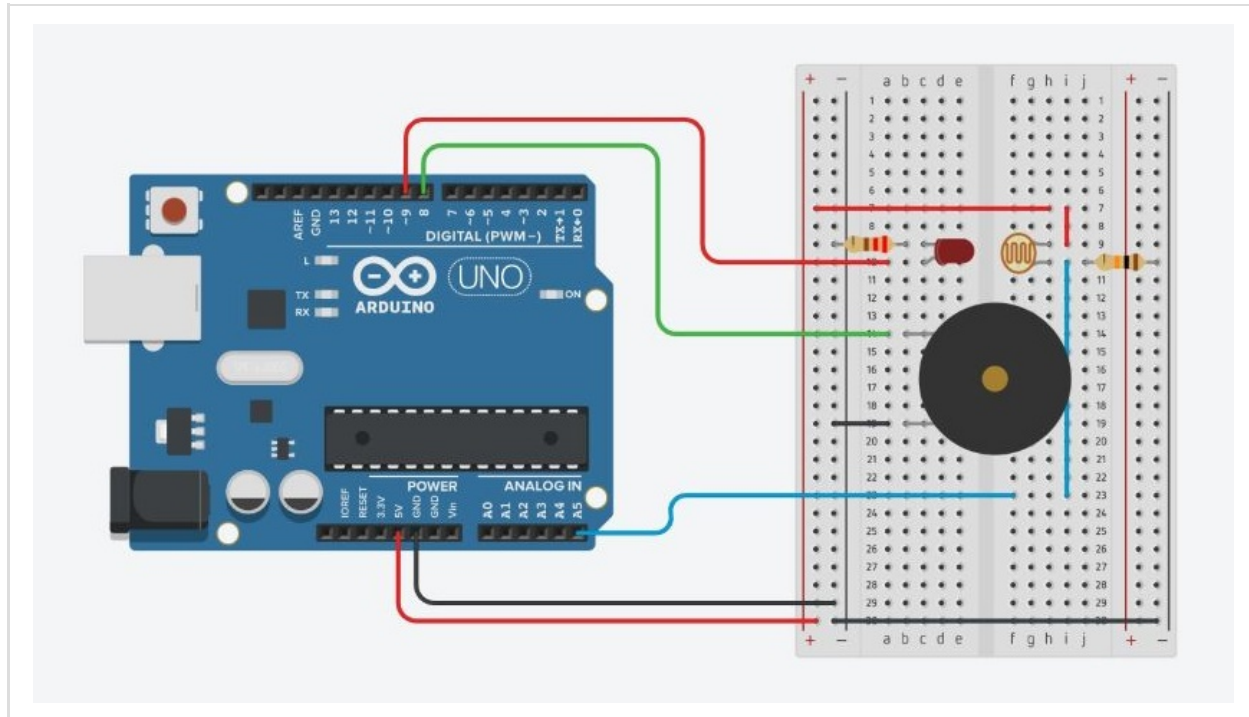


Alarme sonore (par détection de passage)



. Liste des composants :

- . 1 DEL rouge
- . 1 résistance de 220 Ω (résistance de protection de la DEL)
- . 1 photorésistance
- . 1 résistance de 10 k Ω (résistance de la photorésistance)
- . 1 buzzer
- . 1 plaque d'essai
- . Fils de connexion

. Objectif

Dans cette activité, le programme "**beep**" est utilisé comme alarme de détection de passage.

Pour cela, on ajoute au montage du sketch "**beep**", une photorésistance qui sera éclairée par la DEL rouge. La sortie de la photorésistance est connectée à l'entrée analogique A5 de l'Arduino.

La valeur de la broche A5 est alors proportionnelle à l'intensité lumineuse reçue par la photorésistance. En présence d'un obstacle entre la DEL et la photorésistance, la tension mesurée au niveau de la broche A5 diminue et quand celle-ci est inférieure à un seuil (la sensibilité du capteur définie initialement), l'alarme sonore est déclenchée.

. La photorésistance

Une photorésistance est un composant électronique dont la résistance en Ohm (Ω) varie en fonction de l'intensité lumineuse. Plus la luminosité est élevée, plus la résistance est basse. La photorésistance est un capteur résistif.



On peut donc l'utiliser comme capteur lumineux pour :

- Mesure de la lumière ambiante.
- Détecteur de lumière dans une pièce.
- Suiveur de lumière dans un robot.
- Détecteur de passage.
- ...

Ses symboles électroniques sont les suivants :

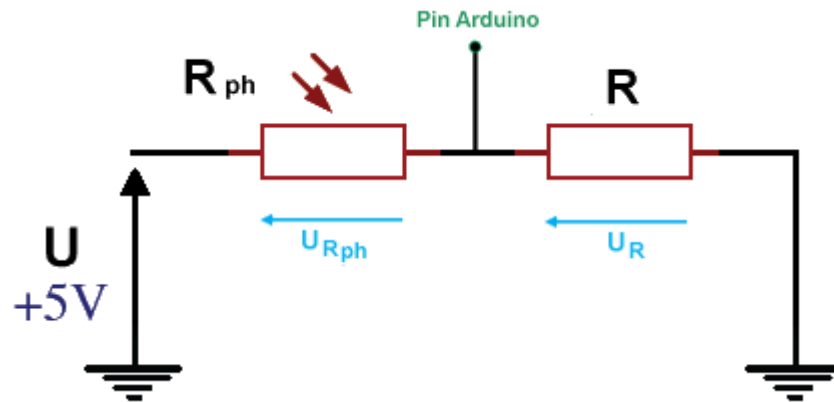


L'avantage des photorésistances est qu'elles sont très bon marché. Par contre, leur réaction à la lumière est différente pour chaque photorésistance, même quand elles ont été produites dans le même lot. On peut ainsi noter une différence de résistance de plus de 50% entre deux photorésistances du même modèle pour la même luminosité. On ne peut donc pas l'utiliser pour une mesure précise de la lumière. Par contre, elles sont idéales pour mesurer des changements simples de la luminosité.

On utilise la photorésistance dans un montage (Cf. ci-dessous), avec une résistance fixe de 10 k Ω , qu'on appelle un **diviseur de tension**.

La photorésistance est alimentée en 5V depuis l'Arduino. Le point entre les deux résistances est relié à une broche analogique de l'Arduino et on mesure la tension de cette broche par la fonction **"analogRead (broche)"**.

Tout changement de la tension mesurée est dû à la photorésistance puisque c'est la seule résistance qui change dans ce circuit, en fonction de l'intensité lumineuse qu'elle reçoit.



D'après la loi d'additivité des tensions dans un circuit en série :

$$U = U_{Rph} + U_R = (R_{ph} + R) I$$

$$U_R = U - U_{Rph} = U - R_{ph} I$$

U_R est la tension appliquée sur l'entrée analogique de l'Arduino. Quand l'intensité lumineuse reçue par la photorésistance augmente, R_{ph} diminue, donc U_R augmente, et au contraire quand la luminosité diminue, R_{ph} augmente et U_R diminue.

On peut exprimer U_R en fonction de U :

$$U = U_{Rph} + U_R = (R_{ph} + R) I$$

Donc :

$$I = \frac{U}{(R_{ph} + R)}$$

Et :

$$U_R = R I = \frac{R}{(R_{ph} + R)} U$$

C'est la raison pour laquelle on appelle ce montage un diviseur de tension.

. Le programme

Le programme de l'activité pourra être modifié pour voir l'influence des variables (sensibilité du capteur, fréquence de l'onde sonore, durée d'émission, durée de silence).

```
Alarme_sonore

// Déclaration des constantes et variables

const int PinLED = 9;
const int PinTone = 8;
const int PinPhotoR = 5;
const int CapteurSensib = 500;
const int TimeSleep = 200;

int ValCapteur = 0;

// Initialisation des entrées et sorties

void setup() {
  pinMode (PinLED, OUTPUT);
  digitalWrite(PinLED, HIGH);
}

// Fonction principale en boucle

void loop() {
  ValCapteur = analogRead(PinPhotoR);
  if (ValCapteur < CapteurSensib) {
    tone(PinTone, 440);
    delay(TimeSleep);
    noTone(PinTone);
    delay(TimeSleep);
  }
  else {
    noTone(PinTone);
  }
}
```

Déroulement du programme :

– Déclaration des constantes et variables :

- . **const int PinLed = 9** (constante nombre entier correspondant au n° de la broche sur laquelle la DEL rouge est connectée)
- . **const int PinTone = 8** (constante nombre entier correspondant au n° de la broche sur laquelle le

buzzer est connecté)

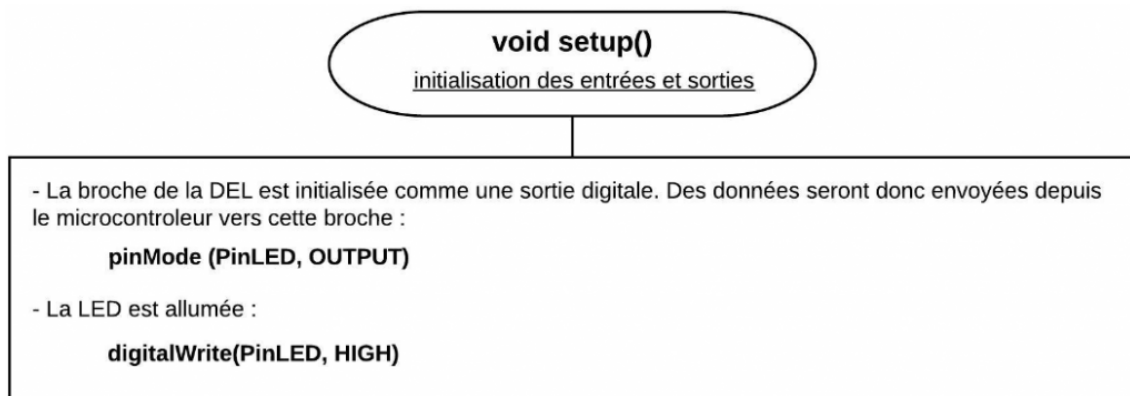
. **const int PinPhotoR = 5** (constante nombre entier correspondant au n° de la broche sur laquelle la photorésistance est connectée)

. **const int CapteurSensib = 500** (constante nombre entier correspondant à la sensibilité du capteur)

. **const int TimeSleep = 200** (constante nombre entier correspondant à la durée d'émission sonore et de silence)

. **int ValCapteur = 0** (variable nombre entier pour stocker la valeur du capteur)

– Initialisation des entrées et sorties :



– Fonction principale en boucle :

