

# Labyrinthe à bille

## Créer une scène 3D interactive

# Document Technique

## Carte de référence **pyFirmata**



[Philippe Roy <philippe.roy@ac-grenoble.fr>](mailto:philippe.roy@ac-grenoble.fr)

<https://forge.aeif.fr/blender-edutech/blender-edutech-tuto>

# Carte de référence pyFirmata



## Entête du programme

Tout programme commence par l'import des bibliothèques, la création de l'objet carte et le démarrage de l'itérateur.

### Création de l'objet carte

C'est à cet endroit qu'il faut spécifier le port de communication (par exemple : 'COM3' '/dev/ttyACM0', ...).

### Démarrage de l'itérateur

C'est nécessaire pour lire les entrées de la carte.

### Destruction de l'objet carte

### Import des Bibliothèques

La bibliothèque `time` n'est obligatoire mais recommandée car elle est très souvent utilisée.

```
import pyfirmata }
import time      }

# Connexion à la carte Arduino
carte = pyfirmata.Arduino('/dev/ttyACM0')
print("Communication Carte Arduino établie")

# Itérateur pour les entrées
it = pyfirmata.util.Iterator(carte)
it.start()
```

## Fin du programme

Tout programme termine par la fermeture du port de communication.

```
# Fermer la connexion à la carte Arduino
carte.exit()
```



# Carte de référence pyFirmata



## Lecture des entrées et écriture des sorties

La lecture des entrées se fait avec la fonction : `entree.read()`

En fonction de la tension au borne de la broche, la valeur retournée est

- pour une entrée binaire : un valeur binaire : 0 (0 V) ou 1 (5 V)
- pour une entrée analogique : une valeur décimale comprise entre 0 et 1

```
bouton = carte.get_pin('d:10:i') # bouton sur la broche 10
cpt_son = carte.get_pin('a:0:i') # capteur sonore sur la broche A0
if bouton.read() == 1:          # test si le bouton est appuyé
    print(cpt_son.read())       # affiche la valeur du capteur sonore
```

L'écriture des sorties fait avec la fonction : `sortie.write(valeur)`

En fonction de la tension qu'on souhaite appliquer à la broche : la **valeur** est

- pour une sortie binaire : une valeur binaire : 0 (0 V) ou 1 (5 V)
- pour une sortie pwm : une valeur décimale comprise entre 0 et 1

```
led_r = carte.get_pin('d:13:o') # led rouge sur la broche 13
led_v = carte.get_pin('d:11:p') # led verte sur la broche 11 en PWM
led_r.write(1)                  # allumer la led rouge (5 V)
led_r.write(0)                  # éteindre la led rouge (0 V)
led_v.write(0.20)               # allumer la led verte faiblement (1 V)
```