

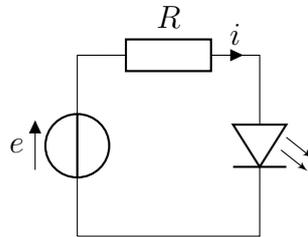
## TP - Émission - Réception (1)

Dans ce TP, nous allons utiliser un micro-contrôleur (carte Arduino) afin de mesurer l'éclairement moyen émis par une LED.

Les bases des scripts Arduino sont données, vous devrez les modifier pour que le système remplisse la fonction demandée. À la fin du TP, vous pourrez explorer plusieurs pistes en autonomie en respectant les branchements introduits dans le TP. Dans la dernière partie, on réalise un filtrage numérique des données : utiliser le TP précédent pour réaliser un traitement des données obtenues (spectres des signaux suivant la valeur du paramètre  $\alpha$  ?).

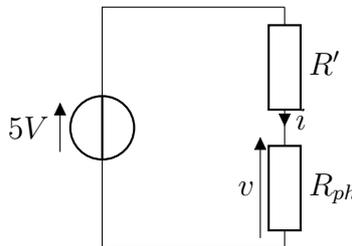
### Montage émetteur

Réaliser le montage ci-dessous. La résistance  $R = 220\Omega$  permet de limiter le courant dans la diode et on utilise la sortie TTL du GBF. Choisir une fréquence assez basse pour pouvoir observer le clignotement de la LED. Éteindre ensuite le GBF le temps de préparer la suite de l'expérience.



### Montage récepteur

1. Mesurer à l'aide d'un ohmmètre les variations de résistances  $R_{ph}$  de la photorésistance suivant l'éclairement auquel elle est soumise.
2. Réaliser le pont diviseur ci-dessous en prenant  $R'$  de l'ordre de la valeur max de  $R_{ph}$ , le montage est alimenté par la sortie 5V du micro-contrôleur. Mesurer la tension  $v$  aux bornes de la photorésistance suivant l'éclairement auquel elle est soumise. Vérifier que les variations de tension  $v$  permettent bien de remonter à l'intensité de l'éclairement (sans pour autant pouvoir déterminer une valeur de l'éclairement).
3. Faire clignoter la LED en face de la photorésistance et observer les variations de la tension  $v$  à l'aide de l'oscilloscope. Commenter.



### Acquisition directe du signal

Les programmes Arduino sont composés de trois parties :

- La déclaration des variables utilisées dans le programme ;
- Le **setup** : instructions qui ne sont réalisées qu'en début de programme ;
- La **loop** : qui tourne "à l'infini", *sauf si on ajoute un while ... avec un compteur qui est incrémenté au cours du fonctionnement, on peut aussi faire en sorte que la mesure / l'affichage ... soit conditionné par l'appui sur un bouton.*

Le programme ci-après permet de mesurer le signal **analogique**  $v$  (instruction `analogREAD`). Ce signal qui doit être compris entre 0 et 5V est alors converti en un nombre compris entre 0 et 1023 (signal **numérique** codé sur 8 bits). Les mesures sont séparées d'un nombre de millisecondes fixé par le nombre `Intervalle`. L'instruction `Serial.println` permet d'afficher la valeur mesurée dans le moniteur série ou dans le traceur série.

Suivant l'expérience réalisée, il peut être nécessaire d'attendre la fin des mesures pour les afficher (pour ne pas perdre de temps entre deux mesures). On peut aussi donner des données brutes (comprises entre 0 et 1023) qui seront converties en V une fois l'expérience terminée (ici c'est le facteur  $k$  qui réalise la conversion).

```
const int photoR=3;
long newMeas;
int Intervalle=100;
const float k=5./1023.;

void setup () {
  pinMode(photoR,INPUT);
  Serial.begin(9600);
}

void loop() {
  newMeas=analogREAD(photoR);
  Serial.println(newMeas*k);
  delay(Intervalle);
}
```

Modifier le programme afin qu'il réalise  $N=20$  mesures. On peut aussi ajouter une mesure de l'instant auquel la mesure a été effectuée :

```
tempsMesure=millis();
Serial.print(tempsMesure);
Serial.print("\t");
Serial.print(newMeas*k);
Serial.print("\n");
```

## Filtre exponentiel

On propose ici de modifier le programme précédent en rajoutant une étape de filtrage numérique. Le filtre que nous allons utiliser est appelé filtre ou lissage exponentiel. On introduit un paramètre  $\alpha$  compris entre 0 et 1. On note  $v_i$  la valeur mesurée à l'instant  $t_i$  et  $s_i$  la valeur filtrée :

$$s_0 = v_0$$

$$s_i = \alpha v_i + (1 - \alpha) s_{i-1}$$

Plus  $\alpha$  est faible plus le lissage est important. Modifier le programme pour qu'il affiche  $s$  en plus de  $v$ . Commencer avec  $\alpha = 0,5$  et réaliser plusieurs mesures en modifiant  $\alpha$ .

- Créer un fichier `.csv` à partir des résultats et tracer le spectre de  $s$ .
- Utiliser le microcontrôleur pour faire clignoter la LED (sortie numérique).
- Le microcontrôleur possède également des sorties analogiques (`pwd`) : expliquer leur fonctionnement. Proposer une application.
- On peut ajouter un bouton poussoir pour commander l'expérience : entrée numérique. Proposer un montage et les instructions correspondantes.