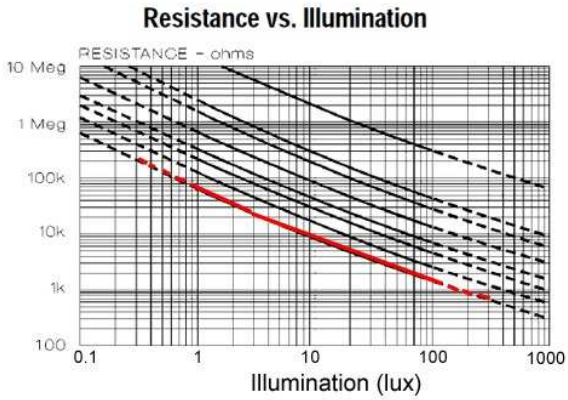
Compétence
- Capteurs

I La photorésistance (1 heure environ)

La photorésistance ou LDR LDR pour Light Dependent Resistor est un dipôle dont la résistance dépend de la lumière qu'il reçoit.
Lien vers une vidéo explicative du fonctionnement avec échange :

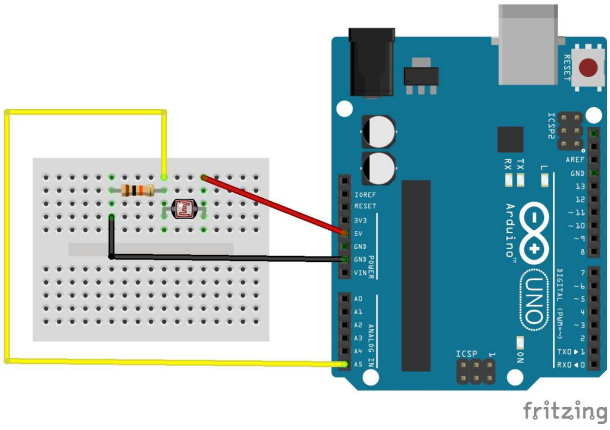
La photorésistance

Le datasheet de la photorésistance précise à l'utilisateur l'évolution de la résistance en fonction de l'éclairement.



Montage ARDUINO

Dans le montage, la résistance de la photorésistance varie en fonction de l'éclairement. La tension aux bornes de la photorésistance va donc évoluer.
Le microcontrôleur récupère cette tension qu'il convertit en valeur numérique allant de 0 à 1023.



Premier code source

```
   int LDRPin=..... ;                          //.........

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  int LDRReading = analogRead(LDRPin);         //.........
  Serial.print("Analog reading = ");           //.........
  Serial.println(LDRReading);                  //.........
  delay(500);                                 //.........
}
```

1. Compléter le numéro de port ***photocellPin*** après avoir réalisé le montage.
2. Commenter le programme afin d'en expliquer le fonctionnement.
3. Mettre en œuvre le programme. Modifier l'éclairement de la photorésistance. Comment la valeur lue évolue-t-elle ?
4. Dans le langage de programmation utilisé, un boucle conditionnelle a pour syntaxe :
   `if ( condition) { procédure} else {procédure}`
   a. Modifier le programme pour qu'il affiche dans le port série "sombre" si la valeur Photocell-Reading est inférieure à 200, sinon il affiche "lumineux".
   b. Améliorer le programme pour qu'il affiche :
      - "Noir" si la valeur LDRReading est inférieure à 10 ;
      - "Sombre" si la valeur LDRReading est inférieure à 200 ;
      - "Lumineux" si la valeur LDRReading est inférieure à 800 ;
      - "Très lumineux" si la valeur LDRReading est supérieure à 800 ;

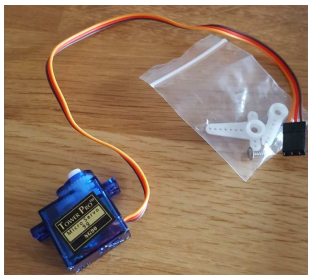
Proposition de correction

Premier code source

```
int LDRPin=5 ;

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  int LDRReading = analogRead(LDRPin);
  Serial.print("Tension lue= ");
  Serial.print(LDRReading);

  if (LDRReading < 10) {
    Serial.println(" Noir");
  } else if (LDRReading < 200) {
    Serial.println(" Sombre");
```

```
  } else if (LDRReading < 800) {
    Serial.println(" Lumineux");
  } else {
    Serial.println(" Très lumineux");
  }
  delay(500);
}
```

II Le servomoteur (1 heure environ)



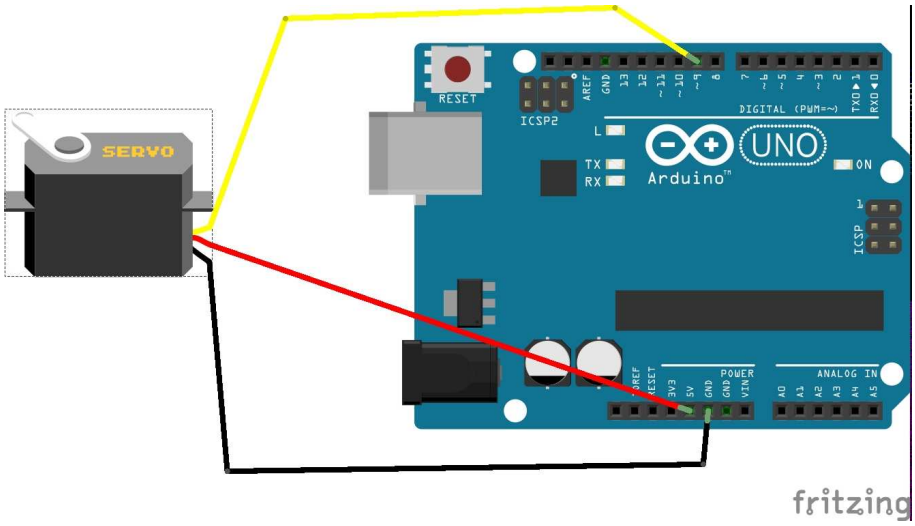
Un servomoteur (souvent abrégé en « servo », provenant du latin servus qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi.

D'après *Wikipédia.fr*

Lien vers une vidéo explicative du fonctionnement avec échange :

Le servomoteur

Montage



Premier code source

```
#include <Servo.h>

Servo servomoteur ;
void setup() {
   servomoteur.attach(9) ;
}

void loop() {
        servomoteur.write(0) ;
        delay(500) ;
        servomoteur.write(180) ;
        delay(1000) ;
}
```

Pour réaliser le programme de commande du servomoteur, il est nécessaire d'importer la bibliothèque *Servo*. On déplace le servomoteur d'un angle α exprimé en degré à l'aide de la commande `servomoteur.write(`α`)` .

1. Réaliser le montage et lancer le programme. Que réalise-t-il ?
2. Proposer un programme qui permette de faire tourner le servomoteur d'un angle de 1° toutes les 10 millisecondes.
3. Améliorer le programme pour qu'il revienne à sa position d'origine avec la même vitesse de rotation.

Facultatif Améliorer le programme pour qu'il signale l'arrivée du servomoteur en butée.

Proposition de correction

```
#include <Servo.h>

int angle_max = 180 ;

Servo servomoteur ;
void setup() {
   servomoteur.attach(9) ;
}

void loop() {
for (int i=0; i<=180; i++){
        servomoteur.write(i) ;
        delay(10) ;
}
for (int i=180; i>=0; i--){
        servomoteur.write(i) ;
        delay(10) ;
}
}
```