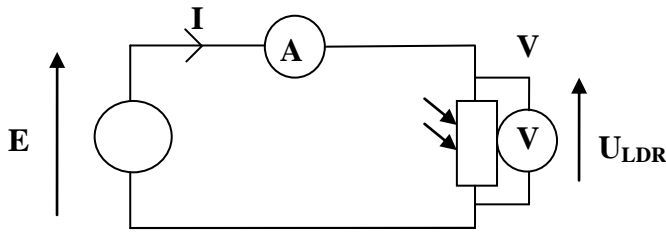


Tracé d'une caractéristique de photorésistance avec Python

1. Mesures



E est la tension aux bornes d'un générateur variable
 U_{LDR} est la tension aux bornes de la photorésistance (capteur) mesurée avec un voltmètre
 I est l'intensité mesurée avec un ampèremètre.

Noter les valeurs mesurées de U pour différentes valeurs de I en faisant varier la tension de l'alimentation. On reporte ensuite ces valeurs dans des listes avec Pyzo.

2. Exécution du code Python

En utilisant un environnement Python adapté (Pyzo par exemple), ouvrir le fichier **Tracé caractéristique $U=f(I)$ avec regression lineaire.py**

```
# -*- coding: utf-8 -*-
"""
Tracé de caractéristique et régression linéaire avec la fonction stats.linregress de scipy
"""
##### IMPORTATION DES BIBLIOTHEQUES ET MODULES #####

import numpy as np # numpy pour les maths, par exemple pour créer 256 valeurs régulièrement espacées entre 0 et 10 : np.linspace(0,10,256)
import matplotlib.pyplot as plt # pour les tracés de graphique

from scipy import stats # module permettant de faire la régression linéaire à partir d'une liste X et d'une liste Y, stats.linregress(X,Y) renvoie 5 valeurs. Les 3 premières valeurs sont la pente, l'ordonnée à l'origine, et le coefficient de corrélation (à mettre au carré)

##### LES VALEURS MESUREES DE I ET U SONT MISES DANS DES LISTES #####

I=[10.4,15.8,19.7,26.4,33.8,41.4,50.5,61.9,12.0,56.0] # Valeurs mesurées de I en mA
U=[1.255,1.92,2.378,3.174,4.075,4.99,6.1,7.46,2.100,6.190] # Valeurs mesurées de U en V
I=[elt/1000 for elt in I] # conversion de I en A (liste modifiée)

##### REGRESSION LINEAIRE ET TRACE DE GRAPHIQUE #####

eq = stats.linregress (I,U) # pour faire la régression linéaire

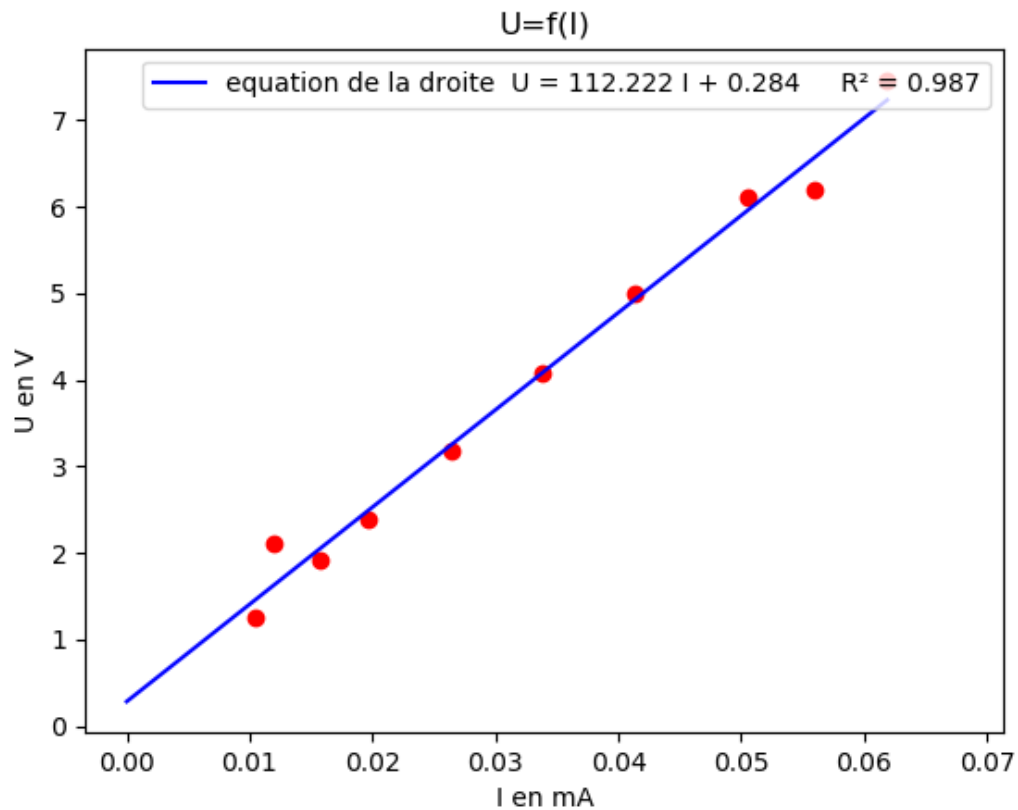
pente = eq[0] # pente
ordorig = eq[1] # ordonnée à l'origine
coeff2 = eq[2]**2 # coefficient de corrélation au carré r²

Xcalc = np.linspace(0,max(I) , 256) # création de points pour le tracé du modèle : on crée 256 points régulièrement espacés entre 0 et la valeur max de I
Ycalc = pente*Xcalc+ordorig # on fait calculer U avec les paramètres de la régression linéaire pour ces valeurs de I
texte = 'equation de la droite U = '+str(round(pente,3))+ ' I + '+str(round(ordorig,3))+ ' R² = '+str(round(coeff2,3))
# on affiche l'équation de la droite avec 3 décimales

print (texte)
plt.title('U=f(I)') # titre du graphique
plt.scatter(I,U, color = 'r', marker = 'o') # On affiche les points de coordonnées (I,U) avec des points rouges
plt.plot(Xcalc,Ycalc,color = 'b',label = texte) # Affichage de la courbe modélisée en bleu
plt.xlabel('I en mA') # nommer l'axe des abscisses
plt.ylabel('U en V') # nommer l'axe des ordonnées
plt.legend() # pour afficher les légendes (label)
plt.show() # afficher le graphique (ne rien mettre dans la parenthèse)
```

Tracé d'une caractéristique de photorésistance avec Python

Exécuter le script pour obtenir la représentation graphique de $y=f(x)$ et le tracé des vecteurs sur Pyzo (raccourci **ctrl+ E**):



Quelques liens utiles :

- Pour installer Pyzo et les modules nécessaires (matplotlib, numpy,...) : <http://maths.spip.ac-rouen.fr/IMG/pdf/pyzo-miniconda-windows.pdf>
- Pour comprendre le langage Python : <https://www.chimsoft.com/python>
- Créer graphique scientifique avec Python : <http://apprendre-python.com/page-creer-graphiques-scientifiques-python-apprendre>