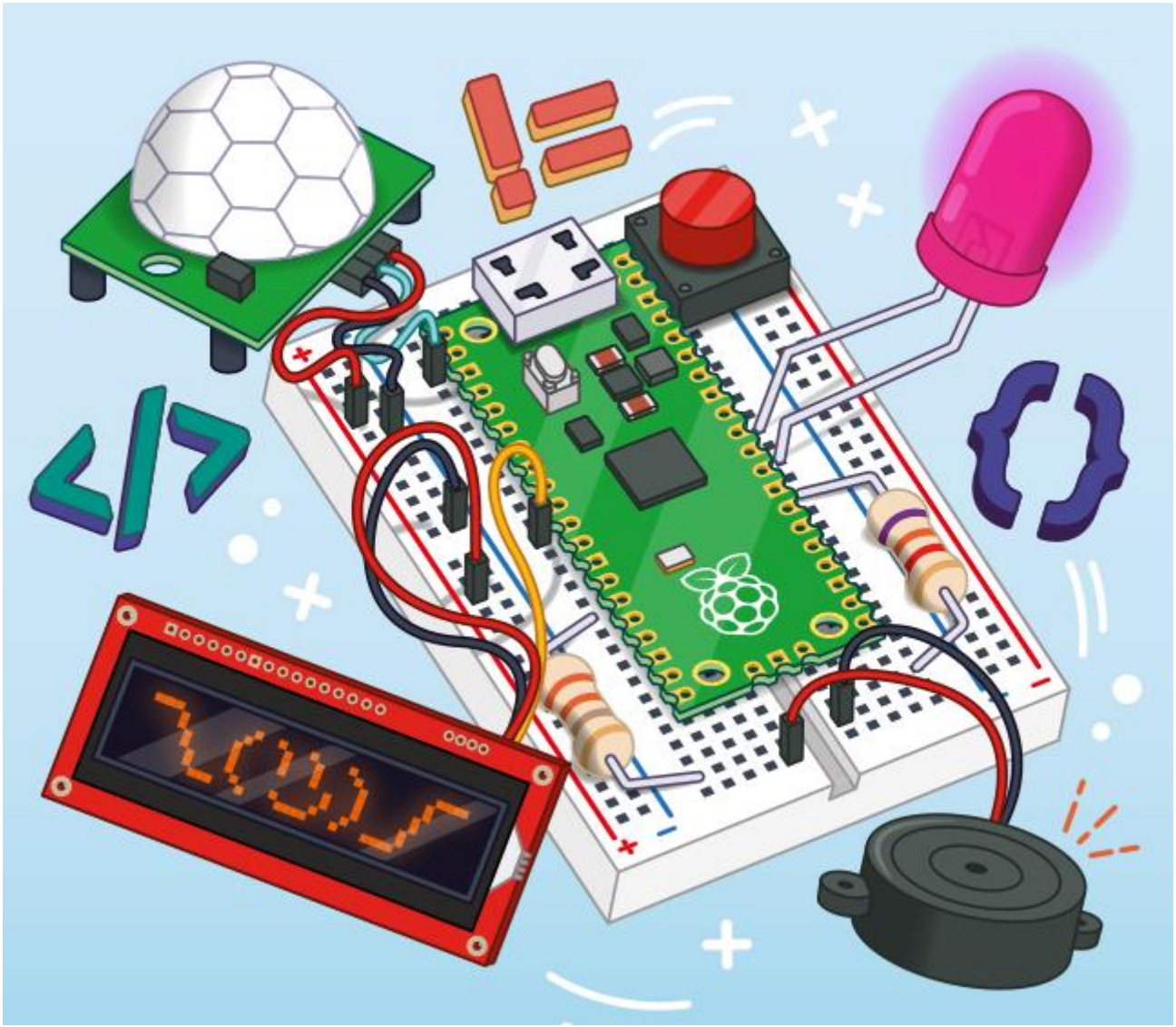
TP N°2 :

Raspberry Pi Pico

Capteur de mouvements et buzzer



I. Présentation du capteur PIR

1. Présentation

Le capteur PIR est un capteur de mouvements.

2. Réglage de la sensibilité

Deux vis permettent de régler la sensibilité ainsi que le temps pendant lequel il est à l'état haut lors d'une détection

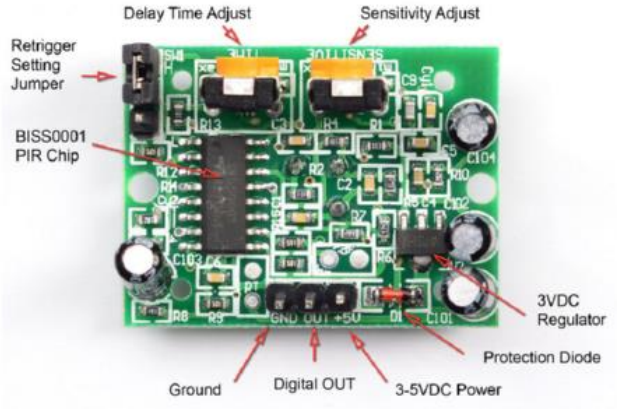
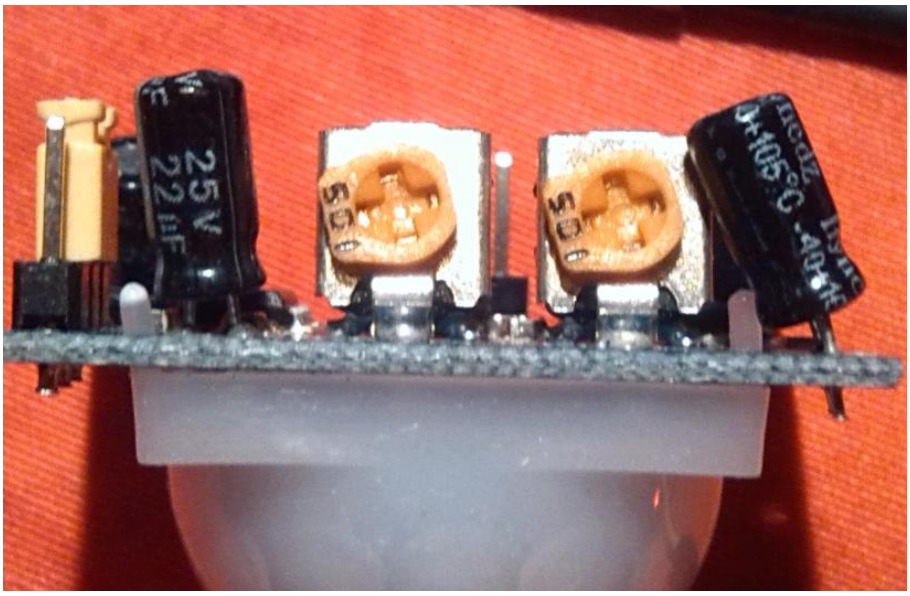


Figure 1 : à utiliser pour une question difficile

Le problème avec les PIRs, c'est leur stabilité : il peut alterner des états haut/bas plusieurs fois à la suite… Après de nombreux essais, la configuration qui apporte le plus de stabilité est la suivante :



3. Principe de fonctionnement

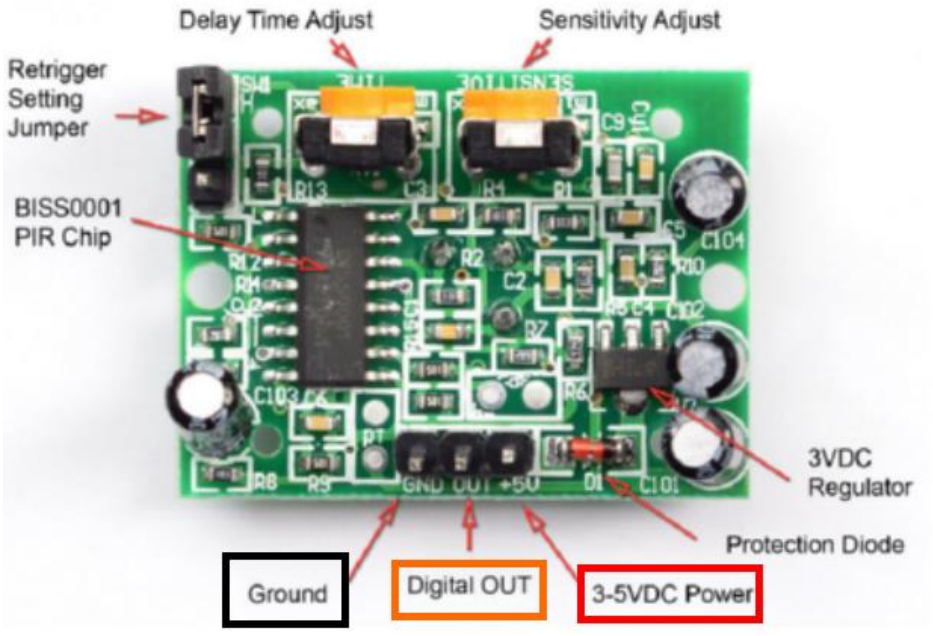
Dès que le PIR détecte un mouvement, il met sa broche OUT à l'état haut pendant un certain temps (réglable avec une des deux vis).

II. Programme de test

1. Câblage

Câbler VCC sur 3.3 V, Ground sur la masse et Digitout OUT sur la broche 28.

Rappel :



2. Test

```
"""
Exemple simple d'utilisation d'un capteur de mouvement
Affiche un message sur la connsole

* capteur PIR sur la broche GP28.
"""
# -*- coding: utf-8 -*-
import board
import digitalio

pir = digitalio.DigitalInOut(board.GP28)
pir.direction = digitalio.Direction.INPUT

while True:
    if pir.value:
        print("AU VOLEUR ! Mouvement détecté!")
        while pir.value:
            pass

```

3. Travail à faire

Tâche N°1. Câbler
Tâche N°2. Régler la sensibilité
Tâche N°3. Tester le code

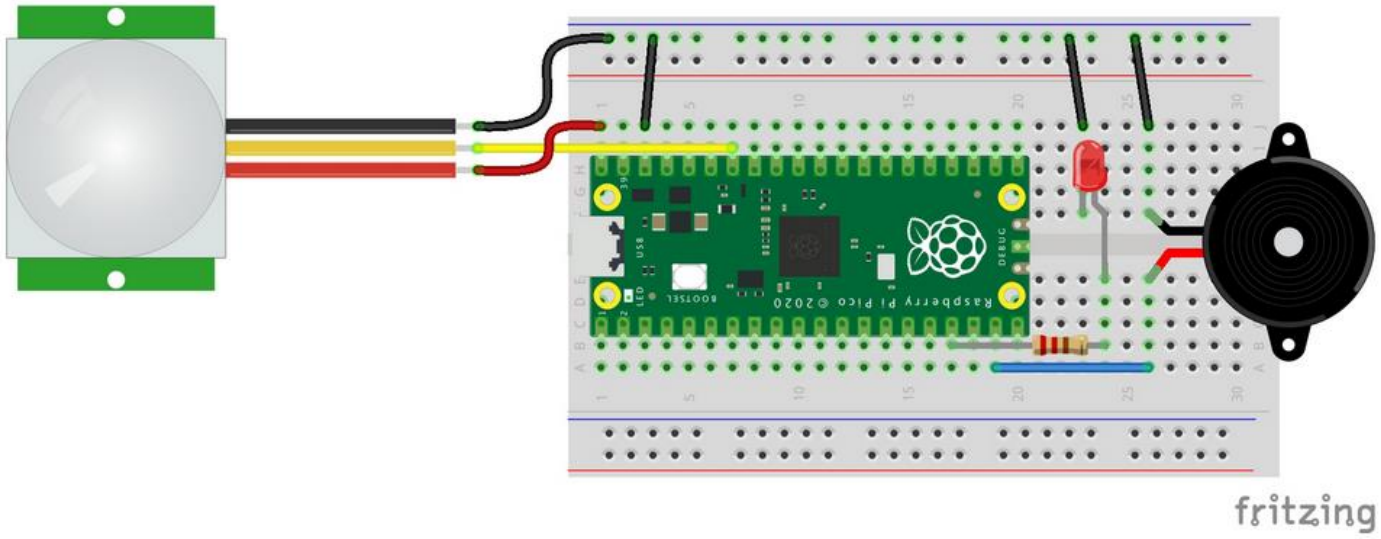
III. Alarme anti-voleurs

1. Travail à faire

En vous inspirant des exemples ci-dessous :

Tâche N°4. Mettez en place une alarme anti-voleurs qui lorsqu'un mouvement est détecté, émet une sonnerie

2. Câblage



3. Exemples de code : Afficher « Mouvement détecté » une seule fois

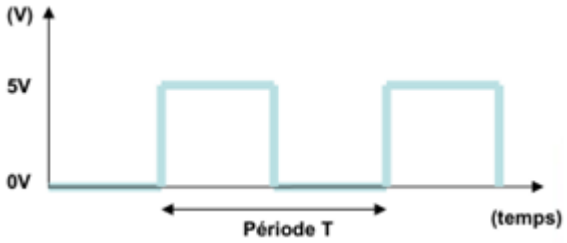
```
# -*- coding: utf-8 -*-
import time
import board
import digitalio

pir = digitalio.DigitalInOut(board.GP28)
pir.direction = digitalio.Direction.INPUT

mouvements_detectes = False
while True:
    if pir.value and not mouvements_detectes:
        print("AU VOLEUR ! Mouvement détecté!")
        mouvements_detectes = True
    mouvements_detectes = pir.value
```

4. Buzzer

a. *Rappel : Jouer une note*

Le buzzer se câble sur une sortie numérique en PWM (ou MLI en français) et le microcontrôleur lui envoie alors un signal périodique dont on fait varier la fréquence en fonction de la note que l'on désire jouer.

f = 1/T ; T=1/f ; f : fréquence; T : période



Exemple : le LA est un signal d'une fréquence f de 440 Hertz soit un signal qui varie 440 fois par seconde. La fréquence du DO est 262 Hz(octave 3) etc.

Le signal doit être carré : c'est-à-dire que la moitié de la période, c'est l'état haut et l'autre moitié, l'état bas.

b. Code correspondant

```
import time
import board
```

```
import pwmio

# Créé un objet buzzer de type PWMOut sur la broche 14
# la configuration initial est duty_cycle=0, muet.
buzzer = pwmio.PWMOut(board.GP14, frequency=660, duty_cycle=0, variable_frequency=True)

while True:
    time.sleep(1)
    buzzer.duty_cycle = 0
    time.sleep(1)
    # le cycle est la moitié du maximum donc PWM = 50%
    buzzer.duty_cycle = 65536//2
```