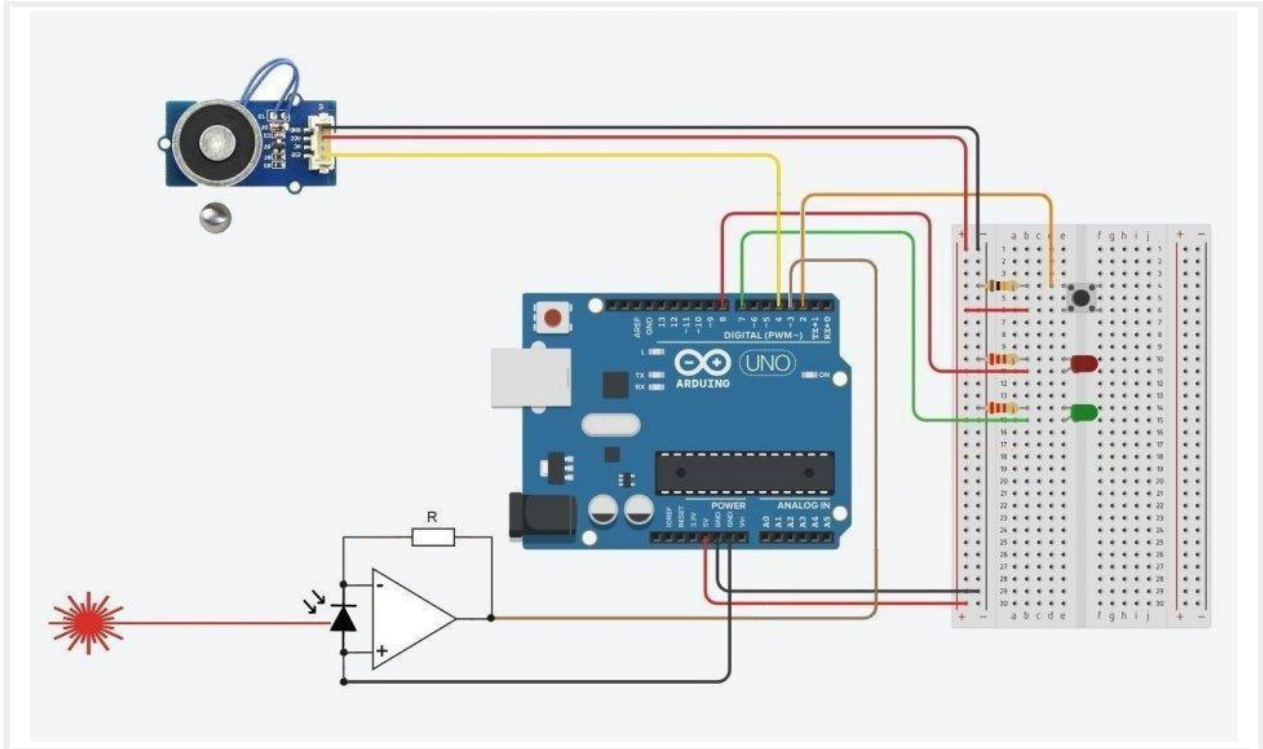


Chute d'une bille (attachInterrupt)

(Déterminer la durée de chute d'une bille)



Liste des composants

- . 1 électroaimant Grove
- . 1 DEL rouge
- . 1 DEL verte
- . 1 bouton poussoir
- . 2 résistances de $220\ \Omega$
- . 1 résistance de $10\ k\Omega$
- . 1 diode laser
- . 1 photodiode
- . 1 montage convertisseur courant-tension
- . 1 plaques d'essais
- . Fils de connexion

. Objectif

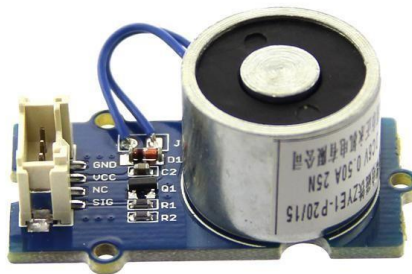
L'objectif du montage est de mesurer la durée en ms de la chute d'une bille à l'aide d'un dispositif de détection constitué d'un laser et d'une photodiode associée à un circuit convertisseur courant - tension.

A $t = 0$, la bille initialement maintenue par l'électroaimant est libérée, un chronomètre est alors déclenché.

Au cours de sa chute, la bille coupe le faisceau laser, provoquant une chute du potentiel (passage d'un niveau haut à un niveau bas) sur la broche numérique 3 de l'Arduino Uno, car la photodiode n'est à ce moment plus éclairée.

Quand la chute de potentiel est détectée, le chronomètre est arrêté et la durée de la chute est affichée dans le moniteur série.

Module électroaimant Grove



Le module électroaimant Grove, pouvant maintenir une masse métallique jusqu'à 1 kg, se raccorde sur une sortie digitale de l'Arduino Uno :

- Sortie digitale à niveau haut, l'électroaimant fonctionne,
- Sortie digitale à niveau bas, l'électroaimant ne fonctionne pas.

Caractéristiques :

Interface : compatible Grove

Alimentation : 5 Vcc

Consommation : 200µA (au repos) et 400 mA (au travail)

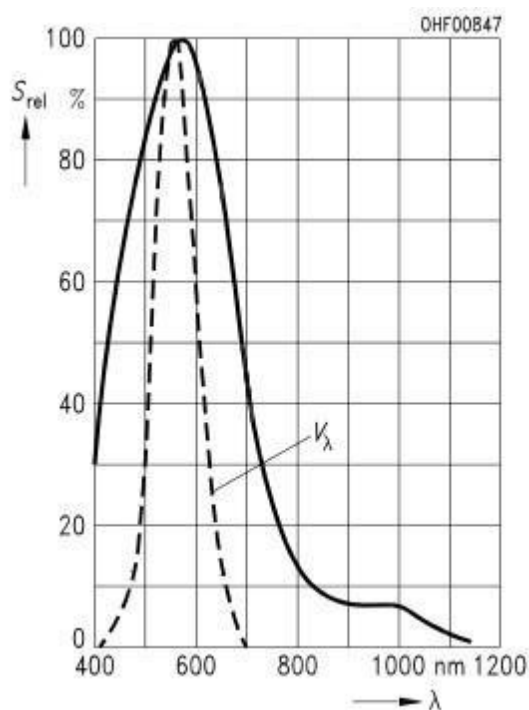
Photodiode – Montage convertisseur courant / Tension

La photodiode utilisée dans cette application est le modèle BPW21.



Une photodiode est un composant semi-conducteur ayant la capacité de capter un rayonnement du domaine optique et de le transformer en signal électrique.

Sensibilité relative spectrale en fonction de la longueur d'onde :



En pointillé : sensibilité de l'œil
En noir : sensibilité de la photodiode

L'intensité du courant délivré par la photodiode est proportionnelle à l'éclairement :

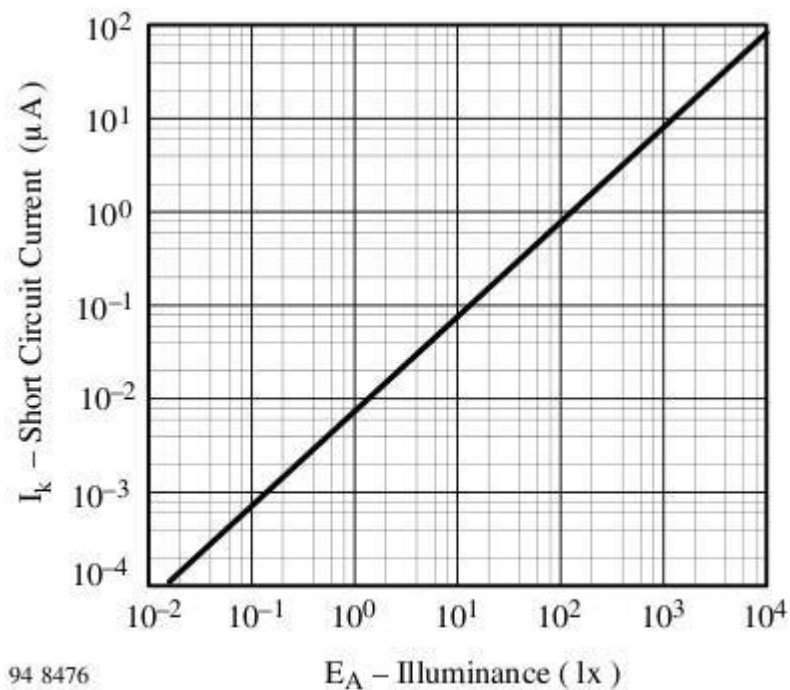
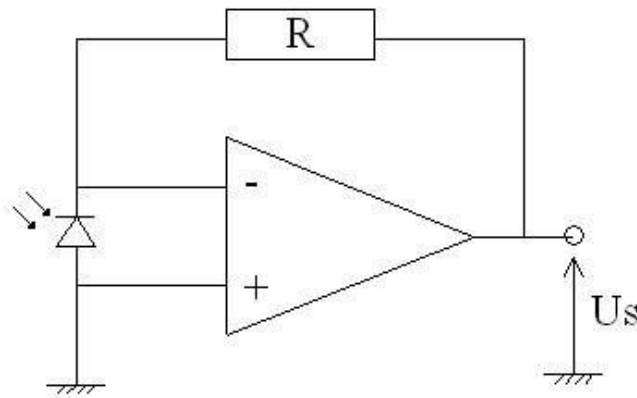


Figure 3. Short Circuit Current vs. Illuminance

On utilise un montage à amplificateur opérationnel convertisseur courant – tension pour mesurer une tension proportionnelle à l'éclairement de la photodiode :



Le courant généré par la photodiode en fonction de l'éclairement circule à travers la résistance R.

La tension de sortie de ce montage vaut :

$$U_s = R.I$$

Or l'intensité du courant est proportionnelle à l'éclairement, donc la tension de sortie est de la forme :

$$U_s = k.Eclairement$$

. Le programme

Voici le code de l'activité :

Chute_Bille_2

```
const byte starter = 2;
const byte detecteur = 3;
const int LEDR = 8;
const int LEDV = 7;
const int EM = 4;

volatile unsigned long debut = 0;
volatile unsigned long fin = 0;
volatile long duree=0;

void setup() {

  Serial.begin(9600);
  pinMode(LEDV, OUTPUT);
  pinMode(LEDR, OUTPUT);
  pinMode(EM, OUTPUT);
  pinMode(starter, INPUT);
  pinMode(detecteur, INPUT);
  attachInterrupt(digitalPinToInterrupt(starter), demarrechrono, RISING);
  attachInterrupt(digitalPinToInterrupt(detecteur), arretchrono, FALLING);
  digitalWrite(LEDV, LOW);
  digitalWrite(LEDR, HIGH);
  digitalWrite(EM, HIGH);
}
```

```

void loop() {

    if (fin !=0){
        duree=fin-debut;
        Serial.print("duree:");
        Serial.println(duree);
        Serial.println("");
        fin =0;
    }
}

void demarrechrono() {

    debut=millis();
    Serial.print("debut:");
    Serial.println(debut);
    digitalWrite(LED_R, LOW);
    digitalWrite(LED_V, HIGH);
    digitalWrite(EM, LOW);

}

void arretchrono() {

    fin=millis();
    Serial.print("fin:");
    Serial.println(fin);
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_V, LOW);
    digitalWrite(EM, HIGH);

}

```

Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . **const int LEDR = 8** (constante nombre entier correspondant à la broche de la DEL rouge)
- . **const int LEDV = 7** (constante nombre entier correspondant à la broche de la DEL verte)
- . **const int starter = 2** (constante nombre entier correspondant à la broche du bouton poussoir)
- . **const int detecteur = 3** (constante nombre entier correspondant à la broche de la diode de détection)
- . **const int EM = 4** (constante nombre entier correspondant à la broche de l'électro-aimant)
- . **volatile unsigned long debut = 0** (variable volatile nombre entier long positif pour stocker l'heure à laquelle la chute de la bille a été déclenchée)
- . **volatile unsigned long fin = 0** (variable volatile nombre entier long positif pour stocker l'heure à laquelle la bille a été détectée)
- . **volatile long duree = 0** (variable volatile nombre entier long pour le calcul de la durée en ms de la chute de la bille)

Remarque :

La déclaration d'une variable volatile est une directive au compilateur. Cela ordonne au compilateur de charger la variable depuis la RAM et non depuis un registre de stockage, qui est un emplacement de mémoire temporaire où les variables du programme sont stockées et manipulées.

Une variable doit être déclarée volatile chaque fois que sa valeur peut être modifiée par quelque chose hors du contrôle de la section de code dans laquelle elle apparaît, comme un thread s'exécutant simultanément. Dans l'Arduino, le seul endroit où cela est susceptible de se produire est dans les sections de code associées aux interruptions, appelées routine de service d'interruption.

– 2. Initialisation des entrées et sorties :

- . **Initialisation de la liaison série à un débit de 9600 bauds,**
- . **Initialisation des broches des DELs en sortie,**
- . **Initialisation de la broche du bouton poussoir en entrée,**
- . **La DEL rouge est allumée et la DEL verte est éteinte,**
- . **L'électro-aimant est activé,**
- . **Déclaration des broches du bouton poussoir (starter) et de la diode de détection (detecteur) en broche d'interruption :**

- `attachInterrupt(digitalPinToInterrupt(starter), demarrechrono, RISING);`
- `attachInterrupt(digitalPinToInterrupt(detecteur), arretchrono, FALLING);`

Les interruptions permettent au microcontrôleur d'exécuter une fonction lorsqu'un évènement survient sur une des broches d'interruption.

Sur la carte Arduino UNO, seules les broches 2 et 3 supportent les interruptions.

Ainsi avec ces déclarations, quand la broche « starter » passe d'un niveau bas à un niveau haut (par appui sur le bouton poussoir), la fonction « demarrechrono » est appelée.

Puis quand la broche de la diode de détection passe d'un niveau haut à un niveau bas (par passage de la bille devant le faisceau laser), la fonction « arrêt chrono » est appelée.

– 3. Fonction principale en boucle :

–> **Attente de la détection du passage de la bille devant la photodiode (fin>0) :**

–> **si fin > 0 :**

- **Calcul de la durée de la chute,**

- Affichage de la durée dans le moniteur série.

– 4. Fonction « demarrechrono » :

Cette fonction est appelée quand on appuie sur le bouton poussoir (la broche « starter » passe d'un niveau bas à un niveau haut) A ce moment :

- l'électro-aimant est désactivé,
- La DEL rouge est éteinte et la diode verte est allumée,
- l'heure d'appui sur le bouton poussoir est stocké dans la variable « debut » et est affichée dans le moniteur série.

– 5. Fonction « arretchrono » :

Cette fonction est appelée quand la bille passe devant la photodiode et coupe le faisceau laser. (la broche « **detecteur** » passe d'un niveau haut à un niveau bas)

A ce moment :

- l'heure de passage de la bille devant la photodiode est stocké dans la variable « fin » et est affichée dans le moniteur série
- La DEL verte est éteinte et la diode rouge est allumée,
- l'électro-aimant est réactivé.