

HAM Radio avec le Raspberry Pi

Partie 10



Sommaire :

Partie 10 : Serveur de streaming avec Rtl sdr.



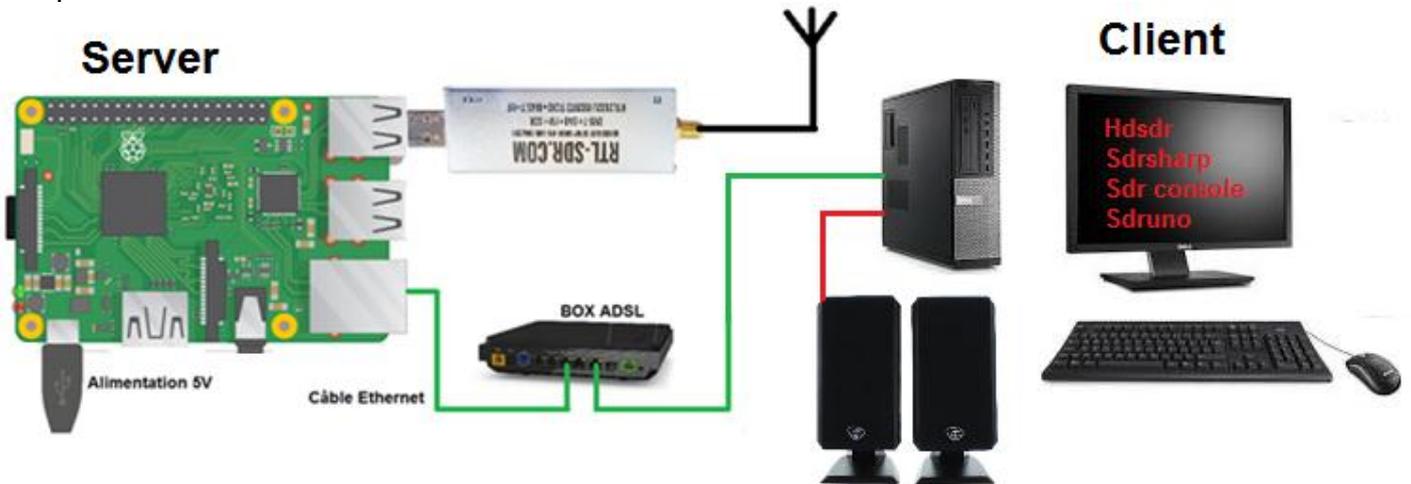
Prérequis : Partie 4 : Utilisation du système d'exploitation Raspbian Lite.

10.1 Présentation de l'ensemble serveur / client	P. 2
10.2 Installation des drivers rtl-sdr	P. 2
10.3 Serveur rtl_tcp avec 4 logiciels SDR	P. 7
10.3.1 Sdrsharp	P. 7
10.3.2 Sdr console	P. 10
10.3.3 Hdsdr	P. 12
10.3.4 Sdruno	P. 14
10.4 Installation du serveur SPY Server	P. 17
10.5 Test avec 2 logiciels clients	P. 19
10.5.1 Sdrsharp	P. 19
10.5.2 Sdr console	P. 20
10.6 Configuration de la box ADSL	P. 21
10.7 Conclusion	P. 21

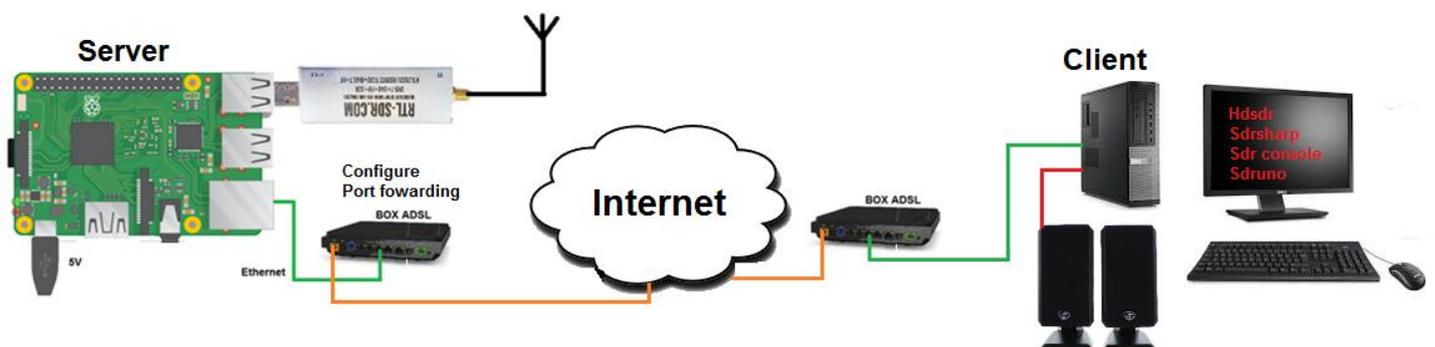
Version du 31/10/2020 V1.0 sur RPI3

10.1 Présentation de l'ensemble serveur / client

Dans ce tutoriel, il s'agit de mettre en œuvre un Raspberry Pi en tant que serveur pour faire de l'écoute à distance sur un PC. A la différence du tutoriel 6 sur OpenWebRX, le client (PC) utilise un des 4 logiciels (Hdsdr, Sdrsharp, Sdr console, Sdruno). L'avantage est que l'écoute sera beaucoup plus fluide qu'avec OpenWebRX, tout en disposant de toutes les fonctionnalités des logiciels de réception SDR. La communication entre le client et le serveur se fera en TCP/IP.



Bien évidemment il est possible d'avoir une connexion entre le serveur et le client via Internet. Dans ce cas il faudra configurer la BOX ADSL comme décrit dans le tutoriel 6.8 : Accéder au web SDR depuis l'extérieur du QRA.



10.2 Installation des drivers rtl-sdr

Cette fois ci, j'ai utilisé les deux liens suivants afin d'installer des pilotes pour la clé RTL-SDR V3. Cette version dispose d'améliorations par rapport au pilote décrit dans le tutoriel 3.

<https://www.rtl-sdr.com/rtl-sdr-blog-v-3-dongles-user-guide/>

<https://github.com/rtlsdrblog/rtl-sdr-blog>

Utiliser Raspbian Lite sans interface graphique. Penser à exécuter ces commandes si cela n'est pas déjà fait.

```
sudo apt update
sudo apt upgrade
sudo apt install git
```

Récupérer les fichiers :

```
git clone git://github.com/rtlsdrblog/rtl-sdr-blog.git
```

```
pi@raspberrypi:~ $ ls
rtl-sdr-blog
pi@raspberrypi:~ $
```

Installer les bibliothèques.

```
sudo apt install build-essential cmake usbutils libusb-1.0-0-dev
```

```
pi@raspberrypi:~/rtl-sdr/build $ sudo apt install build-essential cmake usbutils
libusb-1.0-0-dev
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
build-essential est déjà la version la plus récente (12.6).
usbutils est déjà la version la plus récente (1:010-3).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessair
es :
 libmicrodns0 libqt5charts5 xlog-data
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
 cmake-data libjsoncpp1 librhash0 libusb-1.0-doc libuv1
Paquets suggérés :
 cmake-doc ninja-build
Les NOUVEAUX paquets suivants seront installés :
 cmake cmake-data libjsoncpp1 librhash0 libusb-1.0-0-dev libusb-1.0-doc
 libuv1
0 mis à jour, 7 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 40578 ko dans les archives.
Après cette opération, 23,6 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```

Taper ces commandes dans l'ordre :

```
cd rtl-sdr-blog/

mkdir build

cd build

cmake -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON ../
```

```
pi@raspberrypi:~ $ cd rtl-sdr-blog/
pi@raspberrypi:~/rtl-sdr-blog $ mkdir build
pi@raspberrypi:~/rtl-sdr-blog $ cd build/
pi@raspberrypi:~/rtl-sdr-blog/build $ cmake -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON ../
-- The C compiler identification is GNU 8.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Build type not specified: defaulting to release.
-- Extracting version information from git describe...
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29")
-- Checking for module 'libusb-1.0'
-- Found libusb-1.0, version 1.0.22
```

sudo make install

```
pi@raspberrypi:~/rtl-sdr-blog/build $ sudo make install
Scanning dependencies of target rtl_sdr_shared
[ 3%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/librtlsdr.c.o
[ 6%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/tuner_e4k.c.o
[ 9%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/tuner_fc0012.c.o
[ 12%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/tuner_fc0013.c.o
[ 15%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/tuner_fc2580.c.o
[ 18%] Building C object src/CMakeFiles/rtl_sdr_shared.dir/tuner_r82xx.c.o
[ 21%] Linking C shared library librtlsdr.so
[ 21%] Built target rtl_sdr_shared
Scanning dependencies of target convenience_static
[ 25%] Building C object src/CMakeFiles/convenience_static.dir/convenience/convenience.c.o
[ 28%] Linking C static library libconvenience_static.a
[ 28%] Built target convenience_static
```

**sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
sudo ldconfig**

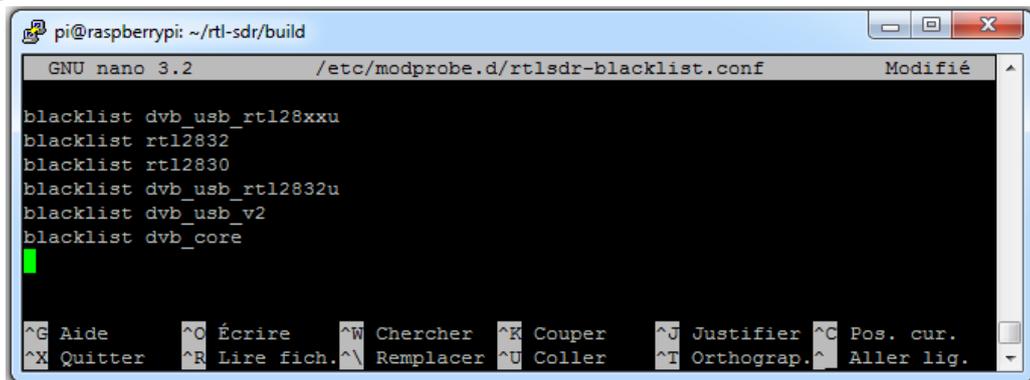
```
-- Set runtime path of "/usr/local/bin/rtl_eeprom" to ""
-- Installing: /usr/local/bin/rtl_adsb
-- Set runtime path of "/usr/local/bin/rtl_adsb" to ""
-- Installing: /usr/local/bin/rtl_power
-- Set runtime path of "/usr/local/bin/rtl_power" to ""
-- Installing: /usr/local/bin/rtl_biast
-- Set runtime path of "/usr/local/bin/rtl_biast" to ""
pi@raspberrypi:~/rtl-sdr-blog/build $ sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
pi@raspberrypi:~/rtl-sdr-blog/build $ sudo ldconfig
pi@raspberrypi:~/rtl-sdr-blog/build $
```

`sudo nano /etc/modprobe.d/rtlsdr-blacklist.conf`

```
pi@raspberrypi:~/rtl-sdr-blog/build $ sudo nano /etc/modprobe.d/rtlsdr-blacklist.conf
```

```
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
blacklist dvb_usb_rtl2832u
blacklist dvb_usb_v2
blacklist dvb_core
```

Copier-coller la liste ci-dessus dans l'éditeur nano. Rappel clic droit : pour coller le texte dans l'éditeur nano.



Enregistrer le fichier à l'aide des touches **Ctrl + o**, puis la touche entrée pour valider l'enregistrement.



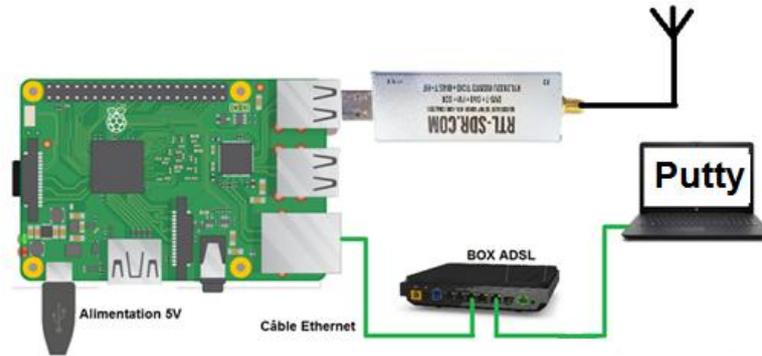
Sortir de l'éditeur à l'aide des touches **Ctrl + x**.



Redémarrer le Raspberry Pi : `pi@raspberrypi:~/rtl-sdr-blog/build $ sudo reboot`

`sudo reboot`

Vérification : Brancher la clé rtl-sdr sur un port USB de la Raspberry Pi. Connecter la clé rtl-sdr à la Raspberry Pi sur un port USB de libre.



Vérifier la présence de la clé rtl-sdr. Avec LXTerminal ou Putty, taper la ligne de commande suivante :

```
lsusb
```

Localiser la clé rtl-sdr : **RTL238 DVB-T**

Cas de la Raspberry Pi 3 :

```
pi@raspberrypi:~$ lsusb
Bus 001 Device 004: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Cas de la Raspberry Pi 4 :

```
pi@raspberrypi:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Avec LXTerminal ou Putty, taper la ligne de commande suivante : (attention **tiret du bas**)

```
rtl_test
```

La clé rtl-sdr devrait être reconnue, si ce n'est pas le cas, débrancher et rebrancher la clé et recommencer le test.

```
pi@raspberrypi:~$ rtl_test
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
^CSignal caught, exiting!
User cancel, exiting...
Samples per million lost (minimum): 0
pi@raspberrypi:~$ ^C
```

Quitter le programme en appuyant simultanément sur les touches Ctrl et la touche c (**Ctrl+c**)

10.3 Serveur rtl_tcp avec 4 logiciels SDR

10.3.1 Sdrsharp

Installer le logiciel Sdrsharp décrit dans le tutoriel 9.2.3, page 7. Cette fois ci la clé de réception RTL_SDR est branchée sur le Raspberry pi. Le but est d'y accéder via le réseau local. L'utilisation s'effectue en deux étapes :

- Exécution du serveur rtl_tcp sur le Raspberry PI ;
- Connexion sur le Raspberry PI avec le logiciel Sdrsharp.

Identifier l'adresse IP de la Raspberry PI :

ifconfig

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.55 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8a7a:b574:ec19:dbce prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:52:7f:d4 txqueuelen 1000 (Ethernet)
    RX packets 1055 bytes 146517 (143.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 238 bytes 32568 (31.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

L'adresse IP de ma RPI est : **192.168.1.55**

Cette commande avec une option « factice » permet d'obtenir toutes les configurations possibles du serveur en lien avec la clé Rtl-sdr.

rtl_tcp -h

```
pi@raspberrypi:~ $ rtl_tcp -h
rtl_tcp: invalid option -- 'h'
rtl_tcp, an I/Q spectrum server for RTL2832 based DVB-T receivers

Usage: [-a listen address]
        [-p listen port (default: 1234)]
        [-f frequency to tune to [Hz]]
        [-g gain (default: 0 for auto)]
        [-s samplerate in Hz (default: 2048000 Hz)]
        [-b number of buffers (default: 15, set by library)]
        [-n max number of linked list buffers to keep (default: 500)]
        [-d device index (default: 0)]
        [-P ppm_error (default: 0)]
        [-T enable bias-T on GPIO PIN 0 (works for rtl-sdr.com v3 dongles)]
        [-D enable direct sampling (default: off)]
pi@raspberrypi:~ $
```

Ecoute en VHF et UHF

Exécuter la commande suivante en veillant bien à remplacer l'adresse IP de la Raspberry Pi par la vôtre.

```
rtl_tcp -a 192.168.1.55
```

```
pi@raspberrypi:~ $ rtl_tcp -a 192.168.1.55
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Tuned to 1000000000 Hz.
listening...
Use the device argument 'rtl_tcp=192.168.1.55:1234' in OsmoSDR (gr-osmosdr) source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
```

Exécuter le logiciel Sdrsharp et exécuter les 4 étapes suivantes dans l'ordre.

The screenshot shows the AIRSPY SDR# v1.0.0.1765 - RTL-SDR TCP interface. The main window displays a frequency of 000.105.500.000 and a spectrum plot. A settings dialog box titled "RTL-TCP Settings" is open, showing the Host IP address 192.168.1.55 and Port 1234. The interface includes various controls for the radio, filter, and audio processing. Red boxes and numbers 1, 2, 3, and 4 highlight specific elements: 1 points to the Radio section, 2 to the settings gear icon, 3 to the Host IP field, and 4 to the play button icon.

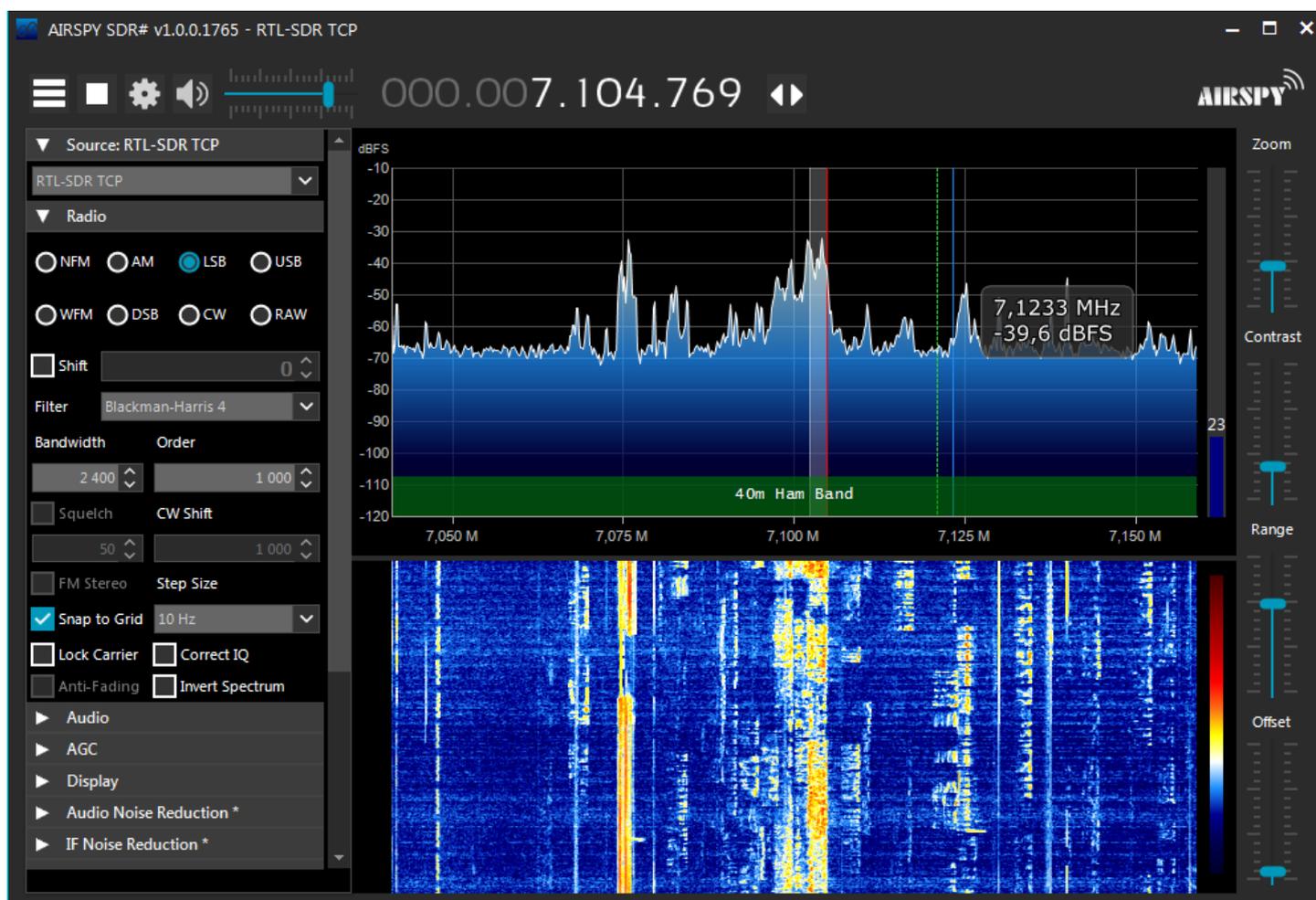
Ecoute en HF

Arrêter l'écoute sur Sdrsharp . Puis stopper le serveur avec les touches CTRL+C.

```
Use the device argument 'rtl_tcp=192.168.1.55:1234' in OsmoSDR (gr-osmosdr) source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
^CSignal caught, exiting!
bye!
pi@raspberrypi:~ $ ctrl+c
```

Exécuter la commande suivante sur le serveur RPI (mode conversion directe) en veillant bien à remplacer l'adresse IP de la Raspberry Pi par la vôtre.

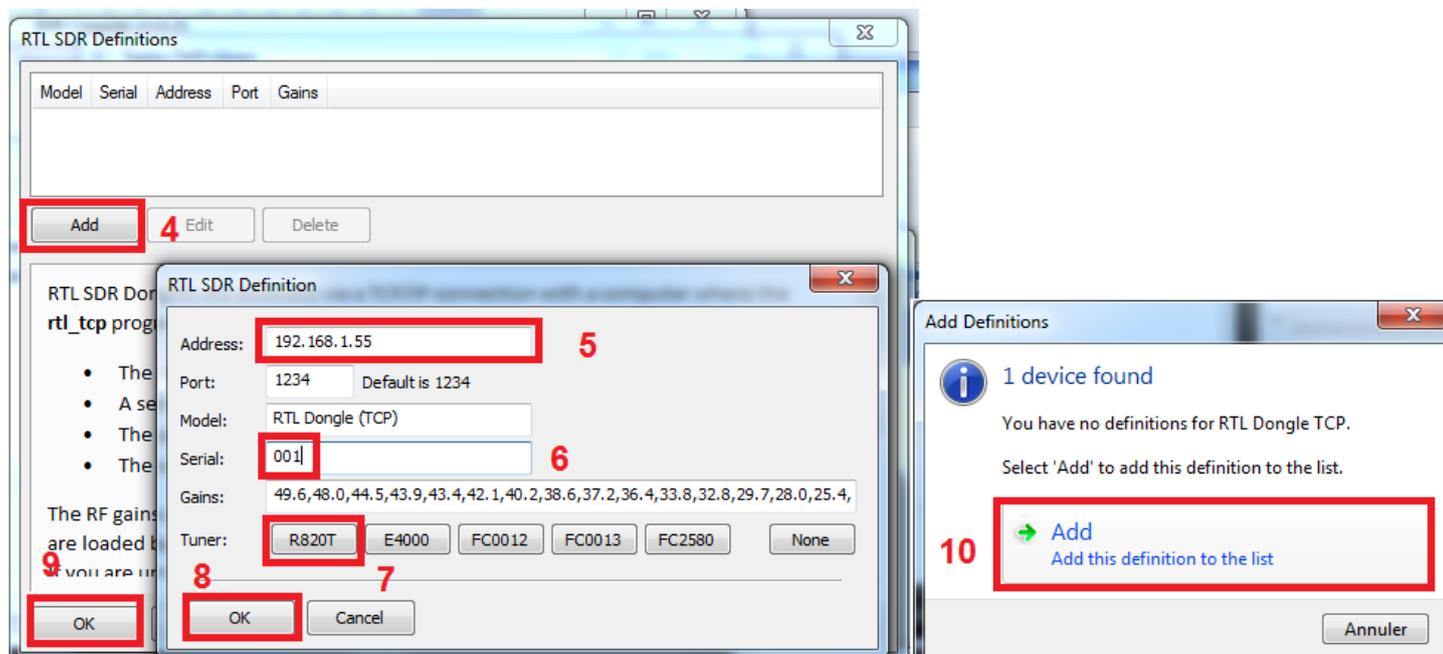
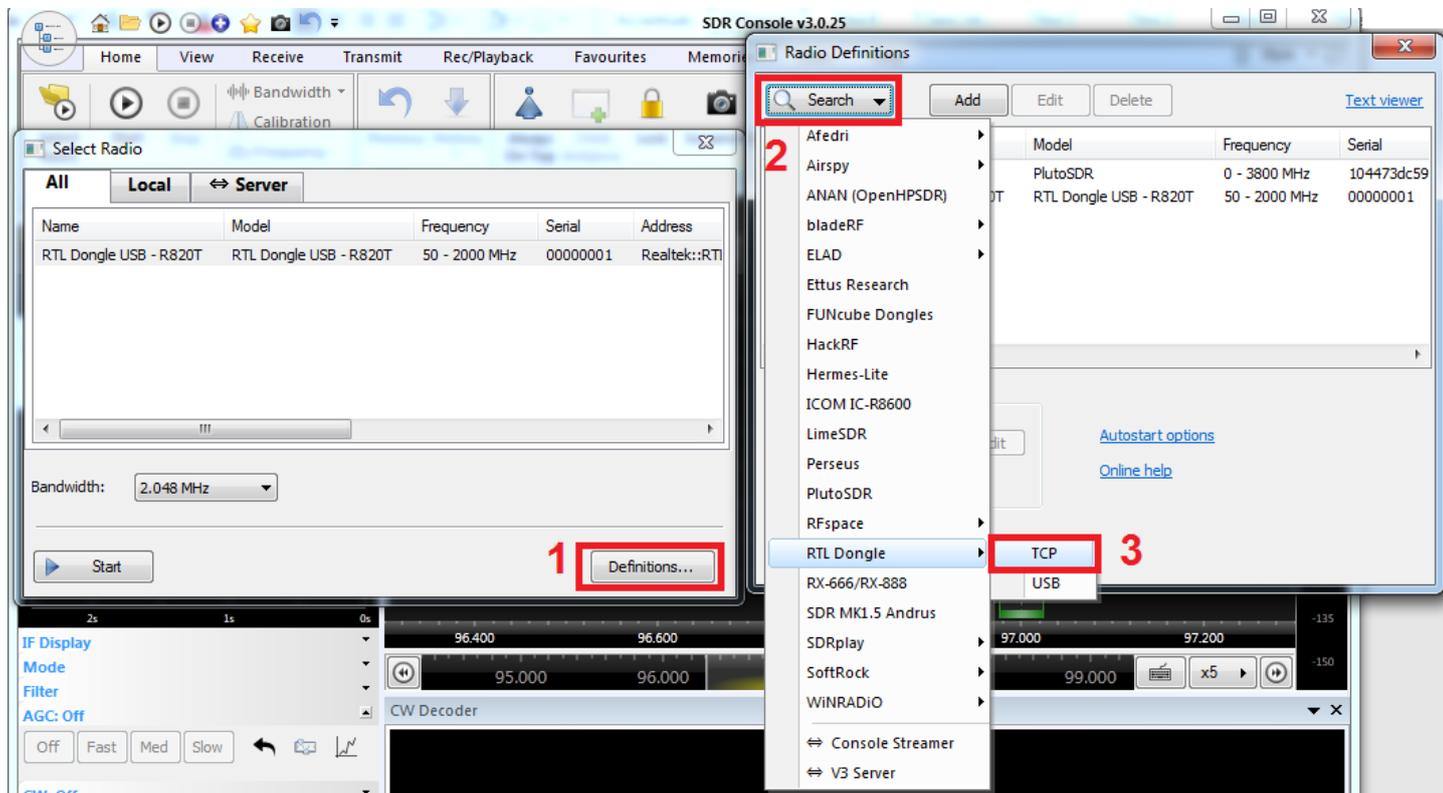
```
rtl_tcp -a 192.168.1.55 -D
```

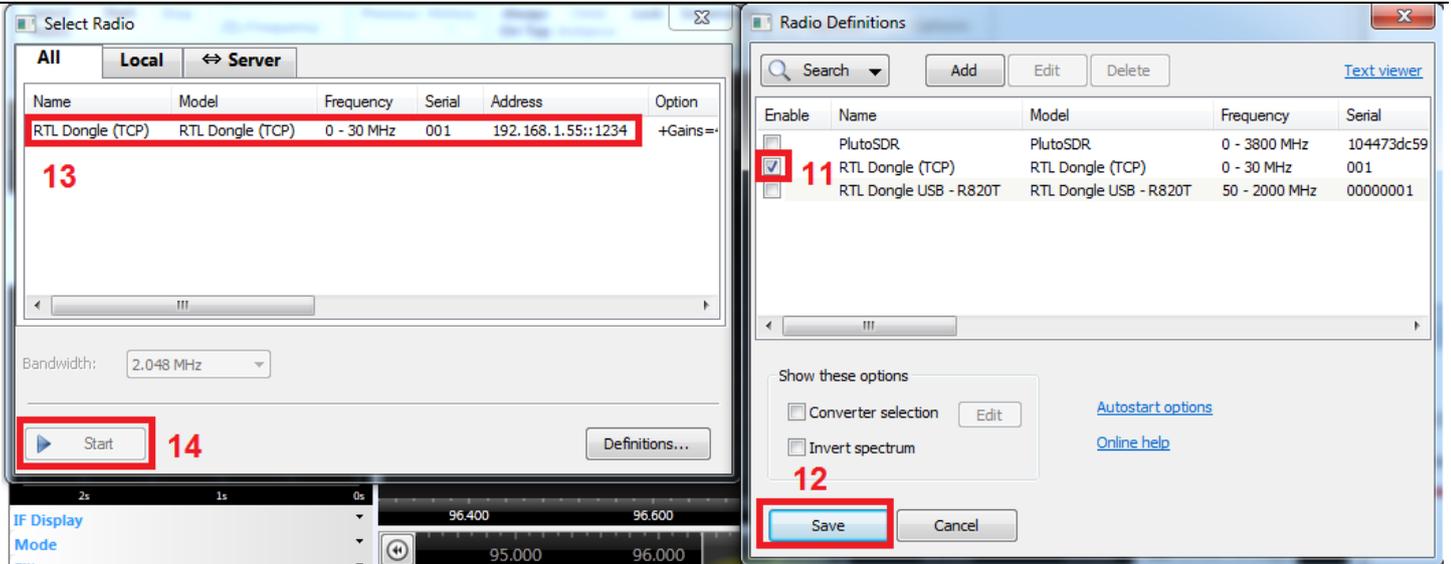


10.3.2 Sdr console

Installer le logiciel Sdr console décrit dans le tutoriel 9.2.4, page 9.

Exécuter Sdr console  Console, puis choisir RTL Dongle TCP





Exécuter la commande suivante sur le serveur RPI en veillant bien à remplacer l'adresse IP de la Raspberry Pi par la vôtre.

```
rtl_tcp -a 192.168.1.55
```

Cliquer sur Start pour démarrer l'écoute. La démarche est identique pour une écoute en HF avec l'option **-D**



10.3.3 Hdsdr

Installer le logiciel Sdr console décrit dans le tutoriel 9.2.2, page 4.

Télécharger le fichier [ExtIO_RTL_TCP_2020-1.zip](#)

Win32 Release

hayguen released this on 29 Aug · 3 commits to master since this release

Precompiled ExtIO-DLL with README for Winrad/HSDR,
supporting several new features as announced here:

<https://www.rtl-sdr.com/new-updates-to-the-librtlsdr-rtl-sdr-driver-fork/>

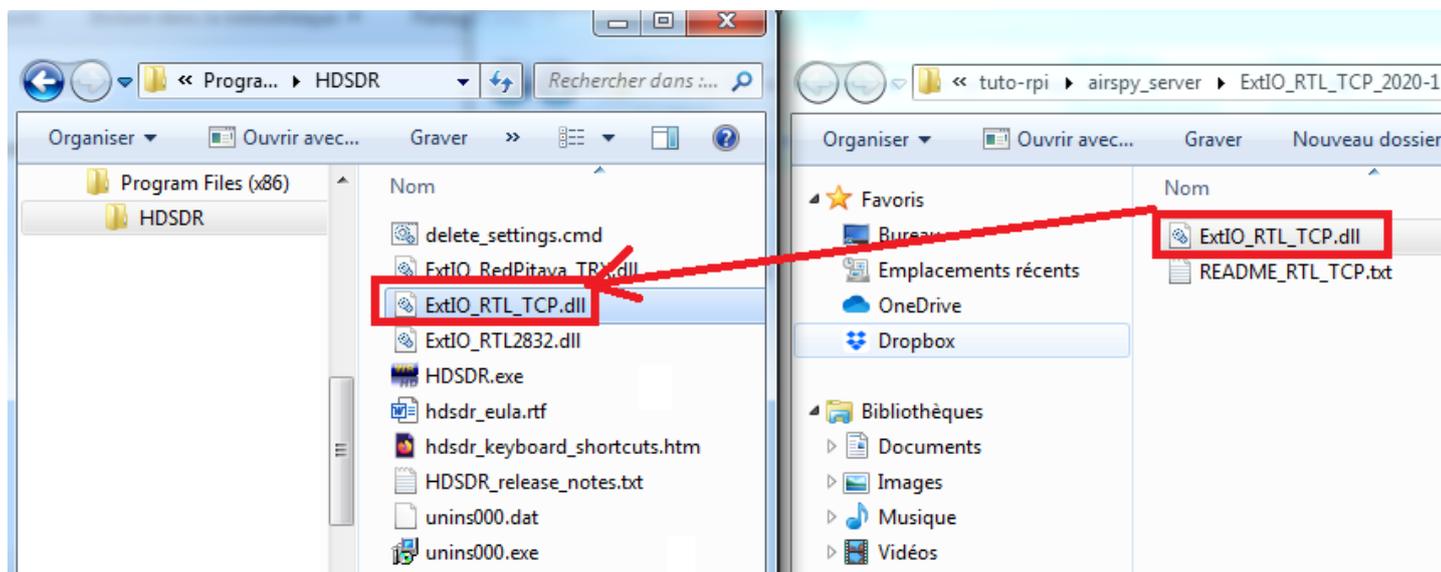
Requires rtl_tcp built from development branch of <https://github.com/librtlsdr/librtlsdr>,
e.g.:

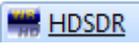
https://github.com/librtlsdr/librtlsdr/releases/tag/dev_2020-08-26

https://github.com/librtlsdr/librtlsdr/files/5130683/librtlsdr-win32_2020-08-26.zip

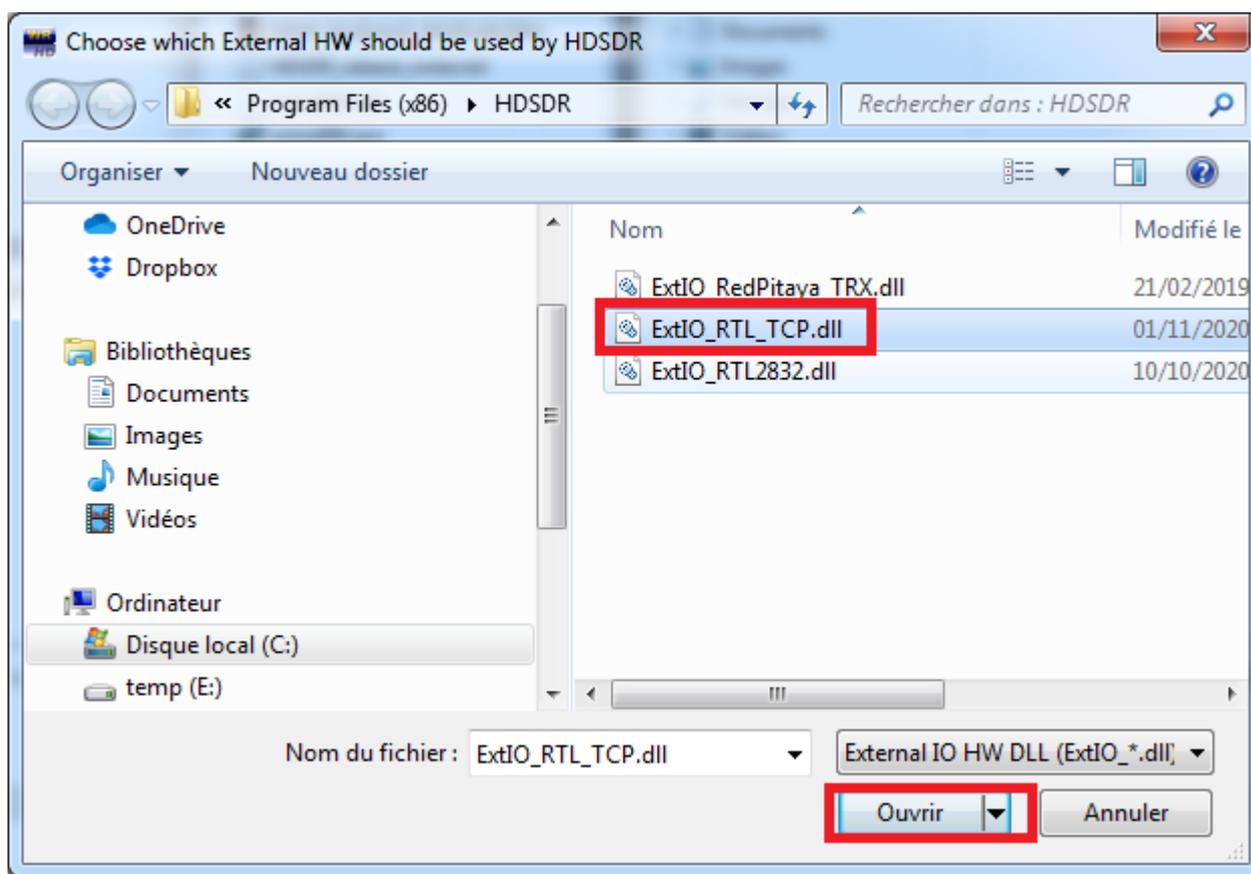
[ExtIO_RTL_TCP_2020-1.zip](#)

Copier le fichier ExtIO_RTL_TCP.dll dans le répertoire d'installation de HSDR

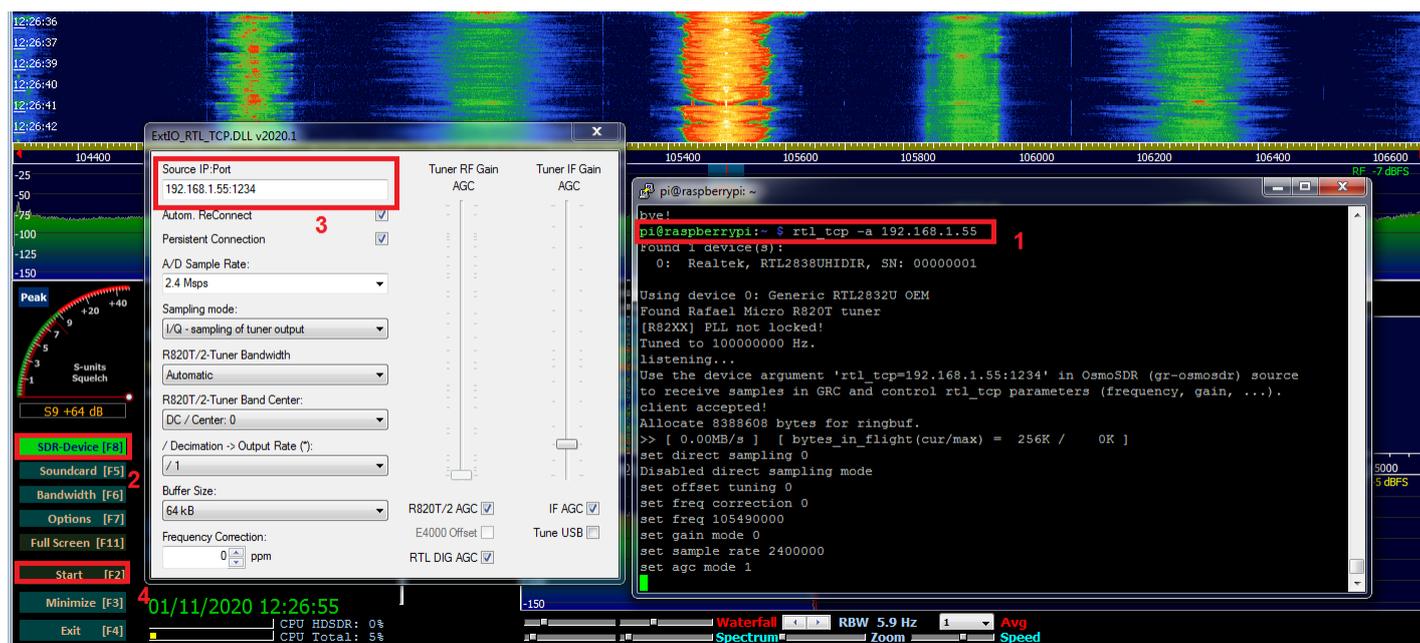


Puis Exécuter le programme HSDR 

Sélectionner le fichier ExtIO_RTL_TCP.dll



Arrêter l'écoute en appuyant sur Stop[F2], puis renseigner la bonne adresse IP dans SDR Device [F8]. Une fois configuré appuyer sur Start[F2].



Pour une écoute en HF, il faudra changer le sampling mode à Q

The screenshot shows the HDSDR software interface. The main window displays a waterfall spectrum with a frequency range from 7020 to 7130 kHz. The interface includes various controls such as 'Tune', 'Volume', and 'Options'. A terminal window is open in the foreground, showing the following output:

```

pi@raspberrypi:~$ rtl_tcp -a 192.168.1.55 -D
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Enabled direct sampling mode, input 2
Enabled direct sampling mode, input 2/Q.
Tuned to 1000000000 Hz.
listening...
Use the device argument 'rtl_tcp=192.168.1.55:1234' in OsmoSDR (gr-osmosdr) source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
client accepted!
Allocate 8388608 bytes for ringbuf.

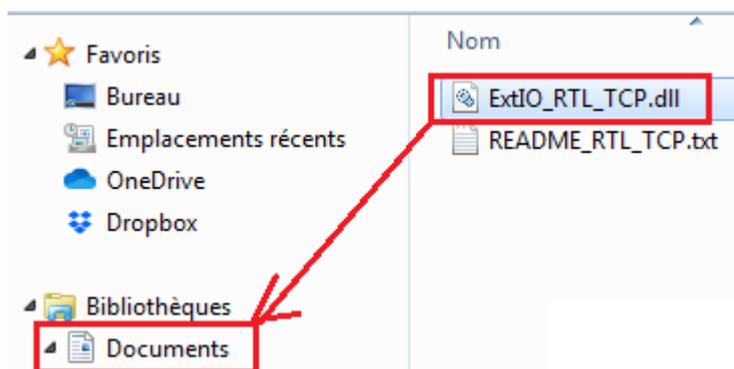
```

10.3.4 Sdruno

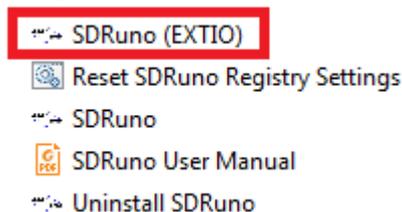
Installer le logiciel Sdruno décrit dans le tutoriel 9.2.5, page 12.

Télécharger le fichier [ExtIO_RTL_TCP_2020-1.zip](#)

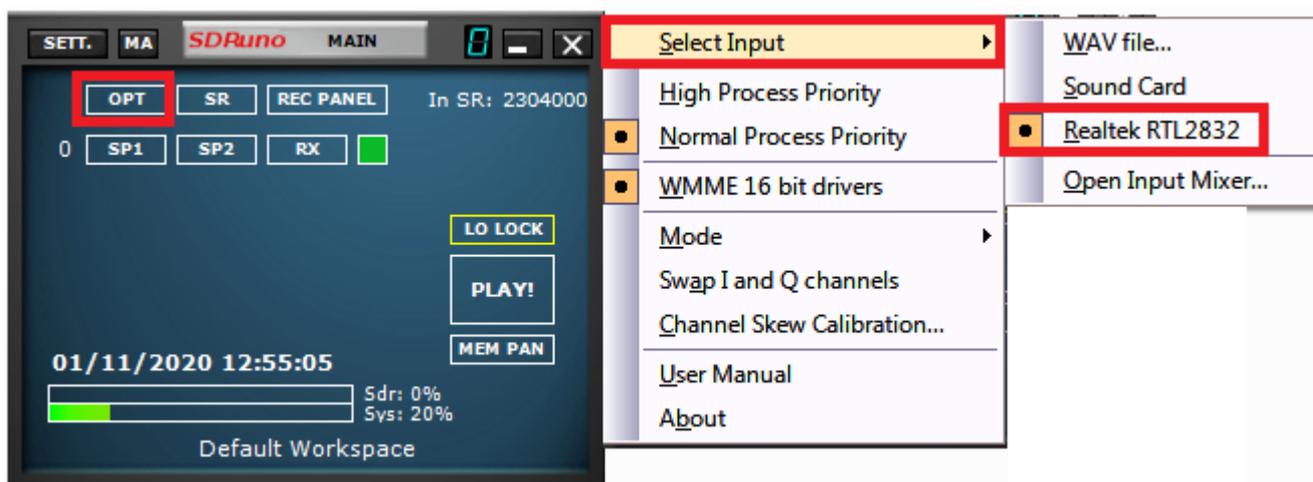
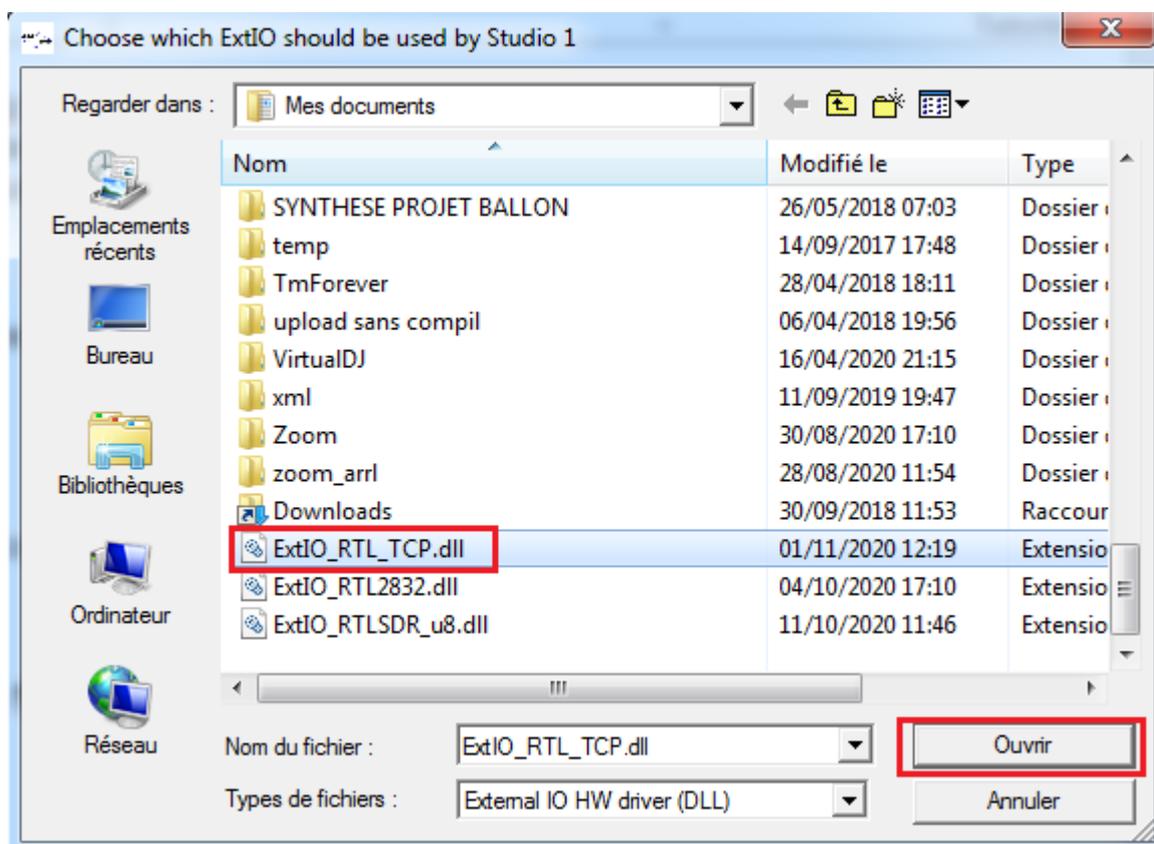
Copier le fichier `ExtIO_RTL_TCP.dll` dans le répertoire Documents



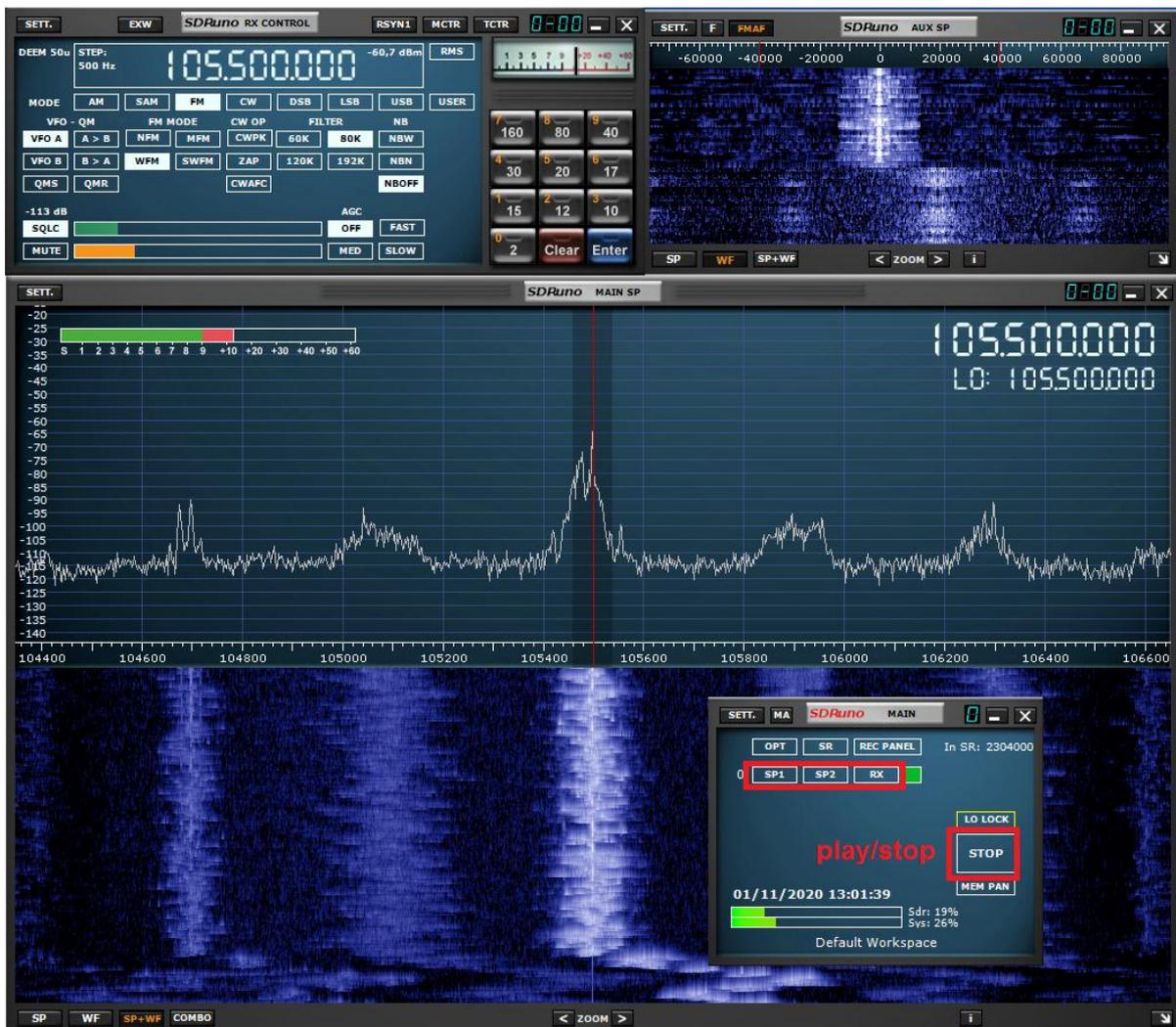
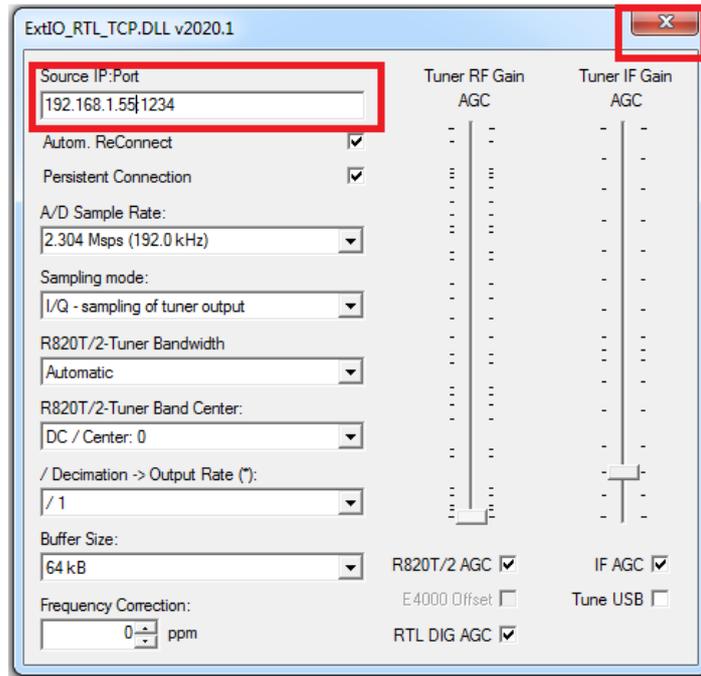
Exécuter le fichier SDRuno (Extio), installé dans le disque dur. (Faire une recherche lorsque l'on ne sait pas où le programme a été installé). Remarquer qu'il y a un manuel en pdf (SDRuno User manual) et c'est bien le seul manuel installé avec le programme parmi les 4 logiciels présentés ici.



Vérifier la bonne prise en compte du fichier `ExtIO_RTL_TCP.dll` afin que le logiciel puisse accéder à la clé rtl-sdr.



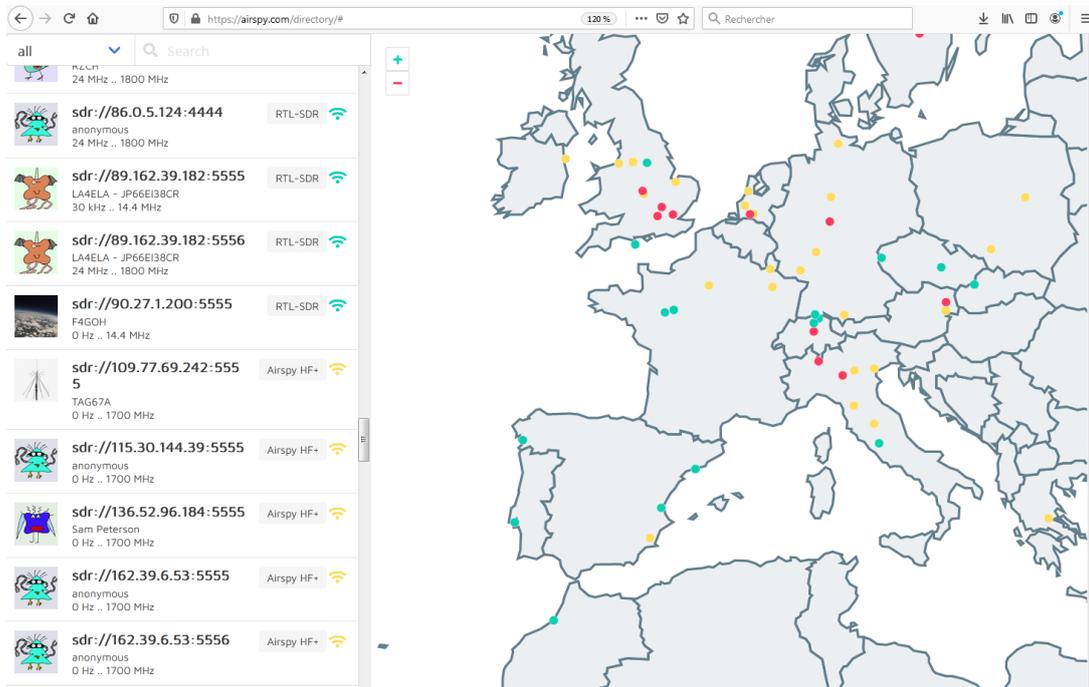
Modifier l'adresse IP en veillant bien à utiliser la vôtre.



A ce jour je n'ai pas réussi à utiliser rtl_tcp avec SDRuno en réception HF (direct sampling mode)

10.4 Installation du serveur SPY Server

[Spyserver](#) fonctionne un peu de la même manière que `rtl_tcp`, mais avec en plus la possibilité de partager sa station d'écoute. Celle-ci apparaît sur la [carte du monde](#). Il n'y a plus qu'à recopier l'adresse IP et le port dans son logiciel SDR préféré et ainsi faire de l'écoute.



Avec l'utilitaire Putty, taper ces commandes dans l'ordre :

```
mkdir spyserver
cd spyserver/
wget -O spyserver.tgz http://airspy.com/?ddownload=4247
ls
```

```
pi@raspberrypi:~ $ mkdir spyserver
pi@raspberrypi:~ $ cd spyserver/
pi@raspberrypi:~/spyserver $ wget -O spyserver.tgz http://airspy.com/?ddownload=4247
--2020-11-01 15:33:24-- http://airspy.com/?ddownload=4247
Résolution de airspy.com (airspy.com)... 172.67.138.28, 104.27.167.24, 104.27.166.24
Connexion à airspy.com (airspy.com)|172.67.138.28|:80... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : https://airspy.com/?ddownload=4247 [suivant]
--2020-11-01 15:33:24-- https://airspy.com/?ddownload=4247
Connexion à airspy.com (airspy.com)|172.67.138.28|:443... connecté.
requête HTTP transmise, en attente de la réponse... 302 Found
Emplacement : /downloads/spyserver-arm32.tgz [suivant]
--2020-11-01 15:33:26-- https://airspy.com/downloads/spyserver-arm32.tgz
Réutilisation de la connexion existante à airspy.com:443.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 63778 (62K) [application/x-gzip]
Sauvegarde en : « spyserver.tgz »

spyserver.tgz      100%[=====>] 62,28K  --.-KB/s  ds 0,1s
2020-11-01 15:33:26 (563 KB/s) - « spyserver.tgz » sauvegardé [63778/63778]

pi@raspberrypi:~/spyserver $ ls
spyserver.tgz
pi@raspberrypi:~/spyserver $
```

Décompresser le fichier, puis éditer le fichier de configuration spyserver.config

```
tar xvzf spyserver.tgz
ls
nano spyserver.config
```

```
pi@raspberrypi:~/spyserver $ tar xvzf spyserver.tgz
spyserver
spyserver_ping
spyserver.config
pi@raspberrypi:~/spyserver $ ls
spyserver spyserver.config spyserver_ping spyserver.tgz
pi@raspberrypi:~/spyserver $ nano spyserver.config
```

Configurer votre serveur en utilisant l'exemple ci-dessous :

<pre># SPY Server Configuration File # TCP Listener bind_host = 192.168.1.55 bind_port = 5555-6666 # List Server in Airspy Directory list_in_directory = 1 # Owner Name # For example: John Doe L8ZEE owner_name = f4goh # Owner email # For example: john@doe.com owner_email = f4goh@orange.fr # Antenna # For example: Random Wire/Magnetic Loop/Mini-Whip/Inverted V/etc. antenna_type = dipole # Antenna Location # For example: 48.858332, 2.294560 antenna_location = 47.89,0.27 # General Description general_description = test rx # User sessions maximum_clients = 1 # Maximum session duration # In minutes. 0 for no limit #maximum_session_duration = 30 # Allow clients to retune and change of gain of the device # allow_control = 1 # Device Type # Possible Values: # AirspyOne (R0, R2, Mini)</pre>	<pre># AirspyHF+ # RTL-SDR # Auto (Scans for the first available device) device_type = RTL-SDR # Device Serial Number as 64bit Hex # For example: 0xDD52D95C904534AD # A value of 0 will acquire the first available device. # device_serial = 0 # Device Sample Rate # Possible Values: # Airspy R0, R2 : 1000000 or 2500000 # Airspy Mini : 6000000 or 3000000 # Airspy HF+ : 768000 # RTL-SDR : 500000 to 3200000 # Comment to use the device's default #device_sample_rate = 2500000 # Force 8bit Compression Mode # The 8bit Compression mode has proven sufficiently good for most streaming use cases. # Use it to same some internet bandwidth. # #force_8bit = 1 # Maximum Bandwidth # Limits the maximum IQ bandwidth the clients can set # Recommended value for WFM is 200000 # Recommended value for narrow band modes is 15000 # #maximum_bandwidth = 15000 # FFT Frames Per Second fft_fps = 15 # FFT Bins # Bins = 2^fft_bin_bits fft_bin_bits = 16</pre>	<pre># Initial Center Frequency initial_frequency = 7100000 # Minimum Tunable Frequency # Comment if using the device default # #minimum_frequency = 0 # Maximum Tunable Frequency # Comment if using the device default #maximum_frequency = 35000000 # Frequency Correction in PPB # #frequency_correction_ppb = 0 # Initial Gain # #initial_gain = 5 # RTL-SDR Sampling mode # Quadrature = 0, # Direct Sampling I Branch = 1 # Direct Sampling Q Branch = 2 rtl_sampling_mode = 2 # Converter Offset # Set to -120000000 to enable the SpyVerter offset #converter_offset = -120000000 # Bias-Tee # For AirspyOne only - Useful for LNA's and SpyVerter #enable_bias_tee = 0 # Buffer Size (in milliseconds) buffer_size_ms = 50 # Buffer Count buffer_count = 10</pre>
---	---	--

Dans l'exemple de configuration `rtl_sampling_mode = 2` pour une écoute en HF. Pour une écoute en VHF/UHF il suffit d'ajouter le caractère # pour mettre la ligne en commentaire (`#rtl_sampling_mode = 2`).

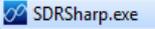
Exécuter le serveur.

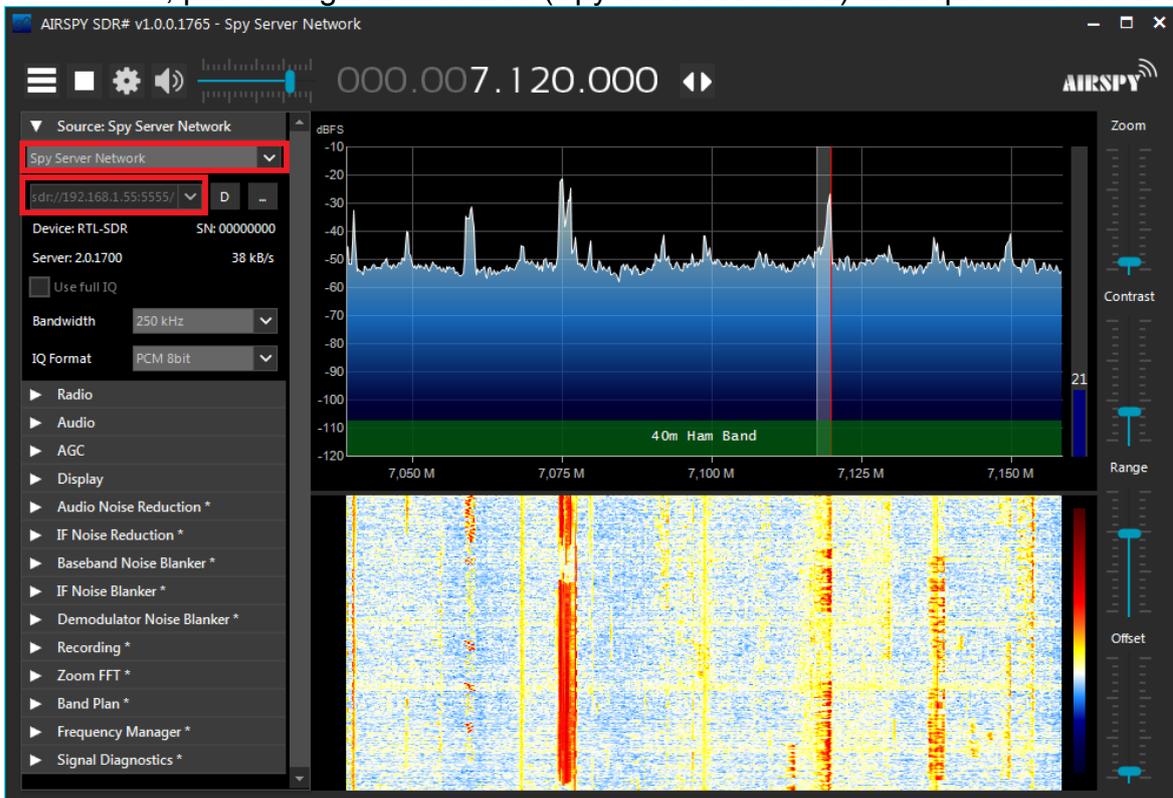
```
./spyserver
```

```
pi@raspberrypi:~/spyserver $ ./spyserver
SPY Server v2.0.1700
Copyright (C) 2016-2018 Youssef Touil - https://airspy.com
Reading the configuration file: spyserver.config
Listening for connections on 192.168.1.55:5555
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Enabled direct sampling mode, input 2
Accepted client 192.168.1.13:26511 running SDR# v1.0.0.1765 on Microsoft Windows
NT 6.1.7601 Service Pack 1
Device was sleeping. Wake up!
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Enabled direct sampling mode, input 2
Acquired an RTL-SDR device
```

10.5 Test avec 2 logiciels clients

10.5.1 Sdrsharp

Exécuter , puis configurer la source (Spy Server Network) ainsi que l'adresse IP.



Dans cette configuration l'écoute est sur le réseau local. Pour pouvoir accéder depuis Internet il faut configurer la BOX ADSL.

10.5.2 Sdr console

Même principe que pour la partie 10.3.2, mais avec SPY Server

1 Add Edit Delete

2 SPY Server

3 Address: 192.168.1.55 Port: 5555

4 Device: - None -

5 Search OK Cancel

6 OK Cancel

7 RTL Dongle (Svr) | RTL Dongle (Svr) | 0 - 14 MHz | 00000000 | 192.168.1.55::5555

8 Start

RTL Dongle (Svr) :: SDR Console v3.0.25

Home View Receive Transmit Rec/Playback Favourites Memories Tools Help

Select Radio Start Stop Bandwidth 100 - 2900 Hz Calibration Visual Gain 0 dB Buffering 100ms Resolution Low (8 bit)

Receive RX 1 0.007.143.800

Haut-parleurs (Realtek High Definition Audio)

IF Display Mode SAM CW-U BFM NFM WFM LSB USB Wide-U

Filter AGC: Slow CW: Off Noise Blanking: Off Noise Reduction: Off Notch: Off

RTL Dongle (Svr), BW = 18.750 kHz CPU: 9.0% Audio: 31ms

10.6 Configuration de la box ADSL

La configuration est identique au tutoriel 6.8 : Accéder au web SDR depuis l'extérieur du QRA.

La différence est dans le numéro du port (5555).

Retour Réseau

DHCP	NAT/PAT	DNS	UPnP	DynDNS	DMZ	NTP	IPv6
------	---------	-----	------	--------	-----	-----	------

spyserver	5555 <small>ex. : 1000</small>	555 <small>ex. : 1000-2000</small>	TCP ▼	raspberrypi-2 ▼	Créer
-----------	-----------------------------------	---------------------------------------	-------	-----------------	-------

Activer	Application/Service	Port interne	Port externe	Protocole	Équipement	
<input checked="" type="checkbox"/>	spyserver	5555	5555	TCP	raspberrypi-2	

Sur la [carte du monde](#), ma station apparait, l'icône « ready » est en vert, il est possible d'accéder à ma station en recopiant l'adresse IP ainsi que le port :

The screenshot shows a web browser displaying the Aircrack-ng directory page for an RTL-SDR station. The URL is `https://airspy.com/directory/#`. The page features a map of Europe with a green dot indicating the station's location. Below the map, there is a table of server statistics and a 'test rx' button.

test rx	
Server version	2.0.1700
Operating system	Linux 5.4.51-v7+ armv7l
Max clients	2
Connected clients	1
Max session duration	Unlimited
Antenna type	dipole
Device type	RTL-SDR
Device resolution	8 bit
Full control allowed	✓
Current center frequency	7.1 MHz
Minimum frequency	0 Hz
Maximum frequency	14.4 MHz
Maximum displayed bandwidth	2 MHz
Display refresh rate	15 FPS
Maximum streamed bandwidth	2 MHz
Maximum IQ sample rate	2.4 MSPS

10.7 Conclusion

Le partage de réception SDR est très facile à réaliser avec un Raspberry pi. Cela pourra rendre de nombreux services dans diverses situations de tests. Je remercie Ladislav **OK1UNL** de m'avoir donné l'idée de ce tutoriel. Les utilisateurs de Linux pourront utiliser [GQRX](#) comme client pour le serveur.