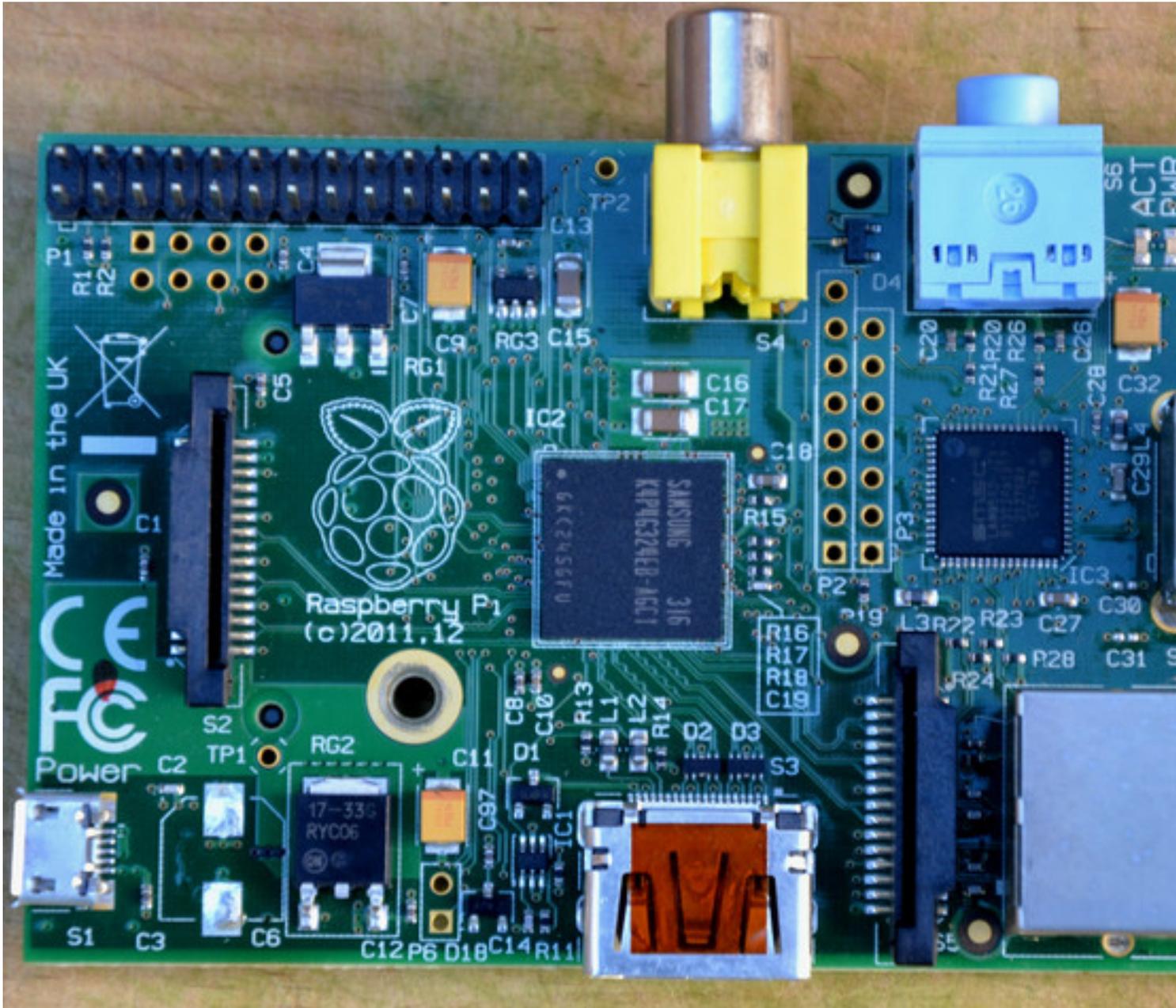


Ci dessous la carte Raspberry pi utilisée dans le nichoir et la mangeoire



L'installation du système d'exploitation sur la carte Raspberry pi est décrite partout, répéter est inutile.

Le but principal ici est de voir les couvées de mésanges, la première description sera donc l'installation de la webcam puis le système de chauffage et réfrigération.

Sur un système fraîchement installé, il faut commencer par faire la mise à jour

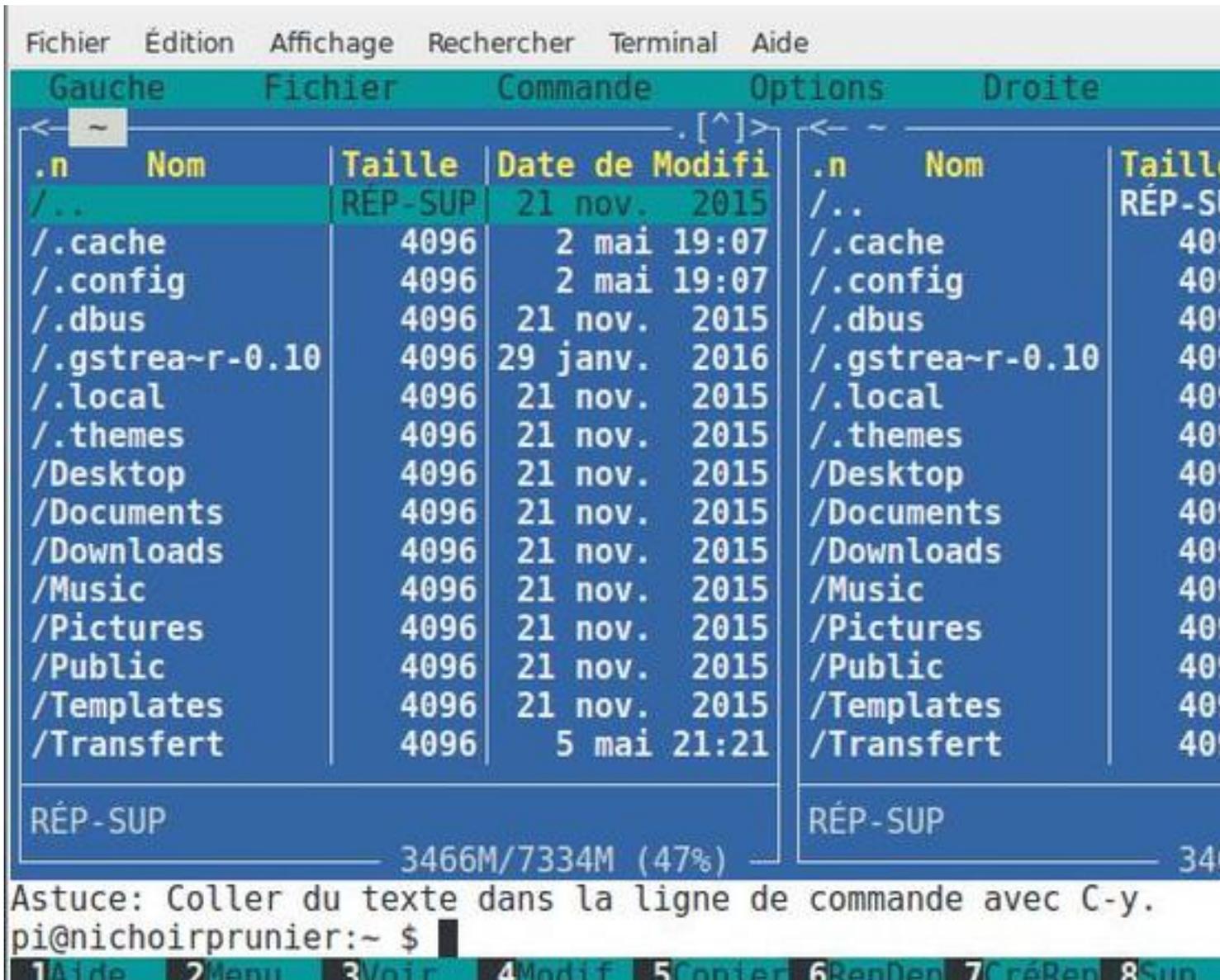
Dans un terminal , identifié en root

```
/home/pi# apt-get update
```

```
/home/pi# apt-get upgrade
```

La navigation dans l'arborescence du système est grandement facilitée par l'utilisation de Midnight Commander. Il est notamment plus facile d'y ouvrir l'éditeur de texte. Pour le lancer à partir du terminal, taper mc.

```
/home/pi# apt-get install mc
```



Pour communiquer avec la Raspberry pi le réseau filaire est utilisé. Le wifi un long moment testé apportait trop de déconnexion et a été abandonné (il a fallu passer du câble ethernet).

Configuration du réseau dans /etc/network/interfaces, remplacer tout par ce qui suit

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.1.111
```

```
netmask 255.255.255.0
```

```
gateway 192.168.1.1
```

Et relancer le réseau avec

```
/etc/init.d/networking restart
```

Il est maintenant possible de travailler sur la carte à distance avec SSH

Installation de la webcam:

La caméra développée pour la carte Raspberry pi a été créée après l'installation des premières webcams , des Logitech sont utilisées. Leur câble usb permet de placer la caméra dans n'importe quelle position, n'importe où. Elles sont toujours utilisées et sont même autofocus.

La transmission des vidéos est assurée par mjpg streamer la source utilisée au début est : <http://www.pobot.org/Vision-par-webcam-avec-une.html>

Tout n'étant pas simple et quelques erreurs y étant présentes, l'installation a été faite comme ce qui suit.

```
/home/pi# aptitude install v4l-utils v4l-conf
```

Une fois fait, on peut lancer différents outils, dont v4l2-ctl et v4l-info

```
/home/pi# aptitude install subversion
```

```
/home/pi# svn co https://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer/
```

```
/home/pi# aptitude install libjpeg8-dev
```

```
/home/pi# aptitude install imagemagick
```

```
/home/pi# cd mjpg-streamer
```

```
/home/pi#/mjpg-streamer# make
```

Lancer la webcam:

```
/home/pi/mjpg-streamer# ./mjpg_streamer -i "./input_uvc.so -y -n -f 15 -r 640x480" -o  
"./output_http.so -n -p 80 -w ./www"
```

Lancer mjpg-streamer au démarrage

Créer un script shell avec gedit ou nano

```
#!/bin/bash  
cd /home/pi/mjpg-streamer  
./mjpg_streamer -i "./input_uvc.so -y -n -f 15 -r 640x480" -o "./output_http.so -n -p 80 -w ./www"
```

enregistrer sous webcam_start.sh dans /etc/init.d
rendre le script exécutable
chmod +x webcam_start.sh

puis

```
update-rc.d webcam_start.sh defaults
```

reboot

Au démarrage de la carte la webcam sera lancée.

Pour afficher la webcam, lancer

<http://192.168.1.111:80> dans n'importe quel navigateur, il affichera ceci

MJPEG-Streamer Demo Pages

a ressource friendly streaming application

Home

Static

Stream

Java

Javascript

VideoLAN

Control

Version info:
v0.1 (Okt 22, 2007)

About

Details about the M-JP

Congratulations

You successfully managed to install this streaming we... you can also access the stream of JPGs, which can o... example. This installation consists of these example... the look and content.



The reason for developing this software was the need for a resource friendly streaming application for Linux-UVC compatible devices. *uvc-streamer* is working well, but I wanted to implement a more flexible instance, plugins can be used to process the images before they are sent to a global variable, multiple output plugins can access the stream. This webpage is served by the *output_http.so* plugin.



The image displayed is the result of the *image_processing* plugin. The HTTP request parameters *action=stream_picture* from the image. For example, just click on...

About the examples

