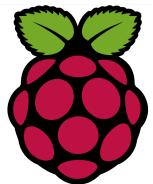


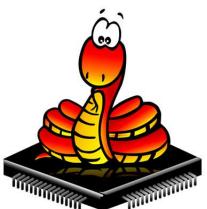


Atelier Raspi

Atelier N°13 Le Sapin

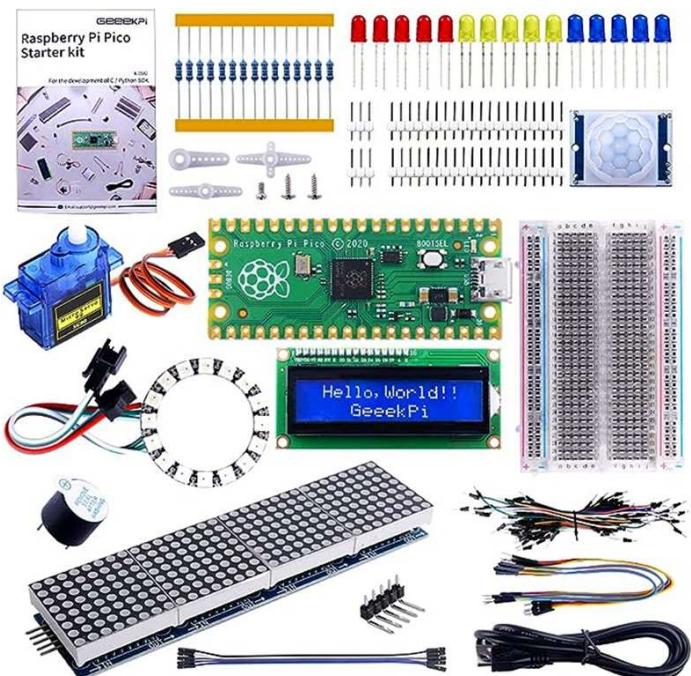


Logo du Raspberry Pico



Logo du MicroPython

L'atelier a pour valeurs, le partage, l'aide, la formation, le faire et construire ensemble à partir de l'expérience des participants



Atelier N°13 Le sapin

Etape 1 Le rayonnement infra rouge

Etape 2 Le capteur de mouvement infra rouge

Etape 3 Le programme du capteur IR de détection de mouvement

Etape 4 Câblage des LEDs et LED du sapin

Etape 5 Utilisation de la librairie Néopixel

Etape 6 Utilisation de la librairie Néopixel suite

Etape 7 La structure du programme sapin

A METTRE A JOUR

A6E1 Le rayonnement infra rouge IR

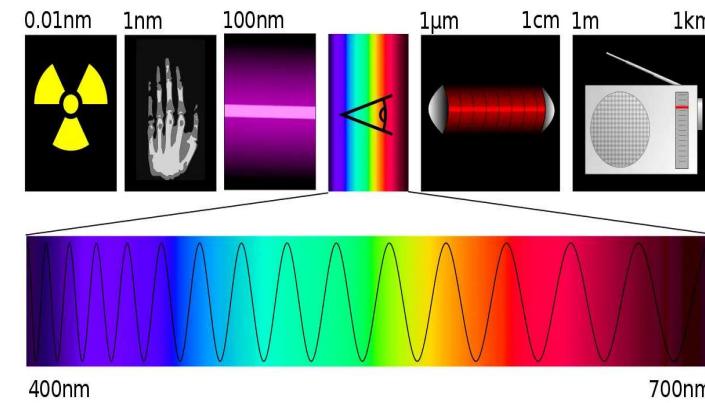
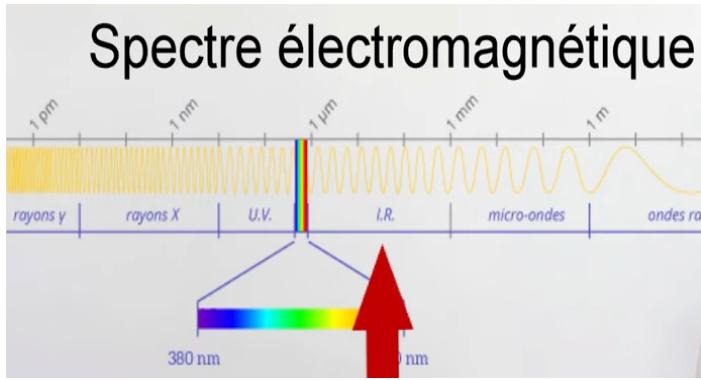


Photo
infra rouge



Télécommande
avec LED
infra rouge

Le rayonnement **infrarouge (IR)** est un rayonnement électromagnétique de longueur d'onde supérieure à celle du spectre visible mais plus courte que celle des micro-ondes ou des ondes radios.

Cette gamme de longueurs d'onde varie dans le vide de 700 nm à 0,1 mm

A6E2 Le capteur de mouvement infra rouge IR



Le capteur IR permet de détecter un mouvement par analyse des rayonnement infra rouge émis par le corps humain. Sa sortie change en fonction du dépassement d'un seuil réglé à l'avance. Il a donc une sortie TOR, Basse (0 volt) ou haute (3,3V) en fonction de sa détection.

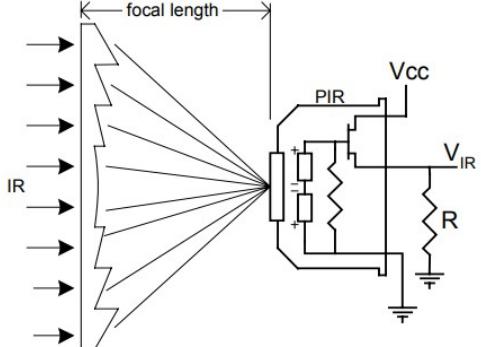
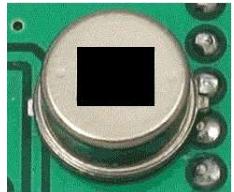
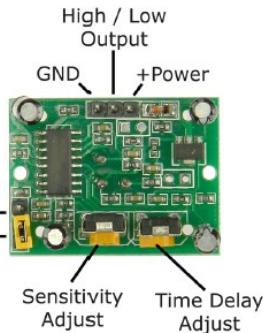
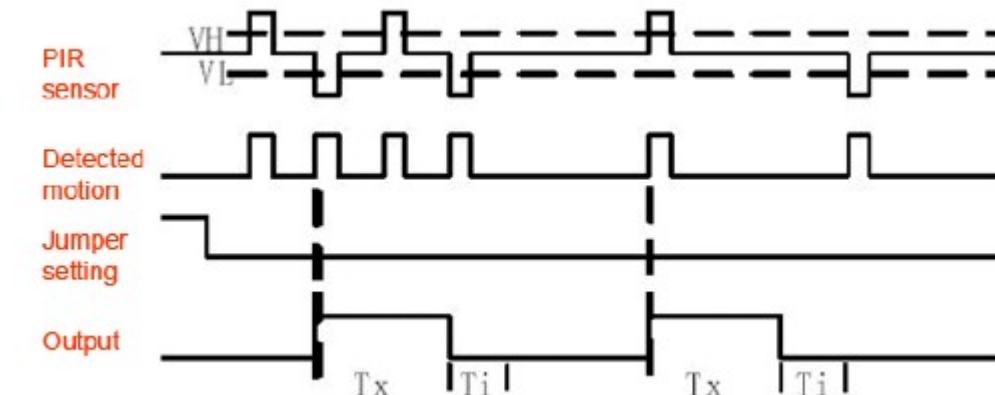


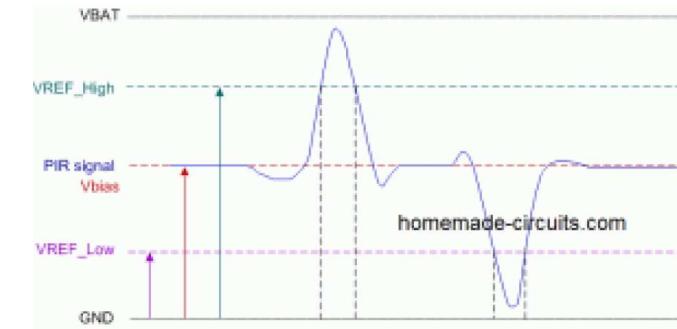
Figure 7: PIR Detector with Fresnel Lens



Capteur IR



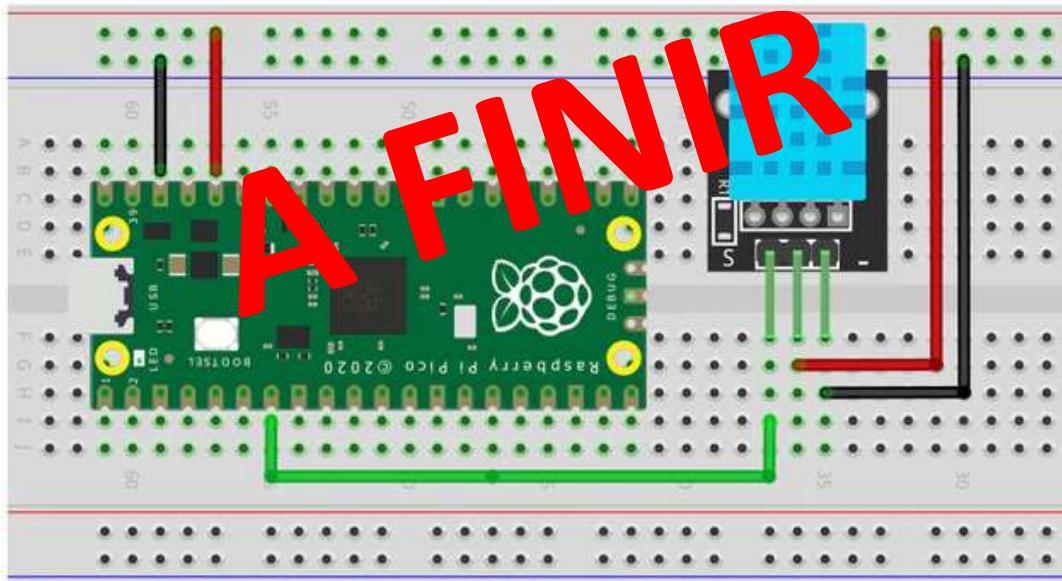
Différents réglages de la sensibilité et du temps de réaction.



Le signal issu du capteur, correspond à l'image simplifiée vue par chacune des zones sensibles du capteurs lorsqu'une personne passe devant.

Output est le signal de sortie TOR de la carte, on va détecter lorsqu'il sera à 1 ou avec une différence de potentiel « haut » soit 3,3v ou 5v.

A6E4 Câblage du PIR



Câblage GPIO 16

A6E3 Le programme capteur IR de détection

```
1 =====
2 # Detection simple d'une presence
3 # Affichage sur la console
4 =====
5 from machine import Pin
6 from time import sleep
7
8
9 # declaration du detecteur infra rouge
10 pir_sensor = Pin(16, Pin.IN)
11
12 while True:
13     reading = pir_sensor.value()
14     #si quelquechose a ete detecte
15     if reading == 1:
16         print("Presence detectée")
17     #si rien n'a ete detecte
18     else:
19         print("Rien a signaler")
20     #on attend 1 seconde
21     sleep(2)
22
23
```

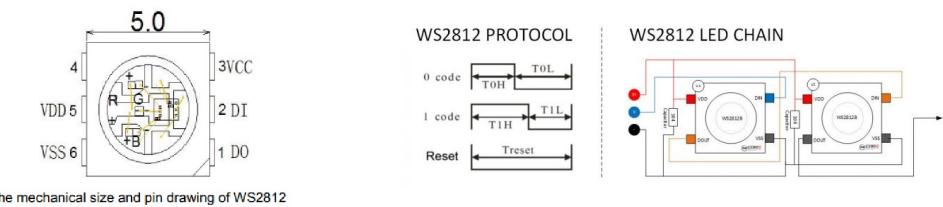
Console ×

```
>>> %Run -c $EDITOR_CONTENT

MPY: soft reboot
Rien a signaler
Rien a signaler
Presence detectée
Rien a signaler
Rien a signaler
Rien a signaler
```

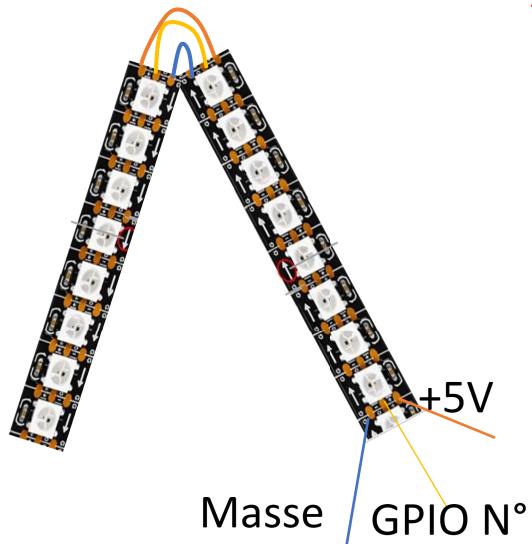
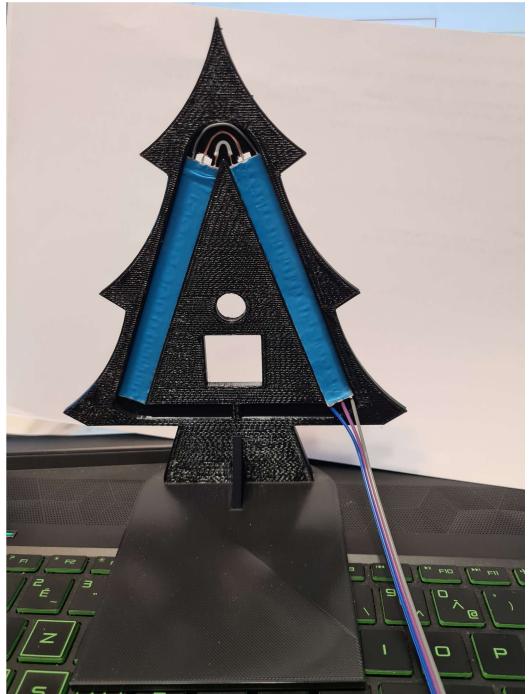
A6E4 Présentation du ruban de leds

A FINIR



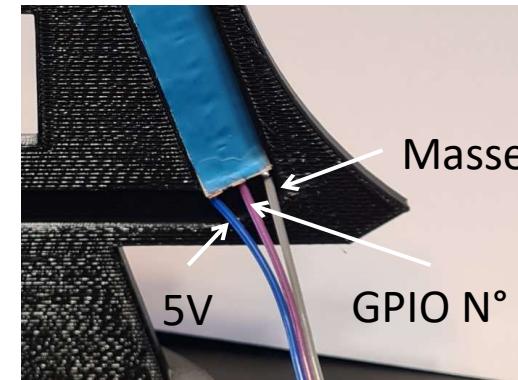
Câblage en série des leds

A6E4 Câblage des bandes LED du sapin

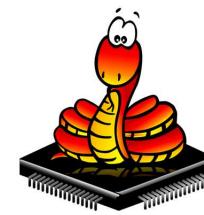


Le 5V peut être une alimentation extérieure de puissance
sinon câbler sur le Vbus pin 40
La masse doit être commune avec la carte Pico

Attention aux branchements, on va voir les fils de derrière
donc la masse sera à droite et le +5V à gauche,
la couleur des fils n'a pas forcément de signification.



A3E11 Utilisation de la librairie Neopixel



Fonctions utilisées:

pixels = Neopixel(i_nb_leds, 0, 0, "RGB") -> dans la variable pixels, on va utiliser la classe Néopixel avec les paramètres suivants: le nombre de LEDS à piloter (*i_nb_leds*) qui est une variable de notre programme, le N° de la machine d'état, le N° de la Pin utilisée, le mode de couleur RVB.

pixels.fill(*i_value*, *i_brightness*): -> on fait appel à la fonction fill() avec un parameter couleur et luminosité
exemple: “rouge avec (255,0,0) et 10 pour 10/255 de luminosité”

pixels.show(): -> on fait appel à la fonction d'afficher les LEDs (éteindre, allumer avec la couleur déjà programmée)

RGB (255, 0, 0)
RGB (255, 127, 0)
RGB (255, 255, 0)
RGB (0, 255, 0)
RGB (0, 0, 255)
RGB (75, 0, 130)

A6E3 Le programme du ruban de LEDs

```
1 from neopixel import Neopixel
2 import time
3
4 # declaration du ruban de leds (Pin 0)
5 NUM_LED = 30
6 PIN_NB = 0
7 leds = Neopixel(NUM_LED, 0, PIN_NB, "GRB")
8
9 # on definit la luminosite des leds / 255
10 leds.brightness(15)
11 # on eteint toute les leds
12 leds.clear()
13 # on affiche
14 leds.show()
15
16
17 while True:
18     # on allume toutes les leds du sapin en rouge pendant 1 sec
19     leds.fill((255,0,0))
20     leds.show()
21     time.sleep(1)
22     # on allume toutes les leds du sapin en rouge pendant 1 sec
23     leds.fill((0,255,0))
24     leds.show()
25     time.sleep(1)
26     # on allume toutes les leds du sapin en rouge pendant 1 sec
27     leds.fill((0,0,255))
28     leds.show()
29     time.sleep(1)
30 
```

A6E7 La structure du programme global du sapin

A partir des 2 programmes précédents, créer le programme suivant :

- Importer les modules nécessaires
- Déclaration du PIR et du ruban de leds
- Initialisations
- Boucle principale
 - Si détection
 - Allumer le sapin.
 - Sinon
 - Eteindre le sapin

A6E7 Correction

```
1 from machine import Pin
2 from neopixel import Neopixel
3 from time import sleep
4
5 # declaration du detecteur infra rouge
6 pir_sensor = Pin(16, Pin.IN)
7
8 # declaration du ruban de leds (Pin 0)
9 NUM_LED = 30
10 PIN_NB = 0
11 leds = Neopixel(NUM_LED, 0, PIN_NB, "GRB")
12
13 # on definit la luminosite des leds / 255
14 leds.brightness(15)
15 # on eteint toute les leds
16 leds.clear()
17 # on affiche
18 leds.show()
19
20 while True:
21     reading = pir_sensor.value()
22     #si quelquechose a ete detecte
23     if reading == 1:
24         print("Presence detectée")
25         # on allume toutes les leds du sapin en rouge
26         leds.fill((255,0,0))
27         leds.show()
28     #si rien n'a ete detecte
29     else:
30         print("Rien a signaler")
31         # on eteint les leds du sapin
32         leds.clear() # equivalent a: leds.fill((0,0,0))
33         leds.show()    #on attends 1 seconde
34         sleep(2)
35
36
```

A6E7 Pour aller plus loin avec la lib Neopixel...

```
1 from neopixel import Neopixel
2 import time
3
4 # declaration du ruban de leds (Pin 0)
5 NUM_LED = 30
6 PIN_NB = 0
7 leds = Neopixel(NUM_LED, 0, PIN_NB, "GRB")
8
9 # declaration de quelques couleurs
10 couleur_rouge = (255,0,0)
11 couleur_vert = (0,255,0)
12 couleur_bleu = (0,0,255)
13 couleur_noir = (0,0,0)
14
15 # on definit la luminosite des leds
16 leds.brightness(15)
17 # on eteint toute les leds
18 leds.clear()
19 # on affiche
20 leds.show()
21
```

```
22 while True:
23     # on allume tour a tour les leds du sapin en bleu
24     for i in range (0,NUM_LED):
25         leds.set_pixel(i,couleur_bleu)
26         leds.show()
27         time.sleep(0.1)
28         leds.clear()
29         leds.show()
30
31     # on allume un bloc de leds
32     leds.set_pixel_line(5,20,couleur_rouge)
33     leds.show()
34     time.sleep(2)
35     leds.clear()
36     leds.show()
37
38     # on fait un gradient de couleur
39     leds.set_pixel_line_gradient(5,20,couleur_vert,couleur_bleu)
40     leds.show()
41     time.sleep(2)
42     leds.clear()
43     leds.show()
```

```
#Rotation d'un bloc de leds dans un sens
leds.set_pixel_line(0,3,couleur_bleu)
for i in range (NUM_LED):
    leds.rotate_right(1)
    leds.show()
    time.sleep (0.2)
    i=i+1
leds.clear()
leds.show()
```

```
#Rotation dans l'autre sens
leds.set_pixel_line(0,3,couleur_vert)
for i in range (NUM_LED):
    leds.rotate_left(1)
    leds.show()
    time.sleep (0.2)
    i=i+1
leds.clear()
leds.show()
```

A6E7 Les couleurs

1		RGB(0,0,0)	29		RGB(128,0,128)
2		RGB(255,255,255)	30		RGB(128,0,0)
3		RGB(255,0,0)	31		RGB(0,128,128)
4		RGB(0,255,0)	32		RGB(0,0,255)
5		RGB(0,0,255)	33		RGB(0,204,255)
6		RGB(255,255,0)	34		RGB(204,255,255)
7		RGB(255,0,255)	35		RGB(204,255,204)
8		RGB(0,255,255)	36		RGB(255,255,153)
9		RGB(128,0,0)	37		RGB(153,204,255)
10		RGB(0,128,0)	38		RGB(255,153,204)
11		RGB(0,0,128)	39		RGB(204,153,255)
12		RGB(128,128,0)	40		RGB(255,204,153)
13		RGB(128,0,128)	41		RGB(51,102,255)
14		RGB(0,128,128)	42		RGB(51,204,204)
15		RGB(192,192,192)	43		RGB(153,204,0)
16		RGB(128,128,128)	44		RGB(255,204,0)
17		RGB(153,153,255)	45		RGB(255,153,0)
18		RGB(153,51,102)	46		RGB(255,102,0)
19		RGB(255,255,204)	47		RGB(102,102,153)
20		RGB(204,255,255)	48		RGB(150,150,150)
21		RGB(102,0,102)	49		RGB(0,51,102)
22		RGB(255,128,128)	50		RGB(51,153,102)
23		RGB(0,102,204)	51		RGB(0,51,0)
24		RGB(204,204,255)	52		RGB(51,51,0)
25		RGB(0,0,128)	53		RGB(153,51,0)
26		RGB(255,0,255)	54		RGB(153,51,102)
27		RGB(255,255,0)	55		RGB(51,51,153)
28		RGB(0,255,255)	56		RGB(51,51,51)