

## Configurations électroniques



Le script ci-dessous définit une fonction qui reçoit comme argument un numéro atomique inférieur à 18 et renvoie la configuration électronique correspondante. Une fois le script exécuté, il suffit d'appeler la fonction dans la console Python.

```
def configuration(Z):
    config = " " #Création de la variable configuration de type chaîne de caractères "string"
    if Z <= 2 : # condition de type if (si..)
        config = "(1s)" + str(Z) #la variable configuration renvoie à la chaîne de caractère (1s) suivi du caractère associé à Z
    elif Z <=4 : # elif est la contraction de else if (sinon si ..)
        config = "(1s)" + str(2) + " (2s)" + str(Z-2)
    elif Z <= 10 :
        config = "(1s)" + str(2) + " (2s)" + str(...)+ " (2p)" + str(...)
    print("La configuration électronique de l'atome de numéro atomique",Z,"est:", config) # écriture du résultat
    return
```

Remplacez les ... par ce qui convient pour permettre d'afficher la configuration électronique des éléments jusqu'à Z = 10 puis complétez ce script pour aller jusqu'à Z = 18.



## Tracé des courbes d'évolution des quantités de matières des espèces chimiques lors d'un titrage

(D'après Physique-Chimie Terminale – Hatier avril 2020)

On titre un volume  $V_1 = 20,0$  mL d'une solution de sulfate de fer (II) par de concentration  $c_1 = 0,10$  mol.l<sup>-1</sup> par une solution d'hydroxyde de sodium de concentration  $c_2 = 0,25$  mol.l<sup>-1</sup>.

On obtient alors un précipité d'hydroxyde de fer (II).

- 1) Établir l'équation de la réaction support du titrage.
- 2) Identifier les ions spectateurs.
- 3) Rappeler la définition de l'équivalence.
- 4) Exprimer puis calculer la valeur du volume équivalent  $V_E$  attendu.
- 5) Exprimer puis calculer la quantité de matière des espèces en solution dans le mélange réactionnel pour un volume de solution titrante versé égal à :

- a)  $V = 0$  mL
- b)  $V = V_E$
- c)  $V = V_{\max} = 25$  mL

6) Le programme ci-contre permet de tracer l'évolution de la composition du mélange réactionnel.

- a) Compléter le programme afin qu'il calcule le volume équivalent du titrage.
- b) Compléter le programme pour qu'il fournisse les listes nhydroxyde, nfer et nproduit contenant chacune trois valeurs de quantité de matière pour :  $V = 0$  mL,  $V = V_E$  et  $V = V_{\max} = 25,0$  mL.
- c) Exécuter le programme et enregistrer le graphique.

7) Modifier le programme pour qu'il permette de visualiser également les quantités de matière des ions spectateurs.

```
import matplotlib.pyplot as plt
print("*****")
print("* Quantités de matière et titrage *")
print("* Réaction support du titrage : *")
print("* Fe2+ + 2HO- --> Fe(OH)2 *")
print("*****Hatier 2020**")

print("")
print("Attention : le séparateur décimal est le point")
print("")

c1=0.1 # Concentration solution titrée en mol/L
V1=20.0 # Volume de solution titrée en mL
c=0.25 # Concentration solution titrante en mol/L
Vmax=25.0 # Volume maximal affiche en mL
Ve= # Calcul du volume équivalent en mL
### Listes des quantités de matière, en mol
### [initiale, à Ve, à Vmax]

### HO-
nhydroxyde=

### Fe2+
nfer=

### Fe(OH)2
nproduit=

V=[0,Ve,Vmax]
### Ecriture des résultats
print("")
print("Volume équivalent :",round(Ve,2),"mL")

### Trace du graphique
plt.plot(V,nhydroxyde,"r",label="Réactif titrant HO-")
plt.plot(V,nfer,"b",label="Réactif titré Fe2+")
plt.plot(V,nproduit,"g",label="Produit Fe(OH)2")
plt.xlabel("V (mL)")
plt.ylabel("quantités de matière (mol)")
plt.legend()
plt.grid(True)
plt.show()
```

## Représentation de vecteurs vitesse et accélération



En reprenant les exercices préparatoires 1 et 2, on peut représenter, pour chaque position le vecteur vitesse à l'aide de la fonction suivante :

```
def representation_vecteur_vitesse(abcisses,ordonnees,vx,vz):
    """ Cette fonction représente un vecteur.
    Elle demande en entrée les coordonnées du point d'application du vecteur
    ainsi que les coordonnées du vecteur.
    L'échelle est à adapter en modifiant le paramètre scale. """
    for i in range(len(vx)):
        plt.quiver(abcisses[i+1], ordonnees[i+1], vx[i], vz[i], scale_units='xy', angles='xy', scale=15)
```

Pour faire afficher le graphique, il faut ensuite ajouter l'instruction `plt.show()` en fin de programme.

1) Comme indiqué en vert, cette fonction utilise l'instruction `plt.quiver` pour tracer une flèche.

- a) Quelle est la signification de la ligne : `for i in range(len(vx)):`  
 b) Pourquoi l'instruction `quiver` utilise-t-elle tantôt les indices `[i+1]` et `[i]` ?

2) Proposer une fonction permettant de calculer les coordonnées du vecteur accélération du système.

3) Proposer une fonction permettant de représenter le vecteur accélération.

## Tracé du diagramme de distribution d'un acide faible



On veut utiliser un script Python pour tracer le diagramme de distribution d'un acide faible. Cet acide faible sera noté AH et sa base conjuguée sera notée A<sup>-</sup>.

- 1) Etablir l'équation de la réaction entre cet acide faible et l'eau et donner l'expression de la constante d'acidité.
- 2) En déduire l'expression du rapport  $[A^-]/[AH]$ .
- 3) Exprimer le pourcentage de forme basique A<sup>-</sup> dans le mélange, ainsi que le pourcentage de forme acide AH.
- 4) Compléter la partie « Calculs préliminaires ».
- 5) Compléter la partie « Tracé du diagramme » en traçant le pourcentage de forme basique et de forme acide en fonction du pH. Attribuer aux axes les noms « pH » et « % ».

```
import matplotlib.pyplot as plt
import numpy as np
pKa = 3.75 # Valeur du pKa de l'acide étudié (ici, l'acide méthanoïque)
def diagramme(pKa):
    ##### Calculs préliminaires #####
    pH = np.arange(0,14.1,0.5) # Crée un tableau de valeurs allant de 0 à 14 par pas de 0,5
    R = # Calculer ici le rapport des concentrations [A-]/[AH]
    P_base = # Calculer ici le pourcentage de base [A-]
    P_acide = # Calculer ici le pourcentage d'acide [AH]

    ##### Tracé du diagramme #####

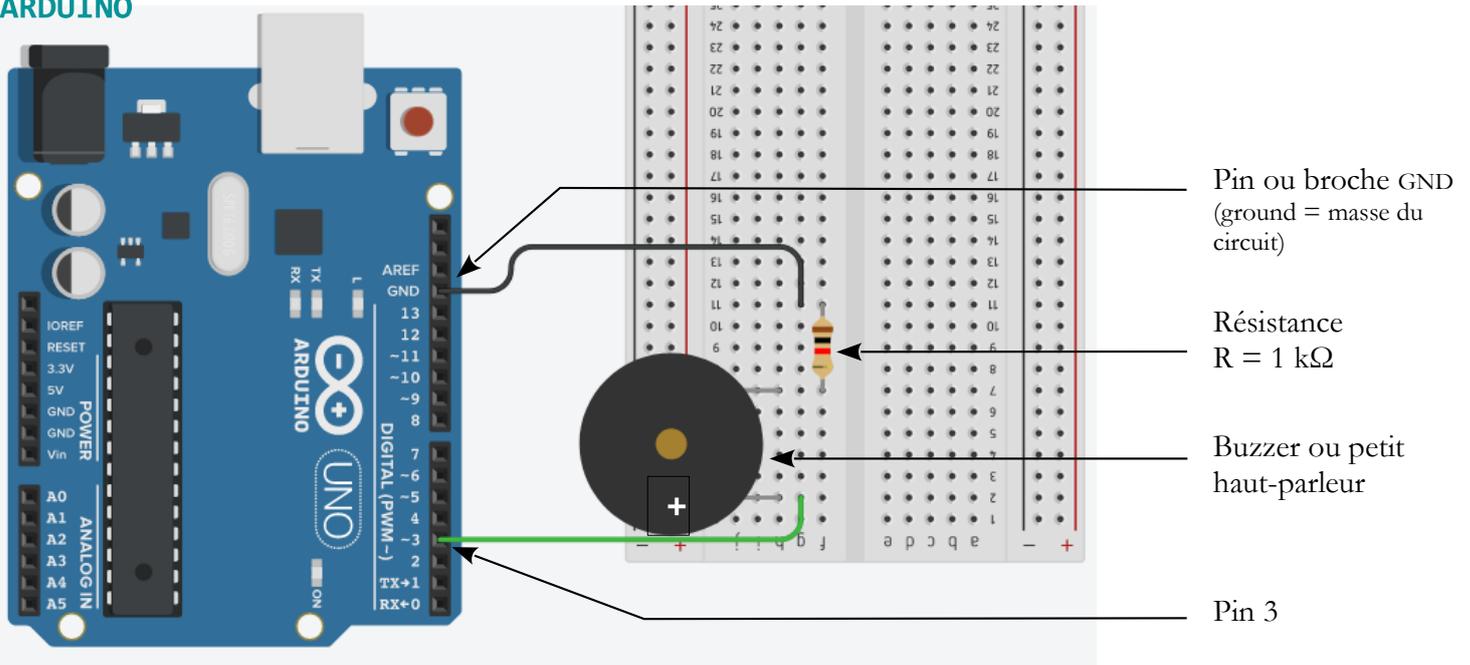
    plt.figure(figsize = (8,4)) # Fixe les dimensions du graphique
    # Tracé du pourcentage de forme basique en fonction du pH
    # Tracé du pourcentage de forme acide en fonction du pH
    # Nom de l'axe des abscisses
    # Nom de l'axe des ordonnées

    plt.legend()
    plt.grid()
    plt.title ("Diagramme de distribution d'un couple acide faible/base faible")
    plt.show()
    return
```

diagramme (pKa)



## Produire un son avec un microcontrôleur Arduino



Pour produire un son avec un microcontrôleur Arduino, on réalise le montage suivant :

Connecter la breadboard à la carte Arduino en reliant la résistance à la masse (pin GND) et le buzzer au pin 3.

Une fois les circuits réalisés, connecter la carte Arduino à l'ordinateur et téléverser le programme `Prod_son.ino` situé dans le dossier de la classe.

Le code de ce programme est donné ci-dessous.

Ce code est commenté pour vous permettre de comprendre le rôle de chaque instruction.

```
void setup() // Cette partie du code s'exécute une seule fois au lancement du programme
{
  pinMode(3,OUTPUT); // On déclare le pin 3 comme étant une borne de sortie
}
void loop() // Cette partie du code s'exécute en boucle
{
  digitalWrite(3,LOW); // Le pin 3 est fixé à l'état bas, c'est à dire à une tension de 0 V
  delayMicroseconds(1136); // On attend 1136 microsecondes
  digitalWrite(3,HIGH); // Le pin 3 est fixé à l'état haut, c'est à dire à une tension de 5 V
  delayMicroseconds(1136); // On attend 1136 microsecondes
}
```

Code de programmation

Commentaires

- 1) Qu'observez-vous une fois le code Arduino téléversé ?
- 2) Représenter l'allure de la tension fournie par la carte Arduino en fonction du temps.
- 3) Déterminer la période puis la fréquence du son produit.
- 4) Comment modifier le code pour entendre un son de fréquence  $f' = 523,25$  Hz.
- 5) Modifier le code et observer.
- 6) À la fin du TP, vider la mémoire du microprocesseur en téléversant un programme vierge.



# Capteur de lumière

Il existe de nombreux dispositifs d'éclairage automatique qui s'allument lorsque la nuit tombe. Quel dispositif permet ce fonctionnement autonome ?

## Doc 1 : La photorésistance

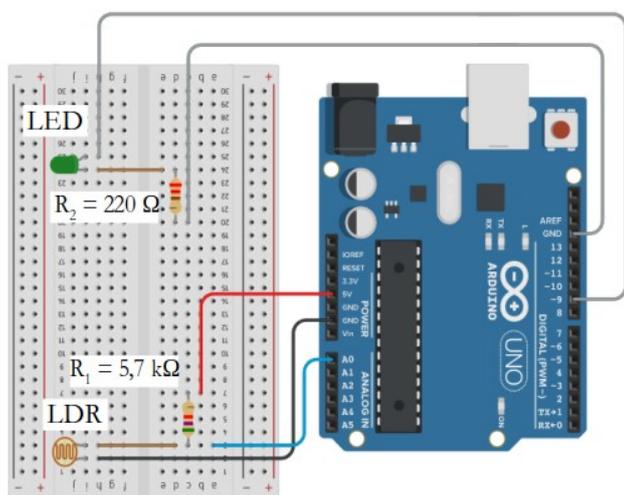
Une photorésistance, appelée aussi LDR (Light Dependent Resistor), est un dipôle dont la valeur de la résistance varie en fonction de l'intensité de la lumière qu'il reçoit. La valeur de la résistance diminue lorsqu'elle est éclairée et elle est maximale en l'absence de lumière.

Les LDR sont couramment présentes dans de nombreux circuits électroniques, en particulier dans les alarmes, les horloges, les luminaires...



## Doc 2 : Montage expérimental :

Les possibilités offertes par le microcontrôleur sont utilisées pour réaliser un circuit où une photorésistance commande l'allumage d'une diode en fonction de l'intensité lumineuse ambiante. Ce circuit est généralement appelé interrupteur crépusculaire et se présente de la façon suivante.



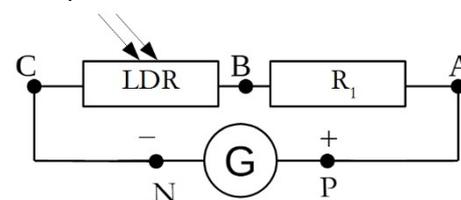
## Doc 3 : Approche de programmation de microcontrôleur Arduino

On utilise le programme ci-dessous et le microcontrôleur Arduino pour réaliser un circuit.

```

1 // définition et initialisation des variables
2 int analogPin = 0 ; //utilisation de l'entrée A0
3 int ledPin = 9 ; //utilisation du pin 9 pour connecter la LED
4 int Valeur = 0 ;
5 float Tension = 0 ;
6
7 void setup() {
8   Serial.begin(9600) ; // connexion carte-ordinateur
9   pinMode(ledPin, OUTPUT) ; // le pin connecté à la LED est une sortie digitale
10  digitalWrite(ledPin, LOW) ;
11 }
12
13 void loop() {
14   //mesure de la valeur pour l'entrée analogique A0
15   Valeur = analogRead(analogPin) ;
16   Tension = Valeur*5.0/1023 ; // calcul de la valeur de la tension
17
18   Serial.print(Tension) ;
19   Serial.println(" V" ) ;
20   delay (1000) ; //1000 ms entre deux mesures successives
21
22   // choix du seuil de basculement de la LED ici 3,5 V
23   if (Tension < 3.5) {
24     digitalWrite(ledPin, LOW) ;
25   }
26   else{
27     digitalWrite(ledPin, HIGH) ;
28   }
29 }
    
```

- 1) Le montage ci-dessus est déjà partiellement réalisé :  
la LED et la résistance  $R_2$  ont déjà été connectés à la breadboard et au microcontrôleur. Compléter le montage en plaçant la photorésistance et la résistance  $R_1$ .
- 2) Téléverser le programme dans le microcontrôleur et vérifier qualitativement que le dispositif obtenu fonctionne comme un interrupteur crépusculaire.
- 3) Prévoir comment varie la valeur de la résistance de la LDR (notée  $R_{LDR}$ ) si la luminosité diminue.
- 4) Dans ce circuit,  $U_{LDR}$  varie dans le même sens que  $R_{LDR}$ . Comment doit varier la tension  $U_{LDR}$  aux bornes de la photorésistance quand l'éclairage diminue ? Vérifier en masquant la LDR et en notant les valeurs de tensions  $U_{LDR}$  pour différents éclairages (pleine lumière, légère ombre, obscurité).
- 5) Pour déclencher l'allumage de la LED, le microcontrôleur Arduino mesure la tension entre les bornes de la photorésistance. Ouvrir le "Moniteur Série" Arduino et visualiser la tension mesurée par le microcontrôleur.
- 6) Déterminer la valeur de la tension  $U_{LDR}$  à partir de laquelle la diode s'éclaire. Cette valeur est-elle cohérente avec la valeur de seuil donnée dans le programme (doc. 3) ?
- 7) a) Comment modifier le programme pour qu'une ombre légère suffise à provoquer l'allumage de la LED.  
b) Effectuer la modification et vérifier que la nouvelle valeur de seuil choisie est bien respectée.
- 8) La partie détection du montage peut être schématisée ainsi :  
La tension  $U_{PN} = 5,0 V$  est la tension fournie par le microcontrôleur.  
a) Exprimer littéralement la valeur de la tension  $U_{AB}$  en fonction de  $U_{PN}$  et de  $U_{LDR}$ . Justifier en explicitant la loi utilisée.  
b) En supposant  $U_{LDR} = 3,5 V$ , déterminer la valeur de  $U_{AB}$ .  
c) En déduire, en justifiant, la valeur de l'intensité traversant la résistance  $R_1$  et la photorésistance.
- 9) Dans la seconde partie du montage, déterminer par une mesure de tension, l'intensité traversant la LED lorsque celle-ci est allumée.





## Tracé du diagramme de distribution d'un diacide

Le programme suivant permet de tracer le diagramme de distribution de l'acide méthanoïque.

Adaptez ce programme pour qu'il puisse tracer le diagramme de distribution d'un diacide faible comme l'acide carbonique ( $pK_{a1} = 6,37$  et  $pK_{a2} = 10,32$ ).

```
import matplotlib.pyplot as plt
import numpy as np
pKa = 3.75
def diagramme(pKa):
    pH = np.arange(0,14.1,0.5) # Crée un tableau de valeurs allant de 0 à 14 par pas de 0,5
    R = 10**(pH-pKa)
    P_base = 100*R/(R+1)
    P_acide = 100/(R+1)

    ##### tracé du diagramme
    #####

    plt.figure(figsize = (8,4))
    plt.plot(pH,P_acide,"rx-",label="acide")
    plt.plot(pH,P_base,"b+-",label="base")
    plt.xlabel("pH")
    plt.ylabel("%")
    plt.legend()
    plt.grid()
    plt.title ("Diagramme de distribution d'un couple acide faible/base faible")
    plt.show()
    return
diagramme (pKa)
```