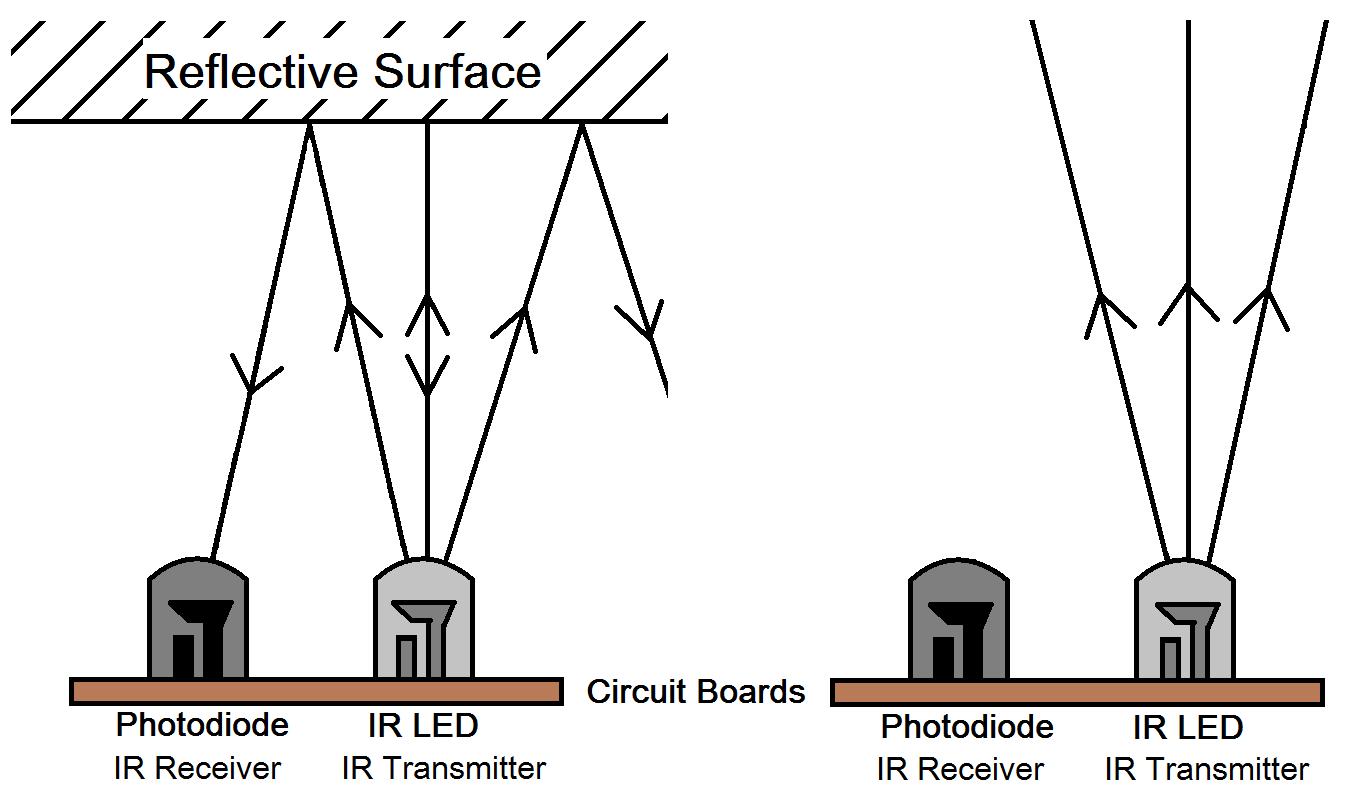
DOC Capteur IR

# Le capteur

## Fonctionnement

Comme son nom l’indique, ce capteur utilise un ***émetteur et un récepteur infrarouge***.



Ce schéma permet de mieux visualiser le fonctionnement du capteur.

Sur la gauche on voit que la ***LED infrarouge émet une lumière vers la surface réfléchissante***.

Et la ***photodiode reçoit plus ou moins la lumière réfléchit***.

Selon la **couleur** de la surface ainsi que sa **distance**.

#### Exemple

Une surface blanche à 30 cm du capteur sera mieux qu’une surface verte à 30 cm également.

## Utilisation

Ce capteur peut avoir différentes fonctions :

* ***Calculer une distance***
* ***Détecter un changement de contraste***
* ***Mesurer la réflectivité d’un milieu en fonction de la distance***

Dans notre cas nous allons avoir besoin de détecter un contraste et de déterminer la « ***profondeur de ce contraste*** ».

#### Une image contenant horloge, objet Description générée avec un niveau de confiance très élevéNotre map

Comme le montre cette photo, les ***surfaces vertes représentent des places de parking***.

Notre but et donc de ***détecter ces places*** en détectant le ***contraste entre la route et la place*** ; une fois le contraste détecté, nous allons pouvoir calculer la profondeur de la place grâce au capteur IR ainsi que le capteur de vitesse présent sur le disque du moteur. Pour ce faire, nous allons mesurer la vitesse et le temps passé à côté de la place.

Avec le ***temps*** et la ***vitesse*** nous allons pouvoir ***déterminer la taille et la profondeur.***

# Le code

## Une image contenant capture d’écran Description générée avec un niveau de confiance très élevéInclude

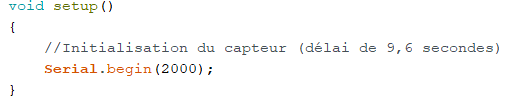
Le #define IR\_PROXIMITY\_SENSOR A1 permet de ***définir le pin*** sur lequel le capteur est branché.

Le #define ADC\_REF 5 permet de ***déterminer le voltage envoyé vers le capteur***, ici 5 Volts.

Cependant nous utilisons un Shield pour Arduino, la tension sera donc de 3,3 Volts

Float voltage est la variable qui ***va stocker le voltage envoyé par le capteur***. Cette variable va changer de valeur en fonction de la réflectivité du milieu.

## Void setup



Void setup permet ***d’initialiser toutes la valeurs et paramètres utilisés dans le programme Arduino***.

Serial.begin(2000) effectue une initialisation de la ***rapidité de modulation*** (en bauds) à 2000.

## Une image contenant capture d’écran Description générée avec un niveau de confiance très élevéVoid loop

Void loop est la partie du programme qui ***va se répéter à l’infini dans l’Arduino***.

## Voltage =getVoltage() permet d’obtenir les différentes valeurs envoyés par le capteur à un rythme régulier et non en continu.

Serial.print (‘’sensor voltage = ’’) permet ***d’afficher le message entre ‘’ ‘’***.

Ici on affiche « sensor voltage = »

Serial.print (voltage) permet ***d’afficher les valeurs envoyées par le capteur***.

Serial.print (‘’\n’’) permet de ***mettre un espace entre chaque ligne*** d’informations du capteur.

Delay(200) permet de mettre un ***délai de 200 millisecondes entre chaque affichage***.

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Float getVoltage() permet ***d’obtenir les valeurs du capteur de façon continue***.

Int sensor\_valeur permet d’obtenir une ***moyenne des sum*** envoyées par le capteur.  
La ***boucle for va demander 20 fois la valeur au capteur*** puis sensor\_value = sum/20 permet de ***faire une moyenne***.

On va finir par ***renvoyer le voltage au Serial.print*** pour ***l’afficher sur le moniteur Arduino***.