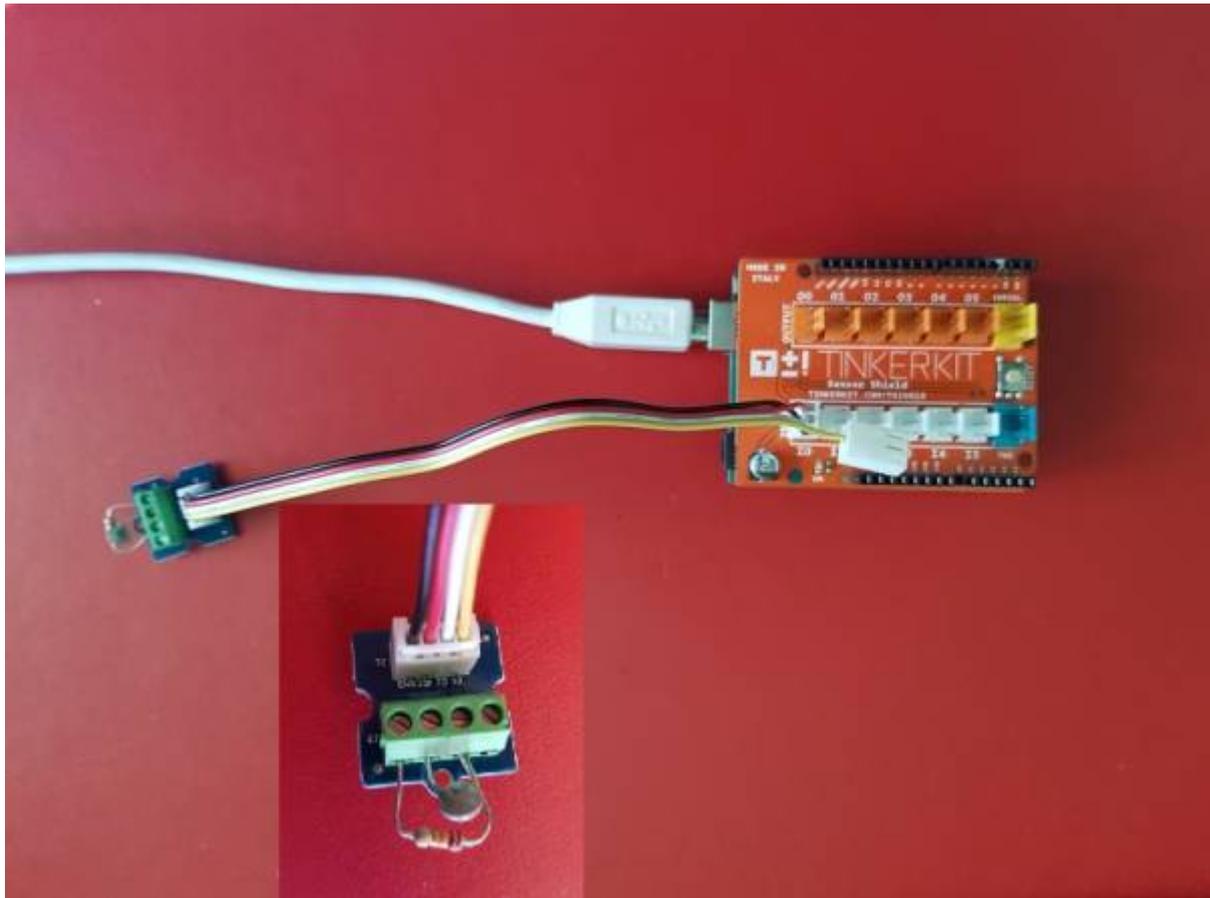
- *Documentation*
 - Fichier Acrobat Reader à télécharger [ici](#)
- *Modèle*
 - Résistance à 25°C : 10 kΩ
 - B=4300.
 - Tolérance: ±10%.



- *Aide pour la **simulation de la chaîne de mesure***
 - Les équations de la chaîne de mesure sont téléchargeables [ici](#) 
 - Le modèle à simuler est téléchargeable [ici](#)
- *Bibliothèques à installer dans l'IDE : aucune*



- *Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno.*



- Un premier exemple pour tester le capteur
 - Traitement à réaliser : $T=f(N)$

ctn.cpp

```
/*  
Mesure de la température ambiante avec une CTN  
Bibliothèque math.h : https://www.arduino.cc/en/math/h  
*/  
// Constantes  
//-----  
// CTN  
const int Beta = 4300; // Kelvin  
const float T0 = 298.15; // Kelvin (25°C)  
const int R0 = 10000; // Résistance du capteur à 25°C  
// Diviseur de tension  
const int Vcc = 5; // Volt  
const int R1 = 12000; // Ohm  
// CAN  
const int n = 10;  
const int VPE = 5;  
  
// Variables  
//-----  
int CTN = A0; // La CTN et son conditionneur sont connectés  
sur la broche A0
```

```

int N = 0; // Image de la température, sortie du CAN
double temperature = 0.0; // Résultat du calcul de la température :
temperature=f(N)
// Coefficient du CAN
float kcan = pow(2, n) / VPE;
// Coefficients utilisés pour simplifier le calcul de la température
// k0, a
double k0 = kcan * Vcc * R1;
double a = R0 / exp(Beta / T0);
// k1, k2
double k1 = k0 / a;
double k2 = R1 / a;
//-----
void setup()
{
  Serial.begin(9600); // Fenêtre "serial" pour la mise au point
}
//-----
void loop()
{
  N = analogRead(CTN);
  temperature = Beta / log((k1 / N) - k2) - 273.15;
  Serial.println(temperature);
}

```



Le projet pour l'IDE **VSCode** de l'exemple ci-dessus est téléchargeable : [ici](#)

Ressource : [wiki](#)

Capteurs intégrés

LM35



- Source : [wiki](#)

Capteur de température intégré.

- Distributeur : [Gotronic](#)
- Caractéristiques
 - Alimentation : 4 à 30 V

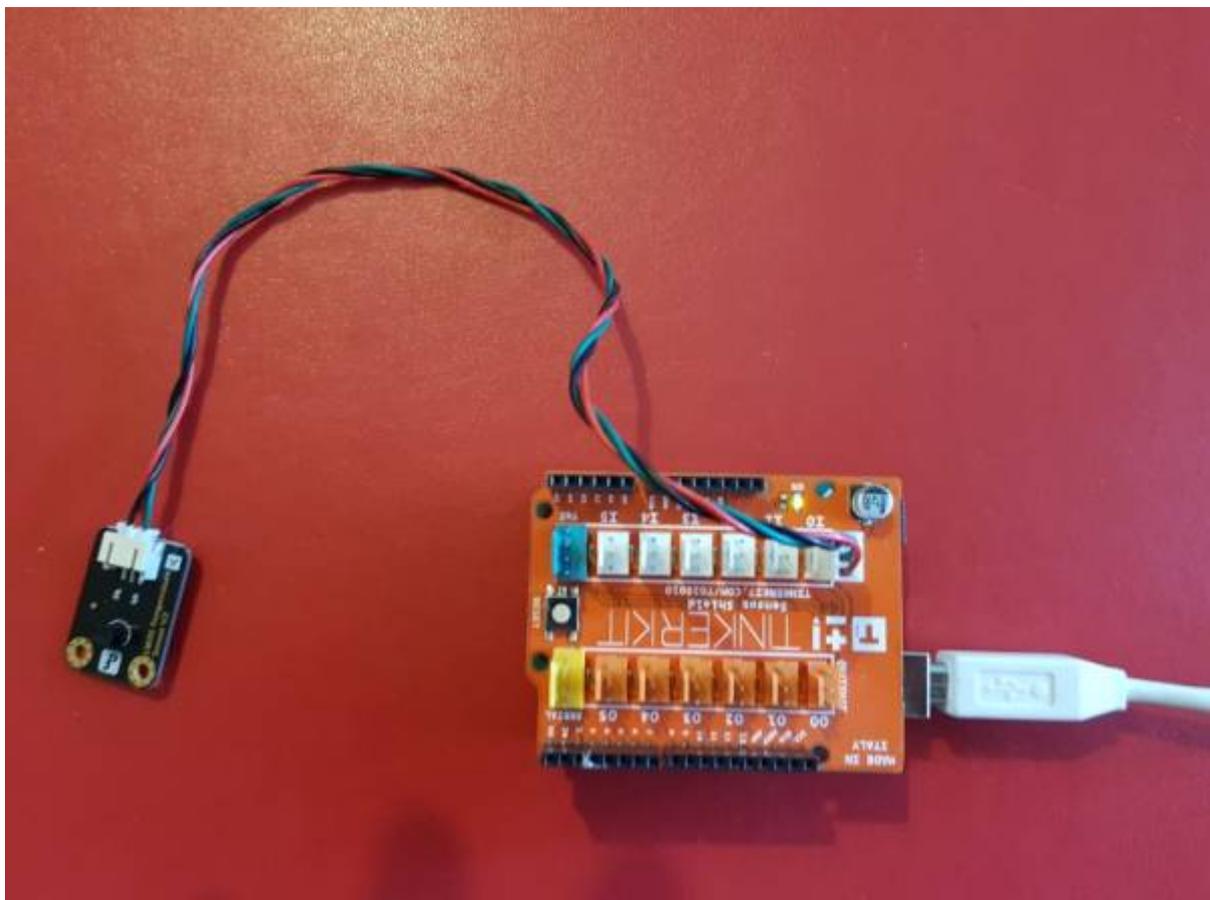
- Plage de mesure : -55 / +150°C
- Sensibilité : 10mV/°C
- Précision : +/-0,5°C (à 25°C)
- Boîtier : TO92



- *Documentation*
 - Fichier Acrobat Reader à télécharger [ici](#)
- *Modèle*
 - Sensibilité : 10mV/°C



- *Aide pour la **simulation de la chaîne de mesure***
 - Les équations de la chaîne de mesure sont téléchargeables [A venir](#) 
 - Le modèle à simuler est téléchargeable [ici](#)
- *Bibliothèques à installer dans l'IDE : aucune*
- *Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno*



- *Un premier exemple pour tester le capteur*

lm35.cpp

```
void setup()
{
  Serial.begin(9600); // Débit binaire : 9600 bps
}

void loop()
{
  uint16_t N;
  double temperature;
  // Lecture
  N=analogRead(A0); //LM35 connecté à Analog 0
  // Traitement
  temperature = (double) N * (5/10.24);
  // Ecriture
  Serial.print("Température:"); // Affiche la température sur le
  moniteur
  Serial.print(temperature);
  Serial.println("C");
  delay(1000);
}
```



Le projet pour l'IDE **VSCode** de l'exemple ci-dessus est téléchargeable [ici](#)

Pour aller plus loin : Mesurer une température négative ([Télécharger](#))

MM111



Capteur de température Velleman basé sur un amplificateur MCP6L01T-E / LT permettant de mesurer la température entre -50 et 150 °C. Il communique avec un microcontrôleur type Arduino ou compatible via une liaison analogique.

- Distributeur : [Gotronic](#)
- Caractéristiques
 - Alimentation: 5 Vcc

- Plage de mesure: -50 à +150 °C
- Sensibilité: 21 mV/°C
- Précision: 0,4 °C
- Dimensions: 22 x 22 x 5 mm



- *Documentation*
 - Fichier Acrobat Reader à télécharger [ici](#)
- *Modèle A venir*



- *Aide pour la **simulation de la chaîne de mesure***
 - Les équations de la chaîne de mesure sont téléchargeables [A venir](#) 
 - Le modèle à simuler est téléchargeable [A venir](#)
- *Bibliothèques à installer dans l'IDE : aucune*
- *Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno.*
- *Un premier exemple pour tester le capteur : A venir*

Le projet pour l'IDE VSCode de l'exemple ci-dessus est téléchargeable : [A venir](#)

Capteurs numériques

MLX90614



- *Source : [wiki](#)*

Ce module capteur de température IR sans contact est basé sur un MLX90614 et comporte un convertisseur analogique-numérique et un DSP (Digital Signal Processor) pour des résultats fiables et précis.

- *Distributeur : [Gotronic](#)*
- *Caractéristiques*
 - Alimentation: 3,3 à 5 Vcc
 - Consommation: 1,2 mA
 - Plage de mesure: -70 à 382 °C
 - Résolution: 0.01 °C

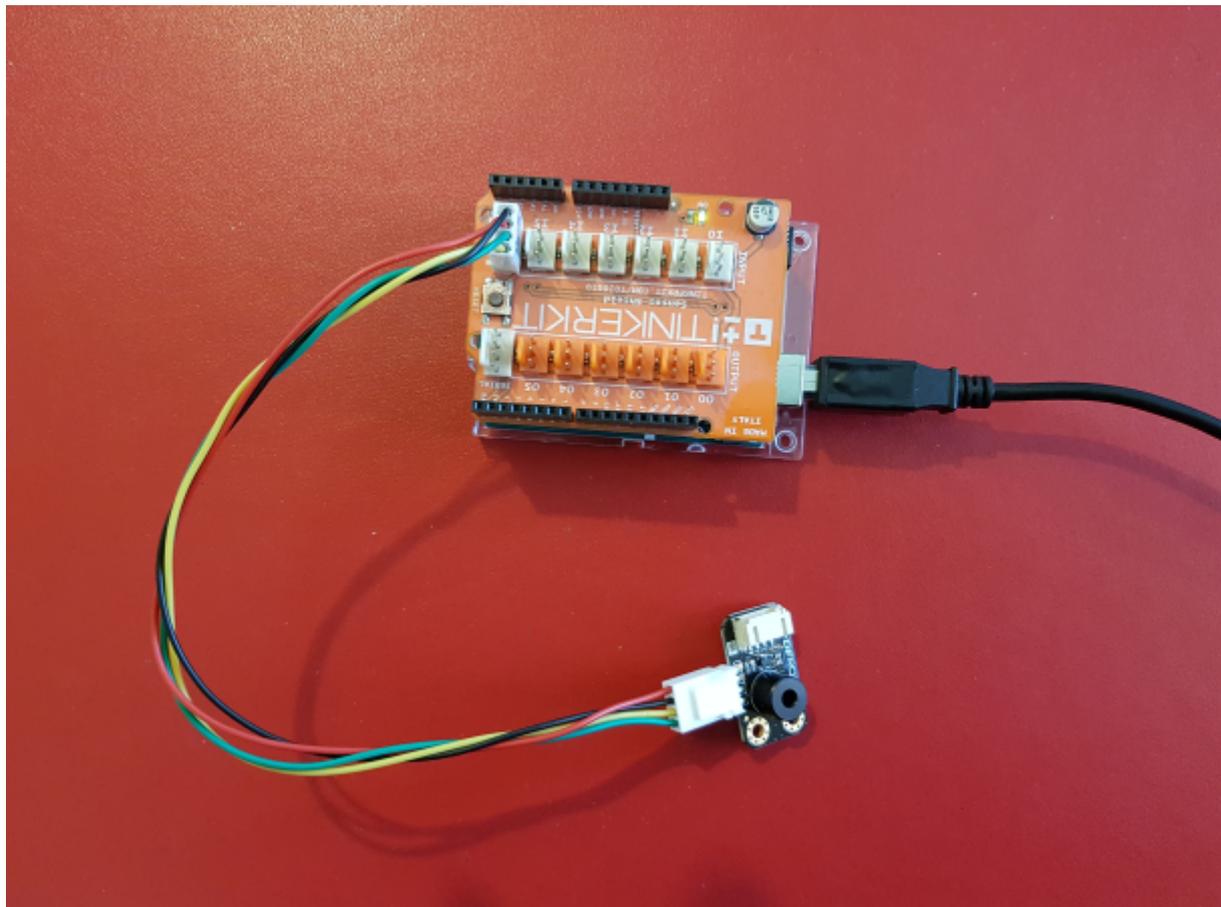
- Interface: I2C
- Dimensions: 32 x 18 mm



- *Documentation*
 - Fichier Acrobat Reader à télécharger [ici](#)
- *Bibliothèques à installer dans l'IDE*



- *Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno*



- *Un premier exemple pour tester le capteur*

[mlx90614.cpp](#)

```
/******  
This is a library example for the MLX90614 Temp Sensor  
  
Designed specifically to work with the MLX90614 sensors in the  
adafruit shop  
----> https://www.adafruit.com/products/1748  
----> https://www.adafruit.com/products/1749*/
```

*These sensors use I2C to communicate, 2 pins are required to interface
Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!*

*Written by Limor Fried/Ladyada for Adafruit Industries.
BSD license, all text above must be included in any redistribution
*****/*

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

void setup() {
  Serial.begin(9600);

  Serial.println("Adafruit MLX90614 test");

  mlx.begin();
}

void loop() {
  Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());
  Serial.print(" *C\tObject = "); Serial.print(mlx.readObjectTempC());
  Serial.println(" *C");
  Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempF());
  Serial.print(" *F\tObject = "); Serial.print(mlx.readObjectTempF());
  Serial.println(" *F");

  Serial.println();
  delay(500);
}
```

TMP102



- Source : [wiki](#)

Le TMP102 est capable de lire des températures avec une résolution de 0,0625 °C et une précision

allant jusqu'à 0,5 °C. La sortie possède des résistances intégrées de 4,7kΩ pour les communications I2C et fonctionne de 1,4V à 3,6V. La communication I2C utilise une signalisation à drain ouvert, il n'est donc pas nécessaire d'utiliser le décalage de niveau.

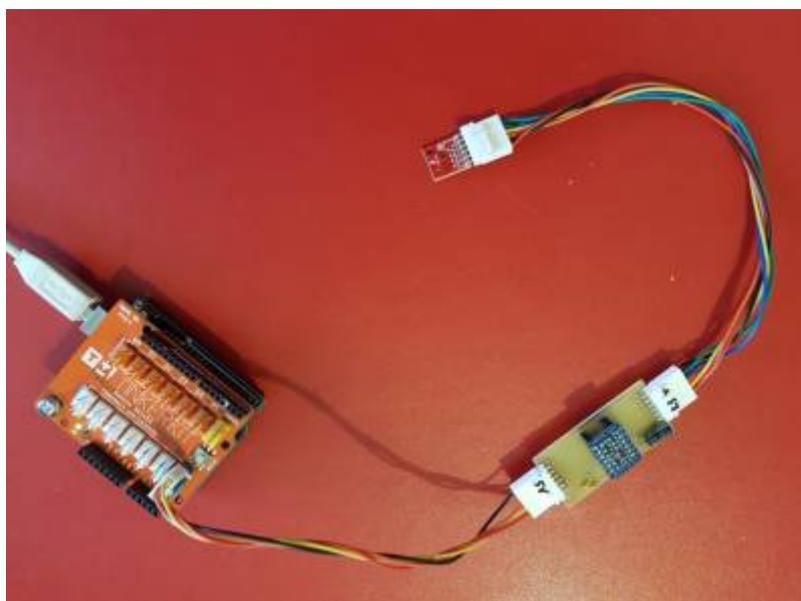
- *Distributeur* : [Mouser](#)
- *Caractéristiques*
 - Alimentation: 1,4 à 3,6 Vcc
 - Consommation: 10 µA maxi (1 µA en veille)
 - Plage de mesure: -40 °C à +125 °C
 - Précision: 0,5 °C (de -25 °C à +85 °C)
 - Résolution: 12bits, 0,0625 °C
 - Interface série 2 fils (I2C)
 - Dimensions: 16 x 16 mm



- *Documentation*
 - Fichier Acrobat Reader à télécharger : [ici](#)
 - Schéma à télécharger [ici](#)
- *Bibliothèques à télécharger dans l'IDE*

```
SparkFun TMP102 Breakout by SparkFun Electronics  
<techsupport@sparkfun.com> Version 1.0.0 INSTALLED  
A library to drive the Texas Instruments TMP102 by I2C. Communicates  
with the TMP102 over I2C to quickly integrate a temperature sensor into your  
project.
```

- *Lien vers la documentation de la bibliothèque* [Sparkfun TMP102 Breakout v1.0.0](#)
- *Connexion à un shield* [Tinkerkit v2](#) monté sur une Arduino Uno.



- *Un premier exemple pour tester le capteur*

Arduino Examples → Examples from Custom Libraries → SparkFun_TMP102_Library → **SparkFun_TMP102_Breakout_Example.ino**

Exemple de résultat attendu

```
CONSOLE DE DÉBOGAGE  SORTIE  TERMINAL
Temperature: 21.94 > Alert Pin: 1 >
Temperature: 21.94 > Alert Pin: 1 >
Temperature: 21.94 > Alert Pin: 1 >
Temperature: 21.87 > Alert Pin: 1 >
```



Le projet contenant l'exemple pour la carte **Netduino**, à programmer avec l'IDE **Visual Studio Community 2015**, est téléchargeable [ici](#)

From:
<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:
<http://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:temperature:temperature&rev=1628666358>

Last update: **2021/08/11 09:19**

