



NOS PROJETS COMPLETS

UN RÉCEPTEUR AIRPLAY AVEC VOTRE RASPBERRY PI



CE QU'IL VOUS FAUT

ShairPort

Où le trouver ? :

<https://github.com/abrasive/shairport>

- Un Raspberry Pi (toutes versions) sous Raspbian
- Une carte micro SD de 8 Go minimum

Difficulté :

Les utilisateurs d'iTunes sous Mac, PC ou iPhone connaissent bien la technologie AirPlay permettant de partager des contenus sur différents appareils. Avec relativement peu de manipulation, votre Raspberry Pi sous Raspbian peut lui aussi devenir un récepteur AirPlay...



AirPlay est un protocole créé par Apple facilitant l'échange de contenus multimédias. Depuis iTunes, un iPhone, un iPod ou un iPad, vous pouvez envoyer des fichiers sur une télévision ou une chaîne Hi-Fi compatible, en un tour de main. Il suffit juste que les différents appareils soient connectés au même réseau WiFi. Lorsqu'il est possible de partager un fichier, l'icône AirPlay est affichée et vous n'avez qu'à cliquer sur le nom du récepteur.

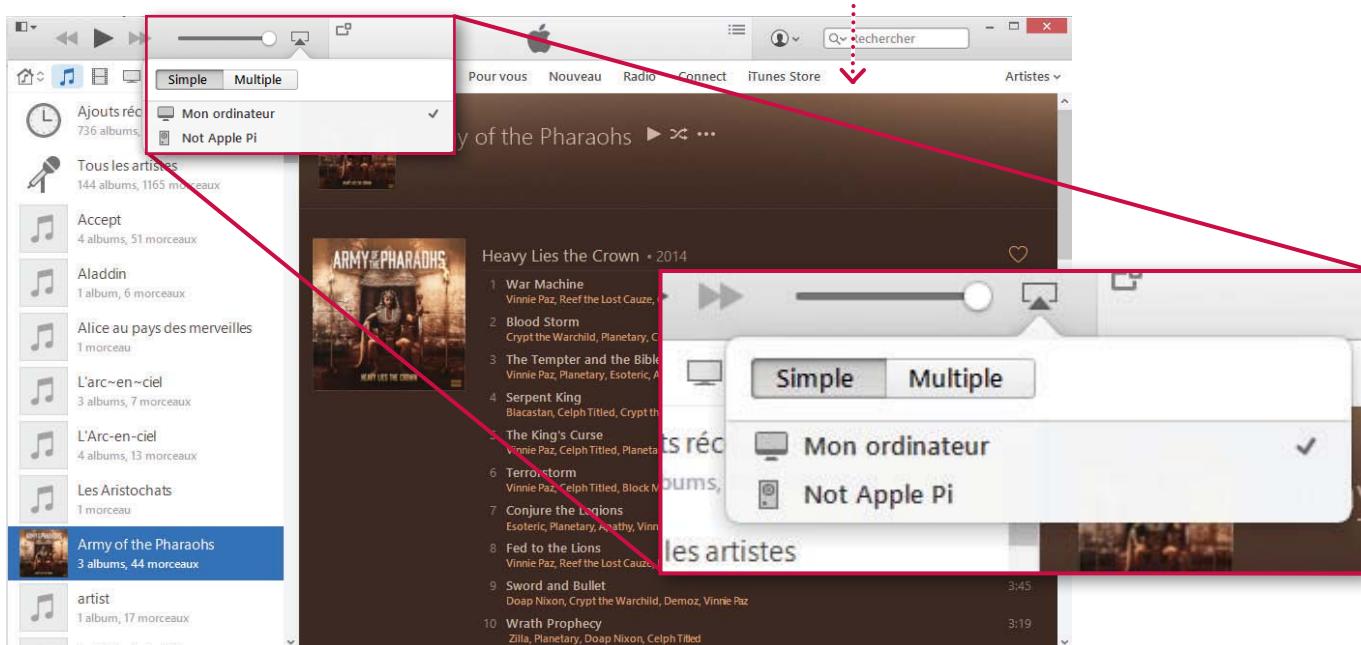
UN SYSTÈME BON MARCHE POUR ÉCOUTER SA MUSIQUE

Avec un Raspberry Pi sous Raspbian, vous pouvez très facilement vous confectionner un récepteur bon marché. La seule limitation c'est qu'il ne fonctionnera qu'avec de la musique. Vous pourrez soit opter pour une version «bureau» avec un Raspbian fonctionnel qui pourra le cas échéant diffuser la musique soit pour une version «sans écran» avec juste

des enceintes et un pilotage via SSH (voir page 24). Pas besoin de payer une fortune pour un appareil certifié par Apple...

PAS BESOIN DE PAYER UNE FORTUNE POUR UN APPAREIL À LA POMME (ENFIN PLUTÔT «À LA NOIX»).

Envoyez vos morceaux sur le Raspberry Pi transformé en jukebox !



L'AUTEUR



RICHARD WATERWORTH

a 16 ans, vit en Angleterre et collabore parfois avec MagPi, le magazine officiel de la fondation Raspberry Pi. Passionné par la photo, Richard a réalisé ce petit projet pour simplement profiter de la musique de son téléphone sur des enceintes «standards».

Lien :
<http://richardtech.net>
https://youtu.be/N_NlhAIMkaQ



ShairPort en action !

PAS À PAS



Mise à jour des paquets et de Raspbian

Comme d'habitude, commençons par mettre à jour la liste des fichiers disponibles dans les dépôts APT présents dans le fichier de configuration et mettre à jour tous les paquets installés sur le système avec les commandes

sudo apt-get update

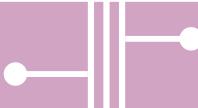
et

sudo apt-get upgrade

Notez que le Raspberry Pi peut être connecté au réseau par câble Ethernet ou par WiFi.

Fichier Édition Onglets Aide

```
pi@raspberrypi:~ $ sudo apt-get update
Réception de : 1 http://mirrordirector.raspbian.org jessie InRelease [15,0 kB]
Réception de : 2 http://archive.raspberrypi.org jessie InRelease [13,2 kB]
Réception de : 3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8 963 kB]
Réception de : 4 http://archive.raspberrypi.org jessie/main armhf Packages [142 kB]
Atteint http://archive.raspberrypi.org jessie/ui armhf Packages
Atteint http://archive.raspberrypi.org jessie/staging armhf Packages
Ign http://archive.raspberrypi.org jessie/main Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/main Translation-fr
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/staging Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/staging Translation-fr
Ign http://archive.raspberrypi.org jessie/staging Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/ui Translation-fr
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Réception de : 5 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37,5 kB]
Réception de : 6 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70,3 kB]
```



Installation des dépendances

Maintenant que votre Raspberry Pi est à jour, nous allons installer des logiciels additionnels qui ne sont pas inclus dans Raspbian. Voici la liste.

- avahi-utils
- build-essential
- chkconfig
- git
- libao-dev
- libavahi-client-dev
- libcrypt-openssl-rsa-perl
- libio-socket-inet6-perl
- libssl-dev
- libwww-perl
- pkg-config

Pour les installer il va falloir saisir cette commande longue comme le bras :

**sudo apt-get install avahi-utils build-essential
chkconfig git libao-dev libavahi-client-dev libcrypt-openssl-rsa-perl libio-socket-inet6-perl libssl-dev libwww-perl pkg-config**

pi@raspberrypi: ~

```
Setting up libfile-listing-perl (6.04-1) ...
Setting up libfont-afm-perl (1.20-1) ...
Setting up liburi-perl (1.64-1) ...
Setting up libhtml-tagset-perl (3.20-2) ...
Setting up libhtml-parser-perl (3.71-1+b4) ...
Setting up libio-html-perl (1.001-1) ...
Setting up liblwp-mediatypes-perl (6.02-1) ...
Setting up libhttp-message-perl (6.06-1) ...
Setting up libhtml-form-perl (6.03-1) ...
Setting up libhtml-tree-perl (5.03-1) ...
Setting up libhtml-format-perl (2.11-1) ...
Setting up libhttp-cookies-perl (6.01-1) ...
Setting up libhttp-daemon-perl (6.01-1) ...
Setting up libhttp-negotiate-perl (6.00-2) ...
Setting up libsocket6-perl (0.25-1+b1) ...
Setting up libio-socket-inet6-perl (2.72-1) ...
Setting up libnet-ssleay-perl (1.65-1+b1) ...
Setting up libio-socket-ssl-perl (2.002-2+deb8u1) ...
Setting up libnet-http-perl (6.07-1) ...
Setting up libwww-robotrules-perl (6.01-1) ...
Setting up libnet-smtp-ssl-perl (1.01-3) ...
Setting up libmailtools-perl (2.13-1) ...
Setting up libssl-dev:armhf (1.0.1k-3+deb8u4) ...
Setting up libssl-doc (1.0.1k-3+deb8u4) ...
Setting up rsync (3.1.1-3) ...
Setting up libwww-perl (6.08-1) ...
Setting up liblwp-protocol-https-perl (6.06-2) ...
Processing triggers for libc-bin (2.19-18+deb8u3) ...
Processing triggers for systemd (215-17+deb8u3) ...
pi@raspberrypi: ~
```



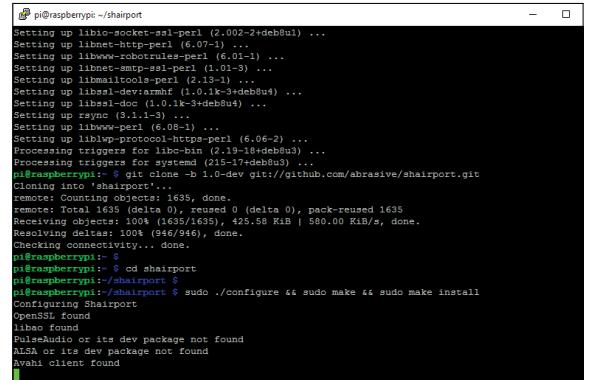
Installation de ShairPort

Nous allons aussi télécharger le programme ShairPort qui va nous permettre de streamer le flux audio sur notre Raspberry Pi :
sudo git clone -b 1.0-dev git://github.com/abrasive/shairport.git

Allons ensuite dans le répertoire de ShairPort pour l'installer :

```
cd shairport
sudo ./configure && sudo make && sudo make install
```

Après cela, ShairPort est prêt à l'action !

```
pi@raspberrypi:~/shairport
Setting up libio-socket-ssl-perl (2.002-2+deb8u1) ...
Setting up libhttp-negotiate-perl (6.07-1+deb8u1) ...
Setting up libwww-mechanize-perl (6.01-1) ...
Setting up libnet-async-ssl-perl (1.01-3) ...
Setting up libmailtools-perl (2.13-3) ...
Setting up libssl-dev-armhf (1.0.1k-3+deb8u4) ...
Setting up libssl-doc (1.0.1k-3+deb8u4) ...
Setting up rsyslog (3.1.1-3) ...
Setting up libio-socket-perl (1.06-1) ...
Setting up libhttp-protocol-https-perl (6.06-2) ...
Processing triggers for libc-bin (2.19-18+deb8u3) ...
Processing triggers for systemd (215-17+deb8u3) ...
pi@raspberrypi:~$ git clone -b 1.0-dev git://github.com/abrasive/shairport.git
Cloning into 'shairport'...
remote: Counting objects: 1635, done.
remote: Total 1635 (delta 0), reused 0 (delta 0), pack-reused 1635
Receiving objects: 100% (1635/1635) 425.58 KiB | 500.00 KiB/s, done.
Resolving deltas: 100% (966/946), done.
Checking connectivity... done.
pi@raspberrypi:~$ pi@raspberrypi:~$ cd shairport
pi@raspberrypi:~/shairport$ pi@raspberrypi:~/shairport$ sudo ./configure && sudo make && sudo make install
Configuring Shairport
OpenSSL found
libao found
PulseAudio or its dev package not found
ALSA or its dev package not found
Avahi client found
```

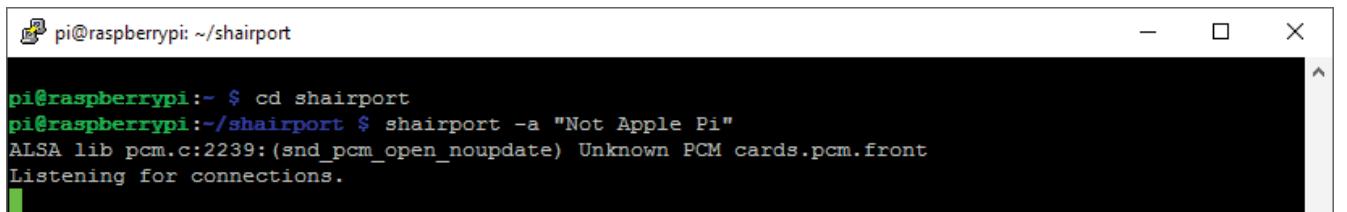


Démarrage de ShairPort

Pour démarrer ShairPort faites :

```
shairport -a 'Not Apple Pi'
```

Cette ligne de commande va démarrer le serveur ShairPort. Nous avons décidé de l'appeler Not Apple Pi mais vous pouvez mettre ce que vous voulez entre les apostrophes. Si tous se passe bien vous devriez voir **Listening for connections**. Cela veut dire que le Raspberry Pi est prêt à recevoir de la musique. Sur le menu AirPlay de votre iPod ou d'iTunes, Not Apple Pi devrait apparaître sur la liste. Si ce n'est pas le cas, mettez à jour l'iOS, la version d'iTunes et redémarrez les appareils qui doivent communiquer entre eux. Si votre Rapsberry Pi est connecté sur le port HDMI d'un téléviseur, utilisez notre astuce de la page 76.



```
pi@raspberrypi: ~$ cd shairport
pi@raspberrypi:~/shairport$ shairport -a "Not Apple Pi"
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.front
Listening for connections.
```

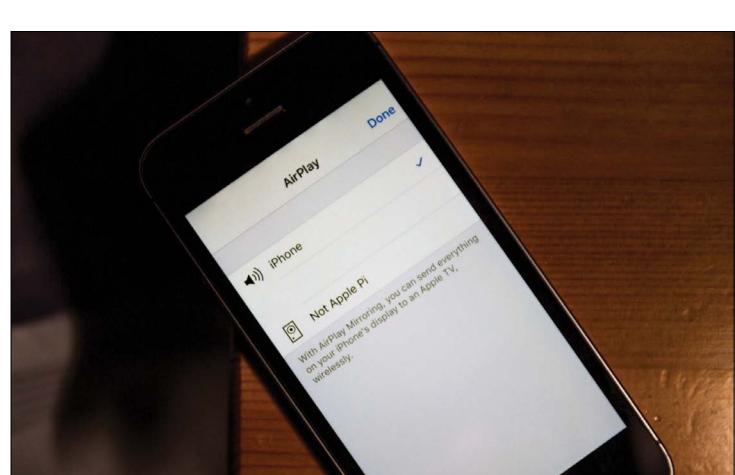


Ajuster le volume

Si vous trouvez que le son est trop bas et que vous n'avez pas moyen de modifier le son d'une autre manière, faites **Ctrl+C** pour arrêter ShairPort et tapez :

```
alsamixer
```

Avec les flèches haut et bas du clavier, vous pouvez ajuster le niveau du volume. Nous recommandons ne ne pas dépasser **80** sans quoi le son sera déformé. Faites **Esc** pour sauvegarder les changements et relancez ShairPort.





NOS PROJETS COMPLETS

VOTRE FILM EN «STOP MOTION»



CE QU'IL VOUS FAUT

Pi-mation.py

Où le trouver ? :

<https://github.com/russb78/pi-mation>

- Raspberry Pi
(toutes versions)

- Raspberry Pi Camera Board (toutes versions)

Difficulté :

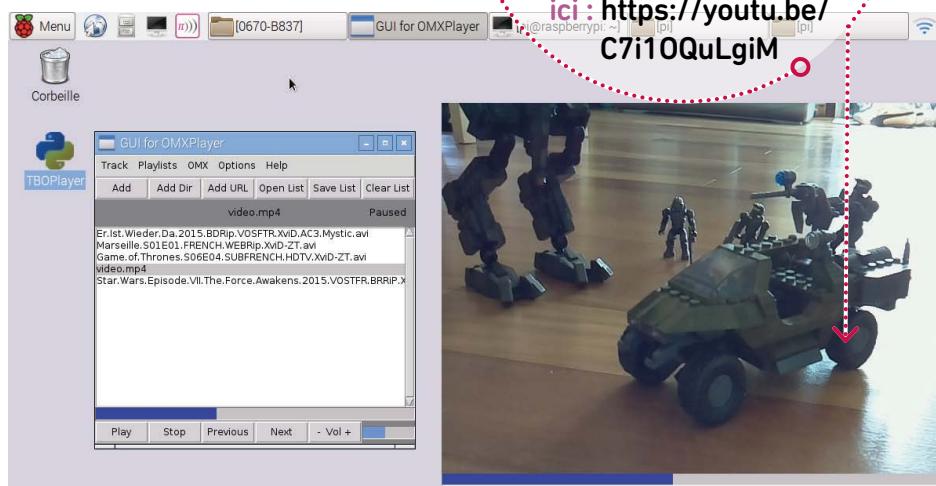
Connaissez-vous la technique cinématographique du «stop motion» ? Il s'agit de faire bouger des éléments (pâte à modeler, jouet, etc.) et de prendre une photo entre chaque manipulation pour donner l'illusion d'une animation. Simple comme de jouer avec des Legos, mais aussi très exigeant si l'on veut bien faire, le stop motion est à la portée de tous avec un Raspberry Pi !



Savez-vous ce qu'ont en commun Chicken Run, le clip de Peter Gabriel Sledgehammer, Wallace & Gromit, Chapi Chapo et l'icône soviétique «Cheburashka» ? Tous ont été réalisés en stop motion (on «animation en volume» dans la langue de Nabilla). Même le dernier Star Wars a eu le droit à sa séquence en stop motion lorsque Finn allume par mégarde le jeu d'échecs dans le Faucon Millenium.

PI-MATION S'OCCUPE DE TOUT

Même si cette technique d'animation a toujours été la plus accessible, elle le devient encore plus lorsque le Raspberry Pi s'en mêle. Avec



Raspbian, le Raspberry Pi Camera Board (Rapicam) et le script Python Pi-mation de Russell Barnes vous pouvez laisser libre cours à votre imagination. Le script s'occupe de tout : enregistrement, prévisualisation, correction des erreurs et encodage. Aidez-vous de vos corps, visage, jouets, peluches et envoyez-nous les liens vers vos créations !



→ LE MODULE CAMERA V2 EST ARRIVÉ !

Ce projet a été réalisé avec le Raspberry Pi Camera Board de première génération. Or juste après notre achat, la version 2 est sortie. Cette dernière intègre un capteur Sony de 8Mpixels (au lieu de 5 pour le précédent) permettant d'afficher 3280x2464 pixels en photo et jusqu'à 1080p en 30 fps en vidéo. À vous de voir si le prix légèrement plus élevé vaut de passer à cette v2.0. Pour info notre vieux modèle dispose de caractéristiques tout à fait honorables : 2592x1944 pixels en mode photo et une résolution de 1080p pour la vidéo avec le même rafraîchissement. Regardez les promotions régulières chez notre partenaire.

Lien : www.kubii.fr

→ PAS DE RASICAM ? ENVIE DE PLUS D'OPTIONS ?

Alors que nous avions commencé l'article sur Pi-mation, nous avons découvert le programme stopmotion. Celui-ci est un peu plus complexe à prendre en main même s'il est en français et qu'il dispose d'une interface graphique. L'installation nécessite un peu plus d'éléments et quelques réglages, mais il dispose aussi de beaucoup plus d'options : masque transparent pour afficher la précédente prise de vue, réglage du nombre d'images par seconde, collage de plusieurs séquences, etc. Sachez aussi que si vous n'avez pas de Rapicam, rien ne vous empêche d'importer vos clichés pris avec un vrai appareil photo. Si ce programme vous intéresse : <http://goo.gl/Dij4Oy>

Offert aux lecteurs de www.framboise314.fr par IDPRESSE

63



NOS PROJETS COMPLETS

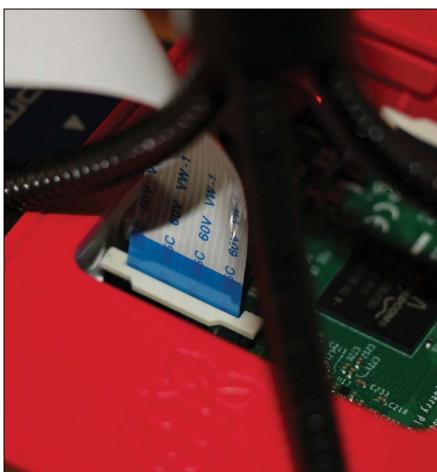
Votre studio de cinéma «stop motion»

PAS À PAS



Branchements

Commençons par brancher le Raspberry Pi équipé de Raspbian (voir page 14) sur votre écran et votre duo souris/clavier. Utilisez la prise Ethernet si vous n'avez pas le module WiFi. Avant de brancher l'alimentation, branchez le Raspicam comme sur la photo : la partie bleue de la nappe doit être en face du verrou en plastique blanc. Regardez la notice pour plus de précision.



Installation des modules

Allumez la machine et allez dans **Menu>Préférences>Raspberry Pi Configuration** puis dans l'onglet **Interfaces**, activez **Camera**. Même si la machine vous le conseille, attendez avant de redémarrer, car nous n'avons pas fini avec les réglages. Ouvrez un terminal puis tapez :

sudo apt-get update && sudo apt-get upgrade pour chargez les dernières versions du système et des paquets. Installez ensuite les composants Python et les paquets qui seront nécessaires (même si normalement certains sont préinstallés sous Jessie) :

sudo apt-get install python-setuptools
sudo apt-get install python-picamera
sudo apt-get install python-pygame
sudo apt-get install libav-tools && sudo apt-get install omxplayer

Redémarrez :

sudo reboot



```
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
Réception de : 1 http://mirrordirector.raspbian.org jessie InRelease [15,0 kB]
Réception de : 2 http://archive.raspberrypi.org jessie InRelease [13,2 kB]
Réception de : 3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8 964 kB]
Réception de : 4 http://archive.raspberrypi.org jessie/main armhf Packages [144 kB]
Réception de : 5 http://archive.raspberrypi.org jessie/ui armhf Packages [8 311 kB]
Ign http://archive.raspberrypi.org jessie/main Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/main Translation-fr
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/ui Translation-fr
```



Téléchargement du script

Installons ensuite le script Pi-mation :

git clone https://github.com/russb78/pi-mation.git

Si vous n'avez pas changé le répertoire par défaut, le dossier comprenant les éléments doit se trouver dans votre dossier perso. Pour lancez le script il faudra aller dans ce dernier.

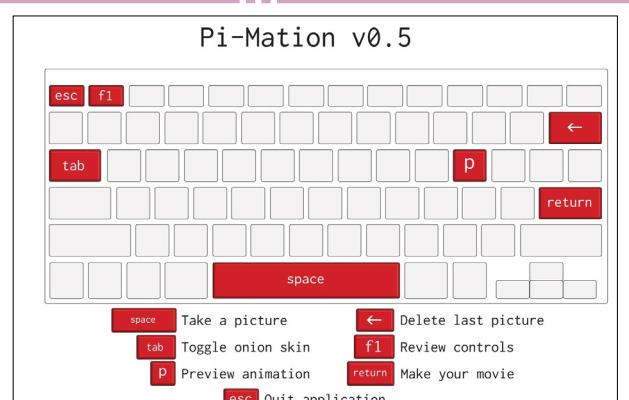
Faites :

cd pi-mation/pi-mation

puis

python pi-mation.py

Vous devriez être accueilli par un écran d'aide vous expliquant les commandes. Faites **F1** pour commencer l'aventure.





Votre studio d'enregistrement

Pour réussir votre film, il va falloir bricoler un petit trépied ou en tout cas bloquer le capteur de la caméra dans une position fixe. Mettez autant de lumière que possible sur la scène et prenez garde aux variations de lumière ! En extérieur, il suffit qu'un nuage passe pour ruiner vos efforts. Votre studio est prêt ! La barre d'espace va prendre un cliché, **F1** permet de revenir aux explications, **Tabulation** va vous montrer la dernière image fixe, la touche **Retour** va effacer la dernière image prise tandis que **P** va lancer un aperçu de votre vidéo depuis le début.



L'encodage

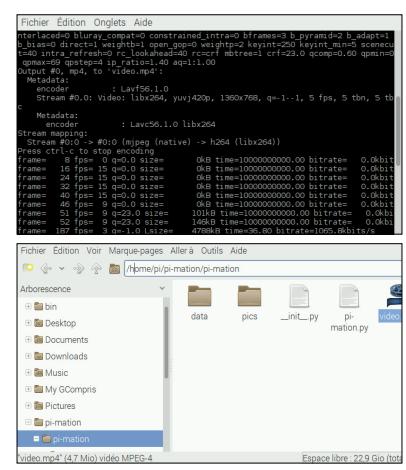
Une fois que vous aurez terminé (nous vous conseillons de commencer par un film d'une cinquantaine de photos pour vous entraîner) appuyez sur **Entrée** pour que le script commence à encoder votre vidéo. Cette dernière prendra place dans le répertoire de Pi-motion. Libre à vous de la lire avec TBOPlayer (article complet dans le prochain numéro !), de la mettre sur Youtube ou de la transférer sur une clé USB (regardez dans **/media/pi/**). Vous pouvez aussi y ajouter une bande-son !



Modifiez le code !

Ouvrez le fichier Pi-motion avec Leafpad (clic droit) et modifiez-le ! En scrutant les lignes de codes (commentées par nos soins), vous verrez que vous pouvez changer plein de choses comme le nombre d'image par seconde. Si les débutants commenceront avec 5 il est possible de s'approcher du fps «cinéma» avec 24 images

secondes. Si le codec x264 ne vous donne pas satisfaction, libre à vous de le changer. Si vous faites une bêtise vous pouvez toujours récupérer la version d'origine du code avec la commande :
git checkout pi-motion.py



```
Fichier Édition Rechercher Options Aide
# global variables
pics_taken = 0
current_alpha, next_alpha = 128, 255

# set your desired fps (~5 for beginners, 10+ for advanced
fps = 5

# Initialise Pygame, start screen and camera
pygame.init()
res = pygame.display.list_modes() # return the best resolution
width, height = res[0] # Having trouble getting the right
print "Reported resolution is:", width, "x", height
start_pic = pygame.image.load(os.path.join('data', 'start.jpg'))
start_fix = pygame.transform.scale(start_pic, (width, height))
screen = pygame.display.set_mode([width, height])
pygame.display.toggle_fullscreen()
pygame.mouse.set_visible = False
play_clock = pygame.time.Clock()


```



Pi-mation : le code complet

Ce code de Russell Barnes est sous GNU/GPLv2, vous pouvez donc le modifier et le redistribuer à condition de respecter la licence. Vous en trouverez une copie ici : <http://goo.gl/PPGG9V> et une traduction non officielle ici : <http://goo.gl/n1s1gq>.

PAS À PAS

```
# pi-mation.py

import pygame, picamera, os, sys

pics_taken = 0
current_alpha, next_alpha = 128, 255

#Changez le nombre d'images par seconde ici
fps = 5

pygame.init()
res = pygame.display.list_modes() # return the best
resolution for your monitor
width, height = res[0]
print "Reported resolution is:", width, "x", height
start_pic = pygame.image.load(os.path.join('data', 'start_
screen.jpg'))
start_pic_fix = pygame.transform.scale(start_pic, (width,
height))
screen = pygame.display.set_mode([width, height])
pygame.display.toggle_fullscreen()
pygame.mouse.set_visible = False
play_clock = pygame.time.Clock()
camera = picamera.PiCamera()
camera.resolution = (width, height)

#Cette boucle concerne l'enregistrement d'une image
def take_pic():
    global pics_taken, prev_pic
    pics_taken += 1
    camera.capture(os.path.join('pics', 'image_' +
str(pics_taken) + '.jpg'), use_video_port = True)
    prev_pic = pygame.image.load(os.path.join('pics',
'image_' + str(pics_taken) + '.jpg'))

#Concerne l'effacement de l'image de prévisualisation
def delete_pic():
    global pics_taken, prev_pic
    if pics_taken >= 1:
        pics_taken -= 1
        prev_pic = pygame.image.load(os.path.join('pics',
'image_' + str(pics_taken) + '.jpg'))

#Concerne la prévisualisation avec la touche P
def animate():
    camera.stop_preview()
    for pic in range(1, pics_taken):
        anim = pygame.image.load(os.path.join('pics',
'image_' + str(pic) + '.jpg'))
        screen.blit(anim, (0, 0))
        play_clock.tick(fps)
        pygame.display.flip()
        play_clock.tick(fps)
    camera.start_preview()

#Concerne le remplacement de l'image de prévisualisation
par la plus récente
def update_display():
    screen.fill((0,0,0))
    if pics_taken > 0:
        screen.blit(prev_pic, (0, 0))
    play_clock.tick(30)
    pygame.display.flip()
```

```
#Concerne l'encodage vidéo
def make_movie():
    camera.stop_preview()
    pygame.quit()
    print "\nQuitting Pi-Mation to transcode your video."
    os.system("avconv -r " + str(fps) + " -i " + str((os.
path.join('pics', 'image_%d.jpg'))) + " -vcodec libx264
video.mp4")
    sys.exit(0)

def change_alpha():
    global current_alpha, next_alpha
    camera.stop_preview()
    current_alpha, next_alpha = next_alpha, current_alpha
    return next_alpha

#Fermeture «propre» de l'application
def quit_app():
    camera.close()
    pygame.quit()
    print "You've taken", pics_taken, " pictures. Don't
forget to back them up!"
    sys.exit(0)

#Affichage de l'écran d'aide
def intro_screen():
    intro = True
    while intro:
        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_ESCAPE:
                    quit_app()
                elif event.key == pygame.K_F1:
                    camera.start_preview()
                    intro = False
        screen.blit(start_pic_fix, (0, 0))
        pygame.display.update()

#Assignation des touches du clavier
def main():
    intro_screen()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_ESCAPE:
                    quit_app()
                elif event.key == pygame.K_SPACE:
                    take_pic()
                elif event.key == pygame.K_BACKSPACE:
                    delete_pic()
                elif event.key == pygame.K_RETURN:
                    make_movie()
                elif event.key == pygame.K_TAB:
                    camera.preview_alpha = change_alpha()
                    camera.start_preview()
                elif event.key == pygame.K_F1:
                    camera.stop_preview()
                    intro_screen()
                elif event.key == pygame.K_p:
                    if pics_taken > 1:
                        animate()
                    update_display()

    if __name__ == '__main__':
        main()
```



SERVO MOTEUR: Y' EN A LÀ- DEDANS !

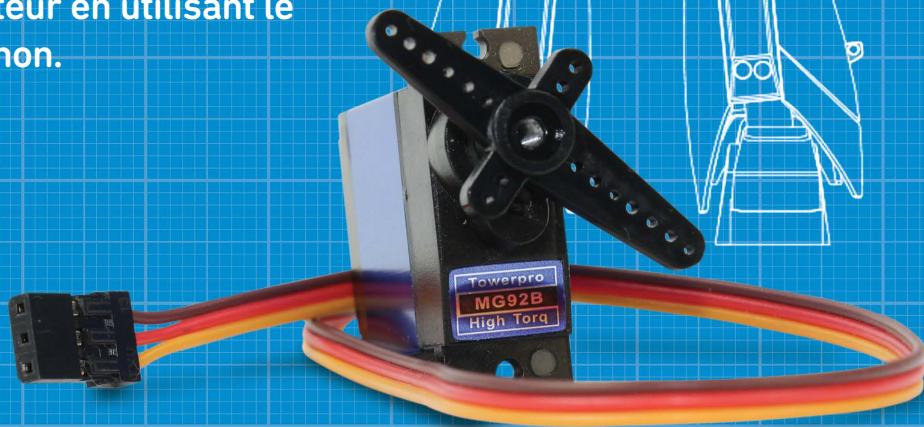
Le servo moteur est un composant indispensable en robotique. Il s'agit d'un moteur électrique à qui on peut envoyer des commandes pour déterminer un angle de rotation. On peut imaginer commander un bras articulé, une roue, un essieu, etc.

Dans cet article nous verrons comment donner des ordres à un servomoteur en utilisant le langage Python.

CE QU'IL VOUS FAUT

- Un Raspberry Pi (toutes versions)
- Un Servomoteur digital ou analogique
- Des fils électriques style «jumper»

Difficulté :





Le Servomoteur

Un servomoteur est composé d'un moteur à courant continu pour actionner l'axe rotatif et d'un circuit de contrôle pour recevoir les ordres de l'ordinateur, ici le Raspberry Pi. Ces ordres donnés sous la forme de signal électrique codé appelé PWM (Pulse Width Modulation ou modulation de largeur d'impulsion). Le signal est répété périodiquement à une fréquence donnée (la plupart du temps 50Hz) et dès qu'il change de largeur, le moteur réagira en conséquence. Par exemple sur un cycle de 20 millisecondes (ms), un signal répété de 0,5 ms commandera au servo un angle de 0°. À 1,5 ms, l'axe restera en position initiale à 90° et à 2,5 ms la rotation sera de 180° (voir notre schéma). Nous verrons comment calculer ces données et les ajuster à votre modèle de servo.

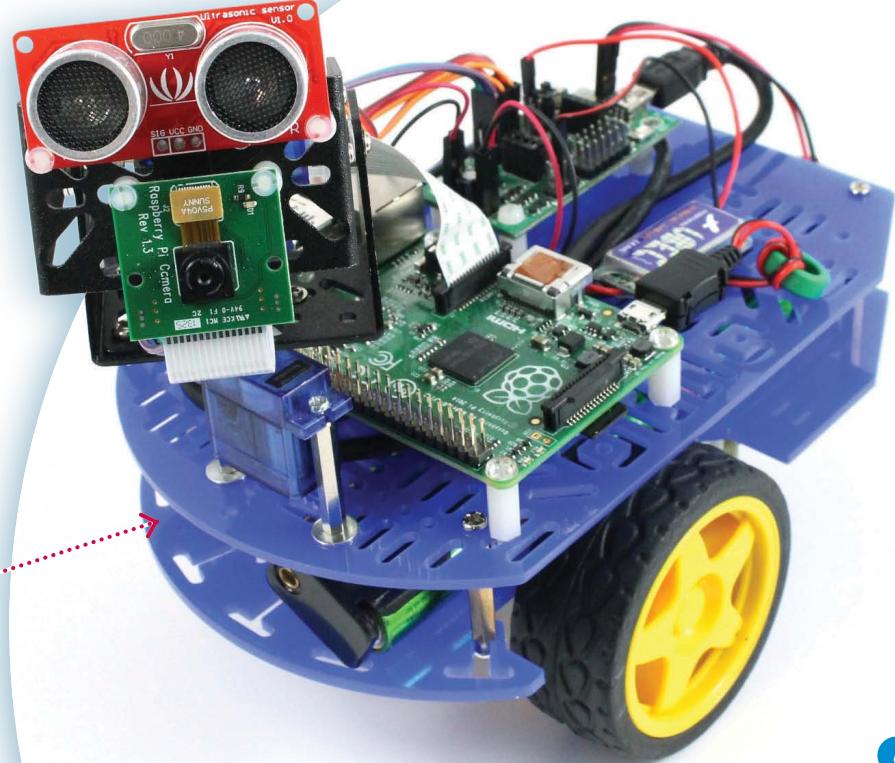
PETIT MAIS COSTAUD !

La bonne nouvelle c'est que, quelle que soit la position de l'axe, le signal qui ordonnera la position initiale enverra le servo à sa position initiale. Il n'y aura pas de «décalage» qui rendrait les choses compliquées. Notons qu'il existe aussi des servos qui permettent des tours complets, mais nous ne

Avant d'imaginer construire et programmer un robot complexe, il faudra un peu de mathématique et beaucoup de programmation...

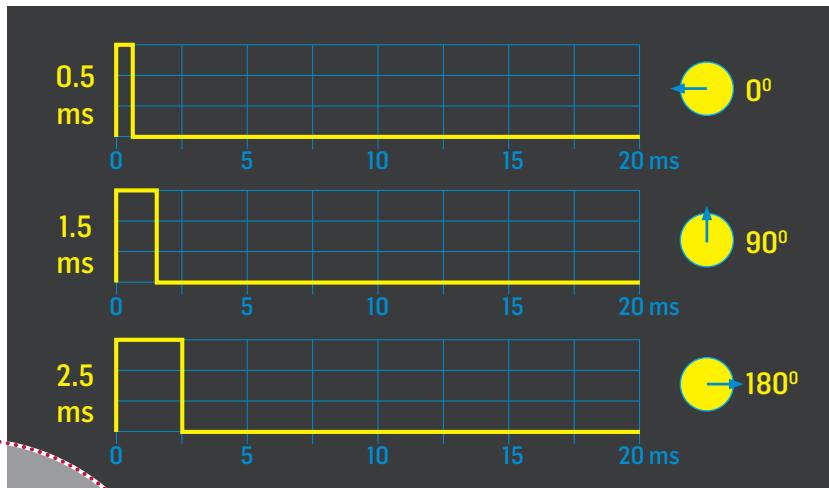
LE RASPBERRY PI N'EST PAS VRAIMENT FAIT POUR CONTRÔLER SEUL UN OU PLUSIEURS SERVOMOTEURS MAIS...

traiterons pas ce cas particulier ici. Pour nos démonstrations, nous avons opté pour un modèle qui permet une rotation de 180° avec un couple de 3,1kg/cm à 5V. Oui vous avez bien lu : pour un poids de 14g et des dimensions très réduites (22.8x12x31 mm), ces petits composants peuvent fournir des quantités très élevées de couples («torque» dans la langue de Donald Trump). Cerise sur le gâteau, vous trouverez ce type de servo sur le marché pour moins de 10€.



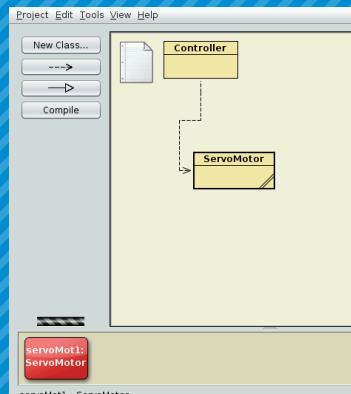


Mais comment le Raspberry Pi peut s'y prendre pour « parler » avec notre servomoteur ? Nous allons simplement utiliser les ports GPIO et le langage Python. Les servos ont 3 fils : un rouge pour le voltage, un noir pour la masse et un jaune pour recevoir les commandes. Il suffira de les brancher sur le Raspberry Pi. Notez qu'il s'agit ici de voir comment tout cela fonctionne, car pour commander plusieurs servos avec la perspective de faire un robot, il faudra un contrôleur dédié (voir encadré). Côté Python, nous lancerons des commandes dans la console pour voir le fonctionnement puis nous écrirons notre premier programme.



→ BLUEJ, JAVA BIEN ?

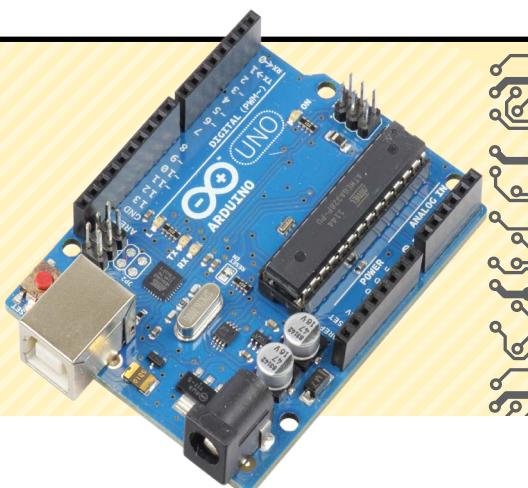
Si vous êtes plus à l'aise avec le Java, vous pouvez opter pour ce langage plutôt que Python. Cela tombe bien puisque BlueJ est intégré à Raspbian Jessie. Cet environnement est plus « graphique » que Python, mais nous avons privilégié ce dernier puisque les bouts de code que vous pourrez « adapter » pour vos projets sont plus nombreux.



Lien : www.bluej.org/raspberrypi

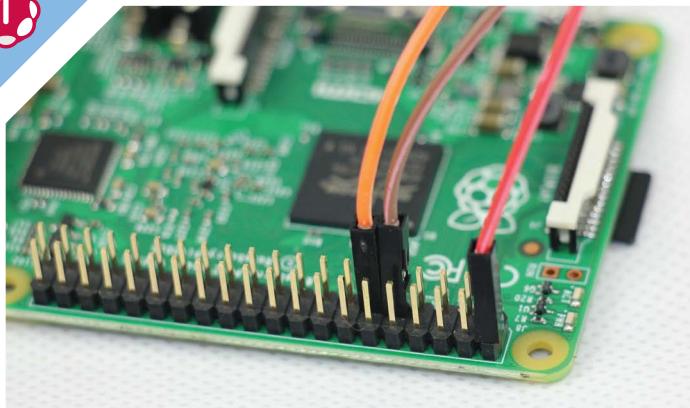
→ ET ARDUINO DANS TOUT ÇA ?

En matière de robot, il est aussi possible d'utiliser les cartes programmables Arduino. Ces dernières sont bien plus à l'aise dans le contrôle des servos que le Raspberry puisque ce dernier ne dispose pas de contrôleur intégré. Par contre notre framboise dispose de plusieurs atouts dans sa manche : WiFi, Bluetooth et bien sûr une puissance de calcul qui manque chez Arduino. Mais pourquoi les mettre dos à dos ? Le Raspberry peut très bien travailler de concert avec une carte Arduino. Ce dernier faisant office de contrôleur PWM et la framboise s'occupant des calculs et de la communication.





Commander un servomoteur avec son Raspberry Pi



Les branchements

Commençons par brancher les fils du servomoteur sur le Raspberry Pi. Dans notre montage, en prenant en compte les spécifications de notre modèle de moteur nous n'avons pas besoin d'une alimentation supplémentaire pour le servo. Branchez directement un servo sur les ports GPIO peut les endommager. Il convient donc de vérifier que votre modèle ne soit pas trop gourmand. Au-dessus de 5V, il faudra une résistance et un schéma de ce type (<http://goo.gl/YmRmzd>). Vous pouvez bricoler une solution sans plaque de prototypage (breadboard). Si votre Raspberry redémarre ou plante, choisissez la sécurité. Nous allons mettre le fil rouge sur le port GPIO numéro 2, le noir sur le 9 et le jaune sur 11 (le port physique 1 étant le plus près du coin de la carte vers l'intérieur). En fonction de la connectique de votre servo, il faudra peut-être des petits fils électriques de type «jumper» (ceux des cours de techno).



Python avec Raspbian

Continuons en branchant notre Raspberry sous Raspbian comme nous en avons maintenant l'habitude : avec clavier/souris/écran ou via SSH (voir page 24). Depuis un terminal, lancez :

sudo python

Nous sommes maintenant dans la console Python permettant de donner des ordres directement sans passer par un script. Cela va nous être utile pour mieux étonner notre servo (tous n'ont pas les mêmes spécifications) et mieux digérer le futur code. Mais avant il va falloir comprendre comment envoyer les impulsions et comprendre la notion de «rapport cyclique»...

```
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```





ServoDatabase.com

Find a servo: Google Custom Search | Search | Advanced Search

Servo Database > TowerPro Servos > SG90

TowerPro SG90 Servo

Basic Information

Modulation:	Analog
Torque:	4.8V: 25.0 oz-in (1.80 kg-cm)
Speed:	4.8V: 0.12 sec/60°
Weight:	0.32 oz (9.0 g)
Dimensions:	Length: 0.91 in (23.0 mm) Width: 0.48 in (12.2 mm) Height: 1.14 in (29.0 mm)
Motor Type:	3-pole
Gear Type:	Plastic
Rotation/Support:	Bushing

Un peu de mathématiques...

La plupart des servos fonctionnent à 50Hz et comme vous vous en souvenez peut-être de vos années collèges, une fréquence de 50Hz signifie que 50 impulsions électriques sont envoyées par seconde. Pour connaître le délai du cycle, il faut faire 1/50 ce qui donne 0,02s ou 20 milliseconde (ms). Nous avons vu qu'il fallait que le Raspberry envoie un signal précis, par exemple 0,5 ms toutes les 20 ms pour aller complètement à gauche. Or le Raspberry Pi ne peut pas envoyer ce signal tel quel, il faut le traduire en rapport cyclique (duty cycle en anglais ou DC). Pour cela on doit faire DC = PWM/20ms. Dans le cas d'une fréquence à 50Hz, 2,5% de DC correspond donc à 0,5 ms (puisque 5% de DC correspond à 1 ms). De même 1,5ms correspondant à 7,5 % de DC (on peut aussi faire $0,075 \times 20 = 1,5$). Bien sûr, un DC intermédiaire de 6 % par exemple enverra un PWM de 1,2 ms ce qui bougera l'axe du servo à 72° (il suffit de faire une règle de trois, si 1,5 ms correspond à 90°). Bien sûr ces chiffres sont donnés à titre indicatif, les fréquences peuvent changer tout comme l'amplitude de l'angle ou le nombre de ms nécessaire à mouvoir le servo à droite comme à gauche. Souvent ce sera même 1ms, 1,5ms et 2ms. C'est pour cela qu'il faudra commencer par «jauger» votre servo et déterminer les DC optimums pour mieux l'utiliser ensuite...



Jauger son matériel

Comme pour un programme, les lignes commençant par # ne sont pas à taper, elles servent juste à commenter les commandes. Tapez :

#Pour importer la librairie permettant de contrôler les ports GPIO
>>>import RPi.GPIO as GPIO

#Pour permettre de donner un chiffre à chaque port en fonction de leur position physique
>>>GPIO.setmode(GPIO.BARD)

#Il faut ensuite déclarer que le port 11 sera le port de sortie (celui qui commande les PWM)

>>>GPIO.setup(11, GPIO.OUT)

#Il faudra ensuite donner le même port avec la fréquence utilisée
>>>pwm=GPIO.PWM(11,50)

>>>pwm.start(5)

Après avoir entré cette dernière commande et tapé Entrée, le servo devrait aller complètement à gauche. Ce ne sera sûrement pas le cas, car les spécifications des servos ne sont pas les mêmes. Il va falloir bidouiller le DC avec par exemple :

>>>pwm.ChangeDutyCycle(2)

Allez-y à taton pour trouver les valeurs qui vont permettre de faire aller le servo complètement à gauche et complètement à droite. Par exemple :

**>>>pwm.ChangeDutyCycle(7)
>>>pwm.ChangeDutyCycle(10)
etc.**

Avec ces deux données (le plus à gauche et le plus à droite), il ne sera pas difficile de déterminer le point neutre (correspondant à 90°) et tout le reste des angles que vous pourriez souhaiter. Attention, certains servos ont des angles plus réduits (de 0 à 160°) et d'autres, numériques, fonctionnent à des fréquences plus exotiques. Par exemple notre Towerpro MG92B a parfois donné de meilleurs résultats à 300Hz qu'à 50... Si vous désirez en savoir plus sur votre servo ou en prendre un qui correspond le mieux à vos attentes, vous trouverez pas mal de données techniques et de comparatifs ici : www.servodatabase.com

À la fin, tapez les commandes suivantes pour arrêter le signal PWM et réinitialiser l'utilisation des ports GPIO :

**>>>pwm.stop()
>>>GPIO.cleanup()**

```
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import RPi.GPIO as GPIO
>>> GPIO.setmode(GPIO.BARD)
>>> GPIO.setup(11, GPIO.OUT)
>>> pwm=GPIO.PWM(11,50)
>>> pwm.start(2.5)
>>> pwm.start(2.5)
>>> pwm.start(5)
>>> pwm.start(2.5)
>>> pwm.start(2.5)
>>> pwm.ChangeDutyCycle(7)
>>> pwm.ChangeDutyCycle(10)
>>> pwm.ChangeDutyCycle(12)
>>> pwm.ChangeDutyCycle(9)
>>> pwm.ChangeDutyCycle(4)
```



Le Servomoteur



Notre premier script !

Nous pouvons maintenant faire notre premier programme test.py !

#Les lignes suivantes sont le copier-coller des commandes que nous avons tapées dans la console juste avant

```
import RPi.GPIO as GPIO
import time
```

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
```

#Ici c'est la même chose sauf que nous avons choisi la fréquence de 300Hz, car notre servo est digital et pas analogique. Le 22.5 correspond à notre position de départ avec cette fréquence.

```
p = GPIO.PWM(11,50)
p.start(11.5)
```

#Cette boucle va simplement changer le servo de position toute les secondes entre 90° (neutre), 180° (droite) et 0° (gauche)

```
try:
    while True:
        p.ChangeDutyCycle(11.5)
        time.sleep(1)
        p.ChangeDutyCycle(18)
        time.sleep(1)
        p.ChangeDutyCycle(5)
        time.sleep(1)
```

#Cette ligne permet de tout réinitialiser en faisant Ctrl+C

```
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

Lancez ce programme depuis son répertoire d'origine en faisant :

sudo python test.py

Normalement votre servo devrait réagir et faire les mouvements que nous lui avons demandé de faire. Si ce n'est pas le cas, il faudra refaire les calculs ou changer la fréquence d'utilisation. Attention, car comme nous vous l'avons déjà notifié, sans contrôleur externe, la précision ne sera pas de mise. Pour ouvrir ou fermer un volet d'avion, ce sera suffisant, mais pas si vous désirez faire un robot qui casse les œufs ou qui écrit à la plume...

```
Fichier Édition Rechercher Options Aide
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)

p = GPIO.PWM(11,50)
p.start(11.5)

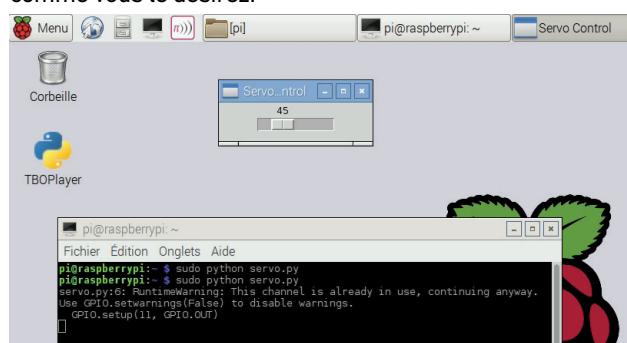
try:
    while True:
        p.ChangeDutyCycle(11.5)
        time.sleep(1)
        p.ChangeDutyCycle(18)
        time.sleep(1)
        p.ChangeDutyCycle(5)
        time.sleep(1)

except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```



Votre servo fait n'importe quoi ?

Si votre servo réagi bizarrement (mauvais angles, comportement erratique, etc.), c'est peut-être parce qu'il ne supporte pas trop le signal envoyé par le Raspberry Pi sans contrôleur. Pour en avoir le cœur net, vous pouvez essayer ce script : <http://tinyurl.com/zok2lzs>. Une fois lancé, ce dernier permet de positionner l'axe du servo avec un repère graphique que vous pourrez déplacer à la souris (cela ne fonctionne pas en SSH). À vous de voir le comportement du servo et d'ajuster les données qui sont à l'intérieur du script pour faire fonctionner votre moteur comme vous le désirez.



→ À SUIVRE...

Que voulez-vous voir aborder dans cette rubrique pour le prochain numéro ? Le fonctionnement d'un nouveau composant robotique (capteur de lumière, de température, d'obstacle, etc.) ou bien une approche plus poussée du servomoteur : avec contrôleur, servos multiples ? Dites-nous ce que vous voulez ici : raspberry@idpresse.com !



LE COIN DES ASTUCES

A l'avenir cette rubrique contiendra des astuces ou des réponses aux problèmes rencontrés par nos lecteurs. Pour ce premier numéro, nous avons dû trouver nous même des sujets à aborder, mais n'hésitez pas à poser vos questions, demander de l'aide ou même proposer vos propres astuces à raspberry@idpresse.com...

#01 MON RASPBERRY PI NE DÉMARRE PLUS DEPUIS QUE JE L'AI OVERCLOCKÉ... JE VOUDRAIS REVENIR À LA NORMALE !

Vous avez un peu forcé la dose avec l'overclock et votre Raspberry Pi refuse de démarrer correctement? Il est possible de revenir aux paramètres de base en restant appuyé sur **Shift** (la touche pour les majuscules) lorsque vous démarrez la machine. Le Raspberry Pi va alors booter normalement. Si vous n'avez pas de clavier et que vous utilisez le protocole SSH, il suffit de mettre un fil conducteur entre les plots 5 et 6 du port GPIO lors du démarrage... Cette méthode va aussi réinitialiser le choix du boot.



#02 JE N'AI PAS ASSEZ DE PLACE SUR MA CARTE SD POUR MES NOUVEAUX PROJETS/LOGICIELS ET JE N'AI PAS NON PLUS LES MOYENS POUR EN ACHETER UNE NOUVELLE...

Il est conseillé d'avoir une carte d'au moins 8 Go pour profiter pleinement de Raspbian: installer des logiciels, nouveaux paquets ou transférer des fichiers. Seulement ces 8 Go sont bien vite remplis au fur et à mesure que vous découvrirez le système. Nous avons vu que la dernière version de Raspbian intègre le jeu Minecraft, le langage Wolfram, le logiciel musical Sonic Pi et la suite LibreOffice. Si vous ne les utilisez pas, vous pourrez gagner 1 Go en les effaçant! Faites:



RASPBIAN JESSIE LITE
Minimal image based on Debian Jessie

Version:	May 2016
Release date:	2016-05-10
Kernel version:	4.4
Release notes:	Link

[Download Torrent](#) | [Download ZIP](#)

SHA-1: 333bfc855e8944ecb1142337ead8c928dc6c9d95

sudo apt-get autoremove --purge wolfram-engine minecraft-pi sonic-pi libreoffice*

Puis:

sudo apt-get clean

Si vous avez installé des paquets en remplacement d'autres plus ancien, vous pouvez aussi faire le ménage et gagner quelque Mo avec cette ligne:

sudo apt-get autoremove

Notez aussi que vous trouverez des versions de Raspbian «allégées» en suivant nos liens ci-dessous...

Lien: www.raspberrypi.org/downloads/raspbian

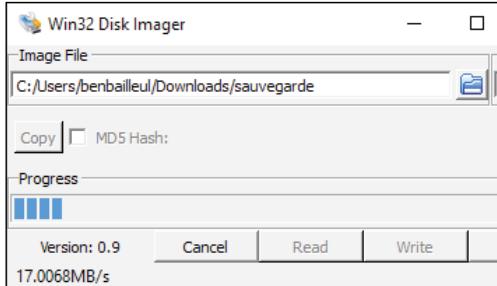
Lien: <https://minibianpi.wordpress.com>

```
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo apt-get clean
pi@raspberrypi:~ $ sudo apt-get autoremove
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 2 non mis à jour.
pi@raspberrypi:~ $
```



#03 J'AI PEUR DE FAIRE UNE BÉTISE AVEC RASPBIAN ET DE CORROMPRE MON SYSTÈME. COMMENT PUIS-JE SAUVEGARDER ?

Si vous avez les droits super utilisateur sur votre Raspberry Pi (l'équivalent des droits administrateurs sous Windows) vous n'êtes pas à l'abri d'une erreur de manipulation. Pour éviter de «casser» Raspbian et d'avoir à réinstaller tous vos logiciels et paquets, vous pouvez sauvegarder l'intégralité de votre carte SD. Il suffit de faire exactement l'opposé de ce que vous avez fait à l'installation: extraire l'image système de votre carte SD pour la placer dans votre PC. Sous Linux,



placez la carte SD dans le lecteur de votre PC et tapez:

dd if=/dev/sdx of=/path/sauvegarde.bin bs=1M

où le **x** de **sdx** est le nom de votre carte et où **/path/sauvegarde.bin** est le chemin du fichier de sauvegarde. Pour la restauration, mettez **of** à la place de **if**.

Sous Windows, suivez le lien ci dessous et utilisez Win32 Disk Imager. Renseignez le chemin et le nom du fichier de sauvegarde dans **Image File** et sélectionnez le lecteur qui contient la carte SD. Faites **Read** et attendez la fin du processus.

Lien: <https://sourceforge.net/projects/win32diskimager>

#04 LE SSH C'EST BIEN JOLI, MAIS COMMENT JE FAIS POUR ACCÉDER À L'INTERFACE GRAPHIQUE DE RASPBIAN DEPUIS UN PC SOUS WINDOWS ?

Dans notre article de la page XX, nous avons vu comment accéder au Raspberry Pi depuis Windows ou Linux. Idéal pour de nombreuses utilisations, mais si vous êtes allergique aux lignes de commande et que vous n'avez pas d'écran supplémentaire, vous pouvez très bien accéder au GUI (interface graphique) avec un VNC. Commencez par vous connecter avec PuTTY comme nous l'avons vu dans notre article et tapez:

sudo apt-get install tightvncserver

Puis lancez le serveur avec la commande

tightvncserver

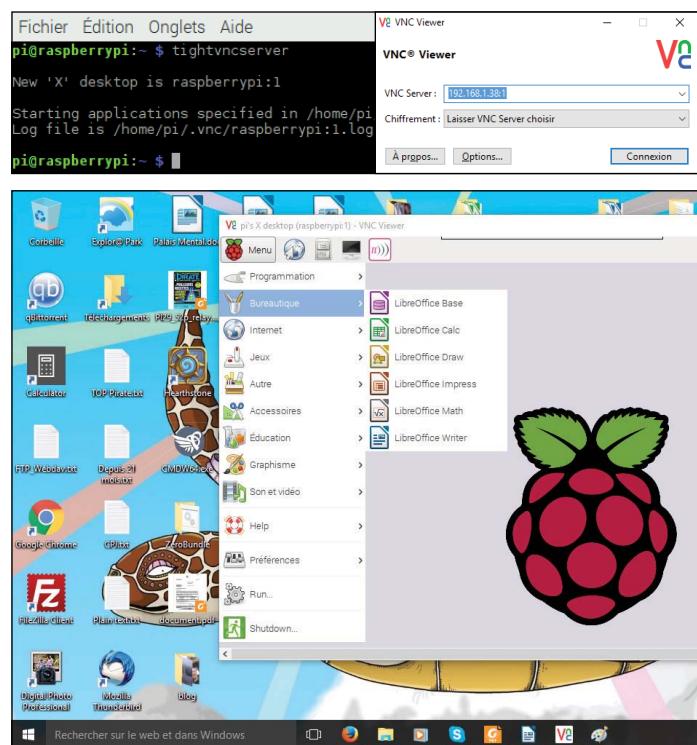
Paramétrez le mot de passe et dites non (**n**) lorsqu'on vous demandera **Would you like to enter a view-only password (y/n)?**

Notez que vous devrez vous souvenir de l'IP locale de votre Raspberry. Pour la partie cliente, utilisez VNC Viewer sur Windows, Mac ou Linux. Dans VNC Server mettez l'IP de votre Raspberry Pi avec le port 1 à la fin. Par exemple:

192.168.1.38:1

Entrez votre mot de passe et passez les avertissements de sécurité. Bravo, le bureau de Raspbian se retrouve sur votre PC de bureau!

Lien: www.realvnc.com/download/viewer



#05 J'AI BRANCHÉ MON RASPBERRY PI SUR UNE TÉLÉVISION, MAIS JE N'AI PAS DE SON !

Si vous branchez votre Raspberry Pi sur une télévision avec un câble HDMI, vous imaginez sans doute avoir le son sur les enceintes intégrées. Ce n'est malheureusement pas automatique, il faudra «expliquer» à la machine que vous voulez que le flux sonore passe par ce câble et pas par le jack RCA. Ouvrez le fichier **config.txt** contenu dans le dossier **/boot/** avec Leafpad par exemple (en utilisant la commande **sudo nano /boot/config.txt**) et retirez le **#** en face de **hdmi_drive=2**. Cela aura pour effet de dé-commenter la ligne et de l'activer. Sauvegardez, rebootez et vous devriez avoir le son!

```
Fichier Édition Rechercher Options Aide
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force
#hdmi_group=1
#hdmi_mode=1

# uncomment to force a HDMI mode rather than DVI. This can
# DMT (computer monitor) modes
hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interfe
# no display
#config_hdmi_boost=4

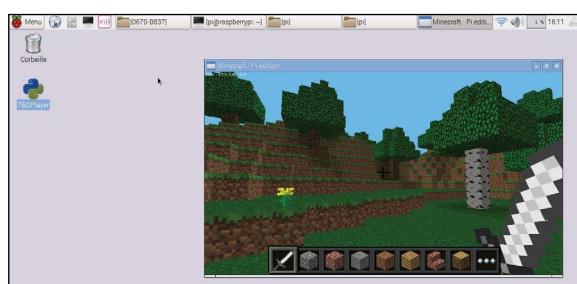
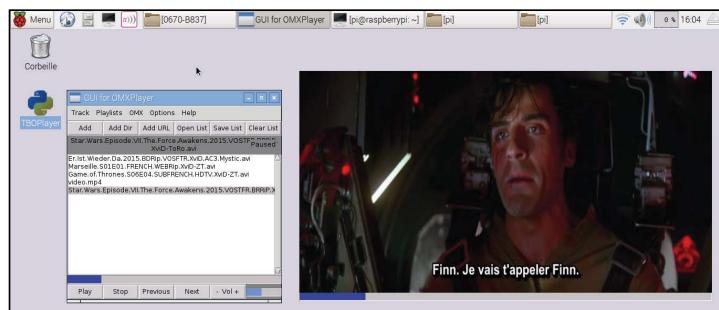
# uncomment for composite PAL
#sdtv_mode=2
```

#06 LORSQUE JE TENTE DE PRENDRE UNE CAPTURE D'ÉCRAN AVEC L'UTILITAIRE «SCROT», LES VIDÉOS OU LA FENÊTRE DE MINECRAFT APPARAÎSSENT EN NOIR. COMMENT FAIRE UNE CAPTURE DIGNE DE CE NOM ?

Lorsque nous avons commencé le magazine, nous avons rencontré ce problème. La commande **scrot** permettant de prendre des captures d'écran sous Linux fonctionnait mal avec les fenêtres affichant de la vidéo ou des jeux. Une vilaine fenêtre noire ou blanche apparaissait en lieu et place de ce que nous voulions capturer. Après moult recherches nous avons découvert le logiciel raspi2png qui fait le travail mieux que les autres. Lancez un terminal et tapez:

```
git clone https://github.com/
AndrewFromMelbourne/raspi2png/
```

Pour le téléchargement, il faudra se créer un compte sur GitHub et donner vos identifiants lorsque le terminal vous le demandera. Après le téléchargement, il faudra déplacer le logiciel pour que le système puisse l'utiliser:



```
sudo cp -a raspi2png/raspi2png /usr/local/bin
```

Avec cette modification, lorsque vous taperez raspi2png, le fichier PNG prendra place dans le dossier du logiciel. Attention, car en prenant une deuxième capture sans préciser son nom, la première sera écrasée. Pour prendre une capture avec un délai de 5 secondes, tapez:

```
raspi2png -d 5 -p screenshot1.png
```

Lien: <https://github.com/AndrewFromMelbourne/raspi2png>



JE SUIS UN PEU PERDU AVEC LINUX, VOUS N'AURIEZ PAS UNE LISTE DES COMMANDES OU DE RACCOURCIS LES PLUS UTILISÉS ?

Intimidé par le fameux prompt **pi@raspberrypi:~ \$** qui vous accueille lorsque vous ouvrez un terminal dans le GUI ou lorsque vous démarrez votre Raspberry Pi en ligne de commande (en SSH ou pas)? C'est vrai que cela peut paraître très obscur aux utilisateurs qui n'ont connu que le duo fenêtres/souris. Voici de quoi vous débrouiller... Notez que ces commandes sont toutes valides sous d'autres distributions Linux!

startx:

Va afficher le bureau graphique (GUI) depuis le CLI (l'interface en ligne de commande). Normalement Raspbian Jessie «boot» sur le GUI, mais on ne sait jamais...

df:

Permet de connaître l'espace disponible sur la carte SD (plusieurs partitions) et les éventuels disques.

```
Fichier Edition Onglets Aide
pi@raspberrypi:~ $ df
Système de fichiers blocs de 1K Utilisé Disponible Utile Monté sur
/dev/root      751380  4335452  2815478  61% /
/dev/pts        646708     0    469708  0% /dev
tmpfs          474028     0    474028  0% /dev/shm
tmpfs          474028   6452    467576  2% /run
tmpfs          5120      4    5116  1% /run/lock
tmpfs          474028     0    474028  0% /sys/fs/cgroup
/dev/mmcblk0p1  613288  20328    41656  20% /boot
tmpfs          94808     0    94808  0% /run/user/1000
/dev/sda1      30266592 5172992  25093600  18% /media/pi/0670-B837
pi@raspberrypi:~ $
```

sudo adduser bibiche:

L'admin crée un compte utilisateur pour **bibiche**.

su bibiche:

Permet de changer d'utilisateur courant pour **bibiche**.

ifconfig :

a afficher des informations concernant le réseau. Pratique pour trouver son IP locale (ligne **inet adr** dans **wlan0** ou **eth0** selon votre connexion à la box: WiFi ou Ethernet)

```
Fichier Edition Onglets Aide
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Boucle locale
          inet adr:127.0.0.1 Masque:255.0.0.0
          adr inet6::1/128 Scope:Hôte
          brd 0.0.0.0
          Mask:0.0.0.0
          MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:(0.0 B) TX bytes:(0.0 B)
lo       Link encap:Boucle locale
          inet adr:127.0.0.1 Masque:255.0.0.0
          adr inet6::1/128 Scope:Hôte
          brd 0.0.0.0
          Mask:0.0.0.0
          MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:(0.0 B) TX bytes:(0.0 B)
pi@raspberrypi:~ $
```

pwd:

Permet de savoir quel est le dossier actuellement utilisé. Comme avec les lignes de commandes sous Windows, vous devez lancer un programme depuis son emplacement. Si cet emplacement n'est pas le bon, faites **cd bon_dossier**

ls:

La première lettre est un L minuscule. Dresse la liste des fichiers présents dans le dossier actuellement utilisé.

top:

C'est un peu le «gestionnaire des tâches» de Raspbian. Cela liste les processus en cours.

kill xxx:

Va tuer le processus **xxx**. Pratique en cas de bug ou si vous ne savez pas comment arrêter un programme (**Ctrl+X** le plus souvent)

man yyy:

Affichera une explication détaillée de l'utilisation de la commande **yyy**. Celle-là vaut de l'or si vous comprenez un peu l'anglais.

```
Fichier Edition Onglets Aide
SUDO(8)           BSD System Manager's Manual           SUDO(8)
NAME
  sudo, sudoedit – execute a command as another user
SYNOPSIS
  sudo [-h | -k] [-Akn] [-a type] [-g group] [-h host] [-p prompt] [-u user]
  sudo [-l [-Akn] [-a type] [-g group] [-h host] [-p prompt] [-u user]
  sudo [-AHEPS] [-a type] [-c num] [-c class] [-g group] [-h host]
  [-p prompt] [-r role] [-t type] [-u user] [VAR=value] [-i | -s]
  sudoedit [-Akn] [-a type] [-C num] [-c class] [-g group] [-h host]
  [-p prompt] [-u user] file ...
DESCRIPTION
  sudo allows a permitted user to execute a command as the superuser or
  another user, as specified by the security policy.
  sudo supports a plugin architecture for security policies and input/output
  logging. Third parties can develop and distribute their own policy
  and I/O logging plugins to work seamlessly with this sudo front end. The
  plugin code must return a process handle when prompted.
```

vcgencmd measure_temp:

Si vous êtes porté sur l'overclocking, cette commande est pratique pour connaître la température du chipset. Tapez **vcgencmd commands** pour voir tout ce que vous pouvez monitorer (voltage, RAM, HDMI, etc.)

Enfin, il n'y a rien de plus pénible que de taper une commande longue comme le bras et de faire une faute de frappe. Obligé de tout retaper? Bien sûr que non! Essayez de taper sur la **flèche haut** du clavier. Magique! La dernière commande, même erronée apparaît! Tapez autant de fois que vous le voulez. Bien sûr vous pouvez utiliser cette astuce pour refaire une même action en ne changeant qu'un caractère par exemple:

raspi2png -d 5 -p screenshot1.png

raspi2png -d 5 -p screenshot2.png

raspi2png -d 5 -p screenshot3.png

raspi2png -d 5 -p screenshot4.png

etc.

Nous n'avons bien sûr pas pu penser à tout. Si vous voulez ajouter votre astuce, ligne de commande ou raccourci, n'hésitez pas à nous l'envoyer!