

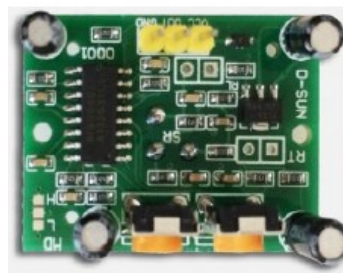
Capteur infrarouge Passif (PIR)

I) Généralités

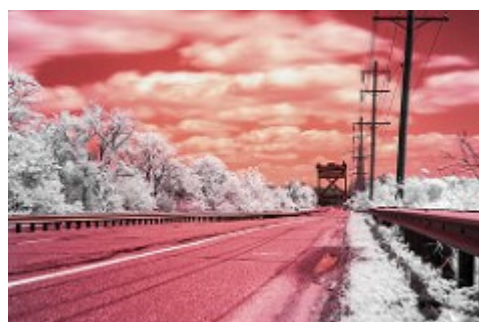
Un capteur PIR est un capteur qui mesure les radiations IR émises par les objets se trouvant dans son champ de vision. Il est généralement utilisé comme détecteur de mouvement ou de proximité. On le rencontre sous différentes appellations:



Voici celui que nous allons étudier le SR501HC



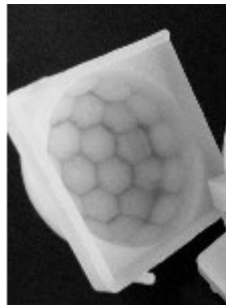
Les capteurs PIR détectent les radiations infrarouges émises par les objets autour de lui. Or la plupart des objet émettent des IR suivant leur température. Ces capteurs sont dits passifs, car ils n'émettent pas de radiation (contrairement aux barrières à infrarouge). Les IR sont invisible par l'œil humain, mais des appareils photos numériques peuvent les détectés.



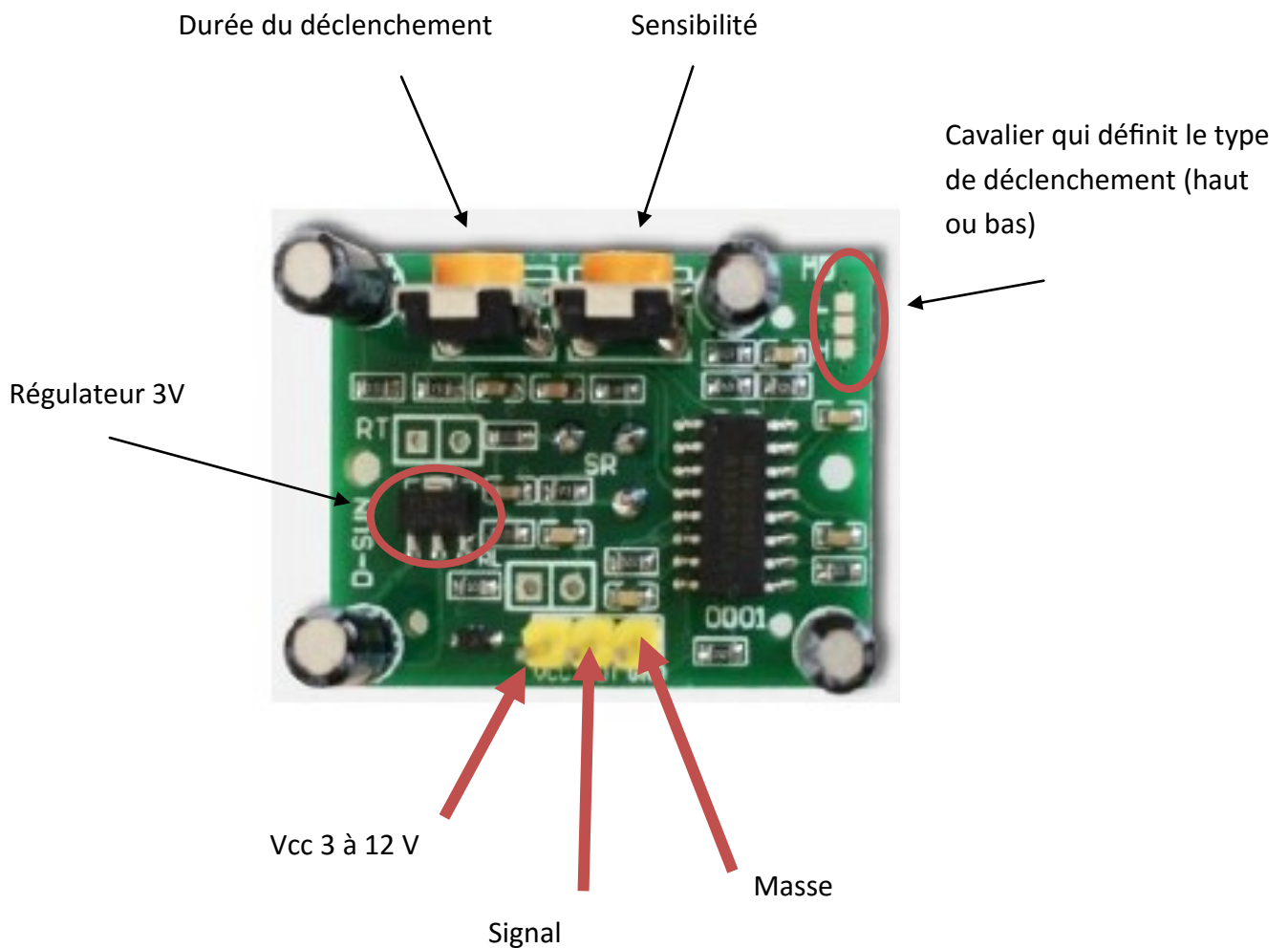
II) Brochage



La zone de détection est un tout petit rectangle. Pour pouvoir une vision à 160° on utilise une lentille de Fresnel (Comme pour les phares).

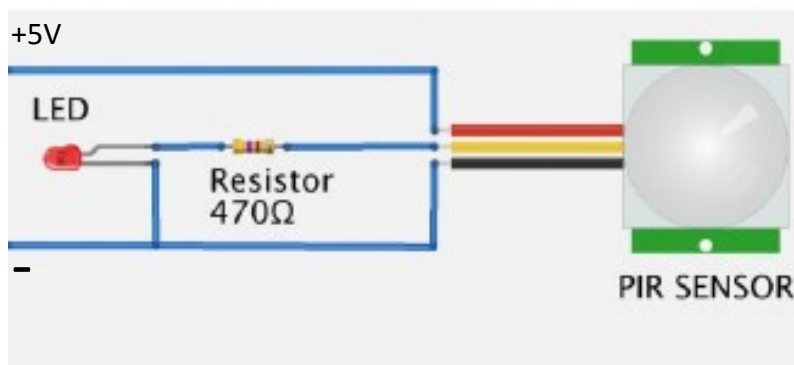


Les zones importantes du capteur



III) Test d'un PIR

Pour tester un PIR



Quand un mouvement est détecté, la LED s'allume.

Remarque:

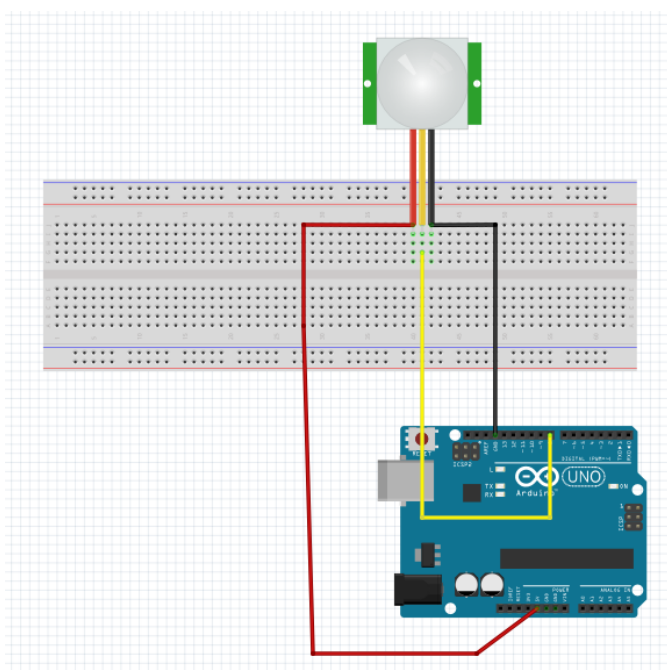
Lorsque l'on branche la pile, le capteur met environ 1min pour se stabiliser. Placer le montage dans un endroit sans mouvement pendant ce temps jusqu'à ce que la LED reste éteinte.

Le capteur peut détecter un mouvement entre 0,6 à 6m.

Réglage de la durée: Elle détermine la durée pendant laquelle la sortie restera à l'état haut.

Réglage de la sensibilité: Faire des essais est la seule méthode valable. Trop de sensibilité peut détecter un animal (il serait dommage qu'une alarme se déclenche avec le chat de la maison). Pour porter à 6m, la sensibilité doit par contre être maximale. Dans un espace restreint, il faut réduire la sensibilité, sinon le capteur peut s'auto-enclencher.

IV) Montage simple avec un arduino UNO



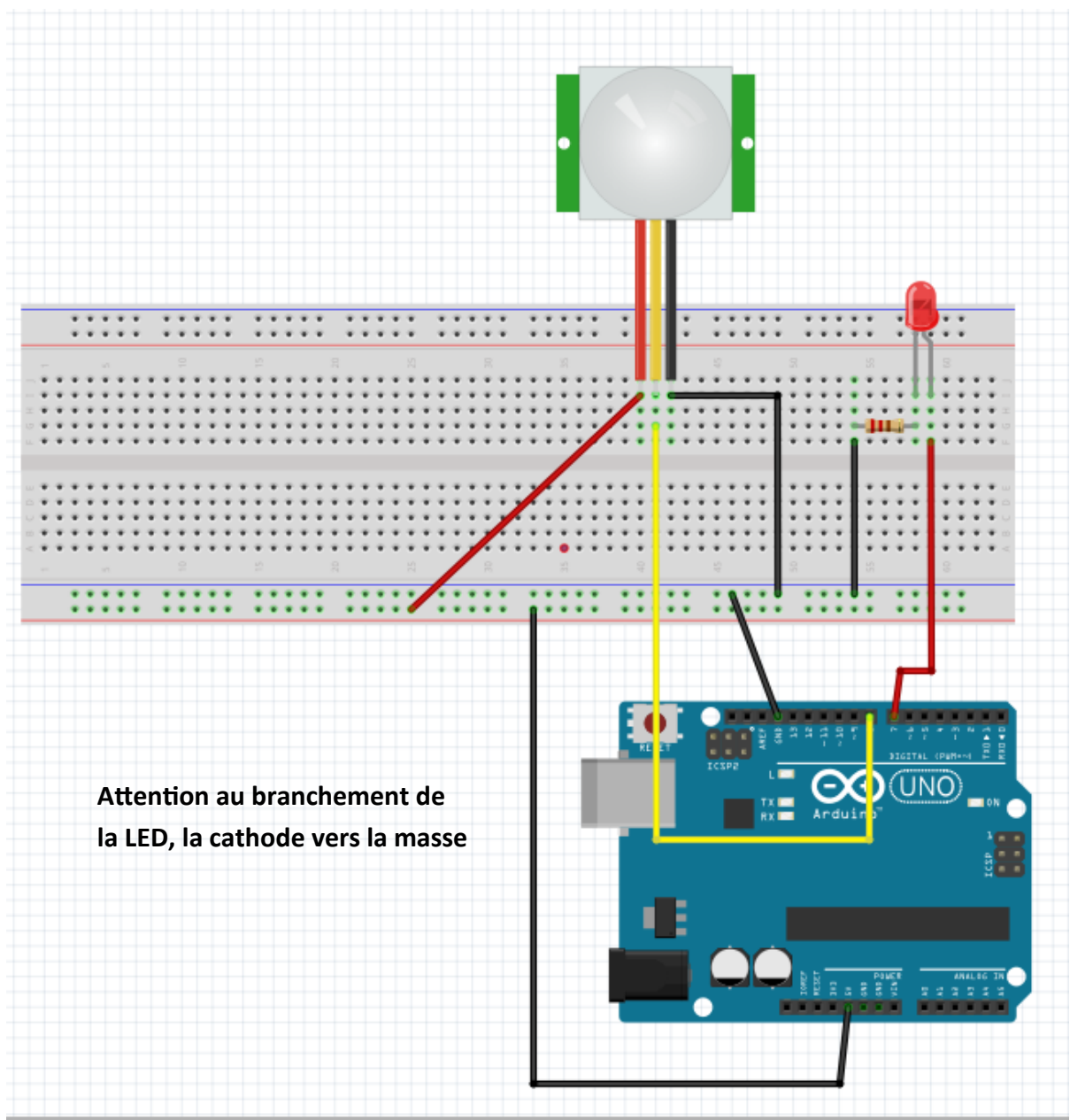
Relier:

| | |
|--------|-------------------------------------|
| PIR | <—> arduino |
| VCC | <—> +5V |
| Signal | <—> Entrée/sortie digitale 8 (DIO8) |
| Masse | <—> Gnd |

Le programme

```
1 // programme test_pir
2 // écrit par F5LLE
3 // essai d'un capteur PIR
4 // Affiche permanence l'état de la broche PIR sur le
5 //moniteur serie
6 #define PIR 8 //chaque fois que les lettres PIR seront rencontrées elles seront remplacées par 8
7 int etat=0;
8 void setup()
9 {
10   Serial.begin(9600); // pour voir le résultat sur le moniteur série
11   pinMode(PIR, INPUT); // La DIO 8 est configurée en entrée
12 }
13
14 void loop() {
15   etat= digitalRead(PIR); // lit l'état de la broche 8
16   Serial.print(etat); //affiche l'état dans le miniteur serie
17   delay(300); // un petit délai pour ne pas écrire tyrop vite
18               // dans le moniteur serie
19 }
```

V) Allumer une LED lors de la détection d'une présence



Le programme

```
1  #define PIR 8
2  #define LED 7 // utilise la LED 13 de la carte arduino
3  #define LED_ARDUINO 13
4  int etat;
5  void setup(){
6      // Serial.begin(9600); moniteur série inutile car LED
7      pinMode(PIR, INPUT);
8      pinMode(LED, OUTPUT);
9      pinMode(LED_ARDUINO, OUTPUT);
10 }
11 void loop(){
12     etat=digitalRead(PIR);
13     digitalWrite(LED, etat);
14     digitalWrite(LED_ARDUINO, etat);
15     Serial.print(etat); // inutile
16     delay(300);
17 }
```

VI) Principe d'une ALARME

C'est un montage similaire au précédent, on rajoute juste un buzzer actif. Un buzzer actif crée lui-même la tonalité émise au contraire d'un buzzer passif où l'on doit fournir une tonalité (donc jouer de la musique).

Capteur ARDUINO UNO

PIR

Vcc 5V

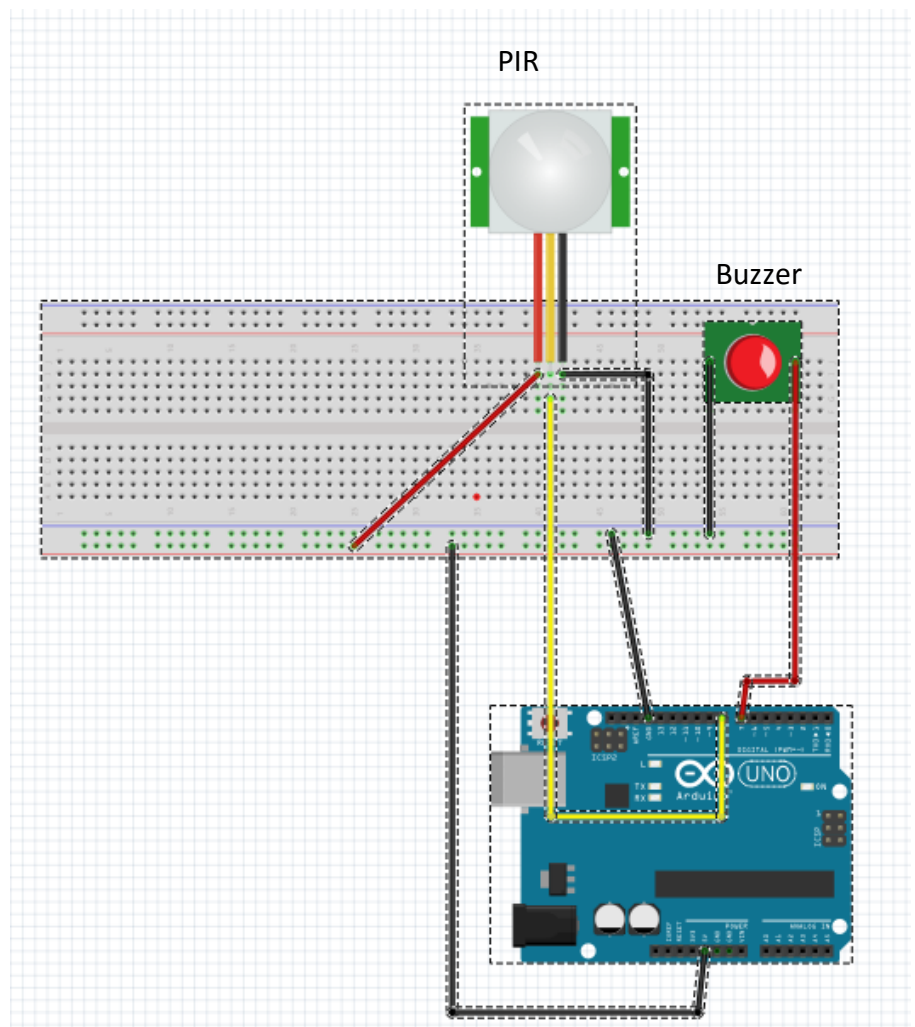
Signal DIO 8

Masse Gnd

Buzzer

Masse Gnd

Signal DIO 7



Le programme

```
1  #define PIR 8
2  #define BUZZER 7 // utilise la LED 13 de la carte arduino
3  #define LED 13
4  int etat;
5  void setup(){
6      // Serial.begin(9600); moniteur série inutile car LED
7      pinMode(PIR, INPUT);
8      pinMode(BUZZER, OUTPUT);
9      pinMode(LED, OUTPUT);
10 }
11 void loop(){
12     etat=digitalRead(PIR);
13     digitalWrite(LED, etat);
14     digitalWrite(BUZZER, etat);
15     delay(300);
16 }
```

Il est maintenant facile de modifier le programme pour différentes applications