

Les microcontrôleurs en Maths-Sciences

-exemples d'utilisation-

-Microcontrôleurs-

qu'est-ce que c'est?

Un microcontrôleur est un circuit intégré associant sur la même puce :

- un micro-processeur
- de la mémoire
- des périphériques (*capteurs/actionneurs*)

Utilisation en Maths-Sciences: Pour quoi?

- Acquisition de données avant traitement
- Génération de signaux
- Réalisation *de simulation*
- Mise en œuvre de *l'algorithmique/programmation*

Où en trouve-t-on?

Aujourd'hui, ils sont présents un peu partout : dans les appareils domestiques, médicaux, de télécommunication, dans les voitures, les avions, *l'industrie*, ...

Comment les utiliser?

Les divers types de microcontrôleurs se différencient entre autres par leur architecture, le nombre et le type de périphériques d'entrées/sorties et le langage de programmation: au lycée, le choix de *Python comme langage de programmation des microcontrôleurs* semble être en adéquation avec les référentiels.

-Quelques microcontrôleurs disponibles au grand public-



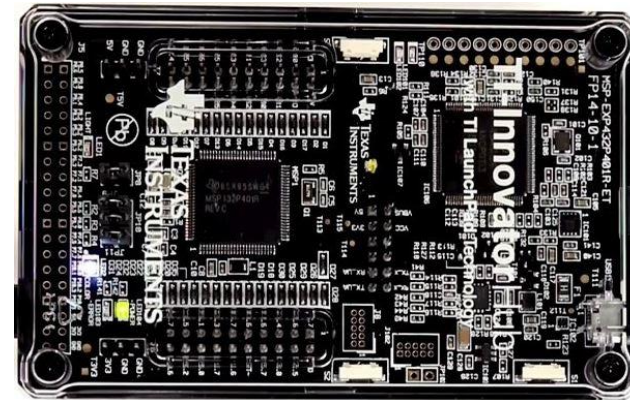
Carte micro:bit (nRF51822)



Carte PyBoard (STM32)



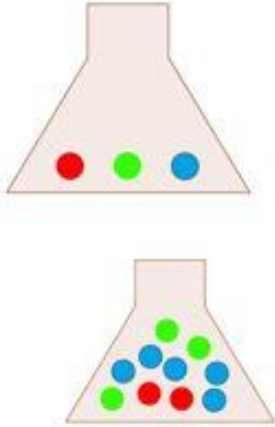
Carte Arduino UNO (Atmel ATMEGA 328)



Carte Ti-HUB (MSP-EXP432P401-ET)

Exemples simples d'utilisation

Simulation de tirages dans une urne



```

ÉDITEUR - URNE1
LIGNE DU SCRIPT 0001
# Hub Project
from ti_system import *
from time import *
from random import *
import color

import ti_plotlib as pit

def tirages(n):
    for i in range(1,n+1):
        ... c=randint(1,3)
        ... if c==1:
        ...     color.rgb(255,0,0)
        ...     wait(3)
        ...     color.off()
        ... elif c==2:
        ...     color.rgb(0,255,0)
        ...     wait(3)
        ...     color.off()
        ... else:
        ...     color.rgb(0,0,255)
        ...     wait(3)
        ...     color.off()
    ... return "fin des tirages"
Fns... | a | # | Outils | Exéc | Script
    
```

Synthèse additive des couleurs

Durant son concert, un chanteur a besoin d'être éclairé à l'aide d'une lumière jaune. Les deux éclairagistes vérifient le matériel dont la salle de concert est pourvue. Il s'avère que les projecteurs ne disposent que de filtres bleus, rouges et verts.

Les éclairagistes peuvent-ils répondre aux besoins du chanteur?



```

ÉDITEUR - MICROCTL
LIGNE DU SCRIPT 0003
# Hub Project
from ti_system import *
from time import *
import ti_plotlib as pit

import color

def couleur(r,g,b):
    ... color.rgb(r,g,b)
    ... wait(5)
    ... color.rgb(0,0,0)
    ... return "fin!"
Fns... | a | # | Outils | Exéc | Script
    
```



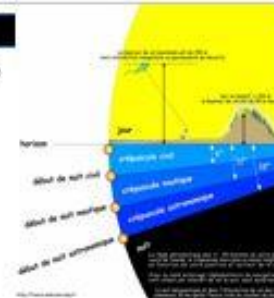
```

ÉDITEUR - UK001
LIGNE DU SCRIPT 0002
# Hub Proje
from ti_system import *
from time import *
import color
import brightness
Mesure
disp_clr()
lum0=brightness.measurement()
disp_at(5,"modifier la luminosité",4,"left")
disp_wait()
lum1=brightness.measurement()
while not lum0==lum1:
    if lum0<lum1:
        ... color.rgb(255,0,0)
        ... sleep(2)
        ... color.off()
    elif lum0>lum1:
        ... color.rgb(0,255,0)
        ... sleep(2)
        ... color.off()
    if lum0<lum1:
        ... color.rgb(255,0,0)
        ... sleep(2)
        ... color.off()
    elif lum0>lum1:
        ... color.rgb(0,255,0)
        ... sleep(2)
        ... color.off()
    else:
        ... color.off()
Fns... | a | # | Outils | Exéc | Script
    
```

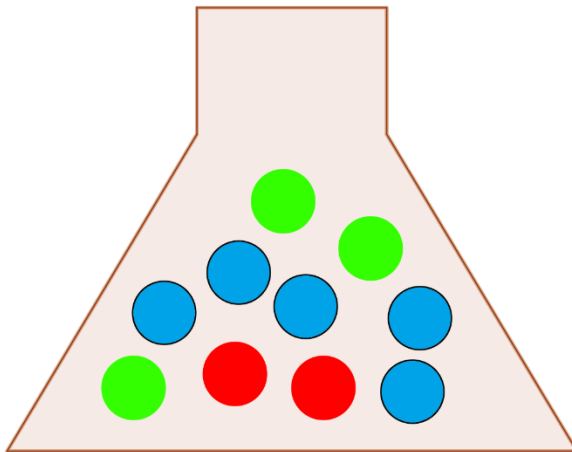
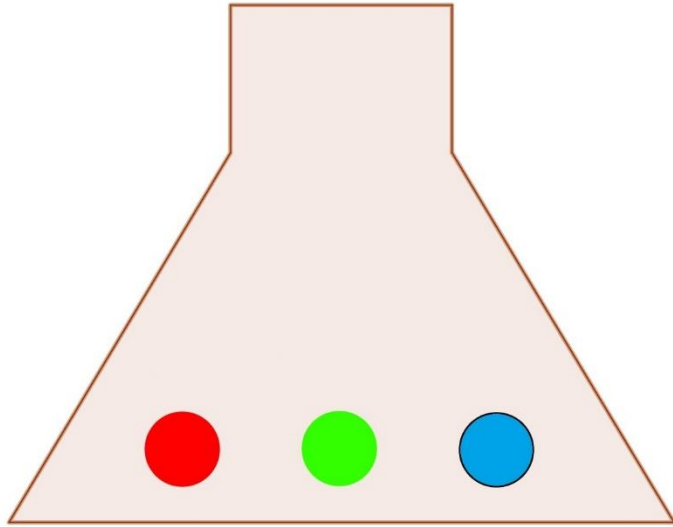
Interrupteur crépusculaire

Projet: réaliser un éclairage automatique dépendant de la luminosité ambiante.

Mesurer l'intensité lumineuse ambiante :
 Lum0 ← mesure ± (tolérance ?)
 Modifier l'intensité lumineuse (temp; cache devant le capteur)
 Lum1 ← mesure
 Si Lum1 > Lum0 :
 Alors allumer la DEL RVB en rouge (2s)
 Sinon Si Lum0 < Lum1 :
 Alors allumer la DEL RVB en vert (2s)
 Sinon : Ne rien faire



Simulation de tirages dans une urne



```
ÉDITEUR : URNE1
LIGNE DU SCRIPT 0001

# Hub Project
from ti_system import *
from time import *
from random import *
import color

import ti_plotlib as plt

def tirages(n):
    for i in range(1,n+1):
        c=randint(1,3)
        if c==1:
            color.rgb(255,0,0)
            wait(3)
            color.off()
        elif c==2:
            color.rgb(0,255,0)
            wait(3)
            color.off()
        else:
            color.rgb(0,0,255)
            wait(3)
            color.off()
    return "fin des tirages"
```

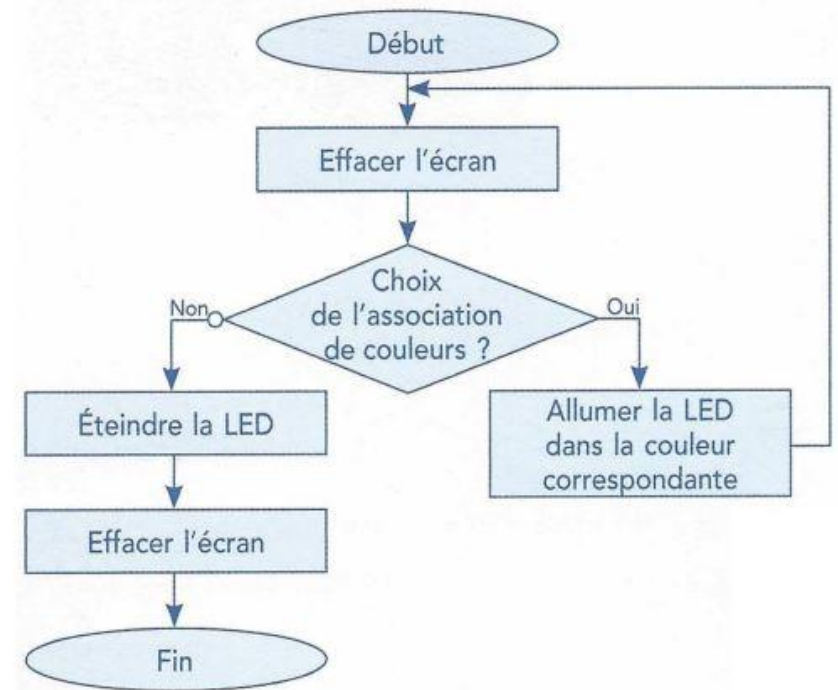
Fns... | a A # | Outils | Exéc | Script



Synthèse additive des couleurs

Durant son concert, un chanteur a besoin d'être éclairé à l'aide d'une lumière jaune. Les deux éclairagistes vérifient le matériel dont la salle de concert est pourvue. Il s'avère que les projecteurs ne disposent que de filtres bleus, rouges et verts.

Les éclairagistes peuvent-ils répondre aux besoins du chanteur?



ANALYSER



REALISER



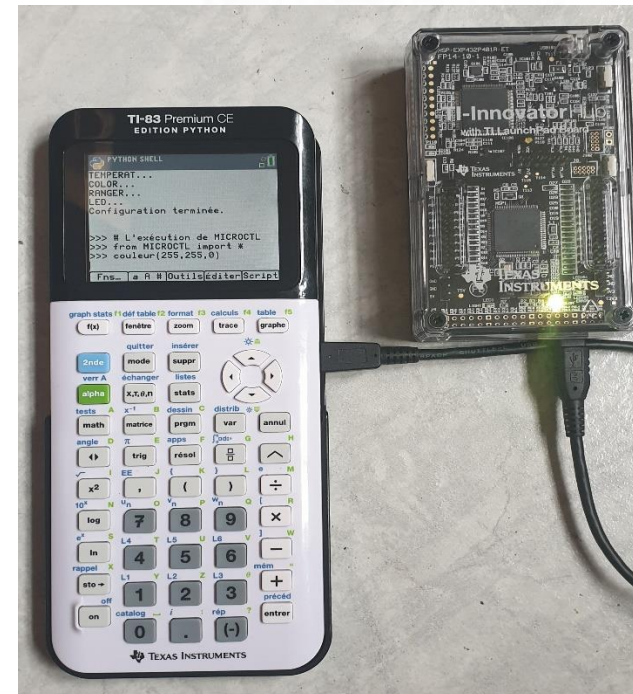
VALIDER

```
ÉDITEUR : MICROCTL
LIGNE DU SCRIPT 0001
_Hub Project
from ti_system import *
from time import *
import ti_plotlib as plt

import color

def couleur(r,g,b):
    color.rgb(r,g,b)
    wait(5)
    color.rgb(0,0,0)
    return "fin!"

Fns... a A # Outils Exéc Script
```



```

ÉDITEUR : U6SB1
LIGNE DU SCRIPT 0002
# Hub Projet
from ti_system import *
from time import *
import color
import brightns
#mesure
disp_clr()
lum0=brightns.measurement()
disp_at(5,"modifier la luminosité", "left")
disp_wait()
lum1=brightns.measurement()
#comparaison
if lum0<lum1:
  **color.rgb(255,0,0)
  **sleep(2)
  **color.off()
elif lum0>lum1:
  **color.rgb(0,255,0)
  **sleep(2)
  **color.off()
if lum0<lum1:
  **color.rgb(255,0,0)
  **sleep(2)
  **color.off()
elif lum0>lum1:
  **color.rgb(0,255,0)
  **sleep(2)
  **color.off()
else:
  **color.off()

```

Fns... a A # Outils Exéc Script

Interrupteur crépusculaire

Projet: réaliser un éclairage automatique dépendant de la luminosité ambiante.

Mesurer l'intensité lumineuse ambiante :
 $Lum0 \leftarrow \text{mesure} \pm (\text{tolérance ?})$
 Modifier l'intensité lumineuse (lampe ; cache devant le capteur)
 $Lum1 \leftarrow \text{mesure}$
 Si $Lum1 > Lum0$:
 | Alors allumer la DEL RVB en rouge (2s)
 Sinon Si $Lum0 < Lum1$:
 | Alors allumer la DEL RVB en vert (2s)
 Sinon : Ne rien faire

