

# Installation de la raspberry Pi

## Utilisation de la version minimaliste de raspbian

Nous avons choisis une version minimaliste de raspbian disponible ici :

<http://www.linuxsystems.it/raspbian-wheezy-armhf-raspberry-pi-minimal-image/>

Pour préparer la carte SD suivre le tuto ici (section command line interface) :

<http://www.raspberrypi.org/documentation/installation/installing-images/linux.md>

## Création du point d'accès wifi

Now that we have a new wifi usb stick : D-Link's DWA 131 E1 nano Wi-Fi

<https://blog.dickwyn.xyz/2015/06/28/setting-up-d-links-dwa-131-e1-nano-wi-fi-adapter-to-work-with-raspbian/comment-page-1/> <https://www.raspberrypi.org/forums/viewtopic.php?t=33588&start=75>

At some point, they speak about some dropbox link, don't use them

Use instead :

```
wget http://fars-robotics.net/8192eu-4.9.35-v7-1014.tar.gz
```

```
tar xzf 8192eu-4.9.35-v7-1014.tar.gz
```

```
./install.sh
```

--- DEPRECATED --- Suivre ce tuto : <http://elinux.org/RPI-Wireless-Hotspot>

Un autre tuto en français :

<http://www.journaldulapin.com/2013/05/05/un-serveur-nabaztag-autonome-avec-le-raspberry-pi/>

---

Note pour la clé wifi Brest-wirless faut installer :

```
sudo apt-get install zd1211-firmware
```

Installer le nécessaire pour avoir du wifi :

```
apt-get install wireless-tools wpasupplicant
```

```
apt-get install hostapd udhcpd
```

Notre fichier */etc/udhcpd.conf* :

```
# Sample udhcpd configuration file (/etc/udhcpd.conf)
```

```
# The start and end of the IP lease block
```

```
start      192.168.42.100    #default: 192.168.0.20
```

```
end        192.168.42.150    #default: 192.168.0.254
```

```
# The interface that udhcpd will use

interface wlan0          #default: eth0

# The maximum number of leases (includes addresses reserved
# by OFFER's, DECLINE's, and ARP conflicts

#max_leases 254          #default: 254

# If remaining is true (default), udhcpd will store the time
# remaining for each lease in the udhcpd leases file. This is
# for embedded systems that cannot keep time between reboots.
# If you set remaining to no, the absolute time that the lease
# expires at will be stored in the dhcpd.leases file.

remaining yes           #default: yes

# The time period at which udhcpd will write out a dhcpd.leases
# file. If this is 0, udhcpd will never automatically write a
# lease file. (specified in seconds)

#auto_time 7200          #default: 7200 (2 hours)

# The amount of time that an IP will be reserved (leased) for if a
# DHCP decline message is received (seconds).

#decline_time 3600       #default: 3600 (1 hour)

# The amount of time that an IP will be reserved (leased) for if an
# ARP conflict occurs. (seconds)

#conflict_time 3600      #default: 3600 (1 hour)

# How long an offered address is reserved (leased) in seconds

#offer_time 60           #default: 60 (1 minute)

# If a lease to be given is below this value, the full lease time is
# instead used (seconds).

#min_lease 60           #default: 60

# The location of the leases file
```

```
#lease_file /var/lib/misc/udhcpd.leases #default:
/var/lib/misc/udhcpd.leases

# The location of the pid file
#pidfile /var/run/udhcpd.pid #default: /var/run/udhcpd.pid

# Everytime udhcpd writes a leases file, the below script will be called.
# Useful for writing the lease file to flash every few hours.

#notify_file #default: (no script)

#notify_file dumpleases # <--- useful for debugging

# The following are bootp specific options, setable by udhcpd.

#siaddr 192.168.0.22 #default: 0.0.0.0

#sname zorak #default: (none)

#boot_file /var/nfs_root #default: (none)

# The remainder of options are DHCP options and can be specified with the
# keyword 'opt' or 'option'. If an option can take multiple items, such
# as the dns option, they can be listed on the same line, or multiple
# lines. The only option with a default is 'lease'.

#Examples
#opt dns 192.168.10.2 192.168.10.10
#option subnet 255.255.255.0
#opt router 192.168.10.2
#opt wins 192.168.10.10
#option dns 129.219.13.81 # appened to above DNS servers for a total
of 3
#option domain local
#option lease 864000 # 10 days of seconds

opt dns 8.8.8.8 4.2.2.2 # The DNS servers client devices will use.
opt subnet 255.255.255.0
opt router 192.168.42.1 # The Pi's IP address on wlan0 which we will set up
shortly.
opt lease 864000 # 10 day DHCP lease time in seconds

# Currently supported options, for more info, see options.c
#opt subnet
#opt timezone
#opt router
#opt timesrv
#opt namesrv
#opt dns
#opt logsrv
#opt cookiesrv
```

```
#opt lprsrv
#opt bootsize
#opt domain
#opt swapsrv
#opt rootpath
#opt ipttl
#opt mtu
#opt broadcast
#opt wins
#opt lease
#opt ntpsrv
#opt tftp
#opt bootfile
#opt wpad

# Static leases map
#static_lease 00:60:08:11:CE:4E 192.168.0.54
#static_lease 00:60:08:11:CE:3E 192.168.0.44

# --- EyeFi Cards ---
static_lease 00:18:56:2e:74:20 192.168.42.100 # EyeFi 0
static_lease 00:18:56:2f:79:98 192.168.42.101 # EyeFi 1
static_lease 00:18:56:2f:73:52 192.168.42.102 # EyeFi 2
static_lease 00:18:56:2f:77:03 192.168.42.103 # EyeFi 3
static_lease 00:18:56:2f:75:35 192.168.42.104 # EyeFi 4
static_lease 00:18:56:2e:73:95 192.168.42.105 # EyeFi 5
```

Nous donnons une adresse IP fixe aux cartes EyeFi pour pouvoir les identifier plus facilement par la suite.

Notre fichier `/etc/network/interfaces` :

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0

up iptables-restore < /etc/iptables.ipv4.nat
```

Notre fichier `/etc/hostapd/hostapd.conf` :

```
# interface wlan du Wi-Fi
interface=wlan0
```

```
# nl80211 avec tous les drivers Linux mac80211
driver=nl80211

# Nom du spot Wi-Fi
ssid=OpenStreetView

# mode Wi-Fi (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g)
hw_mode=g

# canal de fréquence Wi-Fi (1-14)
channel=6

# Wi-Fi ouvert, pas d'authentification !
auth_algs=1

# Beacon interval in kus (1.024 ms)
beacon_int=100

# DTIM (delivery trafic information message)
dtim_period=2

# Maximum number of stations allowed in station table
max_num_sta=15

# RTS/CTS threshold; 2347 = disabled (default)
rts_threshold=2347

# Fragmentation threshold; 2346 = disabled (default)
fragm_threshold=2346

#control interface, will be used by hostapd_cli
ctrl_interface=/var/run/hostapd

wmm_enabled=0
```

## Python3.2

```
apt-get install python3.2
```

## Bluetooth (pour la montre liveview)

Installation du bluetooth :

```
apt-get install bluetooth
```

Démarrer la montre en restant appuyer sur les 2 boutons.

Faire un scan bluetooth :

```
pi@raspberrypi:~$ hcitool scan
Scanning ...
BC:85:56:68:92:8E    ubuntu-0
30:39:26:C4:B0:C2    LiveView
```

Voir ce qu'il manque pour l'apairage

## I2C (boussoles, accéléro ...)

```
apt-get install python-smbus python3-serial
```

Ajouter en fin du ~/.bashrc :

```
export QUICK2WIRE_API_HOME=/home/pi/OpenPathView_wip/includes/quick2wire-
python-api
export PYTHONPATH=$PYTHONPATH:$QUICK2WIRE_API_HOME
```

Puis faire un :

```
source .bashrc
```

Permissions, editer le fichier /lib/udev/rules.d/60-i2c-tools.rules :

```
KERNEL=="i2c-[0-9]*", GROUP="i2c", MODE="0666"
```

## Liaison série avec l'arduino (via USB)

```
sudo usermod -a -G dialout pi
```

From:  
<https://wiki.openpathview.fr/> - **Open Path View**

Permanent link:  
<https://wiki.openpathview.fr/doku.php?id=rpiinstall>

Last update: **2018/01/27 16:02**

