



Les animations de la Vilaine Bidouille  
à la Médiathèque de Redon

Saison 02 - Episode 02

Raspberry Pi pour Yéti

par [Cyrille Laulier](#)



## Objectif :

C'est une création originale de Lever le Rideau : Un yéti mécanique qui grandit au fur et à mesure qu'il est nourri.

La partie basse du corps du yéti est fixe par rapport au sol. La partie haute est mobile. Elle peut monter et descendre pour donner l'illusion que le yéti grandit.

Les enfants lanceront des poids à travers sa gueule. De l'autre côté, le poids tomberont dans un sceau faisant contrepoids avec la partie haute.

Le sceau baisse, le haut du corps monte.

Autre détail, la tête peut s'incliner à gauche et à droite.

Le but de Lever le Rideau est d'équiper le yéti d'un système électronique permettant au yéti de pousser des cris quand il a bien mangé. Eventuellement, il pourrait bouger la tête.

## La solution toute trouvée : la carte Raspberry Pi

C'est un ordinateur complet sous la forme d'une carte électronique grande comme un paquet de cartes.

La Raspberry Pi tourne sous Linux (Raspbian). On y branche son clavier, sa souris, son écran, ses périphériques USB.

Et on l'utilise comme un ordinateur normal.

Mais l'énorme avantage de la Raspberry Pi, c'est qu'elle est équipée de connections avec le monde réelle : les fiches d'entrée/sortie. Comme une carte Arduino, elle peut interagir avec son environnement.

Si le sceau du yéti descend jusqu'à appuyer sur un bouton, la Raspi peut le détecter, l'interpréter via un programme, et, déclencher la lecture d'un son pré-enregistré.



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21

## 1ere étape : Configuration de la carte :

Il faut être connecté à internet. Sinon rien de ce qui suit ne fonctionne.

a) La mise à jour (un classique de Linux) :

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

b) Installation du langage de programmation Python

```
sudo apt-get install python-dev
```

c) Expliquer à Python comment utiliser les connexions entrée/sortie

```
wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.3.1a.tar.gz
```

```
tar xzf RPi.GPIO-0.3.1a.tar.gz
```

```
cd RPi.GPIO-0.3.1a
```

```
sudo python setup.py install
```

d) Lancer Python

```
sudo idle
```

## 2nde Etape : La programmation

Analysons le programme yeti2.py

```
# -*- coding: iso-8859-15 -*-  
import RPi.GPIO as GPIO  
import time  
import pygame
```

```
pygame.mixer.init()
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(17, GPIO.OUT)
```

```
GPIO.setup(25, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```

```
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```

```
GPIO.setwarnings(False)
```

```
fin = 0
```

```
ajoutAngle = 2  
nbrTour = 1  
angle1 = 0  
angle2 = 180
```

```
print("\n+-----/ ServoMoteur Controlleur /-----+")  
print(" |")  
print(" | Le Servo doit etre branche au pin 11 / GPIO 17 |")  
print(" | Le saut doit etre branche au pin 22 / GPIO 25 |")  
print(" | La langue doit etre branche au pin 18 / GPIO 24 |")  
print("+-----+\n")
```

```
pygame.mixer.music.load("Goliath_Online.mp3")  
pygame.mixer.music.play()  
time.sleep(3)  
pygame.mixer.music.load("Goliath_Target.mp3")  
pygame.mixer.music.play()  
time.sleep(3)
```

```
while (fin==0) :  
    cctSaut=GPIO.input(25)  
    cctLangue=GPIO.input(24)  
    if (cctSaut == 0):  
        pygame.mixer.music.load("criYeti.mp3")  
        pygame.mixer.music.play()  
        time.sleep(1)  
        pwm=GPIO.PWM(17,100)  
        pwm.start(5)  
        duty1 = float(angle1)/10 + ajoutAngle  
        duty2= float(angle2)/10 + ajoutAngle  
        i = 0  
        while i <= nbrTour:  
            pwm.ChangeDutyCycle(duty1)  
            time.sleep(0.8)  
            pwm.ChangeDutyCycle(duty2)  
            time.sleep(0.8)
```

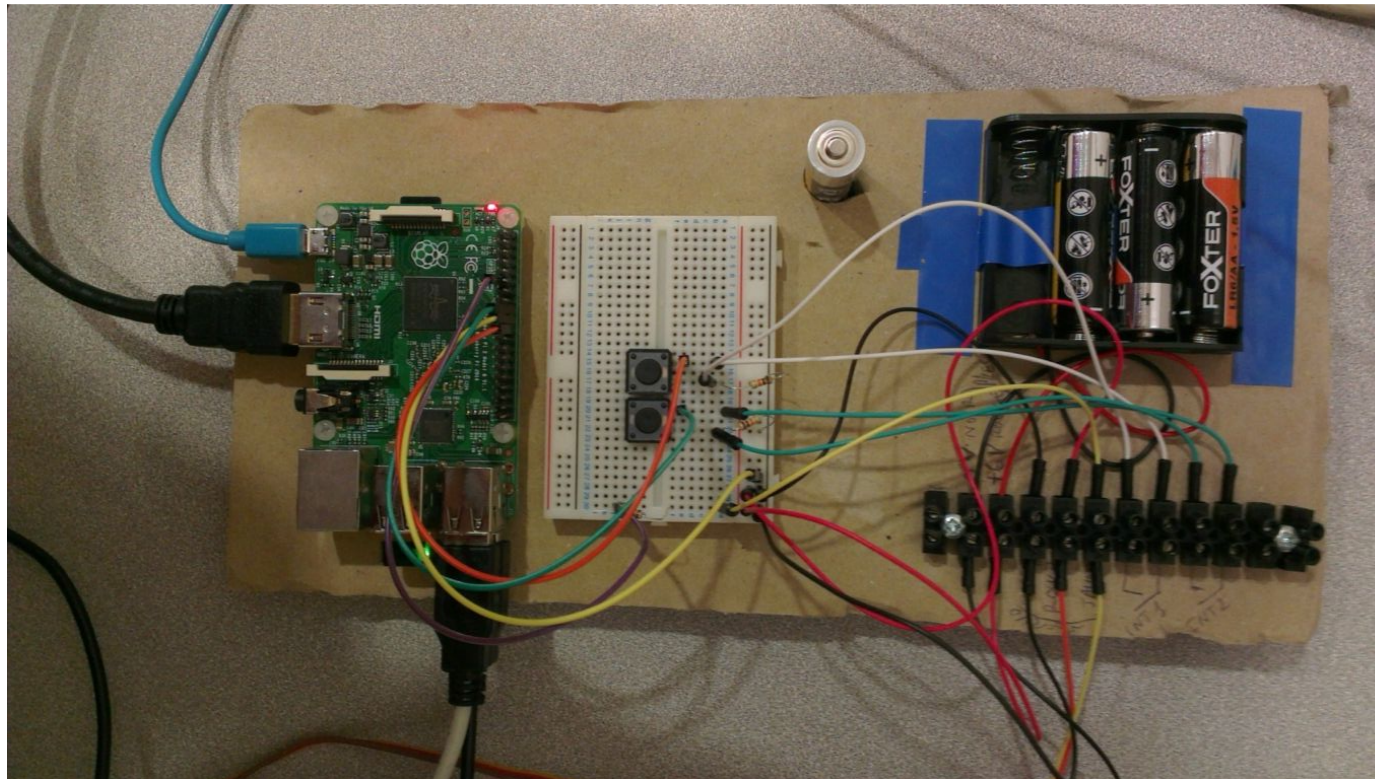
```
    i = i+1
    pwm.stop()
    time.sleep(i)
if (cctLangue == 0):
    pygame.mixer.music.load("faimYeti.mp3")
    pygame.mixer.music.play()
    time.sleep(i)
    pwm=GPIO.PWM(17,100)
    pwm.start(5)
    duty1 = float(angle1)/10 + ajoutAngle
    duty2= float(angle2)/10 + ajoutAngle
    i = 0
    while i <= nbrTour:
        pwm.ChangeDutyCycle(duty1)
        time.sleep(0.8)
        pwm.ChangeDutyCycle(duty2)
        time.sleep(0.8)
        i = i+1
```



```
pwm.stop()  
time.sleep(i)
```

### 3eme étape : le câblage

On notera que la carte Raspi n'est pas assez puissante pour piloter le servomoteur. On lui ajoute donc une alimentation indépendante avec des piles.



## 4eme étape : Se passer de l'écran et du clavier

a) Configurer Linux pour démarrer sans mot de passe

```
sudo raspi-config
```

Puis naviguer dans les options pour démarrer le mode graphique directement sous le login usuel sans mot de passe.

b) Rendre le script exécutable au démarrage

Il faut créer un fichier dans un endroit bien particulier.

```
sudo nano ~/.config/autostart/python.desktop
```

Dans ce fichier, on écrit :

```
[Desktop Entry]
```

```
Type = Application
```

```
Name = x11vnc
```

```
Exec = sudo python /home/pi/yeti2.py
```

Il n'y a plus qu'à rebooter sans l'écran.

## 5eme étape : la frustration

C'est bien beau tout ça, mais ça ne serait pas mieux si :

- le yéti poussait un cri choisi aléatoirement parmi de nombreux sons préenregistrés ?
- on pouvait piloter le yéti via Bluetooth avec une appli sur téléphone développée avec App Inventor 2 ?
- au lieu d'un servo, on pilotait un arbre à came en 230V via une carte de relais ?
- etc ...