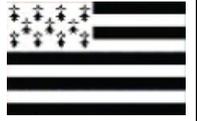


Universal HamRadio Remote

Partie 13



Sommaire :

Partie 13 : Universal HamRadio Remote

Rappel : **Raspberry Pi = RPI**

Remote = distant

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

13.1 Présentation de l'ensemble serveur / client	P. 2
13.2 Installation	P. 2
13.3 Configuration de base	P. 9
13.4 Conclusion	P. 11

Ce document reprend les informations du lien suivant avec des copies d'écran.

https://github.com/F4HTB/Universal_HamRadio_Remote_HTML5/wiki

Version du 30/10/2021 V1.0 sur RPI4

13.1 Présentation de l'ensemble serveur / client

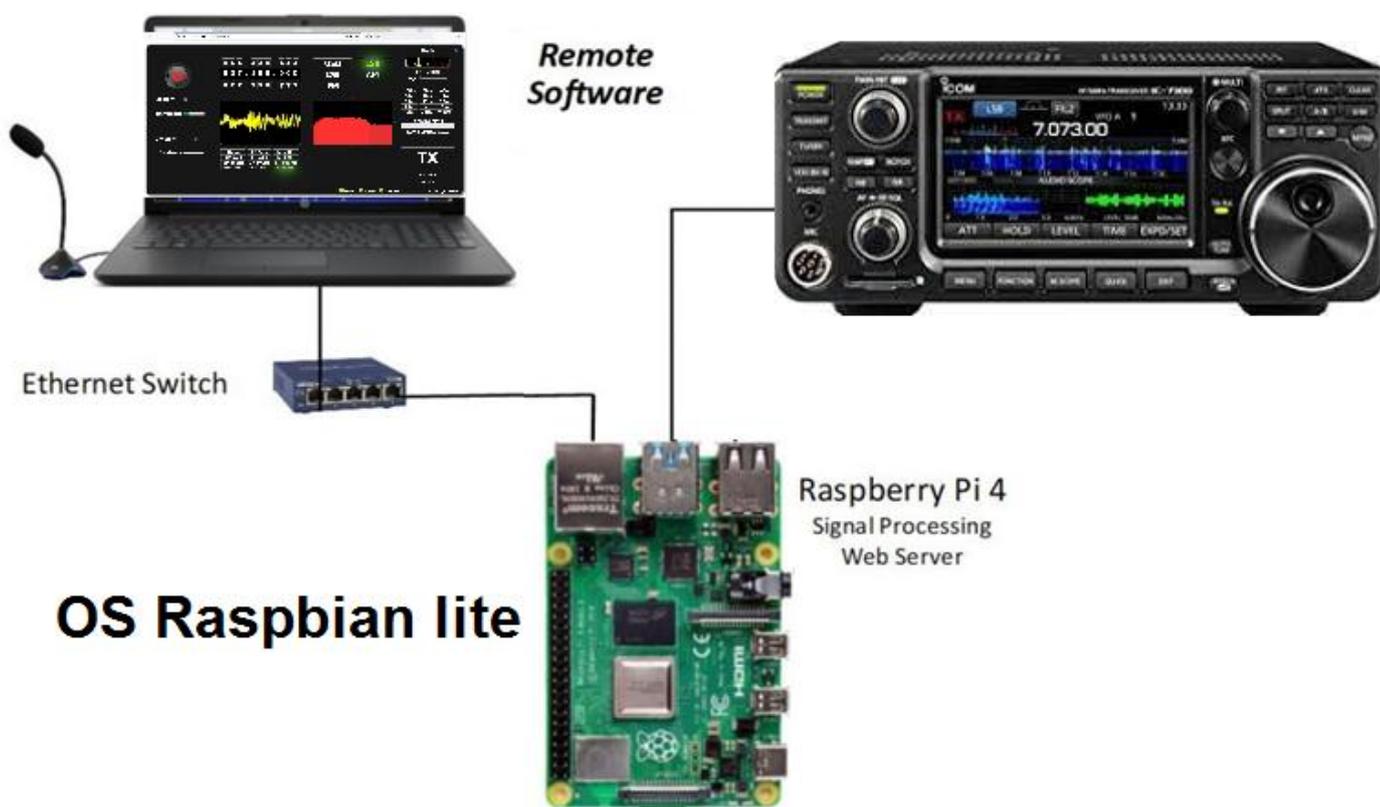
L'objectif est de pouvoir utiliser un émetteur récepteur à distance à l'aide d'un navigateur WEB.

En utilisant le navigateur Firefox (par exemple), l'utilisateur accède au Raspberry PI sur son réseau local sur lequel est connecté le transceiver.

A noter qu'il est possible d'utiliser avec le même environnement, les transceivers supportés par la bibliothèque [Hamlib](#), (qui est considérable).

Le logiciel a été réalisé par [F4HTB](#). L'interface est conviviale et intuitive.

Exemple de configuration avec un PC sur un réseau local avec un IC7300 .L'opérateur pourra trafiquer à distance en phonie avec un PC relié sur son réseau local. Le Raspberry PI est utilisé en tant que serveur WEB.



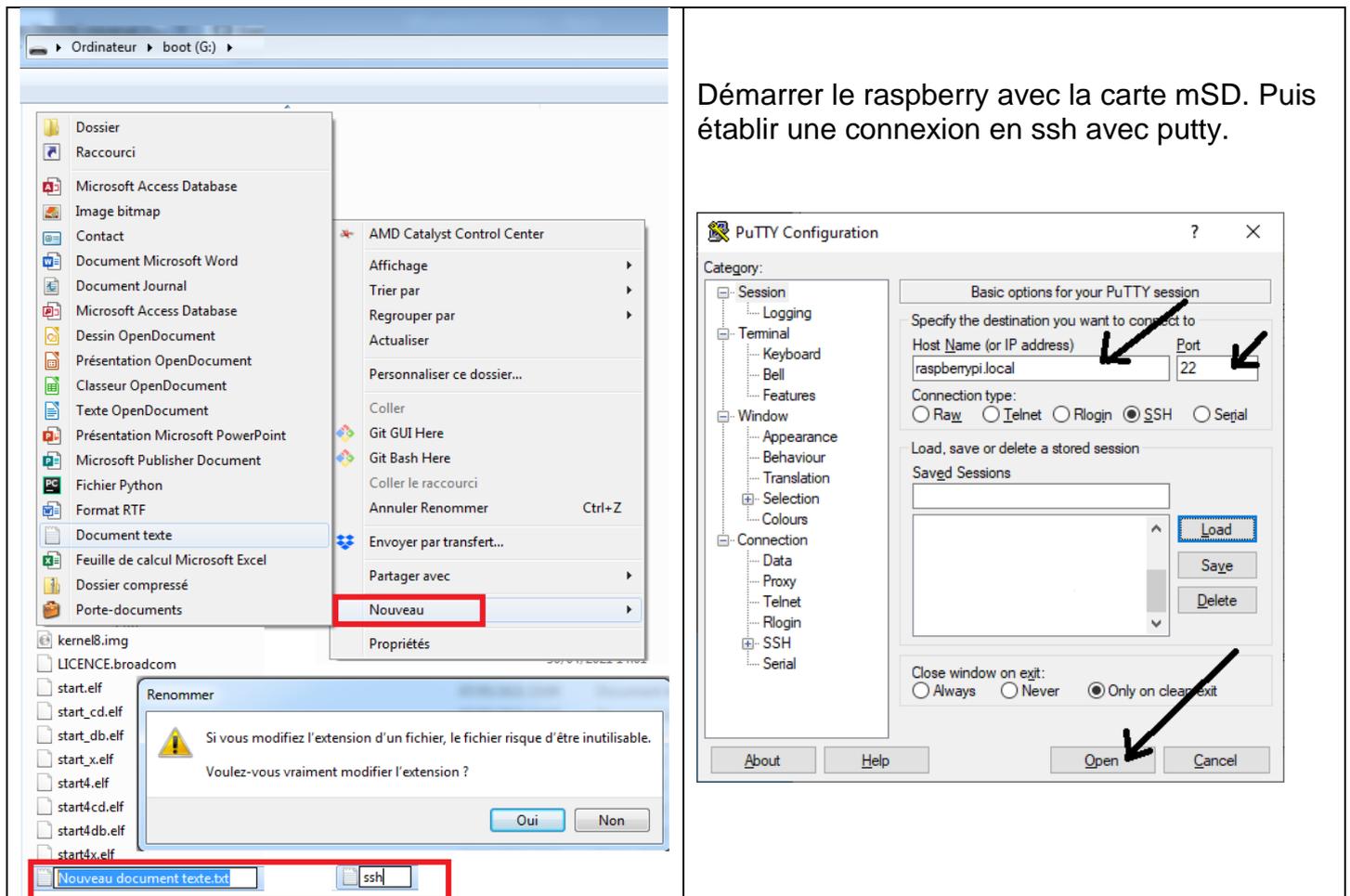
13.2 Installation

F4HTB ne propose pas d'image disque toute prête. Il faudra installer le logiciel manuellement.

Commencer par charger sur une carte mSD le système d'exploitation Raspbian Lite sans interface graphique, comme indiqué dans le tutoriel 4 ou en utilisant le guide de F4HTB.

https://github.com/F4HTB/Universal_HamRadio_Remote_HTML5/wiki/Example-of-complete-installation

Une fois l'image avec le système d'exploitation Raspbian Lite créée, ajouter le fichier vide « ssh » sans extension dans la racine de la carte mSD nommée boot. Cela active directement la connexion ssh sans passer par le menu `raspi-config` expliqué dans le tutoriel 4.



Faire une mise à jour du système d'exploitation

```
sudo apt-get update && sudo apt-get upgrade -y && sudo reboot
```

Compter 15 minutes d'attente.

Après le redémarrage, se reconnecter en ssh avec putty.

Installer l'environnement Python 3 et ses dépendances.

```
sudo apt-get install -y git python3 python3-pip python3-numpy
python3-tornado python3-serial python3-pyaudio rtl-sdr
```

```
pi@raspberrypi:~$ sudo apt-get install -y git python3 python3-pip python3-numpy
python3-tornado python3-serial python3-pyaudio rtl-sdr
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.7.3-1).
python3 set to manually installed.
The following package was automatically installed and is no longer required:
 python-colorzero
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
 dh-python gir1.2-glib-2.0 git-man libblas3 libcurl3-gnutls liberror-perl
 libexpat1-dev libgfortran5 libgirepository-1.0-1 libjack-jackd2-0 liblapack3
 libopus0 libportaudio2 libpython3-dev libpython3.7-dev librtlsdr0
 python-pip-whl python3-asn1crypto python3-cffi-backend python3-crypto
 python3-cryptography python3-dbus python3-dev python3-distutils
```

```
sudo pip3 install pyalsaaudio pam pyrtlsdr
```

```
pi@raspberrypi:~$ sudo pip3 install pyalsaaudio pam pyrtlsdr
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pyalsaaudio
  Downloading https://www.piwheels.org/simple/pyalsaaudio/pyalsaaudio-0.9.0-cp37-cp37
)
 100% |████████████████████████████████████████| 71kB 175kB/s
Collecting pam
  Downloading https://files.pythonhosted.org/packages/65/25/74fcd42f6a76c7d06ca161c1
3ea3354c6d6/pam-0.2.0-py3-none-any.whl
Collecting pyrtlsdr
  Downloading https://files.pythonhosted.org/packages/db/88/6855834021364b38885e9aed
bf39b9405d0/pyrtlsdr-0.2.92-py2.py3-none-any.whl
Collecting python-pam (from pam)
  Downloading https://files.pythonhosted.org/packages/2b/a0/2e2dfdbf74170bc3576a11d4
a533adce62f/python_pam-1.8.4-py2.py3-none-any.whl
Installing collected packages: pyalsaaudio, python-pam, pam, pyrtlsdr
Successfully installed pam-0.2.0 pyalsaaudio-0.9.0 pyrtlsdr-0.2.92 python-pam-1.8.4
pi@raspberrypi:~$
```

```
sudo apt-get autoremove -y --purge python3-libhamlib2
```

```
pi@raspberrypi:~$ sudo apt-get autoremove -y --purge python3-libhamlib2
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'python3-libhamlib2' is not installed, so not removed
The following packages will be REMOVED:
 python-colorzero*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 130 kB disk space will be freed.
(Reading database ... 43295 files and directories currently installed.)
Removing python-colorzero (1.1) ...
pi@raspberrypi:~$
```

```
sudo apt-get install -y autoconf automake libtool swig
```

```
pi@raspberrypi:~ $ sudo apt-get install -y autoconf automake libtool swig
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  autotools-dev libltdl-dev libltdl7 libsigsegv2 m4 swig3.0
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc gettext libtool-doc gfortran
  | fortran95-compiler gcj-jdk m4-doc swig-doc swig-examples swig3.0-examples
  swig3.0-doc
The following NEW packages will be installed:
  autoconf automake autotools-dev libltdl-dev libltdl7 libsigsegv2 libtool m4
  swig swig3.0
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,030 kB of archives.
After this operation, 12.6 MB of additional disk space will be used.
Get:1 http://ftp.igh.cnrs.fr/pub/os/linux/raspbian/raspbian buster/main armhf li
bsigsegv2 armhf 2.12-2 [32.3 kB]
Get:2 http://ftp.igh.cnrs.fr/pub/os/linux/raspbian/raspbian buster/main armhf m4
armhf 1.4.18-2 [185 kB]
Get:3 http://ftp.igh.cnrs.fr/pub/os/linux/raspbian/raspbian buster/main armhf au
toconf all 2.69-11 [341 kB]
Get:4 http://ftp.igh.cnrs.fr/pub/os/linux/raspbian/raspbian buster/main armhf au
totools-dev all 20180224.1 [77.0 kB]
Get:5 http://ftp.igh.cnrs.fr/pub/os/linux/raspbian/raspbian buster/main armhf au
tomake all 1:1.16.1-4 [771 kB]
```

Installer la bibliothèque Hamlib

```
cd ~/
git clone https://github.com/Hamlib/Hamlib.git
```

```
pi@raspberrypi:~ $ cd ~/
pi@raspberrypi:~ $ git clone https://github.com/Hamlib/Hamlib.git
Cloning into 'Hamlib'...
remote: Enumerating objects: 47578, done.
remote: Counting objects: 100% (4863/4863), done.
remote: Compressing objects: 100% (1784/1784), done.
remote: Total 47578 (delta 3542), reused 4235 (delta 3068), pack-reused 42715
Receiving objects: 100% (47578/47578), 18.30 MiB | 333.00 KiB/s, done.
Resolving deltas: 100% (37673/37673), done.
pi@raspberrypi:~ $ cd Hamlib/
pi@raspberrypi:~/Hamlib $ █
```

```
cd Hamlib/
./bootstrap
```

```

pi@raspberrypi:~/HamLib $ ./bootstrap
Running 'autoreconf -i' to process configure.ac
and generate the configure script.
aclocal: installing 'macros/libtool.m4' from '/usr/share/aclocal/libtool.m4'
aclocal: installing 'macros/ltoptions.m4' from '/usr/share/aclocal/ltoptions.m4'
aclocal: installing 'macros/ltsugar.m4' from '/usr/share/aclocal/ltsugar.m4'
aclocal: installing 'macros/ltversion.m4' from '/usr/share/aclocal/ltversion.m4'
aclocal: installing 'macros/lt~obsolete.m4' from '/usr/share/aclocal/lt~obsolete.m4'
aclocal: installing 'macros/pkg.m4' from '/usr/share/aclocal/pkg.m4'
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: copying file 'build-aux/ltmain.sh'
configure.ac:95: installing 'build-aux/ar-lib'
configure.ac:26: installing 'build-aux/compile'
configure.ac:126: installing 'build-aux/config.guess'
configure.ac:126: installing 'build-aux/config.sub'
configure.ac:35: installing 'build-aux/install-sh'
configure.ac:35: installing 'build-aux/missing'
amplifiers/elecraft/Makefile.am: installing 'build-aux/depcomp'
bindings/Makefile.am:106: installing 'build-aux/py-compile'
parallel-tests: installing 'build-aux/test-driver'
pi@raspberrypi:~/HamLib $ █

```

`./configure --with-python-binding PYTHON=$(which python3)`

```

pi@raspberrypi:~/HamLib $ ./configure --with-python-binding PYTHON=$(which python3)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking minix/config.h usability... █

```

`make all && sudo make install && cd bindings && make && sudo make install && sudo ldconfig`

La compilation est assez longue, compter 10 minutes.

```

Hamlib Version 4.4~git configuration:

Prefix          /usr/local
Preprocessor    gcc -E
C Compiler      gcc -g -O2
C++ Compiler    g++ -std=c++11 -g -O2

Package features:
With C++ binding          yes
With Perl binding         no
With Python binding       yes
With TCL binding          no
With Lua binding          no
With rigmem XML support   no
With Readline support     no
With INDI support         no

Enable HTML rig feature matrix no
Enable WinRadio           yes
Enable USRP               no
Enable USB backends       no
Enable shared libs        yes
Enable static libs        yes

-----
pi@raspberrypi:~/Hamlib $ make all && sudo make install && cd bindings && make && sudo make install && sudo ldconfig
Making all in macros
make[1]: Entering directory '/home/pi/Hamlib/macros'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/home/pi/Hamlib/macros'
Making all in include
make[1]: Entering directory '/home/pi/Hamlib/include'
make all-am
make[2]: Entering directory '/home/pi/Hamlib/include'
make[2]: Leaving directory '/home/pi/Hamlib/include'
make[1]: Leaving directory '/home/pi/Hamlib/include'
Making all in lib
make[1]: Entering directory '/home/pi/Hamlib/lib'
CC      termios.lo
CC      dummy.lo

```

Redémarrer le Raspberry PI, puis se reconnecter en ssh avec putty.

sudo reboot

```

- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
/usr/bin/mkdir -p '/usr/local/share/doc/hamlib/examples'
/usr/bin/install -c -m 644 py3test.py '/usr/local/share/doc/hamlib/examples'
make[2]: Leaving directory '/home/pi/Hamlib/bindings'
make[1]: Leaving directory '/home/pi/Hamlib/bindings'
pi@raspberrypi:~ $ sudo reboot

```

Installer le logiciel de F4HTB

```
cd ~/
git clone https://github.com/F4HTB/Universal\_HamRadio\_Remote\_HTML5.git
```

```
pi@raspberrypi:~ $ cd ~/
pi@raspberrypi:~ $ git clone https://github.com/F4HTB/Universal_HamRadio_Remote_HTML5.git
Cloning into 'Universal_HamRadio_Remote_HTML5'...
remote: Enumerating objects: 556, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 556 (delta 2), reused 0 (delta 0), pack-reused 549
Receiving objects: 100% (556/556), 3.79 MiB | 327.00 KiB/s, done.
Resolving deltas: 100% (348/348), done.
pi@raspberrypi:~ $ cd ~/Universal_HamRadio_Remote_HTML5
pi@raspberrypi:~/Universal_HamRadio_Remote_HTML5 $
```

Terminer par démarrer le serveur UHRR

```
cd ~/Universal_HamRadio_Remote_HTML5
PYTHONPATH=/usr/local/lib/python3.7/site-packages:$PYTHONPATH ./UHRR
```

```
pi@raspberrypi:~/Universal_HamRadio_Remote_HTML5 $ PYTHONPATH=/usr/local/lib/python3.7/site-packages:$PYTHONPATH ./UHRR
ALSA lib pcm_hw.c:1822:( _snd_pcm_hw_open) Invalid value for card
(<class 'alsaaudio.ALSAAudioError'>, ALSAAudioError('No such device [plughw:CARD=U0x41e0x30d3,DEV=0]'), <traceback object at 0xb3d394b8>)
HTTP server started.
```

Résumé des commandes

```
sudo apt-get update && sudo apt-get upgrade -y && sudo reboot
sudo apt-get install -y git python3 python3-pip python3-numpy python3-tornado python3-serial python3-pyaudio
rtl-sdr
sudo pip3 install pyalsaaudio pam pyrtlsdr
sudo apt-get autoremove -y --purge python3-libhamlib2
sudo apt-get install -y autoconf automake libtool swig
cd ~/
git clone https://github.com/Hamlib/Hamlib.git
cd Hamlib/
./bootstrap
./configure --with-python-binding PYTHON=$(which python3)
make all && sudo make install && cd bindings && make && sudo make install && sudo ldconfig
sudo reboot

cd ~/
git clone https://github.com/F4HTB/Universal_HamRadio_Remote_HTML5.git

cd ~/Universal_HamRadio_Remote_HTML5
PYTHONPATH=/usr/local/lib/python3.7/site-packages:$PYTHONPATH ./UHRR
```

13.3 Configuration de base

Avec un navigateur (Firefox par exemple) accéder au serveur en utilisant l'URL

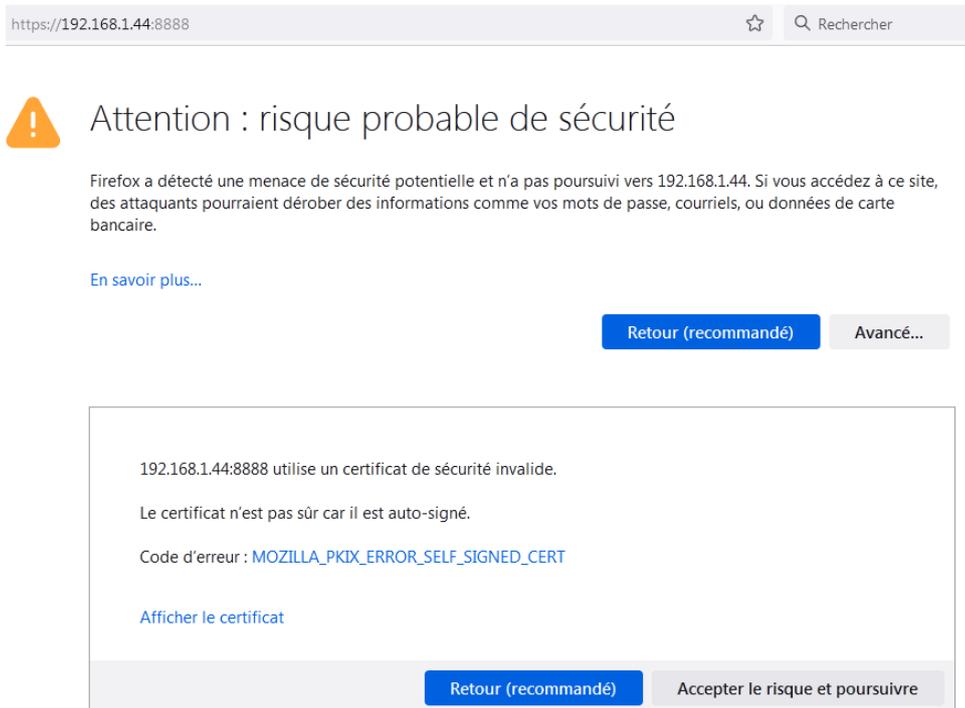
<https://raspberrypi.local:8888/>

ou :

<https://192.168.1.44:8888/>

En spécifiant l'adresse IP de votre Raspberry pi, 192.168.1.44 est donné en exemple.

Ajouter l'exception de sécurité



The screenshot shows a Firefox browser address bar with the URL `https://192.168.1.44:8888`. Below the address bar, a warning icon (orange triangle with an exclamation mark) is displayed next to the text "Attention : risque probable de sécurité". The warning message states: "Firefox a détecté une menace de sécurité potentielle et n'a pas poursuivi vers 192.168.1.44. Si vous accédez à ce site, des attaquants pourraient dérober des informations comme vos mots de passe, courriels, ou données de carte bancaire." Below this message is a link "En savoir plus...". At the bottom of the warning box are two buttons: "Retour (recommandé)" and "Avancé...".

192.168.1.44:8888 utilise un certificat de sécurité invalide.

Le certificat n'est pas sûr car il est auto-signé.

Code d'erreur : [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[Afficher le certificat](#)

At the bottom of the warning box are two buttons: "Retour (recommandé)" and "Accepter le risque et poursuivre".

Une page de configuration apparaît. Configurer les différents menus concernant la carte son et du transceiver utilisé.

Ne pas oublier de cliquer sur « save and restart server » pour enregistrer la configuration

← → ↻ 🏠 🔒 https://192.168.1.44:8888

[SERVER]

SERVER TCP/IP port: Default: **8888**. The server port

SERVER Authentication type: Default: **leave blank**. Else you can use "FILE" or/and "PAM".

SERVER database users file: Default: **UHRR_users.db** Only if you use Authentication type "FILE".

You can change database users file in UHRR.conf.
 To add a user in FILE type, add it in UHRR_users.db (default file name).
 Add one account per line as login password.
 If you plan to use PAM you can add account in command line: `adduser --no-create-home --system thecallsign`.

If you want to change certfile and keyfile, replace "UHRH.crt" and "UHRH.key" in the boot folder, and when the pi boot, it will use those files to start http ssl.

[AUDIO]

AUDIO outputdevice: Output from audio soundcard to the mic input of TRX.

AUDIO inputdevice: Input from audio soundcard from the speaker output of TRX.

[HAMLIB]

HAMLIB radio model: Hamlib trx model.

HAMLIB serial port: Serial port of the CAT interface.

HAMLIB radio rate: Serial port baud rate.

HAMLIB auto tx poweroff: Set to auto power off the trx when it's not in use

HAMLIB serial data bits: Leave blank to use the HAMIB default value.

HAMLIB serial stop bits: Leave blank to use the HAMIB default value.

HAMLIB serial parity: Leave blank to use the HAMIB default value.

HAMLIB serial handshake: Leave blank to use the HAMIB default value.

HAMLIB dtr state: Leave blank to use the HAMIB default value.

HAMLIB rts state: Leave blank to use the HAMIB default value.

[PANADAPTER]

PANADAPTER FI frequency (hz):

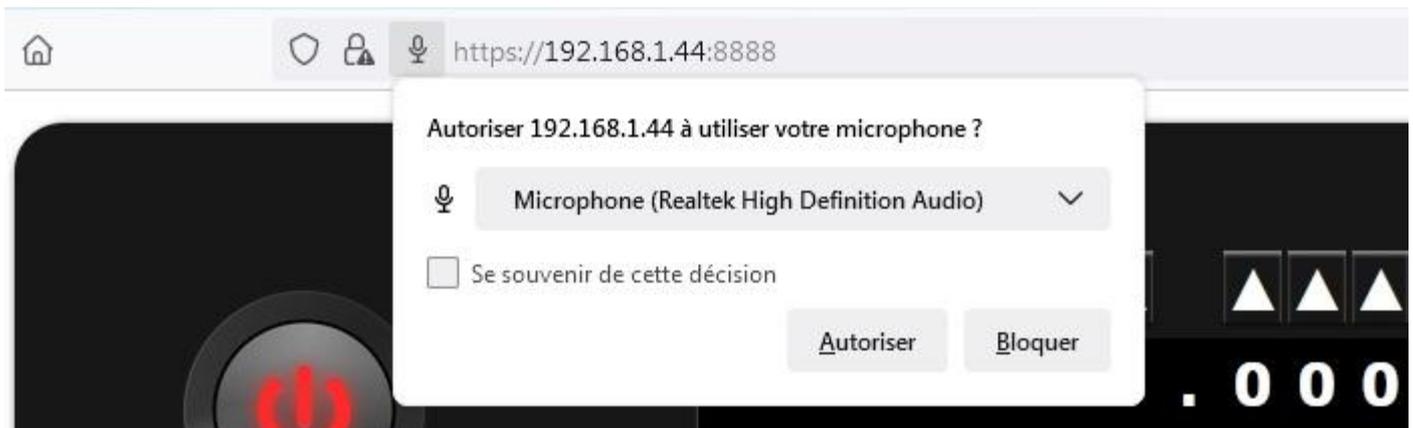
HAMLIB radio rate (samples/s):

PANADAPTER frequency correction (ppm):

PANADAPTER initial gain:

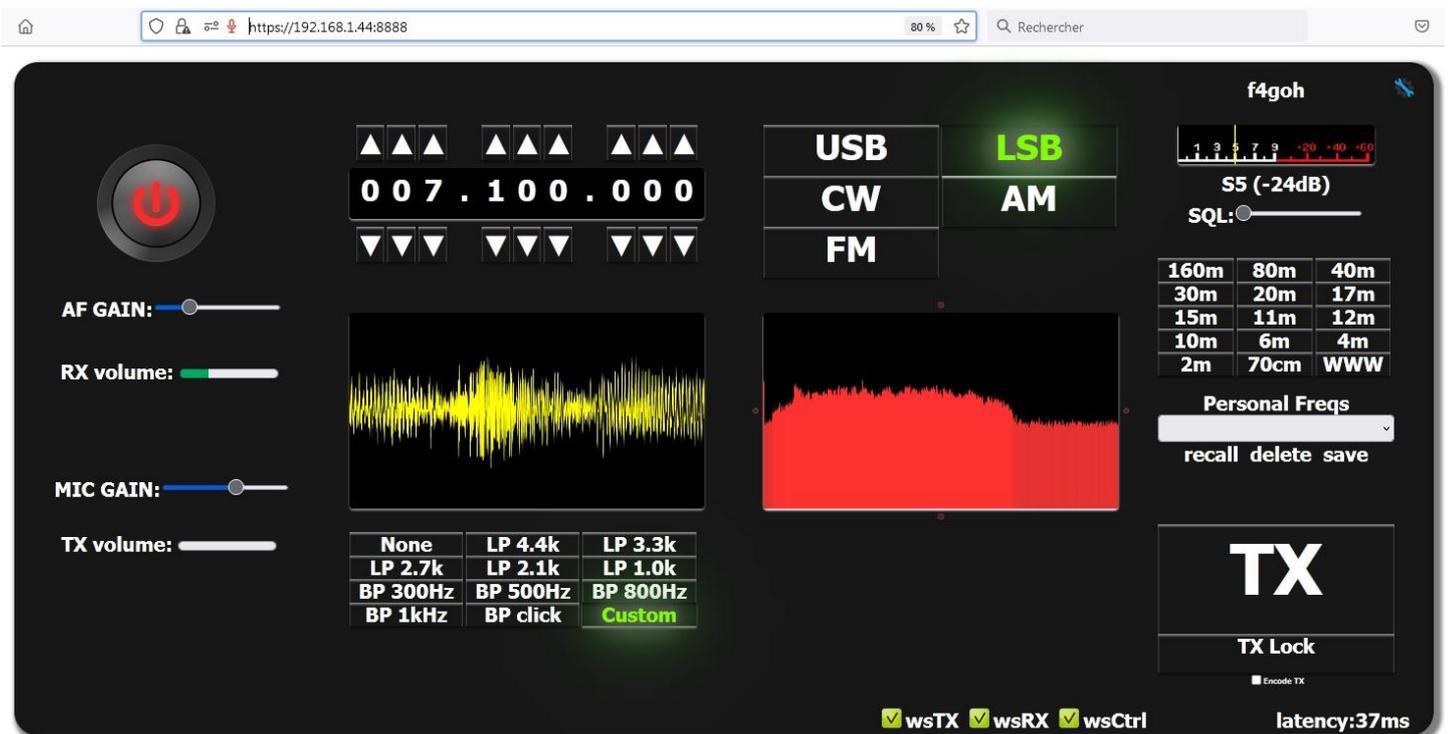
PANADAPTER windowing:

Ne pas oublier de brancher un micro casque sur le PC afin d'autoriser l'utilisation du microphone ou autre source audio.



Enfin l'interface tant attendue

Cliquer sur le bouton de mise en marche. Le son, issu du transceiver, doit être entendu dans le casque. Dans le cas contraire, cliquer sur « la clé à molette bleue » en haut à droite pour revenir au menu de configuration.



13.4 Conclusion

Bravo Olivier (F4HTB) pour cette magnifique réalisation. L'interface graphique est très ergonomique et ravira les OM désirant piloter à distance leur station.

73 et bon remote.