

# TP Capteurs



Dr. Youbi Mohammed Ridha  
département génie biomédical  
Faculté de Technologie  
Université Abou bekr Belkaid  
- Tlemcen

Email: *mohammedridha.  
youbi@univ-tlemcen.dz*

1.0  
Mars 2024

# Table des matières

<b>Objectifs</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>I - CONDITIONNEMENT DES CAPTEURS ACTIFS</b>	<b>5</b>
1. Principe de conditionnement .....	5
<b>II - Travaux pratiques</b>	<b>7</b>
1. DOCUMENTATION .....	7
1.1. DOCUMENT ASSOCIÉ AU TRAVAUX PRATIQUE .....	7
1.2. Matériel .....	7
2. Étapes de mesure .....	7
3. Ressources .....	8
4. Exemple Arduino Temperature Sensor .....	8
<b>III - Exercice :</b>	<b>14</b>

# Objectifs

Ils existent plusieurs grandeurs physiques notamment : température, flux optique- position- débit- vitesse- humidité...etc. Le rôle du capteur est de rendre exploitable ces différentes grandeurs physiques en vue de leur traitement ultérieur.

L'objectif de ce TP est de vous permettre de comprendre le fonctionnement de base des capteurs et d'acquérir des données à partir de ces capteurs pour les traiter.

# Introduction

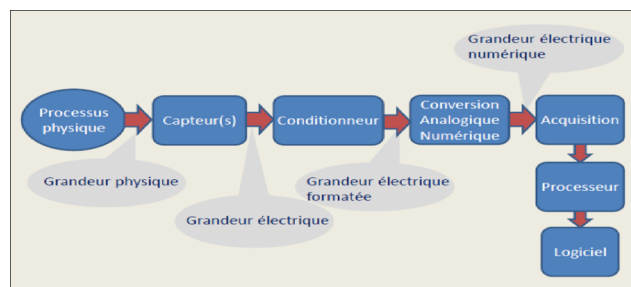
Un capteur est un dispositif qui détecte et répond à un changement dans son environnement. Il convertit la grandeur physique ou chimique mesurée en un signal électrique ou autre qui peut être interprété par un système de traitement de données.

Les capteurs fonctionnent selon différents principes, en fonction du type de grandeur qu'ils mesurent. Par exemple, les capteurs de température mesurent la température en détectant les changements de résistance électrique ou de tension, tandis que les capteurs de pression mesurent la pression en détectant les changements de capacité ou de déformation.

## ***Chaîne de mesure***

La chaîne de mesure est constituée de trois étapes :

- Corps d'épreuve : celui en contact direct avec la mesurande.
- Capteur intermédiaire : ce transducteur est associé à un conditionneur qui fournit une grandeur électrique de sortie exploitable.
- Conditionneur : qui fournit une grandeur électrique de sortie exploitable et d'autre part, il détermine la nature finale du signal électrique.



*chaîne d'acquisition de signal du capteur*

# I CONDITIONNEMENT DES CAPTEURS ACTIFS

## 1. Principe de conditionnement

### *Les conditionneurs de signaux :*

Le conditionnement de signaux est le processus de transformation d'un signal électrique ou mécanique (signal d'entrée) en un autre (signal de sortie). Les conditionneurs de signaux, les instruments utilisés pour effectuer cette tâche, jouent un rôle crucial dans la fourniture de mesures précises, essentielles pour l'acquisition de données et le contrôle des machines.

### *Fonctionnement des conditionneurs de signaux*

**Conversion de signaux :** Les conditionneurs de signaux captent les signaux et les convertissent en signaux électriques plus puissants. Cela est essentiel dans les applications industrielles où divers capteurs sont utilisés pour effectuer des mesures. Les signaux provenant de ces capteurs sont convertis en signaux de processus standard pour une utilisation par les instruments connectés.

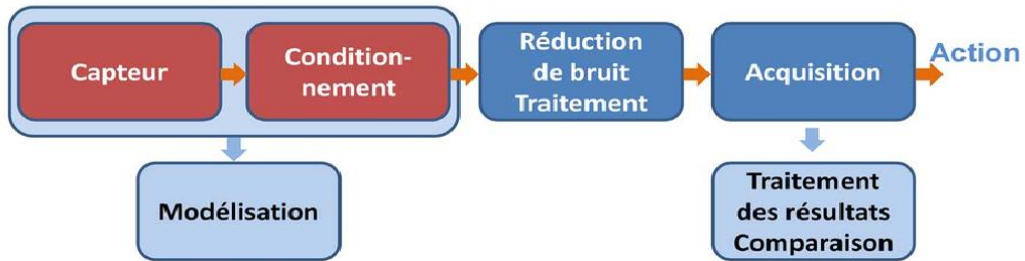
**Linéarisation :** Certains conditionneurs de signaux peuvent linéariser les signaux lorsque la relation entre le signal du capteur et la mesure physique n'est pas linéaire. Ce processus, réalisé par logiciel, interprète le signal et garantit une plus grande précision.

**Amplification :** L'amplification du signal augmente l'amplitude du signal pour le traitement ou la numérisation. Cela peut être réalisé en augmentant la résolution du signal d'entrée ou le rapport signal/bruit. Les conditionneurs de signaux utilisent différents amplificateurs, tels que les amplificateurs d'instrumentation et les amplificateurs d'isolement, pour l'amplification.

**Filtrage :** Le filtrage supprime le bruit du signal et ne laisse passer que les données valides. Les filtres peuvent être passifs (utilisant des composants tels que des condensateurs et des résistances) ou actifs (utilisant des composants actifs tels que des amplificateurs opérationnels). Les conditionneurs de signaux modernes utilisent des filtres numériques pour un réglage facile et une flexibilité.

**Fonctions intelligentes :** Les conditionneurs de signaux avancés offrent des fonctions supplémentaires pour l'évaluation du signal et le prétraitement des données. Ces fonctions permettent la surveillance et l'évaluation des alertes et des alarmes, ainsi que des fonctions mathématiques telles que l'ajout de signaux de capteur ou le contrôle PID.

**Interfaces :** Les conditionneurs de signaux transmettent les signaux des capteurs via des interfaces standard. Les interfaces analogiques (par exemple, signaux de tension ou de courant) nécessitent un câblage séparé pour chaque signal. Les interfaces numériques (par exemple, Ethernet industriel) permettent de connecter plusieurs composants avec un seul câble et de transmettre des informations supplémentaires pour une maintenance plus efficace.



*conditionnement des capteurs*

# II Travaux pratiques

## 1. DOCUMENTATION

### 1.1. DOCUMENT ASSOCIÉ AU TRAVAUX PRATIQUE

- Capteur photoresistance (LDR)\_photodiode (cf. capteur photoresistance \_photodiode (1))
- Capteur ultason (cf. capteur us (2))
- Capteur de presion (cf. Mesure de pression (3))

### 1.2. Matériel

Arduino Uno ou compatible

Capteur de température (thermocouple,NTC,PTC, LM35)

Capteur de pression (mpx2400, BMP280)

Capteur à ultrasons ( HC-SR04)

Capteur optique (par exemple, LDR)

Câbles de connexion

Logiciel de programmation (par exemple, Arduino IDE)

## 2. Étapes de mesure

**Partie 1** : Configuration du matériel

Connectez les capteurs à l'Arduino selon les schémas de câblage fournis.

Vérifiez vos connexions et alimentez l'Arduino.

**Partie 2** : Programmation de l'Arduino

Ouvrez l'Arduino IDE et créez un nouveau croquis.

Copiez et collez le programme Arduino fourni dans le croquis.

Modifiez le programme si nécessaire pour vos propres besoins.

Téléchargez le programme sur l'Arduino.

**Partie 3** : Acquisition et traitement des données

Exécutez le programme sur l'Arduino.

Observez les données lues par les capteurs sur l'écran ou via la connexion série.

Traitez les données pour en extraire des informations utiles (par exemple, calculer la température moyenne, détecter les mouvements).

#### **Partie 4 : Analyse des résultats**

Analysez les données traitées pour identifier les tendances ou les modèles.

Tirez des conclusions sur le fonctionnement des capteurs et les informations qu'ils fournissent.

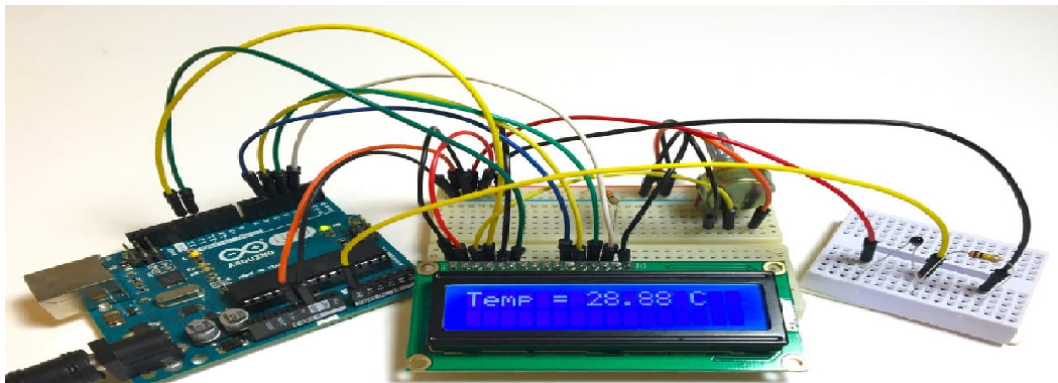
### **3. Ressources**

#### **Ressources :**

\* *Arduino IDE* \* *Bibliothèque Adafruit\_BMP280* \* *Bibliothèque NewPing* \* *Tutoriel sur les capteurs avec Arduino*

#### **Fiches techniques des capteurs :**

\* *Capteur de température LM35* \* *Capteur de pression BMP280* \* *Capteur à ultrasons HC-SR04* \* *Capteur optique LDR*



*TP capteur avec arduino*

### **4. Exemple Arduino Temperature Sensor**

**Thermistors** are simple, inexpensive, and accurate components that make it easy to get temperature data for your projects. Remote weather stations, home automation systems, and equipment control and protection circuits are some applications where thermistors would be ideal. They're analog sensors, so the code is relatively simple compared to digital temperature sensors that require special libraries and lots of code.

In this article, I'll explain how thermistors work, then I'll show you how to set up a basic thermistor circuit with an Arduino that will output temperature readings to the serial monitor or to an LCD.

#### **HOW A THERMISTOR WORKS**

Thermistors are variable resistors that change their resistance with temperature. They are classified by the way their resistance responds to temperature changes. In Negative Temperature Coefficient (NTC) thermistors, resistance decreases with an increase in temperature. In Positive Temperature Coefficient (PTC) thermistors, resistance increases with an increase in temperature.

NTC thermistors are the most common, and that's the type we'll be using in this tutorial. NTC thermistors are made from a semiconducting material (such as a metal oxide or ceramic) that's been heated and compressed to form a temperature sensitive conducting material.



The conducting material contains charge carriers that allow current to flow through it. High temperatures cause the semiconducting material to release more charge carriers. In NTC thermistors made from ferric oxide, electrons are the charge carriers. In nickel oxide NTC thermistors, the charge carriers are electron holes.

### ***A BASIC THERMISTOR CIRCUIT***

Let's build a basic thermistor circuit to see how it works, so you can apply it to other projects later.

Since the thermistor is a variable resistor, we'll need to measure the resistance before we can calculate the temperature. However, the Arduino can't measure resistance directly, it can only measure voltage.

The Arduino will measure the voltage at a point between the thermistor and a known resistor. This is known as a voltage divider.

The equation for a voltage divider is:  $V_{out} = V_{in} * (R1 / (R1 + R2))$

In terms of the voltage divider in a thermistor circuit.

### ***Materials***

Arduino Uno

Thermistor

10k Ohm resistor

Breadboard

Jumper wires

### ***Circuit Diagram***

The *circuit* (cf. p.12)

### ***CODE FOR SERIAL MONITOR OUTPUT OF TEMPERATURE READINGS***

After connecting the circuit above, upload this code to your Arduino to output the temperature readings to the serial monitor in Fahrenheit:

```
int ThermistorPin = 0;

int Vo;

float R1 = 10000;

float logR2, R2, T;

float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

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

void loop() {
  Vo = analogRead(ThermistorPin);
  R2 = R1 * (1023.0 / (float)Vo - 1.0);
  logR2 = log(R2);
```

```
T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));  
T = T - 273.15;  
T = (T * 9.0) / 5.0 + 32.0;  
Serial.print("Temperature: ");  
Serial.print(T);  
Serial.println(" F");  
delay(500);  
}
```

To display the temperature in degrees Celsius, just comment out line 18 by inserting two forward slashes (“//”) at the beginning of the line.

This program will display Celsius and Fahrenheit at the same time:

```
int ThermistorPin = 0;  
int Vo;  
float R1 = 10000;  
float logR2, R2, T, Tc, Tf;  
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;  
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  Vo = analogRead(ThermistorPin);  
  R2 = R1 * (1023.0 / (float)Vo - 1.0);  
  logR2 = log(R2);  
  T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));  
  Tc = T - 273.15;  
  Tf = (Tc * 9.0) / 5.0 + 32.0;  
  Serial.print("Temperature: ");  
  Serial.print(Tf);  
  Serial.print(" F; ");  
  Serial.print(Tc);  
  Serial.println(" C");  
  delay(500);  
}
```

## CODE FOR LCD OUTPUT OF TEMPERATURE READINGS

To output the temperature readings to a 16X2 LCD, follow our tutorial, How to Set Up an LCD Display on an Arduino, then upload this code to the board:

```
#include <LiquidCrystal.h>

int ThermistorPin = 0;

int Vo;

float R1 = 10000;

float logR2, R2, T;

float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

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

void loop() {
  Vo = analogRead(ThermistorPin);
  R2 = R1 * (1023.0 / (float)Vo - 1.0);
  logR2 = log(R2);
  T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
  T = T - 273.15;
  T = (T * 9.0) / 5.0 + 32.0;
  lcd.print("Temp = ");
  lcd.print(T);
  lcd.print(" F");
  delay(500);
  lcd.clear();
}
```

**Instructions**

Connect the thermistor to pin A0 on the Arduino.

Connect the 10k Ohm resistor to pin A1 on the Arduino.

Connect the ground pin of the Arduino to the ground rail on the breadboard.

Connect the 5V pin of the Arduino to the power rail on the breadboard.

Upload the code to the Arduino.

**Circuit Testing** (cf. p.12)

Once the code is uploaded, you can test the temperature sensor by placing it in a warm or cold environment. The temperature reading should change accordingly.

Remplir le tableau suivant:

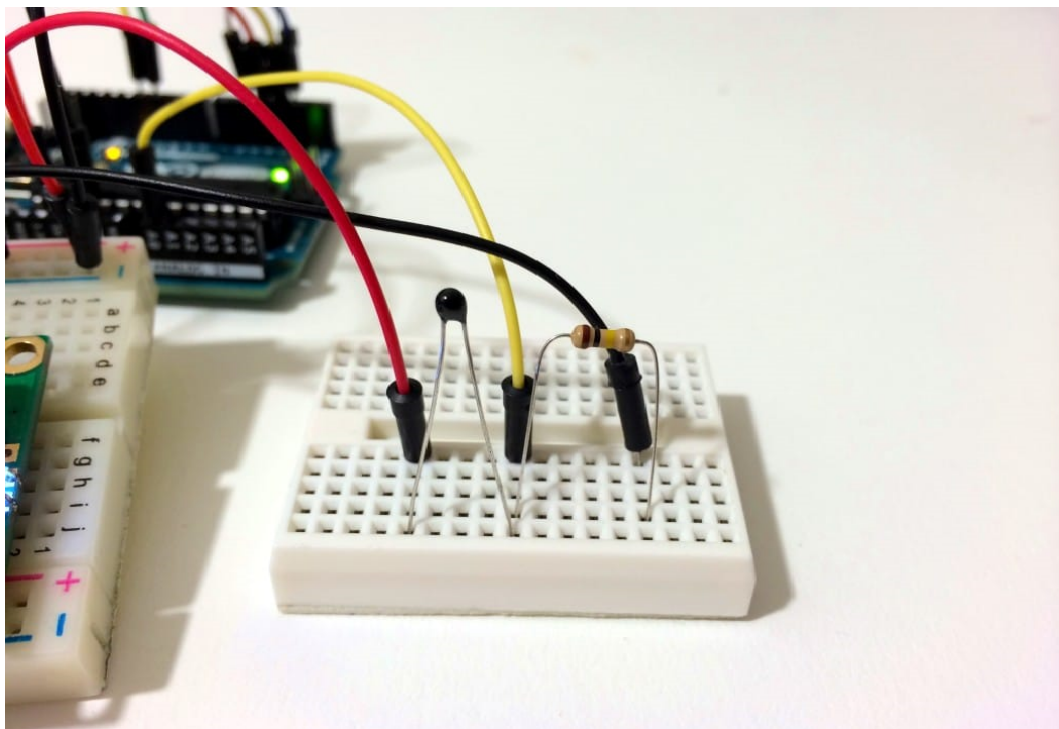
Vout									
Rth									
Temp									

*Tableau:*

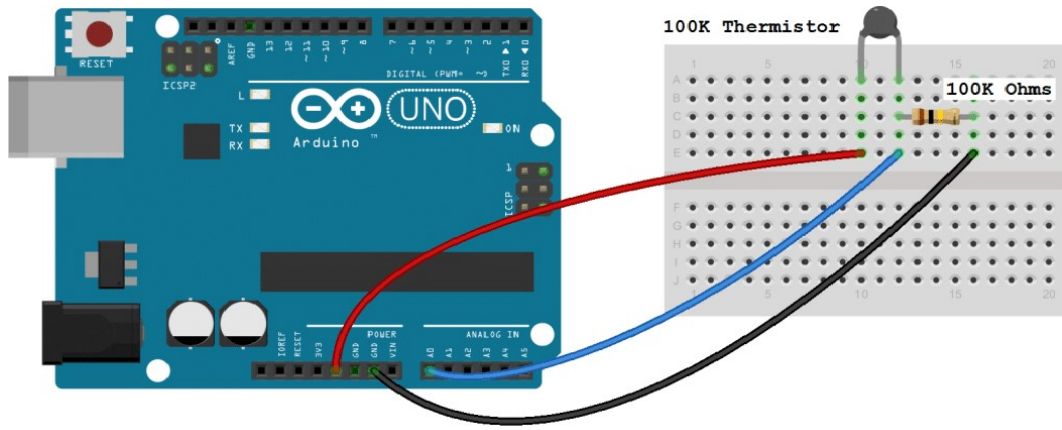
***Troubleshooting:***

If the temperature reading is not changing, check the connections between the thermistor, resistor, and Arduino. If the temperature reading is inaccurate, try changing the value of the resistor.

*Here's a video of the temperature sensor so you can watch me set it up and see how it works (cf. p.)*



*circuit testing*



*circuit*

### III Exercice :

Après avoir effectué les travaux pratiques, formulez une conclusion en résumant ce que vous avez découvert lors des travaux pratiques et en répondant aux questions:

***-capteur photodiode***

quel intérêt a porté d'utilisation la photo diode comme capteur

***-capteur de pression***

Proposer une amplification de signal de sortie du capteur pour avoir une sensibilité de l'ensemble égale à 1mV /mb (la tension de sortie de capteur VS) qui va nous permettre d'effectuer la mesure pour démontrer la loi de boyle-mariotte

***-Objective d'expérience capteur ultrason***

Mettre en place un protocole expérimental (schéma bloc) permettant d'émettre un signal ultrasonore et mesurer son écho dans une mesure expérimentale.

voici un exemple de retour d'expérience d'utilisation des capteur (cf. Réponses aux questions (2))