THÈME 5

# PROJET EXPÉRIMENTAL ET NUMÉRIQUE

SOMMAIRE



## A LES CAPTEURS

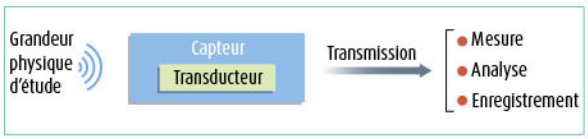
### 1 Qu'est-ce qu'un capteur?

Un capteur est un dispositif qui délivre, à partir d'une grandeur physique, une autre grandeur, souvent électrique, fonction de la première grandeur et directement utilisable pour la mesure ou la commande.

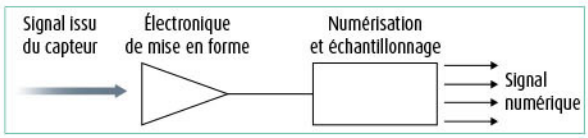
La brique élémentaire du capteur est appelée transducteur. C'est le dispositif qui réagit à la grandeur physique que l'on souhaite mesurer et qui fournit un signal usuellement électrique. Autour du transducteur, une électronique minimale permet une mise en œuvre effective (alimentation, amplification éventuelle, etc.) pour former le capteur proprement dit (Doc 1).

Les capteurs sont utilisés pour quantifier des grandeurs physiques et éventuellement pour les enregistrer au cours du temps. Le signal issu du capteur peut prendre des formes très variées, et il peut être nécessaire de le mettre en forme et de le numériser (Doc 2).

On distingue deux grandes familles de capteurs : les capteurs à réponse analogique et les capteurs à réponse numérique.




**DOC 1** Principe d'un capteur.




**DOC 2** Transformation du signal issu du capteur en signal numérique.

## 2 Les capteurs à réponse analogique

Le signal de sortie fourni par un capteur à réponse analogique est une tension dont la valeur est directement liée à la grandeur physique d'intérêt.

Pour la mesure de la température, la thermistance (Doc 3) possède une résistance dont la valeur dépend de la température. Cette valeur peut être mesurée par un dispositif électronique simple. Pour la mesure de la lumière, la photodiode (Doc 4) fournit un courant lié à l'éclairement incident. Pour mesurer un son, le microphone fournit une tension liée au signal sonore reçu.

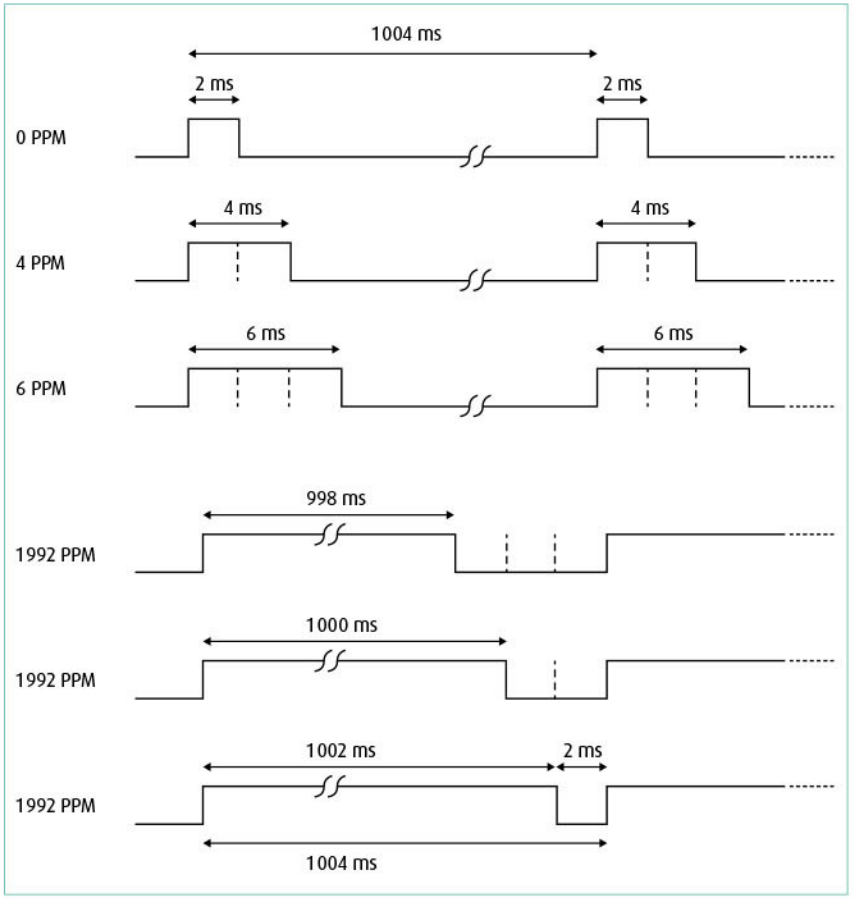
**DOC 3** Une thermistance.

## 3 Les capteurs à réponse numérique

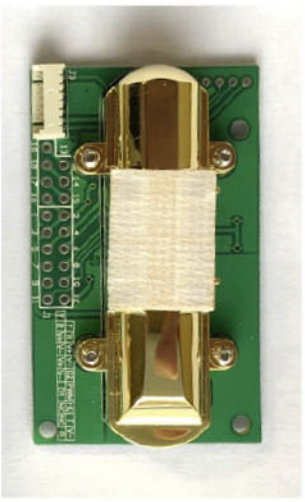
Le signal de sortie fourni par un capteur à réponse numérique peut prendre deux formes différentes :

- Un signal carré dont l'une des propriétés est liée à la grandeur physique d'intérêt. Par exemple, un capteur de $CO_2$ (Doc 6) fournit des impulsions numériques dont la largeur dépend du taux de $CO_2$ (Doc 5).
- Un nombre lié à la grandeur physique d'intérêt. C'est le cas lorsque le transducteur est directement intégré à un dispositif de numérisation. Par exemple, un capteur CCD (Doc 7) fournit la représentation numérisée d'une image, c'est-à-dire un nombre associé à chaque pixel.

**DOC 4** Une photodiode.




**DOC 5** Exemples d'impulsions numériques fournies par un capteur de $CO_2$.



**DOC 6** Un capteur de $CO_2$.

**DOC 7** Un capteur CCD.

# B ÉTALONNAGE DE LA RÉPONSE DU CAPTEUR

Le capteur ne fournit pas directement la grandeur physique d'intérêt avec son unité. Il faut faire correspondre sa réponse (le signal qu'il produit) à la grandeur physique. On parle alors d'étalonnage, ou de calibration, terme anglais souvent mais abusivement utilisé. La procédure d'étalonnage dépend de la connaissance du capteur et de sa réponse.

## 1 La réponse du capteur est une loi connue

Cette loi peut être linéaire ou non, il s'agit de se fier aux données du constructeur disponibles sur la fiche technique. Par exemple, pour une thermistance, on calcule la température T grâce à la mesure de la résistance R et à la formule suivante :

$$T(R) = \frac{1}{\frac{\ln\left(\frac{R}{R_{25}}\right)}{B_{25}} + \frac{1}{T_N}}$$

Avec :

$R_{25}$ = résistance de référence (en Ω)

$T_N$ = température de référence (en Kelvin)

$B_{25}$ = constante de température (en Kelvin)

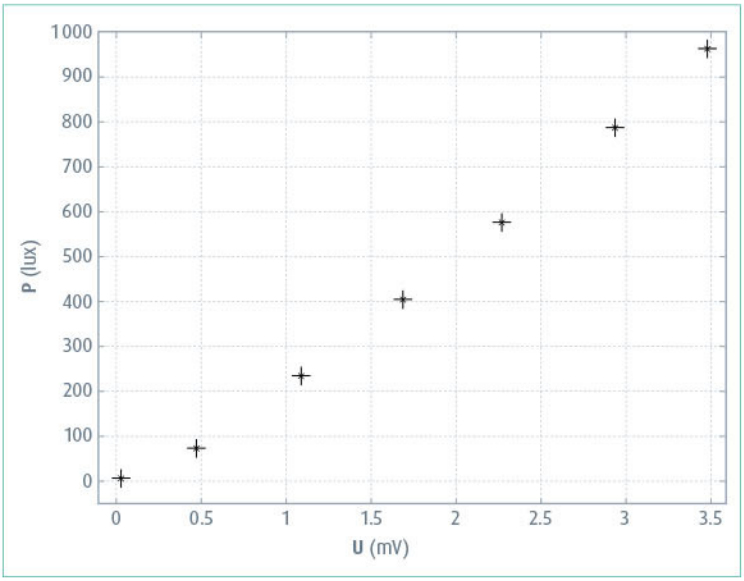
Ces trois paramètres sont fournis par le constructeur dans la fiche technique. Les fiches techniques, ou datasheet (Doc 10), sont le plus souvent rédigées en anglais. On y trouve :

- le domaine d'application typique (applications) ;
- les caractéristiques principales (features) ;
- les dimensions mécaniques ;
- les caractéristiques et limitations ;
- les données nécessaires à l'étalonnage, éventuellement les courbes de réponse du capteur ;
- éventuellement des exemples d'utilisation pratique.

## 2 La réponse du capteur n'est pas connue

Dans ce cas, il faut créer un étalonnage à partir d'un appareil de référence. Supposons que l'on trouve des photodiodes dans les stocks de la salle de travaux pratiques, mais que la documentation du constructeur ait été égarée. Si l'on dispose d'un luxmètre (Doc 8), il est possible d'établir une correspondance pratique (Doc 9) entre la réponse de la photodiode et la mesure effectuée par le luxmètre qui sert d'appareil de référence.


DOC 8 Un luxmètre.




DOC 9 Un exemple de courbe d'étalonnage.

EPCOS

**Temperature Measurement** B57164

**Leaded Disks** K 164

### Applications

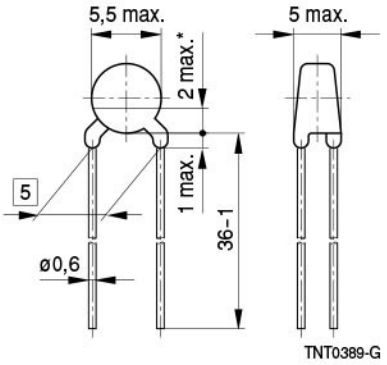
- Temperature compensation
- Temperature measurement
- Temperature control

### Features

- Wide resistance range
- Cost-effective
- Lacquer-coated thermistor disk
- Tinned copper leads
- Lead spacing 5,0 mm
- Marked with resistance and tolerance

### Delivery mode

Bulk (standard),
cardboard tape, reeled or in Ammo pack




*May be free of lacquer

Dimensions in mm
Approx. weight 0,4 g

$T_N = 25\,°C + 273{,}15 = 298{,}15\,K$

| | | | |
|---|---|---|---|
| Climatic category (IEC 60068-1) | | 55/125/21 | |
| Max. power at 25 °C | $P_{25}$ | 450 | mW |
| Resistance tolerance | $\Delta R_N/R_N$ | ± 5 %, ± 10 % | |
| Rated temperature | $T_N$ | 25 | °C |
| *B* value tolerance | $\Delta B/B$ | ± 3 % | |
| Dissipation factor (in air) | $\delta_{th}$ | approx. 7,5 | mW/K |
| Thermal cooling time constant (in air) | $\tau_c$ | approx. 20 | s |
| Heat capacity | $C_{th}$ | approx. 150 | mJ/K |

| $R_{25}$ Ω | No. of *R/T* characteristic | $B_{25/100}$ K | Ordering code |
|---|---|---|---|
| 15 | 1203 | 2900 | B57164K0150+000 |
| 22 | 1203 | 2900 | B57164K0220+000 |
| 33 | 1203 | 2900 | B57164K0330+000 |
| 47 | 1302 | 3000 | B57164K0470+000 |
| 68 | 1303 | 3050 | B57164K0680+000 |
| 100 | 1305 | 3200 | B57164K0101+000 |
| 150 | 1305 | 3200 | B57164K0151+000 |
| 220 | 1305 | 3200 | B57164K0221+000 |
| 330 | 1306 | 3450 | B57164K0331+000 |
| 470 | 1306 | 3450 | B57164K0471+000 |

Pour ce modèle de thermistance :
$R_{25} = 68\ \Omega$
$B_{25} = 3050\ K$

**DOC 10** Datasheet des thermistances de type B57164K.

# C LA CARTE ARDUINO ET SA PROGRAMMATION

Une carte Arduino est un exemple de microcontrôleur : c'est un composant électronique (circuit intégré) que l'on peut programmer afin d'acquérir et de produire ces signaux électriques.

Il existe de nombreux microcontrôleurs différents et une gamme très large de cartes (microbit, arduino, STM32, ESP32 etc.). Les cartes de type Arduino sont très répandues, peu chères et une grande quantité d'information est disponible à leur sujet sur internet. Leurs propriétés et leur prix varie

**LEDs de fonctionnement :**
**ON :** Alimentation correcte de la carte
**TX et RX** s'allument lors du fonctionnement de la communication série.
**L** s'allume lorsque le pin D13 est à 5V et s'éteint lorsqu'il est à 0V.

**BUS SPI** pour communiquer avec certains **capteurs digitaux.**

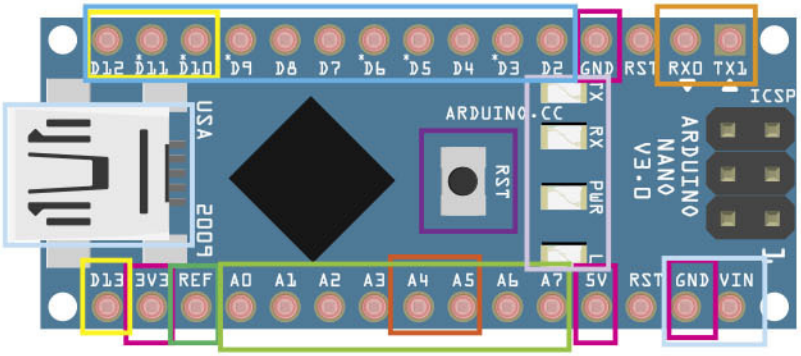
**Entrées et Sorties digitales 2 à 13**

Elles peuvent servir de **sortie,** c'est à dire d'alimentation. Elles sont utilisables en **tout ou rien** (0 ou 5V) ou en **PWM** (modulation de largeur d'implusion) permettant de piloter l'intensité lumineuse d'une LED par exemple.

Elles peuvent aussi servir d'**entrée.**

**BUS UART** pour la communication série.

Attention, si on utilise l'un de ces pins, la communication série par l'USB ne fonctionne plus.




**Alimentations** utilisables pour les circuits extérieurs en **5V** et **3,3 V.** Les différents pins GND sont reliés et peuvent donc être utilisés indifféremment comme **masse.**

Si on ne branche rien ici, la numérisation est réalisée entre 0 et 5V. Si on alimente ce pin avec 1V, celui-ci pourra servir de **référence** et on pourra alors numériser avec 1024 niveaux de quantification entre 0 et 1 V.

**Entrées analogiques 0 à 7**
Elles permettent de mesurer une tension (entre le pin utilisé et le GND de la carte) dans la gamme 0 – 5V. Elles sont utiles pour les **capteurs analogiques.** Attention à ne pas utiliser de tension en dehors de cette gamme.
Elles sont toutes reliées à un même convertisseur analogique numérique de 10 bits. La tension est donc quantifiée sur $2^{10} = 1024$ niveaux ou le niveau 0 correspond à 0V et le niveau 1023 à 5V.
La fréquence d'échantillonnage est au maximum de **9600 Hz.**

**Bouton Reset**
Il permet de redémarrer la carte. Le programme présent dans la carte redémarre mais n'est pas modifié.

**Pour alimenter la carte,** on peut utiliser le **port USB** ou les pins **VIN** et **GND.** Mais dans ce deuxième cas, il faudra utiliser une alimentation **entre 6V et 20V.**

**BUS I2C** pour communiquer avec certains **capteurs digitaux.**

**DOC 11** Carte Arduino NANO.

*Logiciel de programmation des cartes Arduino*

selon le modèle (Arduino UNO, DUE, MEGA, etc..). Nous nous intéresserons ici uniquement aux cartes Arduino UNO et NANO (Doc 11), les moins chères et les moins puissantes, qui suffisent largement pour démarrer et mener à bien de nombreux petits projets.

Une carte Arduino se programme avec un logiciel spécifique accessible en ligne. L'interface de programmation est présentée dans le Doc 12, et la structure d'un programme Arduino est présentée dans le Doc 13.

**Bouton «vérifier»** qui permet de vérifier le code

**Bouton « téléverser»** qui permet d'envoyer le code compilé vers la carte arduino

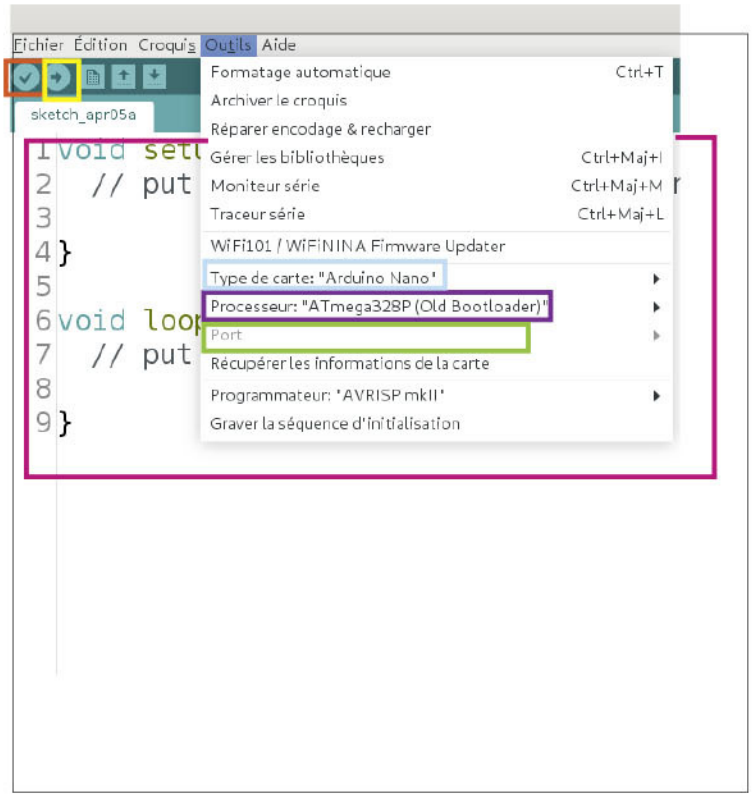
**La communication** entre l'ordinateur et la carte ne peut se faire que si la carte est branchée via un câble USB et si :

**« Type de carte »**

**« Processeur »**

**« Port »**

du menu « outils » sont bien renseignés.
En cas de problèmes de téléversement, vérifiez bien ces trois paramètres.
Le port utilisé peut être vérifié dans le gestionnaire de périphériques sous Windows.




**Toute cette zone permet d'écrire le programme.**
Lorsque le logiciel arduino reconnaît des mots clés, ils changent de couleur. Si ce n'est pas le cas, vous avez sans doute fait une erreur. Vérifiez les majuscules par exemple.

Il est possible d'enregistrer ou de charger un programme à partir du menu « fichier » puis « ouvrir » ou « enregistrer sous ».

**DOC 12** Interface de programmation Arduino.

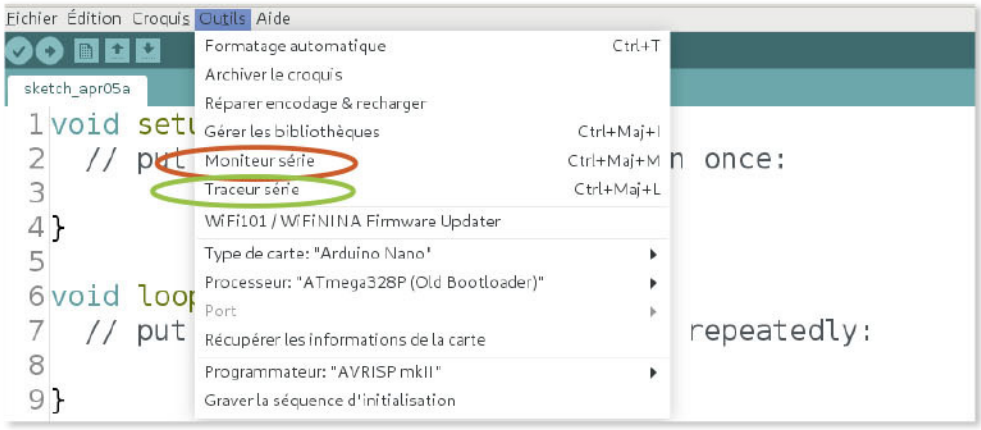

**DOC 13** **Structure d'un programme Arduino.**

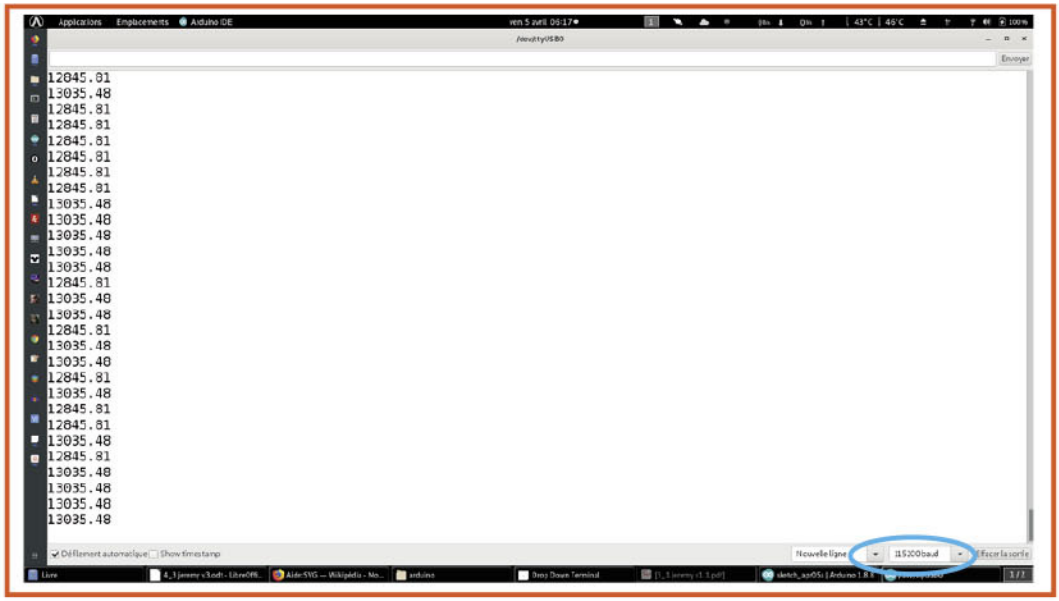
# D LE TRAITEMENT DES DONNÉES

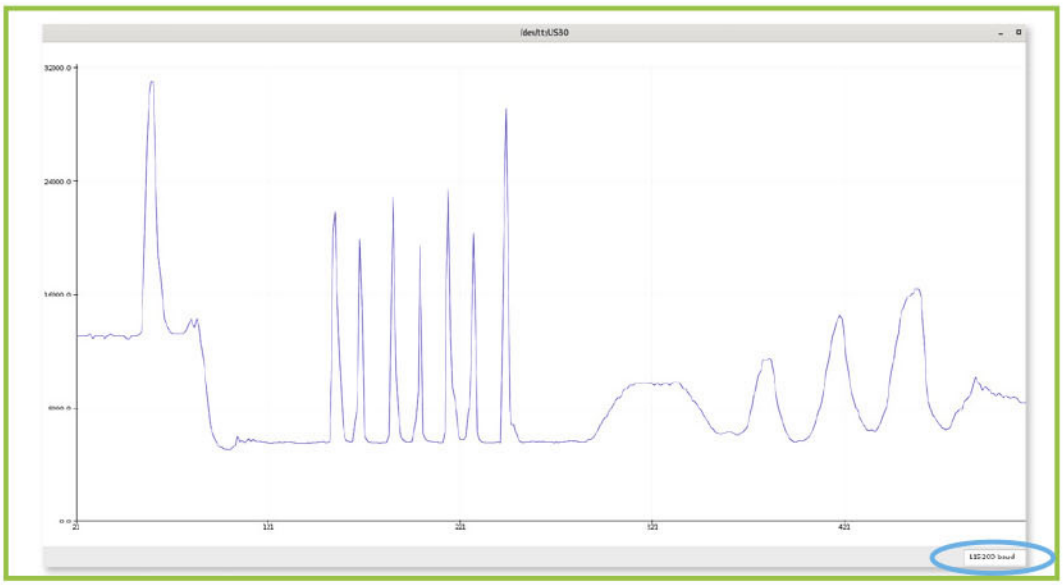
## 1 Afficher les données renvoyées par la carte Arduino

Une fois le type de carte, le processeur et le port sélectionnés, le logiciel Arduino permet d'afficher les données renvoyées par la carte de deux façons :
– le moniteur série affiche la liste des valeurs renvoyées;
– le traceur série affiche une courbe sur une fenêtre temporelle glissante de 5 minutes.

Toutes deux sont accessibles via le menu «Outils». Que ce soit pour le moniteur série ou le traceur série, il ne faut pas oublier de régler le débit pour que les informations s'affichent correctement (Doc 14).




**DOC 14** **Exemple d'affichage des données.**

## 2 Récupérer les données au format csv avec *CoolTerm*

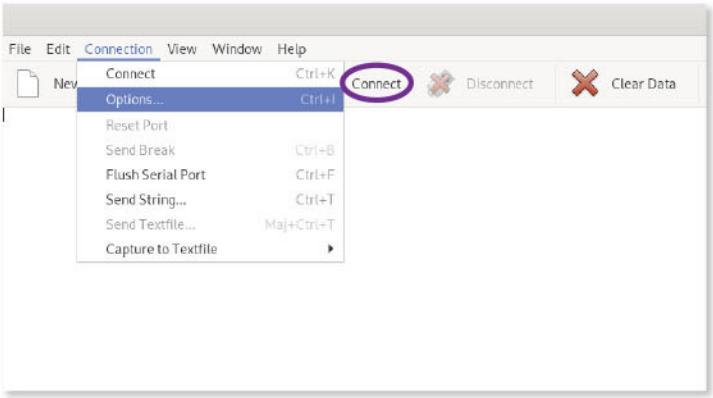
Afin de récupérer les données issues d'un capteur Arduino pour un traitement complémentaire, il est possible soit de sélectionner et de copier les valeurs depuis le moniteur série, soit d'utiliser un logiciel de terminal série comme CoolTerm.

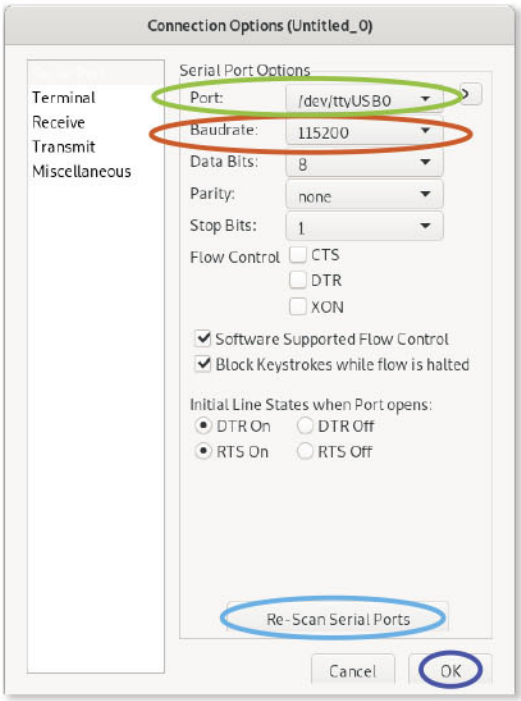
Commencez par brancher la carte Arduino à l'ordinateur en utilisant un cable USB. Puis, dans le menu «Connection», sélectionnez «Options...». Réglez Baudrate et Port d'après le débit de la connexion série qui a été programmé dans la carte Arduino et le port attribué par le système (**Doc 15a**). Si le bon port COM n'apparaît pas, cliquez sur Re-Scan Serial Ports. Une fois les réglages terminés, cliquez sur OK, la fenêtre d'options disparaît, puis cliquez sur Connect. Les données doivent alors défiler comme dans le moniteur série du logiciel Arduino (**Doc 15b**).

Pour réaliser une acquisition, utilisez le menu Connection, puis Capture to Textfile, et enfin Start. L'acquisition pourra être arrêtée en utilisant le Stop (**Doc 15b**).

Réglez le nom du fichier puis cliquez sur Enregistrer. Ne pas oublier de sélectionner «All Files» à la place du «text File» par défaut en bas de la fenêtre et d'ajouter l'extension «.csv» à la fin du nom du fichier (**Doc 15c**).

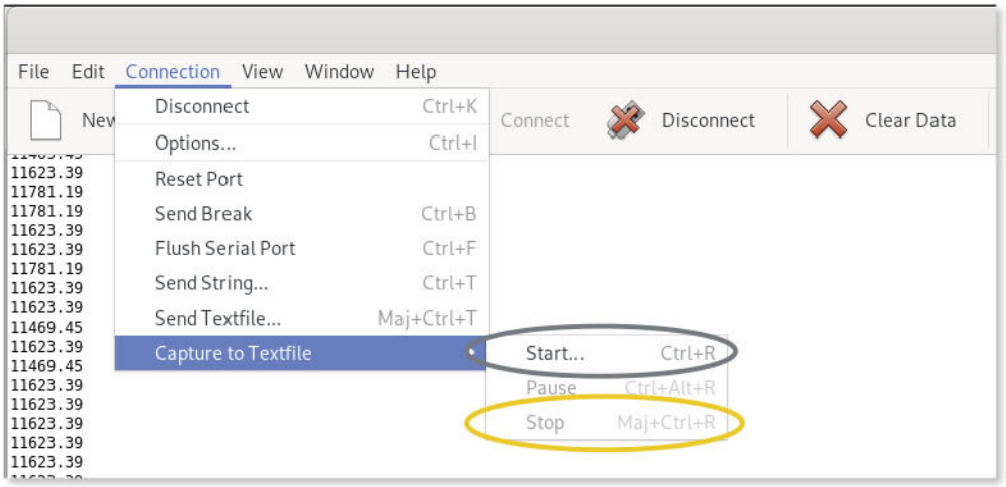
Le fichier csv obtenu est lisible avec un tableur, ou peut être ouvert avec Python pour des traitements ultérieurs (moyenne, écart type, tracé de courbes, d'histogrammes etc.).



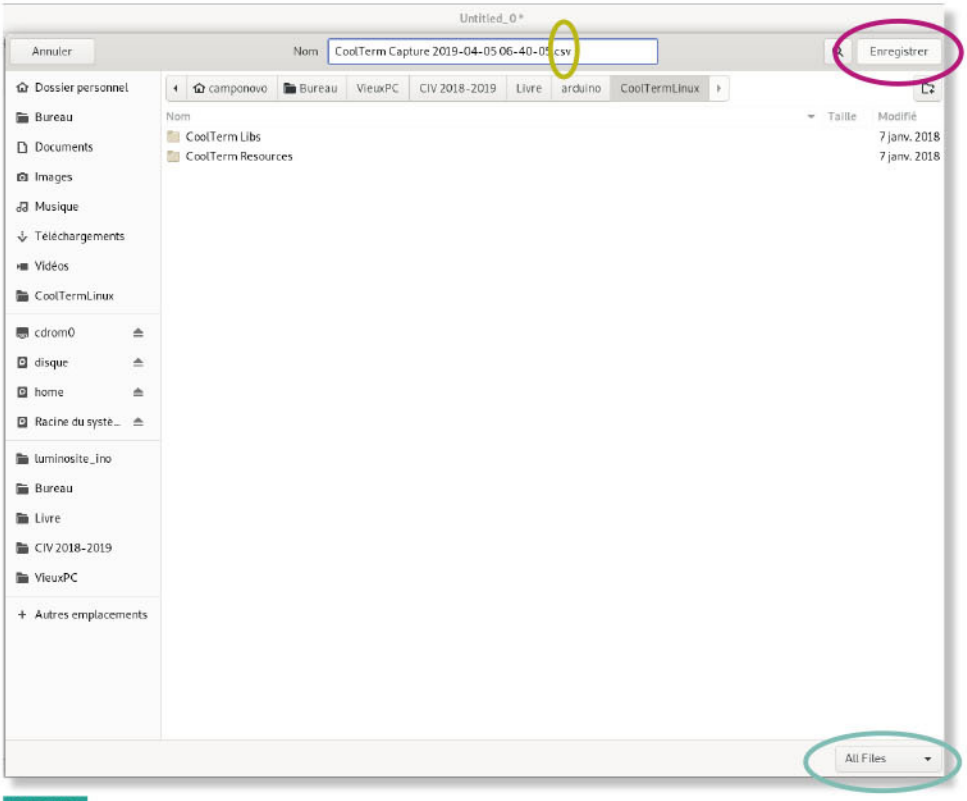




**DOC 15a**

*Logiciel gratuit CoolTerm*




**DOC 15b**

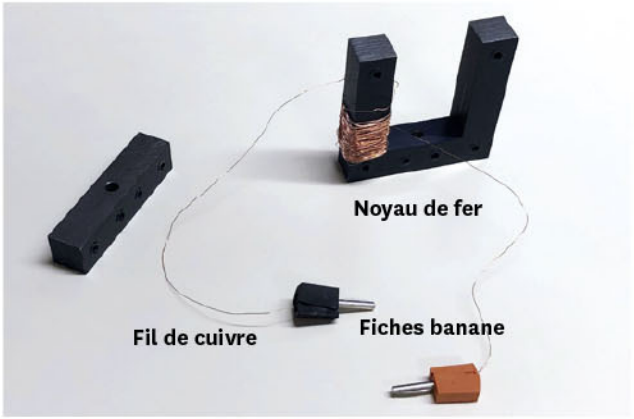



**DOC 15c**

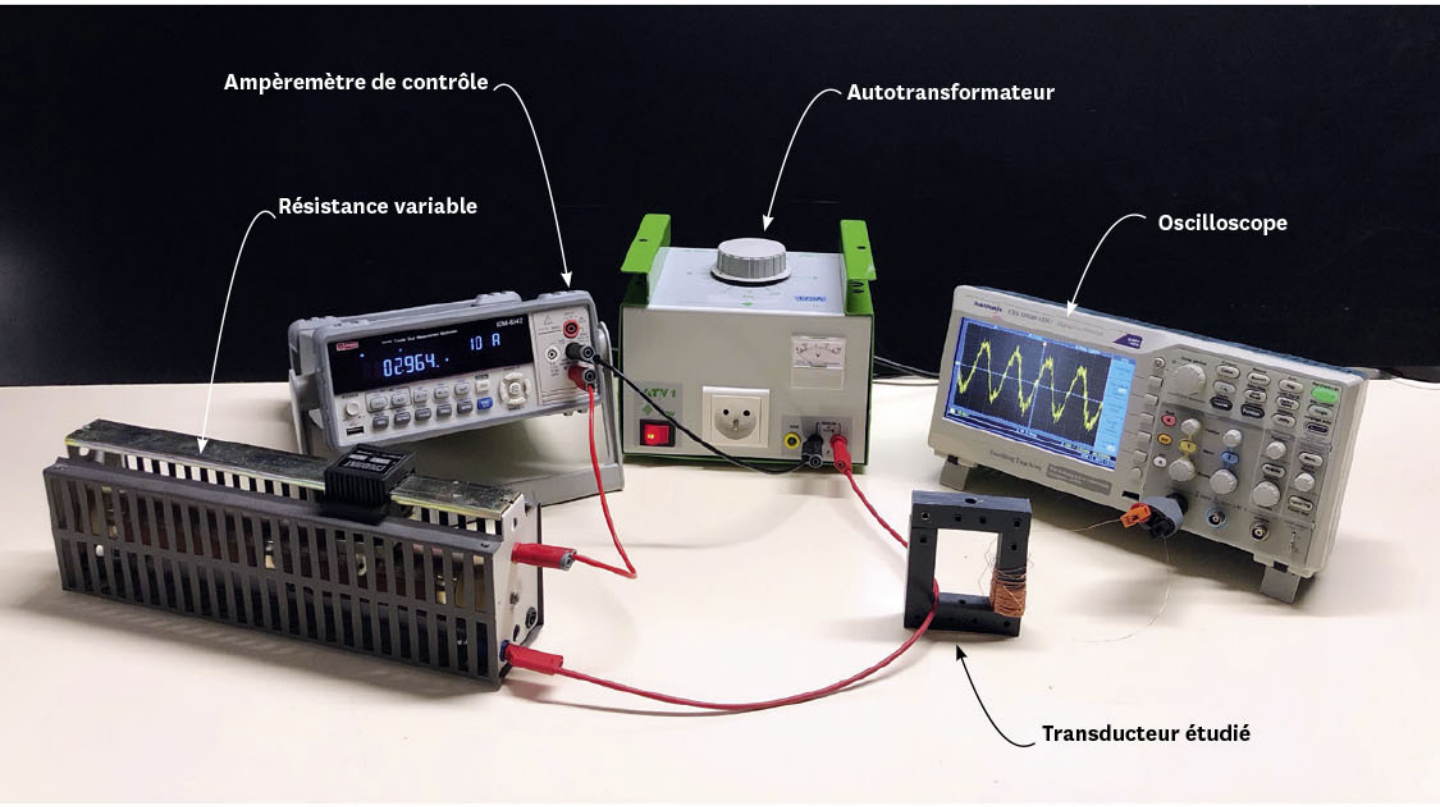
**DOC 15** **Récupération des données d'un capteur Arduino avec le logiciel *CoolTerm*.**

# E FABRIQUER UN CAPTEUR

On peut réaliser très facilement un capteur de courant alternatif, composé d'un transducteur et d'une électronique de mise en forme. Pour le tranducteur courant alternatif/tension, on enroule 100 spires de fil de cuivre autour d'un noyau de fer (Doc 16). On referme alors le noyau et l'on fait passer en son centre un fil dans lequel circule un courant alternatif à 50 Hz (fil rouge sur le Doc 17). En mesurant à l'oscilloscope la tension disponible aux bornes du fil de cuivre, on visualise une image du courant qui circule dans le câble rouge. La réponse du transducteur ainsi créé peut être étalonnée avec un ampèremètre externe.

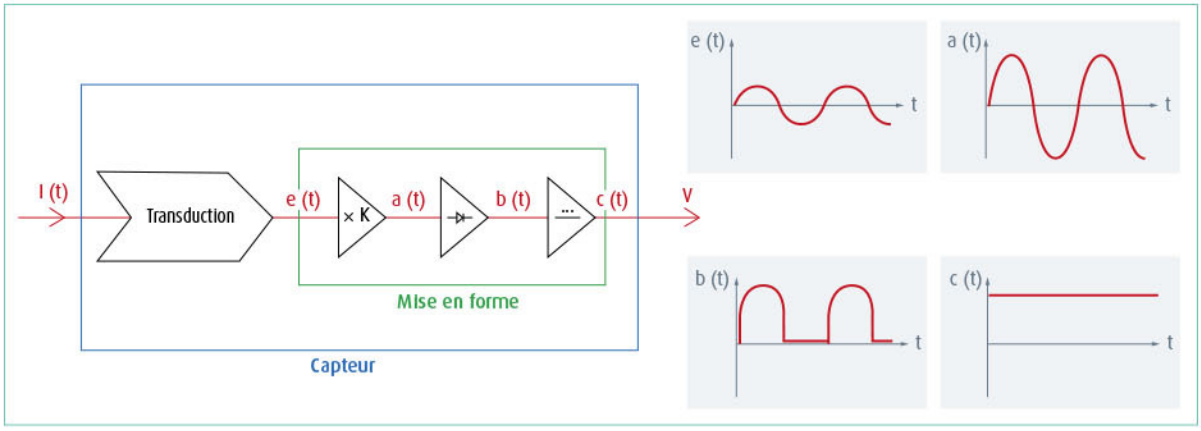



DOC 16 Fabrication du transducteur.

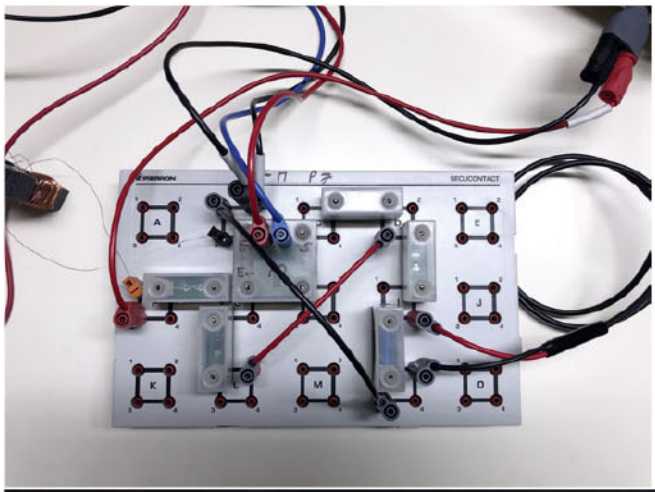


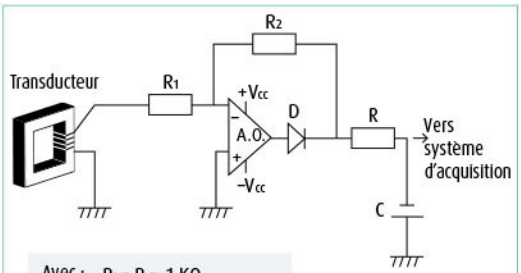

DOC 17 Branchement du transducteur et étalonnage.

Pour que les données du transducteur puissent être exploitées par une carte Arduino, il faut ensuite mettre en forme le signal de sortie. Le circuit de mise en forme (Docs 19 et 20) amplifie le signal, le redresse, et détecte l'amplitude maximale (Doc 18). On obtient ainsi, en sortie du capteur, une tension continue, image de l'intensité efficace circulant dans le circuit.



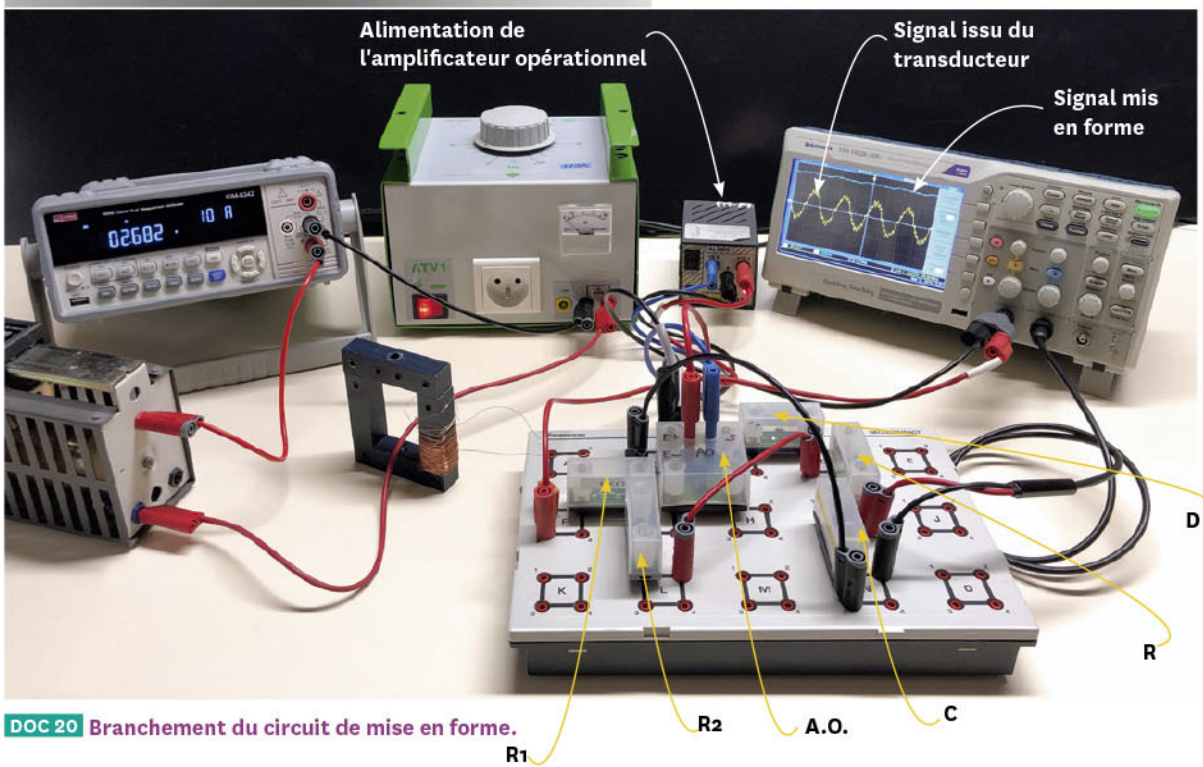

**DOC 18** Forme des signaux à chaque étape du capteur.






Avec :
- $R_1 = R = 1\ K\Omega$
- $R_2 = 22\ K\Omega$
- $C = 100\ \mu F$
- A.O. type 071
- Diode D type 1N4005

**DOC 19** Mise en œuvre pratique et composants de la mise en forme.




**DOC 20** Branchement du circuit de mise en forme.

# F PISTES DE PROJETS

## 1 Matériel

- 1 carte arduino UNO ou Nano
- 1 platine d'expérimentation
- 1 résistance de 1 kΩ (à choisir au plus proche de la valeur de la résistance de la photorésistance dans les conditions de luminosité de la mesure souhaitée)
- 1 photorésistance
- quelques fils

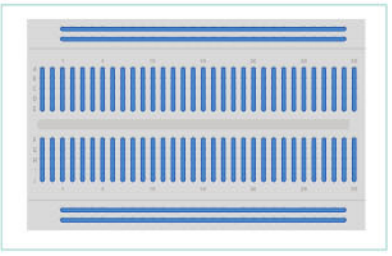
## 2 Mise en place

Sur une platine d'expérimentation, les points sont reliés comme représenté Doc 21. Ainsi, les deux schémas Doc 22 avec platine et sans platine représentent le même montage.

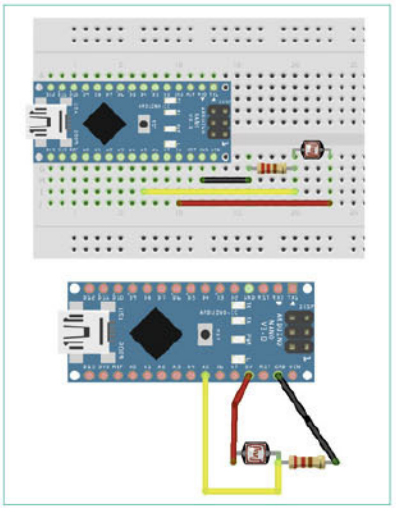
## 3 Programmation

En fonction des besoins, on peut faire calculer à la carte:

- la valeur de la résistance (en ohm) de la photorésistance, grâce au programme du Doc 23.
- une valeur en lux, après calibration de la réponse du capteur, grâce au programme du Doc 24.



DOC 21 Platine d'expérimentation.



DOC 22 Montage avec ou sans platine.

```
int pinPhotoresistance = 5;
int valeurPhotoresistance;

float R = 1000.; //Valeur de la résistance fixe utilisée

void setup(void) {
  Serial.begin(115200);
}

void loop(void) {
  valeurPhotoresistance = analogRead(pinPhotoresistance);  //lecture pin analogique A5
  Serial.println((5. * R / (valeurPhotoresistance*0.00489)  - R));  //calcul et affichage valeur resistance en ohm
  delay(50);  //temps d'attente de 50ms donc il y aura environ 20 mesures par seconde car le reste est très rapide
}
```

DOC 23 Programme pour renvoyer la valeur de la résistance.

```
#include <math.h>
int pinPhotoresistance = 5;
int valeurPhotoresistance;

float R = 1000.;

float calib1 = 1500000.; //Valeurs à modifier lors de la calibration avec un luxmètre
float calib2 = -1.01; //Valeurs à modifier lors de la calibration avec un luxmètre

void setup(void) {
  Serial.begin(115200);
}

void loop(void) {
  valeurPhotoresistance = analogRead(pinPhotoresistance);  //lecture pin analogique A5
  Serial.println(pow(10,(log10(5. * R / (valeurPhotoresistance*0.00488) - R)-log10(calib1))/calib2)); //calcul et affichage valeur en lux
  delay(50);  //delai de 50ms donc mesure à environ 20 Hz
}
```

DOC 24 Programme pour renvoyer une valeur en lux (après calibration).

## 4 Exemples de démarches de projets

Un tel capteur peut être utilisé pour différents projets, par exemple:

1) Mesurer et modéliser la décroissance de l'éclairement en fonction de la distance. Cela se fait très bien dans un tube en carton avec une source lumineuse de type lampe de bureau. On peut ainsi travailler sur la puissance lumineuse disponible pour les sondes spatiales par exemple.

2) Transmettre de l'information sous forme lumineuse. L'information transmise peut être binaire (présence ou absence de lumière) ou un code morse par exemple.

3) Mesurer l'éclairement sur une station météo afin de suivre l'évolution du maximum d'éclairement et de la durée du jour en fonction des saisons.

4) Réaliser une carte de luminosité dans un jardin. Cela nécessite toutefois d'utiliser plusieurs photorésistances et d'adapter les programmes précédents.

5) Réaliser une porte optique en éclairant la photorésistance avec un laser. On peut alors mesurer les coupures du faisceau lumineux pour détecter une ouverture de porte, compter les passages de personnes, mesurer la vitesse de rotation d'une hélice de drone, simuler un spectrophotomètre ou mesurer la vitesse d'une bille dans une gouttière en utilisant deux portes optiques. On peut également imaginer utiliser un tel dispositif pour mesurer une fréquence de vibration d'une table vibrante ou d'oscillation d'un haut parleur.

6) Ajouter une led (ou un relai) aux côtés de la photorésistance afin de piloter son allumage en fonction des conditions de luminosité (modèle de lampadaire automatique).

7) Ajouter une led rouge à côté de la photorésistance afin de mesurer la lumière réfléchie par une surface (et différencier le blanc du noir par exemple). Le même dispositif avec une led RGB peut servir à mesurer une couleur.