

Caractéristique d'un conducteur ohmique

La photorésistance - Luxmètre

1- Documents

Document 1 - Les capteurs

Un capteur est un circuit électronique ou un simple composant électronique (éventuellement une simple résistance). Il permet de faire le lien entre une grandeur physique que l'on souhaite mesurer (température, lumière, humidité, intensité...) et une grandeur électrique. Chaque capteur possède ses spécificités, ses propres caractéristiques: la courbe d'étalonnage d'un capteur sert à établir une relation mathématique entre la tension mesurée à ses bornes et la grandeur physique que l'on souhaite étudier de manière à modéliser l'évolution de la tension en fonction des valeurs prises par le capteur.

Document 2 - Caractéristique tension intensité

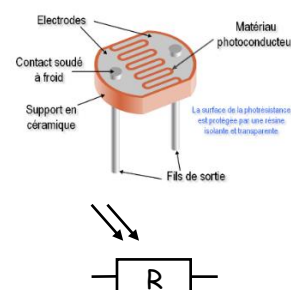
Afin de définir les propriétés électriques d'un dipôle, il est nécessaire d'étudier sa caractéristique tension-intensité $U=f(I)$.

Cette caractéristique est la courbe représentant la tension U entre les bornes du dipôle étudié en fonction de l'intensité I du courant qui le traverse.

Document 3 - La photorésistance LDR

LDR pour Light Dependent Resistor, la photorésistance est un dipôle dont la résistance varie en fonction de l'éclairement E qu'elle reçoit d'une source de lumière: plus elle est éclairée, plus la valeur de sa résistance baisse.

La partie sensible du capteur est une piste de sulfure de cadmium en forme de serpent: l'énergie lumineuse reçue déclenche une augmentation de porteurs de charges libres dans ce matériau, de sorte que sa résistance électrique évolue.



Document 4 - Eclairement et luxmètre

L'éclairement E (lux) est une grandeur physique se mesurant à l'aide d'un luxmètre et qui permet de rendre compte de la luminosité plus ou moins forte d'une source lumineuse. Plus la source paraît intense, plus son éclairement E est élevé.

Un luxmètre est un capteur qui permet de mesurer de façon simple l'éclairement.

On peut aussi utiliser un smartphone comme capteur de lumière.



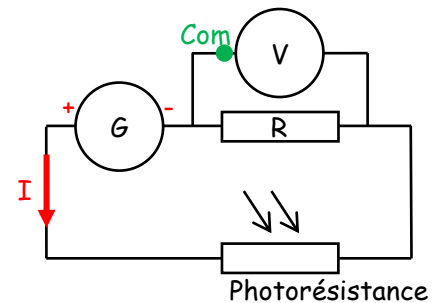
Document 5 - Montage expérimental de base Etalonnage de la photorésistance en luxmètre

Afin d'utiliser la photorésistance en tant que luxmètre, on utilisera un montage pont diviseur de tension.

Le montage pont diviseur de tension est un montage électronique simple qui permet de diviser une tension d'entrée, constitué par exemple de deux résistances en série.

Le voltmètre est placé aux bornes de la résistance.

Remarque: La tension d'alimentation reste constante. On ne fait varier que l'intensité lumineuse arrivant sur la photorésistance. Pour cela on fait varier la distance entre la source lumineuse et la photorésistance.

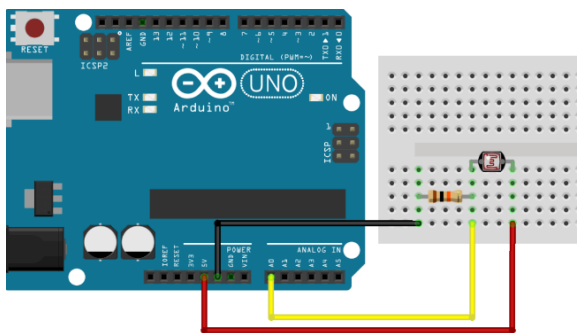


Document 6 - Montage expérimental avec Arduino et Luxmètre

Afin d'utiliser la photorésistance en tant que luxmètre, on peut utiliser une carte microcontrôleur Arduino.

On utilise toujours le principe de montage en série avec une résistance afin d'obtenir un pont diviseur de tension. La tension qui sera relevé aux bornes de la résistance sera affichée via le moniteur série.

Cette tension pourra être convertie en intensité lumineuse si on connaît la relation reliant l'intensité lumineuse E (en lux) à la tension U (en volt) aux bornes de la résistance.



Document 7 - Mesure de tension avec Arduino

Afin de mesurer la valeur de la tension aux bornes du pont diviseur on utilise le port A0 de la carte Arduino. Ce port correspond à la tension U

Document 8 - Matériel

- Ordinateur avec Spyder et Arduino
- Plaque microcontrôleur Arduino avec cordon USB
- Dispositif Ardui-TC-PL de Potorésistance - Luxmètre
- Fils de connexion
- Luxmètre

2- Python

Python est un langage de programmation interprété (les instructions que vous lui envoyez sont "transcrites" en langage machine au fur et à mesure de leur lecture), à ne pas confondre avec un langage compilé, (où avant de pouvoir les exécuter, le logiciel se charge de transformer le code du programme en langage machine).

Il est possible d'associer des bibliothèques à Python afin d'étendre ses possibilités.

Document 9 - Programme en Python pour l'étalonnage de la photorésistance

Le programme à utiliser se nomme: "Etalonnage photoresistance.py"

```
# Prof-TC
# Mai 2021
# Etalonnage d'une photorésistance
# Tracé de U=f(I) pour une photorésistance

#Affichage du nom du programme
print("_____")
print("Etalonnage d'une photorésistance")
print("_____")

#Importation des bibliothèques
import matplotlib.pyplot as plt
import numpy as np
from math import log10

# Valeurs expérimentales à modifier selon l'expérience (E en lux et U en volt)
print("*****")
U=[0.88, 0.80, 0.74, 0.69, 0.64, 0.60, 0.56, 0.53, 0.50, 0.46, 0.44, 0.41, 0.35]
E=[421, 370, 326, 289, 259, 234, 214, 196, 181, 163, 155, 144, 130]
print("*****")

# Nombre de points expérimentaux
K=len(U)

# Détermination des valeurs de L=log(E)
L=[]
for i in range(K):
    L.append(log10(E[i]))

#Affichage des valeurs expérimentales et du nombre de points
print("valeurs de U: ",U)
print("valeurs de L: ",L)
print("Il y a ",K," couples de points")
print("*****")

#Initialisation des variables
Umoy=0.0
Lmoy=0.0
N=0.0
D=0.0
a=0.0
b=0.0
Umin=U[0]
Lmin=L[0]
Umax=U[K-1]
Lmax=L[K-1]

# Initialisation de la figure
fig, ax = plt.subplots()

#Calcul des coefficients de la droite représentant U=f(I)
for i in range(K):
    Umoy=Umoy+U[i]
    Lmoy=Lmoy+L[i]

Umoy=Umoy/(K)
Lmoy=Lmoy/(K)

for i in range(K):
    N=N+(U[i]-Umoy)*(L[i]-Lmoy)
    D=D+(U[i]-Umoy)*(U[i]-Umoy)

a=N/D #Coefficient directeur de la droite
b=Lmoy-a*Umoy #Ordonnée à l'origine de la droite

#Mise en écriture avec 5 décimales pour l'affichage de l'équation
A=round(a,3)
B=round(b,3)
c=10**b
C=round(c,3)

#Tracé des axes, labels et titre du graphique
plt.axis([Umin, Umax, Lmin, Lmax])
plt.xlabel('U (V)',color='green', fontsize=20)
plt.ylabel('L (sans unité)',color='green', fontsize=20)
plt.title("Etalonnage d'une photorésistance", color='red', fontsize=10)
plt.grid()

#Tracé des valeurs expérimentales
plt.scatter(U,L,marker='o',color='r',linewidth = 4)

#Tracé de la droite
x=np.linspace(0,Umax+1,200)
y=a*x+b
plt.plot(x,y,linewidth = 2)

#Ecriture de l'équation de la droite
plt.text((Umax+Umin)/2, Lmax-(Lmax-Lmin)*0.1, "L = "+str(A)+" x U + "+str(B),
color='green', fontsize=10,horizontalalignment = 'center')
plt.text((Umax+Umin)/2, Lmax-(Lmax-Lmin)*0.2, "A = "+str(A)+" B = "+str(B)+" C = "+str(C),
color='red', fontsize=10,horizontalalignment = 'center')

#Affichage du graphique
plt.show()

# Sauvegarde de la figure dans le dossier où se trouve le programme
fig.savefig("Etalonnage d'une photorésistance")

print("*****")
print("L'équation de la droite est: L = ",A," x U + ",B)
print("*****")
print("L'équation de la courbe à utiliser sera donc: E = ",C,"x 10^(",A,"x U)")
print("*****")

#Fin du programme
```

3- Arduino

L'Arduino est une plateforme de prototypage électronique open-source, basée d'une part sur du matériel et d'autre part sur un ensemble de logiciels faciles à utiliser.

Une carte Arduino communique avec son environnement par l'intermédiaire de ses broches d'entrées/sorties. Sur ces broches, des capteurs, dispositifs permettant de transformer une information de l'environnement en signal électrique et des actionneurs, dispositifs permettant de transformer un signal électrique en action mécanique ou lumineuse, vont être connectés.

L'Arduino étant un ordinateur spécialisé dans la gestion de capteurs et d'actionneurs, c'est un programme qui va décider de la manière donc les capteurs et les actionneurs sont utilisés. C'est donc très différent de l'électronique traditionnelle où les fonctions qui relient les capteurs aux actionneurs, l'appui d'un bouton qui entraîne l'allumage d'un DEL par exemple, sont déterminées en dur par le câblage entre les composants, et par les composants eux-mêmes.

Ici les fonctions sont déterminées par un programme. Par conséquent les fonctions peuvent être, d'une part, beaucoup plus élaborées et, d'autre part, modifiables à volonté sans changer la quincaillerie.

Le langage de programmation Arduino peut être divisé en trois parties principales: structures, valeurs (variables et constantes), et fonctions.

Le langage Arduino est très proche du langage C++. Les règles de syntaxe sont communes.

Document 10- Programme Arduino Etalonnage de la Photorésistance

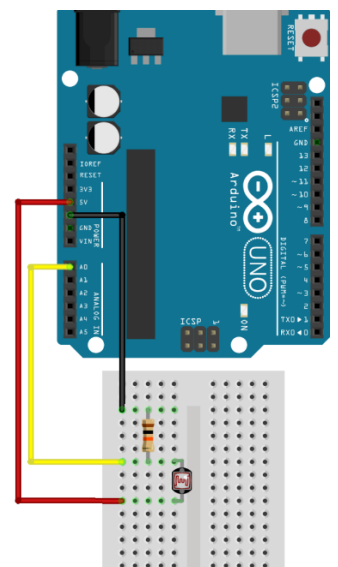
Le programme à utiliser se nomme: "EtalonnagePhotoresistance.ino".

```
*
*EtalonnagePhotoresistance
*18 mars 2022
*Prof-TC
*/
//-----
// Mesure de la tension aux bornes du pont diviseur de la Photoresistance
//-----

float U;      // Potentiel a la borne de la resistance du pont diviseur
//-----

void setup() {
  Serial.begin(9600);      // Paramétrage du port série
  Serial.println("Tension aux bornes du pont diviseur de la Photoresistance"); //Titre du programme
  Serial.println("");      // Saute une ligne
  Serial.println("U (V)");  // Donne la valeur de la tension en Volt
  Serial.println("");      // Saute une ligne
}
//-----

void loop() {
  U = analogRead(A0)*5.0/1023; // Lecture potentiel A0
  Serial.println(U);           // Affichage de la tension U en Volt aux bornes du pont diviseur de la Photoresistance
  delay(5000);                 // Temporisation de 5s
}
//-----
```



Document 11- Programme Arduino Luxmètre

Le programme à utiliser se nomme: "Luxmètre.ino"

```

*
*Luxmètre
*2 juin 2021
*Prof-TC
*/

const char capteur = 0; // Sur la broche A0 est connecté le pont diviseur de tension
float tension = 0; // "Tension" est la variable qui enregistre la tension lue en sortie du capteur
float eclairement = 0; // "Eclairement" est la variable qui donne la valeur de l'éclairement en lux
float A = 1.015; // "A" est une valeur de la conversion à modifier selon les composants utilisés
float B = 1.753; // "B" est une valeur de la conversion à modifier selon les composants utilisés
float C = 56.655; // "C" est une valeur de la conversion à modifier selon les composants utilisés
float L = 0.0; // "L" est une variable intermédiaire permettant de calculer le produit AxU+B

void setup()
{
  Serial.begin(9600); // la voie série pour monitorer
}

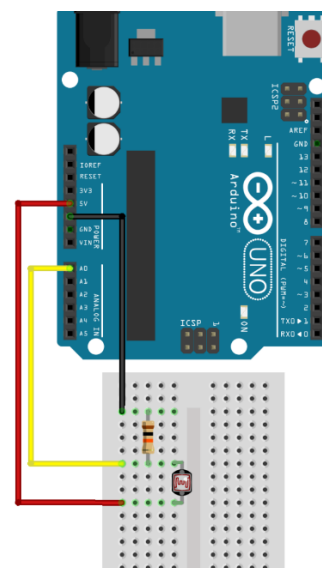
void loop()
{
  tension = (analogRead(capteur) * 5.0) / 1024; // Conversion de la valeur lue par le capteur en tension
  L=A*tension+B;
  eclairement = pow(10,L); // Calcul de la valeur de l'éclairement grace à la formule d'étalonnage
                          // "pow(10,L)" calcule la valeur de 10 à la puissance L

  Serial.print("Tension = "); // Envoie de la valeur de la tension lue vers l'ordinateur via la liaison série
  Serial.print(tension);
  Serial.println(" V");

  Serial.print("Eclairement = "); // Envoie de la valeur de l'éclairement calculée vers l'ordinateur via la liaison série
  Serial.print(eclairement);
  Serial.println(" lux");

  delay(5000); // Délai pour ne prendre des mesures que toutes les 5 secondes
}

```

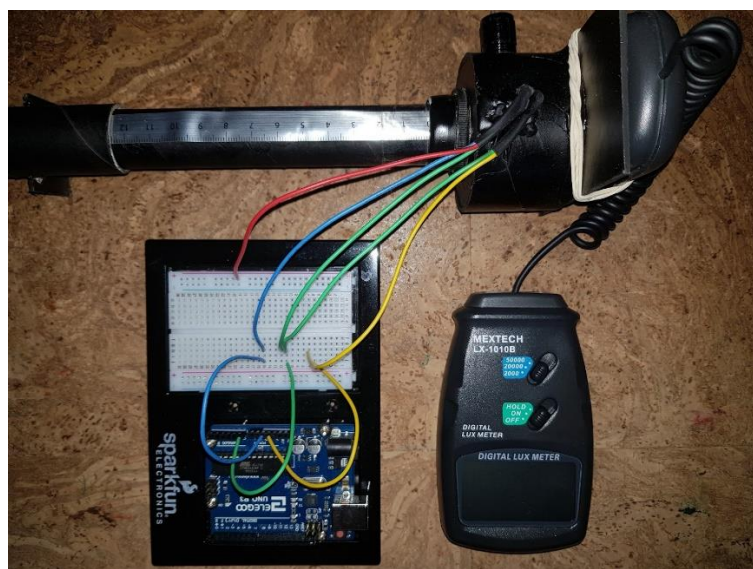


Remarque: Dans ce programme il faut remplacer les valeurs de A, B et C (lignes 10, 11 et 12) par les valeurs calculées avec le programme d'étalonnage de la photorésistance: "Etalonnage photoresistance.py".

4- Expériences

4.1- Etalonnage de la photorésistance

Faire le montage du document 6 afin d'obtenir la courbe d'étalonnage de la photorésistance.



Pour les connexions des fils sur la carte Arduino on devra respecter les codes de couleur suivants:

- Le fil jaune sur l'alimentation "5V".
- Le fil bleu sur la masse "GND".
- Les fils verts au port "A0".
- Le fil rouge n'est pas utilisé.

Faire vérifier le montage par le professeur avant de réaliser les mesures.

- Utiliser le programme Arduino "EtalonnagePhotoresistance.ino" afin de relever les valeurs de la tension U aux bornes du pont diviseur de la photorésistance. Téléverser le programme dans la carte et ouvrir ensuite le moniteur série.
- Allumez la lampe. Faire varier l'éclairage E sur la photorésistance. Pour cela on fait varier la distance D entre la source lumineuse et la photorésistance.
- Relever les différentes valeurs de la tension U et de l'intensité lumineuse E en faisant varier la distance D de 3cm en 3cm entre la source lumineuse et la photorésistance et compléter le tableau ci-dessous.
- Eteignez la lampe.

D (cm)	0	3	6	9	12	15	18	21	24	27	30	33	36
E (Lux)													
U (V)													

- Utiliser le programme Python "Etalonnage photoresistance.py" afin de tracer la courbe d'étalonnage et pour déterminer l'équation spécifique de conversion ainsi que les valeurs des constantes A , B et C utilisées par la suite.
- Pour cela, introduire les valeurs de la tension U et de l'éclairement E du tableau.
- Imprimer ensuite la courbe obtenue (l'équation de la courbe et les valeurs des constantes A , B et C seront indiquées).
- Utiliser le programme Arduino "Luxmètre.ino". Introduire les valeurs des constantes A , B et C . Téléverser le programme dans la carte et ouvrir ensuite le moniteur série.
- Comparer les valeurs relevées avec celle indiquées sur le luxmètre.
- Que constatez-vous?
- Vous venez de fabriquer un luxmètre fonctionnel.