

# Simulation d'un feu de chantier pour la circulation automobile

## Version 0

Créer une simulation de feu dans laquelle

- un feu comporte 2 LEDs (orange et rouge) connectées sur les broches 0 et 1 de la carte. En l'absence de LEDs, afficher le feu sous forme d'un carré de 4 pixels en haut à droite pour rouge et en bas à droite pour orange (une valeur de pixels égale à 4 simulera une couleur différente).
- le feu change toutes les 10 secondes
- un compte à rebours s'affiche sur l'écran

On pourra utiliser les fonctions ci-dessous permettant de simuler un feu rouge, un feu orange et l'affichage du temps restant

*# Version avec LED*

```
def afficheRouge():
    pin1.write_digital(1)
    pin0.write_digital(0)

def afficheOrange():
    pin1.write_digital(0)
    pin0.write_digital(1)

def afficheAttente(attente):
    display.show(str(attente))
```

*# Version avec carte micro:bit seule*

```
def afficheRouge():
    for x in range(3,5):
        for y in range(2):
            display.set_pixel(x,y,9)

    for x in range(3,5):
        for y in range(3,5):
            display.set_pixel(x,y,0)

def afficheOrange():
    for x in range(3,5):
        for y in range(2):
            display.set_pixel(x,y,0)

    for x in range(3,5):
        for y in range(3,5):
            display.set_pixel(x,y,4)

def afficheAttente(attente):
    for x in range(2):
        for y in range(5):
            display.set_pixel(x,y, 0 if 5*x+y >= attente else 9)
```

# version 1 : Simulation d'un système de 2 feux de chantier connectés

Étudier un système de 2 feux connectés dans lequel

- les feux sont en opposition !
- Les 2 feux affichent le même compte à rebours

Étudiez les deux programmes suivants

```
from microbit import *
import radio

radio.config(group=2)
radio.on()

feuRouge = True

def afficheRouge():
    pin1.write_digital(1)
    pin0.write_digital(0)
def afficheOrange():
    pin1.write_digital(0)
    pin0.write_digital(1)
def afficheAttente(attente):
    display.show(str(attente))

while True:
    # Traitement des messages
    incoming = radio.receive()
    if incoming:
        if incoming == "VERT" :
            feuRouge = False
        elif incoming == "ROUGE" :
            feuRouge = True
        elif len(incoming) ==1:
            afficheAttente(int(incoming))

# Affichage Rouge / Orange
if feuRouge :
    afficheRouge()
else :
    afficheOrange()
```

```
from microbit import *
import radio

radio.config(group=2)
radio.on()

DUREE = 10 # 10 secondes
attente = DUREE
feuRouge = True

def afficheRouge():
    pin1.write_digital(1)
    pin0.write_digital(0)
def afficheOrange():
    pin1.write_digital(0)
    pin0.write_digital(1)
def afficheAttente(attente):
    display.show(str(attente))

while True:
    # Affichage Rouge / Orange
    if feuRouge :
        afficheRouge()
    else :
        afficheOrange()

# Gestion de l'attente
sleep(1000)
attente -= 1
afficheAttente(attente)
radio.send(str(attente))

# Alternance du feu
if attente == 0 :
    attente=DUREE
    feuRouge = not feuRouge
    if feuRouge :
        radio.send("VERT")
    else:
        radio.send("ROUGE")
```

Répondez sur feuille aux questions suivantes :

- Les programmes sur les 2 cartes sont-ils identiques ? Pourquoi ?
- Identifiez le programme maître et le programme esclave.
- Décrire le protocole de communication entre les cartes (quels messages sont échangés entre les cartes)
- Est-on certain de la robustesse du système ?
  - les feux peuvent-ils être tous les deux au rouge ? Est-ce gênant ?
  - les feux peuvent-ils être tous les deux au vert ? Est-ce gênant ?
  - quelles sont les conséquences du plantage d'un des feux ?

Ces deux programmes peuvent être simulés dans [le simulateur en ligne](#), mais l'un après l'autre indépendamment. Réglez les paramètres suivant dans l'onglet *Radio* et cliquez sur **UPDATE**

- Channel : 7
- Group : 2
- Address : 0x75626974

Sur la simulation du programme maître vous verrez défiler les message envoyés, sur l'autre programme, vous pourrez utiliser le champ *Message* pour simuler l'envoi d'une commande par le maître (essayez d'envoyer 3 ou VERT et observez le comportement de la carte).

## Version 2 : Le véhicule prioritaire

Un véhicule prioritaire arrive à un feu ce qui doit le faire basculer au vert immédiatement.

- Imaginer par quel(s) moyen(s) on peut interagir avec la carte pour simuler cette situation (IHM)
- la manière de prendre en compte l'arrivée du véhicule prioritaire est-elle identique selon le feu qui reçoit le signal ?
- **situation A** : Le véhicule prioritaire arrive au feu maître. Modifier le programme maître afin qu'un appui sur le bouton A passe le feu maître à l'orange et l'autre feu au rouge.
- **situation B** : Le véhicule prioritaire arrive à l'autre feu. Que faut-il modifier pour prendre en compte cette situation ?
- Décrire les modifications apportées au protocole de communication entre les cartes pour le scénario B.

## Version 3 : Améliorer la robustesse du dispositif

- Quel scénario proposeriez-vous afin de renforcer la robustesse du dispositif (tolérance aux pannes) du dispositif et renforcer la sécurité ?
- Proposez une modification des programmes en ce sens