treescam

raspberry avec une caméra pour faire du stopmotion

matériel

- une vieille carte raspberry 1B révision 000f (cat /proc/cpuinfo |grep Revision)
- carte sd 32Go
- raspberry caméra module V2 8M Pixels
- nappe de connexion camera

installation

Téléchargement du programme rpi-imager sur https://www.raspberrypi.org qui évite la création manuelle de la carte sd. C'est un paquet debian à installer par

```
dpkg -i imager_1.4_amd64.deb
```

Il manque des dépendances à corriger puis refaire l'installation

```
apt --fix-broken install
```

Télécharger l'image de la carte OS-lite 32bits sur la carte SD

Dans la partition boot de la carte sd ajouter un fichier ssh pour pouvoir se connecter en headless

```
touch ssh
```

Exécuter

```
sudo raspi-config
```

Changer le mot de passe de pi, changer le nom en treescam dans la configuration réseau, changer la localisation: locales en fr_FR.UTF-8 UTF-8, langue en fr, fuseau sur Paris, clavier en azerty. Dans les options d'interface activer la caméra. Dans les options avancées étendre la partition système puis reboot.

Efectuer une mise à jour

```
sudo apt update & sudo apt upgrade
```

Date

La date est à initialiser manuellement quand il n'y a pas de connexion internet

```
date -s '2019-10-17 12:00:00'
```

hostapd

Ajouter une clé wifi et installer hostapd pour pouvoir accéder à la machine pour récupérer les images sans la raccorder à un réseau local, notamment en raison de l'insécurité de mjpg-streamer.

Pour activer hostapd

```
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo systemctl status hostapd
```

Affecter un ip fixe à wlan0 dans la plage d'adresse de hostapd dans le fichier /etc/dhcpcd.conf

```
##############  POUR HOSTAPD ###############
interface wlan0
static ip_address=192.168.3.1/24
############################################
```

Désactiver temporairement ipv6 sur l'interface wlan0

```
sysctl -w net.ipv6.conf.wlan0.disable_ipv6=1
```

Pour le mettre en dur dans la configuration, modifier /etc/sysctl.conf

désactiver led camera

Ajouter

```
disable_camera_led=1
```

dans /boot/config.txt

ou en python

```
#!/usr/bin/env python
import time
import RPi.GPIO as GPIO

# Use GPIO numbering
GPIO.setmode(GPIO.BCM)

# Set GPIO for camera LED
# Use 5 for Model A/B and 32 for Model B+
CAMLED = 5

# Set GPIO to output
GPIO.setup(CAMLED, GPIO.OUT, initial=False)

# Five iterations with half a second
# between on and off
for i in range(5):
 GPIO.output(CAMLED,True) # On
 time.sleep(0.5)
 GPIO.output(CAMLED,False) # Off
 time.sleep(0.5)
```

Tester la camera

Sur le RPI

```
raspivid -t 0 -w 1280 -h 720 -o - | nc add_PC_linux 5001
```

Sur un PC linux

```
nc -l -p 5001 | /usr/bin/mplayer -fps 10 -cache 1024 -
```

mjpg-streamer

https://github.com/jacksonliam/mjpg-streamer

installer mjpg-streamer

```
sudo apt-get install git cmake libjpeg-dev libv4l-dev
git clone https://github.com/jacksonliam/mjpg-streamer
cd mjpg-streamer/mjpg-streamer-experimental
make
sudo make install
```

test

```
mjpg_streamer -i "input_raspicam.so -vf -hf -fps 15 -q 50 -x 1024 -y 768" -o "output_http.so -p 8090 -w /usr/local/share/mjpg-streamer/www"
```

dans un navigateur : http://192.168.3.1:8090/stream.html

exécution au démarrage

créer /lib/systemd/system/mjpg.service

```
[Unit]
Description=Job that runs mjpg_streamer
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/mjpg_streamer -i "input_raspicam.so -rot 270  -vf -hf -fps 15 -q 50 -x 1024 -y 768" -o "output_http.so -p 8090 -w
/usr/local/share/mjpg-streamer/www"
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

et le mettre en service

```
sudo systemctl enable mjpg.service
sudo systemctl start mjpg.service
```

nginx

Pour consulter les images ou le stream en http

```
sudo apt install nginx
```

configuration

Modifier le fichier /etc/nginx/site-avalaible/default

```
        location /images {
                alias /home/pi/Images;
                autoindex on;
        }
        location /stream {
            set $pp_d http://localhost:8090/stream_simple.html;
            if ( $args = 'action=stream' ) {
              set $pp_d http://localhost:8090/$is_args$args;
            }
            if ( $args = 'action=snapshot' ) {
              set $pp_d http://localhost:8090/$is_args$args;
            }

            proxy_pass $pp_d;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header X-Forwarded-Port $server_port;
            proxy_set_header X-Request-Start $msec;
        }
        location /cam {
            set $pp_d http://127.0.0.1:8090?action=stream;
            proxy_pass $pp_d;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header X-Forwarded-Port $server_port;
            proxy_set_header X-Request-Start $msec;
        }
        location /snapshot {
            set $pp_d http://127.0.0.1:8090?action=snapshot;
            proxy_pass $pp_d;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header X-Forwarded-Port $server_port;
            proxy_set_header X-Request-Start $msec;
        }
```

Usage

- se connecter au réseau wifi treescam
- se connecter à treescam avec ssh pi@192.168.3.1
- initialiser la date
- la vidéo se consulte sur http://192.168.3.1/cam
- une image se consulte sur http://192.168.3.1/snapshot
- la liste des images se consulte sur http://192.168.3.1/images
- la rotation de l'image est à modifier dans le fichier /lib/systemd/system/mjpg.service
- le rythme des images est configuré par crontab -e
- ou avec terminal

```
wget -O test.jpg 192.168.3.1/snapshot
curl http://192.168.3.1/snapshot --output test.jpg
```

Outils

Stopmotion

Pour prendre des photos à intervalle régulier, il est possible d'utiliser crontab qui exécute un shell sh_cam.sh. Ce shell peut soit appeler directement raspistill ou prendre des snapshot avec mjpg-streamer.

```
#! /bin/sh

#/usr/bin/raspistill -hf -vf  -o /home/pi/Images/out.jpg
/usr/bin/wget http://treescam/stream?action=snapshot -O /home/pi/Images/out.jpg

mv /home/treescam/Images/out.jpg /home/pi/Images/out_`date +"%Y%m%d%H%M"`.jpg
```

ImageMagick

La camera, avec raspistill prend des photos à une résolution de 2592 x 1944, soit 5,038,848 pixels or 5 megapixels. Une photo pèse donc environ 2.4MB, soit 42 photos par GB. Prendre 1 photo par minute occupera 1GB en 7h, soit un ratio de 144Mb par heure ou 3,3GB par jour

https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md

Il peut être nécessaire de recadrer les images, ImageMagick permet de rogner les images : https://codeyarns.com/tech/2014-11-15-how-to-crop-image-using-imagemagick.html

Par exemple pour éliminer la partie gauche d'une image

```
convert out_202012011100.jpg -crop 1728x1944+800+0 out.jpg
```

Pour modifier tout le lot, **attention cela écrase les fichiers d'origine**, donc travailler sur des copies

```
mogrify out_202012*.jpg -crop 1728x1944+800+0
mogrify out_202012*.jpg -crop 1700x1400+800+50
```

Vidéo

Pour créer la vidéo, on peut utiliser MEncoder en ligne de commande ou le logiciel openshot

Voir https://sebsauvage.net/wiki/doku.php?id=mencoder pour des exemples

openshot

En préalable il faut choisir le profil, HD720p 50fps (1280×720) MP4(h.264) Après avoir regroupé toutes les images dans un dossier et s'être assuré que l'ordre des photos est conforme à la chronologie, il faut les importer dans openshot, ensuite, les glisser dans la timelime, chaque photo pour une durée de 2 secondes avec un fondu d'1 seconde en entrée. Pour terminer, exporter la timeline en mp4

mencoder

Il faut au préalable créer la liste des images dans un fichier liste.txt, dans le bon ordre. Le paramètre fps détermine le nombre d'images par secondes et donc la durée de la vidéo, une valeur de 10 est correcte

```
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf scale=1920:1080 -o timelapse.avi -mf type=jpeg:fps=10
mf://@liste.txt
```

mplayer

Pour lire la vidéo en boucle

```
mplayer timelapse.avi -loop 2
```

liens

https://fstoppers.com/originals/creating-time-lapse-videos-mencoder-286926

ServoBlaster

Programme pour RaspberryPi, qui fournit une interface pour piloter plusieurs servomoteurs via GPIO. Le contrôle de la position du servo est assuré par l'envoi de commandes au driver. La position est maintenue jusqu'à l'envoi d'une nouvelle commande

Liens

- https://github.com/richardghirst/PiBits/tree/master/ServoBlaster
- https://fablab.coagul.org/ServoBlaster
- https://www.youtube.com/watch?v=4A7tJ0QH4L4
- https://github.com/skalad/RPi_Cam_Web_Interface_ServoBlaster_pan_tilt

```
git clone https://github.com/richardghirst/PiBits/
cd PiBits/ServoBlaster/user
make servod
chmod +x servod
./servod --help
sudo ./servod &
sudo ./servod --pcm --p1pins=7 &
sudo echo P1-7=50 > /dev/servoblaster
sudo echo P1-11=30% > /dev/servoblaster
echo 0=50% > /dev/servoblaster
echo 1=30% > /dev/servoblaster
sudo echo 0=150 > /dev/servoblaster
sudo killall servod
```

Pan/Tilt Camera Mount

https://www.thingiverse.com/thing:1799905

Écrou de fixation sur un pied d'appareil photo

- Le petit c'est 6,35mm 1/4 pouce au pas de 20 filets au pouce : c'est le pas Kodak (ou pas UNC)
- Le gros c'est 9,5 mm 3/8-ième de pouce au pas de 16 filets au pouce : c'est le pas Congrès (défini par un congrès international autour de 1900)

On trouve ce genre de pas de vis dans les grosses quincailleries (https://www.bricovis.fr/std/visserie-americaine-12.php#Grille) et pour ceux qui veulent les faire eux-mêmes :

- Pas Kodak : Filière Withworth 1/4“, 20 filets au pouce, serie W. Taraud américain 1/4 serie NC
- Pas du Congès (3/8) : Filière Withworth 3/8” serie NC taraud americain NC

Timelapse

https://www.raspberrypi.org/documentation/usage/camera/raspicam/timelapse.md

```
raspistill -t 30000 -tl 2000 -o image%04d.jpg
```

Une photo toutes les 2 secondes pendant 30 secondes

Convertir les images JPEG en vidéo H264

```
avconv -r 10 -i image%04d.jpg -r 10 -vcodec libx264 -vf scale=1280:720 timelapse.mp4
```



! avconf ne semble plus être maintenu

```
ffmpeg -r 10 -i image%04d.jpg -vcodec libx264 -vf scale=1280:720 timelapse.mp4
```

avec mencoder

```
ls *.jpg > liste.txt
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:aspect=16/9:vbitrate=8000000 -vf scale=1920:1080 -o timelapse.avi -mf type=jpeg:fps=24
mf://@liste.txt
```

Article extrait de : http://lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://lesporteslogiques.net/wiki/openatelier/projet/treescam?rev=1632317649**
Article mis à jour: **2021/09/22 15:34**