

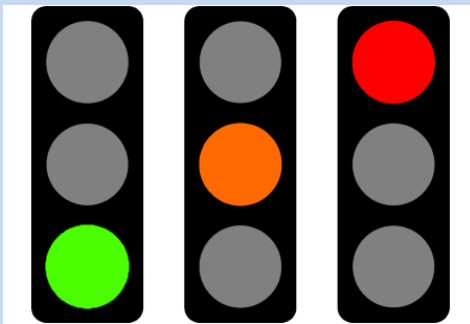
# Activité Feu rouge Micro:Bit

## 1. Introduction

Le feu à 3 états.

C'est le système utilisé en France (mais également dans d'autres pays). Une seule couleur est affichée à la fois :

- Feu vert, les voitures peuvent rouler.
- Feu orange (ou jaune), les voitures doivent s'arrêter si c'est possible.
- Feu rouge, les voitures doivent s'arrêter.



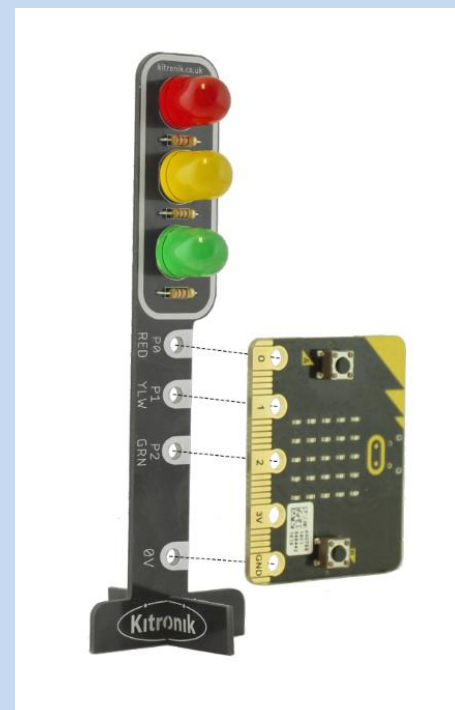
## 2. Assemblage du feu.

La carte Stop:bit Traffic Light est une carte à visser sur la carte micro:bit, reproduisant un feu de circulation.



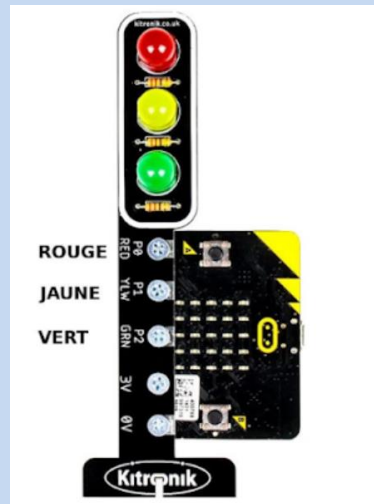
Consulter la page suivante pour l'assemblage du feu.  
<https://www.youtube.com/watch?v=sfh7j-nq8GM&feature=youtu.be>

**Attention au sens de montage de la carte !!!**  
**Attention a ne pas perde les vis et écrous !!!**



**Micro:Bit**





Résultat final.

### 3. Programmation du feu.

On va utiliser l'éditeur python en ligne : <https://python.microbit.org/v/1.1>

```

1 # Add your Python code here. E.g.
2 from microbit import *
3
4
5 while True:
6     display.scroll('Hello, World!')
7     display.show(Image.HEART)
8     sleep(2000)
9

```

### Les broches Stop:bit Traffic Light

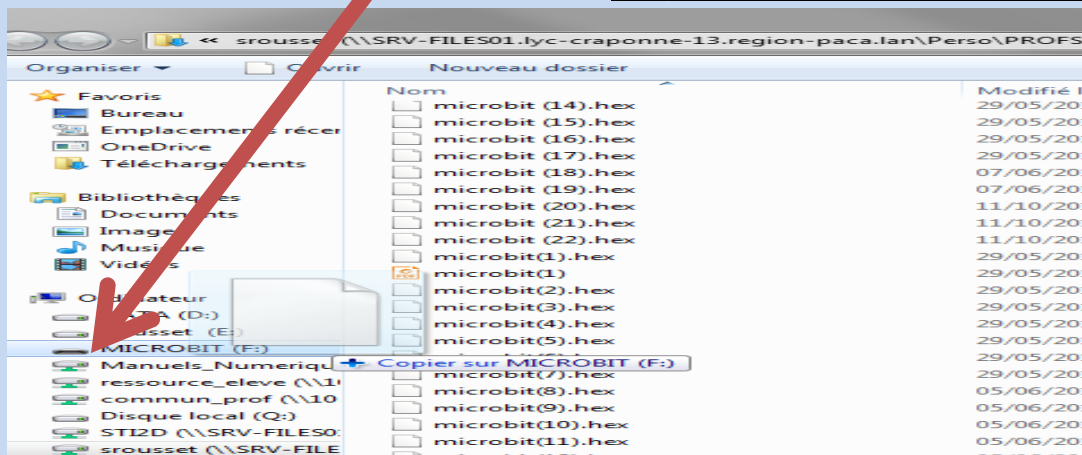
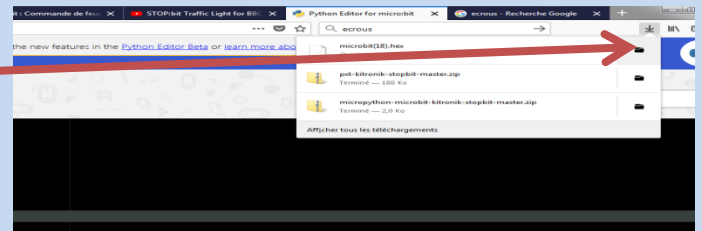
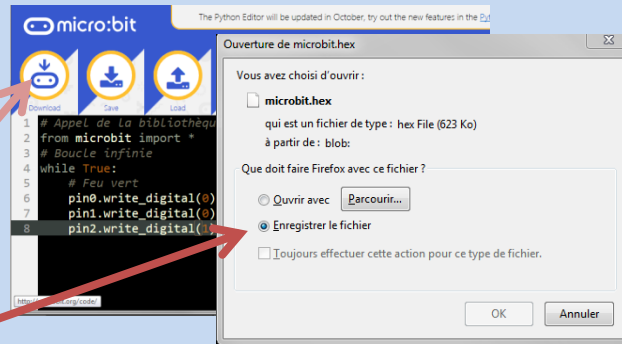
Pinout	
P0	Red LED
P1	Yellow LED
P2	Green LED



- Toujours commencer un programme par la ligne *from microbit import \**
- Ecrire le programme suivant (**Attention à respecter les minuscules/majuscules et les espaces**)

```
# Appel de la bibliothèque "microbit"
from microbit import *
# Boucle infinie
while True:
    pin0.write_digital(0)
    pin1.write_digital(0)
    pin2.write_digital(1)
```

- Cliquer sur Download
- Enregistrer le fichier
- Ouvrir le dossier contenant fichier.
- Copier ce fichier sur la carte MICROBIT



- Que se passe-t-il ?
- Commenter toute les lignes du programme.
- Programmer le feux pour allumer toutes les couleurs. Expliquer votre démarche.



## 4. Exemple du feu tricolore:

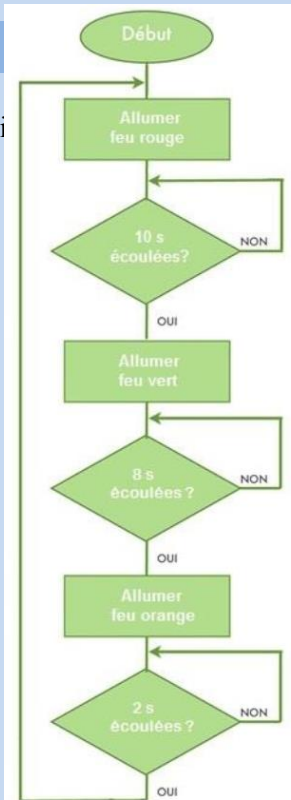
- Réaliser le programme du feu tricolore en fonction de l'algorithme suivant

En langage python une temporisation de 1s s'écrit :

```
Sleep (1000)
```



- Commenter toutes les lignes votre programme.



## 5. Modification du comportement du feu tricolore

Pour fluidifier le trafic la nuit les feux passent orange clignotant.

Lors d'un appuie sur le bouton A les feux passent à l'orange clignotant, si on relâche le bouton A on annule ce fonctionnement.

Aide pour la programmation.

- button\_a.is\_pressed()** : à utiliser dans un test, cette méthode renvoie la valeur booléenne True si le bouton est pressé et False dans le cas contraire

### 4.4.2. L'instruction conditionnelle : IF

Pour savoir si un bouton a été pressé, on va utiliser une instruction conditionnelle : IF (= "Si")



- Pour un test unique (aucune autre alternative n'est traitée) :

```
Si (condition vraie):  
    faire ceci
```

- Pour un test unique (toutes les autres alternatives sont traitées) :

```
Si (condition vraie):  
    faire ceci  
Sinon:  
    faire celà
```

- Pour un test multiple (ici on teste trois conditions, avec en plus le traitement de toutes les autres alternatives) :

```
Si (condition1 vraie):  
    faire ceci  
SinonSi (condition2 vraie):  
    faire cela  
SinonSi (condition3 vraie):  
    faire ça  
Sinon:  
    faire cette chose
```

Pour rédiger le code en Python il suffira de remplacer :

- **Si par if**
- **Sinon par else**
- **SinonSi par elif (contraction de else if)**

L'utilisation de l'instruction conditionnelle if, associée à cette méthode permet alors d'orienter le programme selon le bouton appuyé :

```
from microbit import *  
  
display.show('?')  
  
while True: # création d'une boucle infinie  
    if button_a.is_pressed():  
        display.show('A')  
    elif button_b.is_pressed():  
        display.show('B')
```

Faire valider par le grand manitou de la prog.

