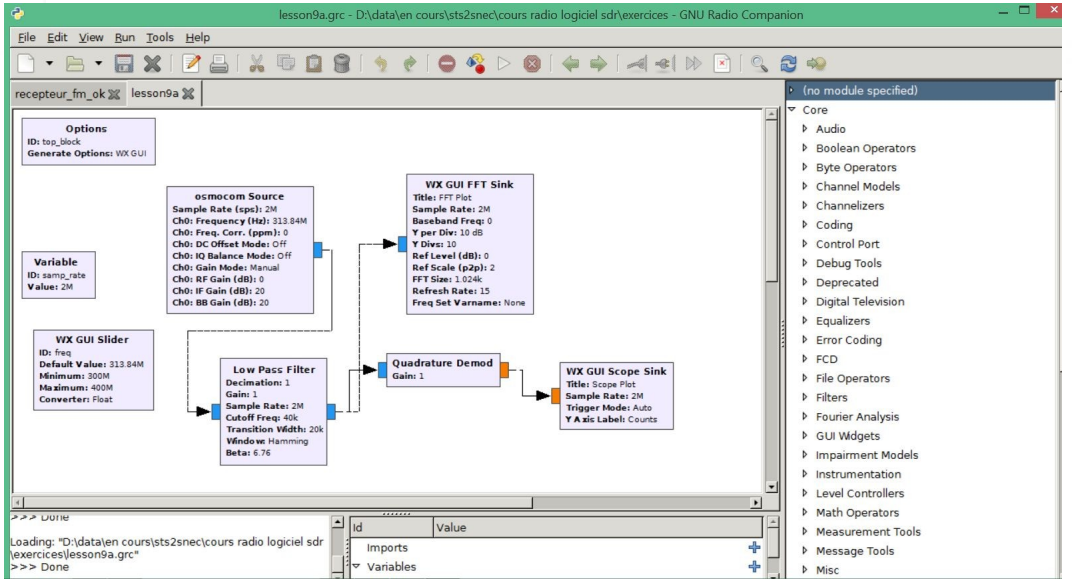
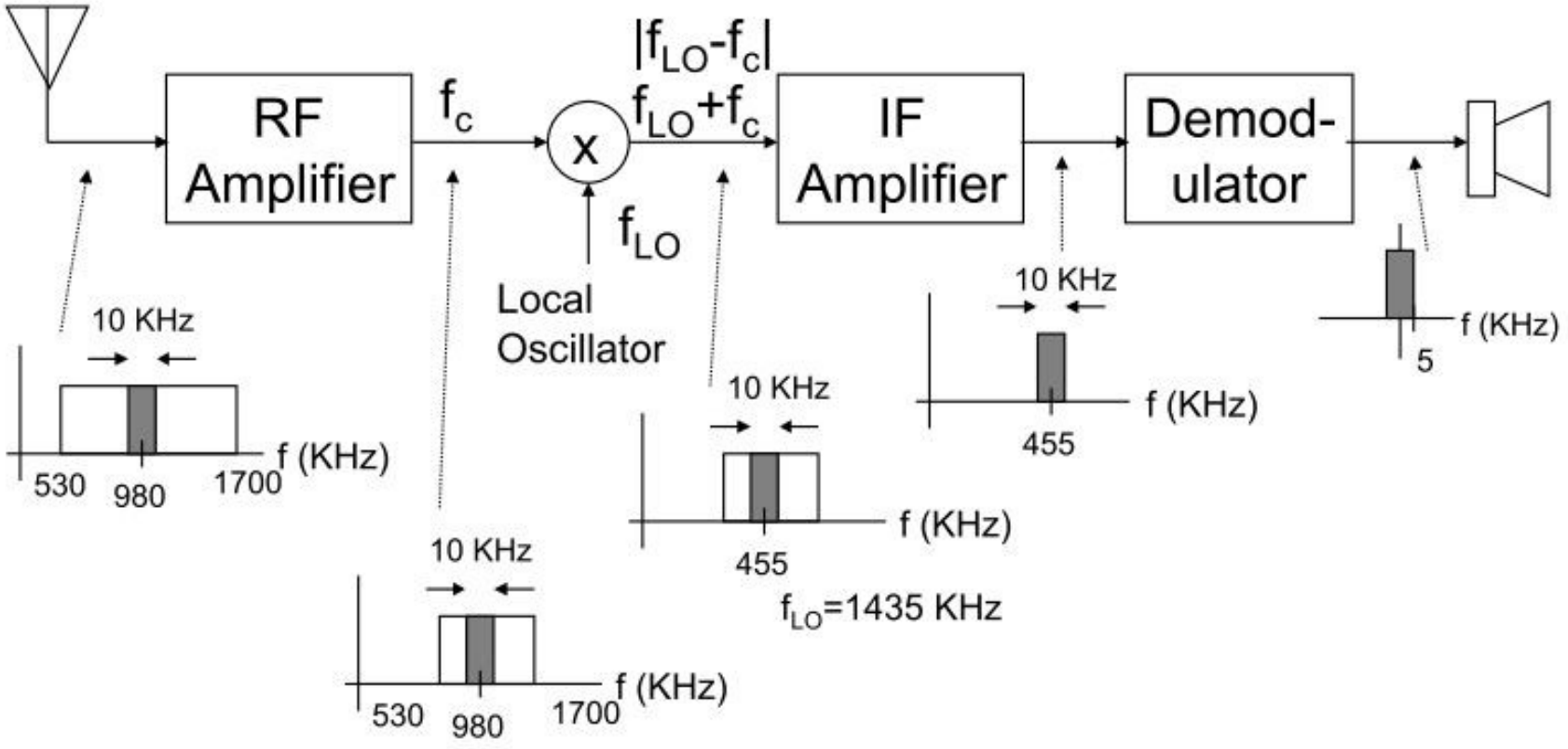
Radio logiciel
SDR + GNU Radio

- Software Defined Radio : radio définie logiciellement

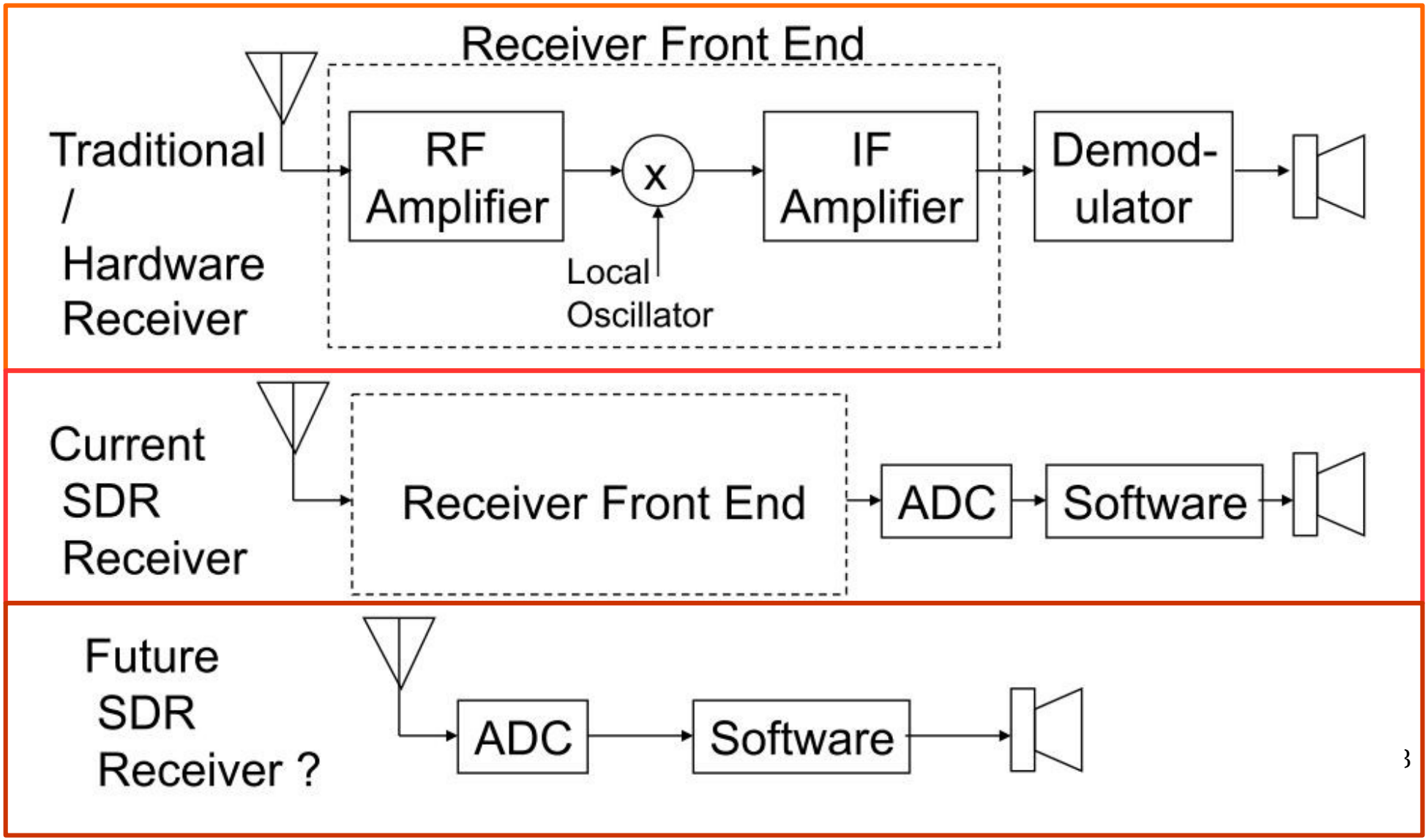


Radio logiciel
Récepteur traditionnel

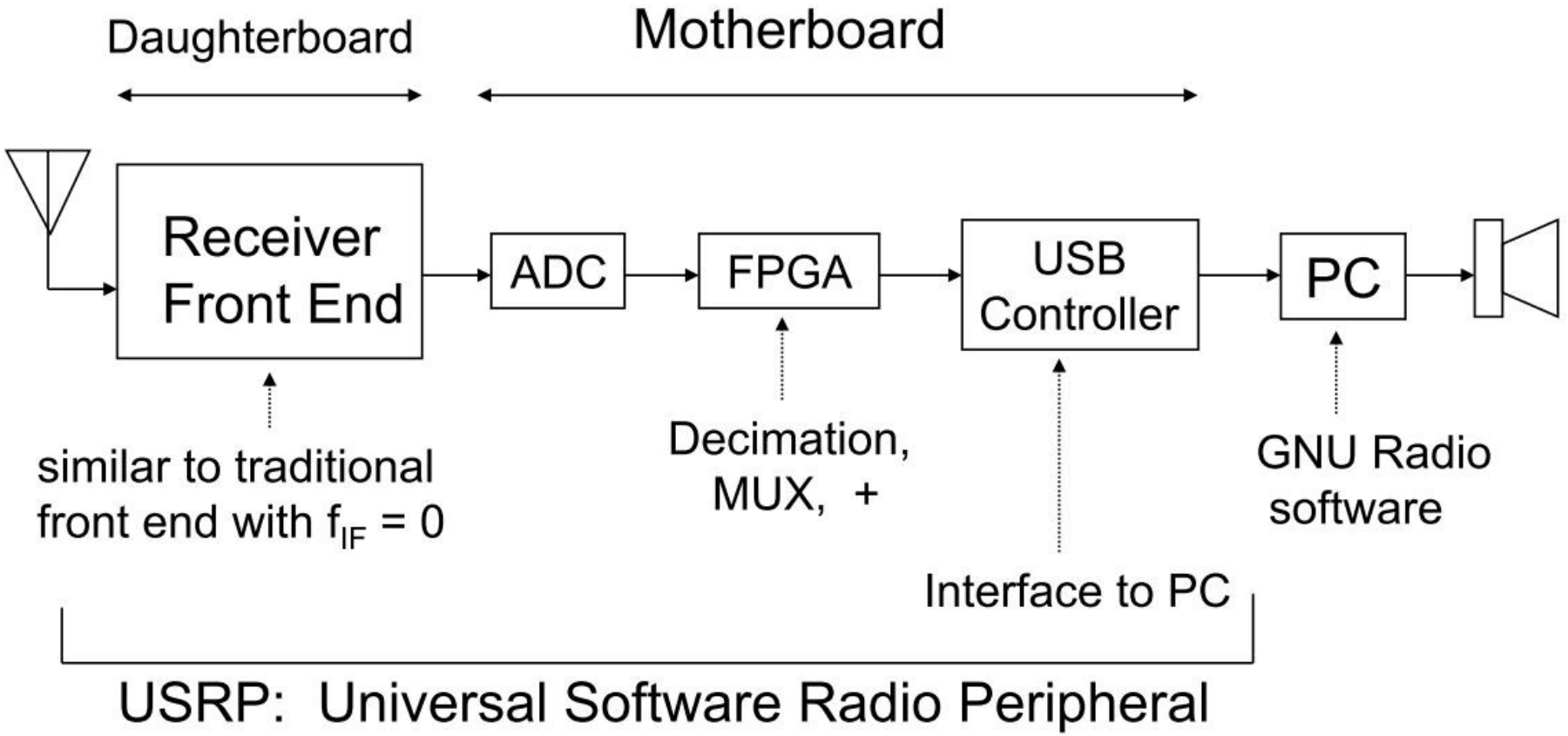


Radio logiciel

Comparaison Traditionnel / USRP



Radio logiciel
Récepteur USRP



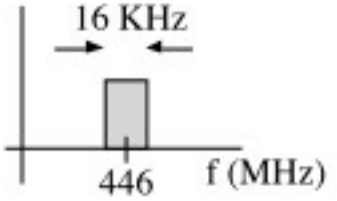
Radio logiciel
Représentation signaux en quadrature

- L'USRP fournit les composantes I et Q du signal ramené à la bande de base (F = 0)
- GNU Radio travaille sur ces composantes I et Q.

The received signal, S(t), may be represented as follows:

$$S(t) = I(t)\cos(2\pi f_c t) + Q(t)\sin(2\pi f_c t)$$

f_c = carrier frequency

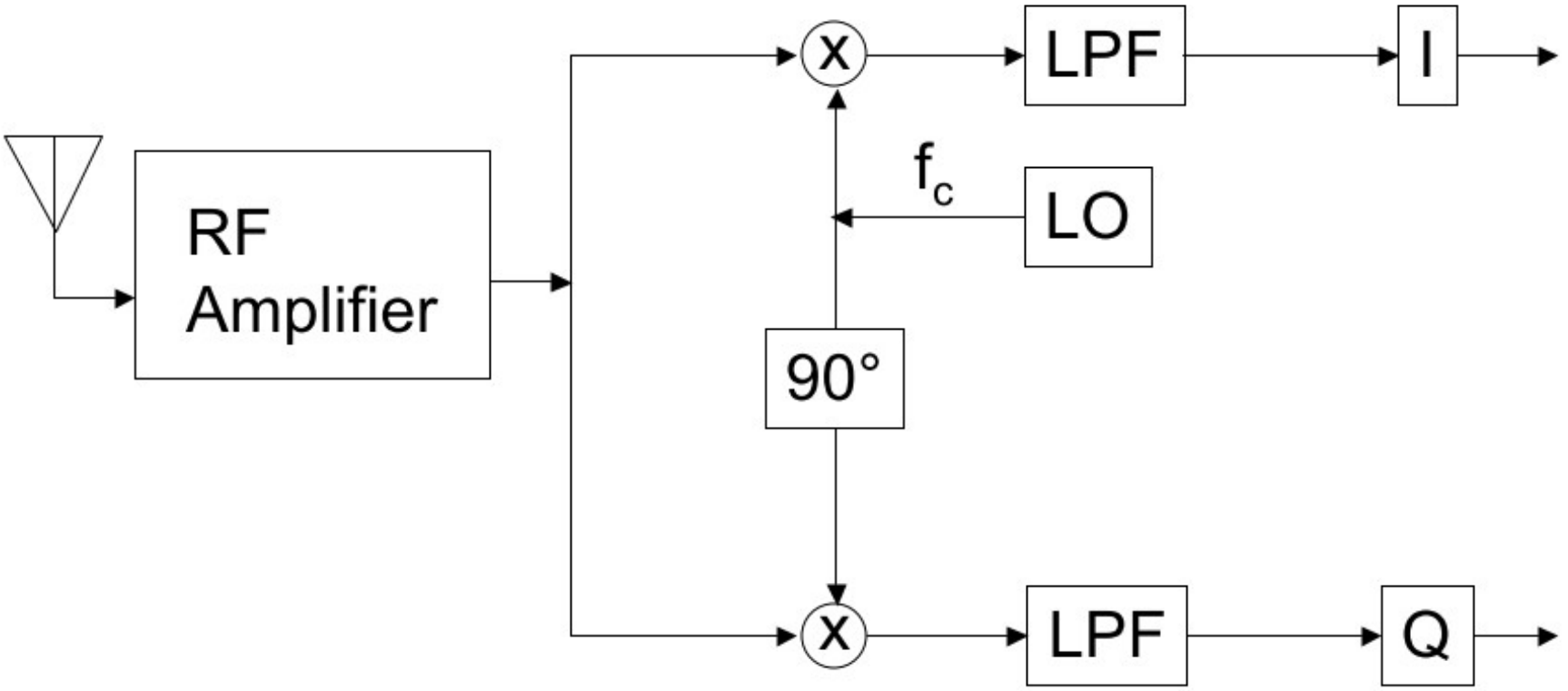


I(t) = in-phase component

Q(t) = quadrature component

Contain amplitude and phase information of baseband signal

Radio logiciel
Extraction de I et Q

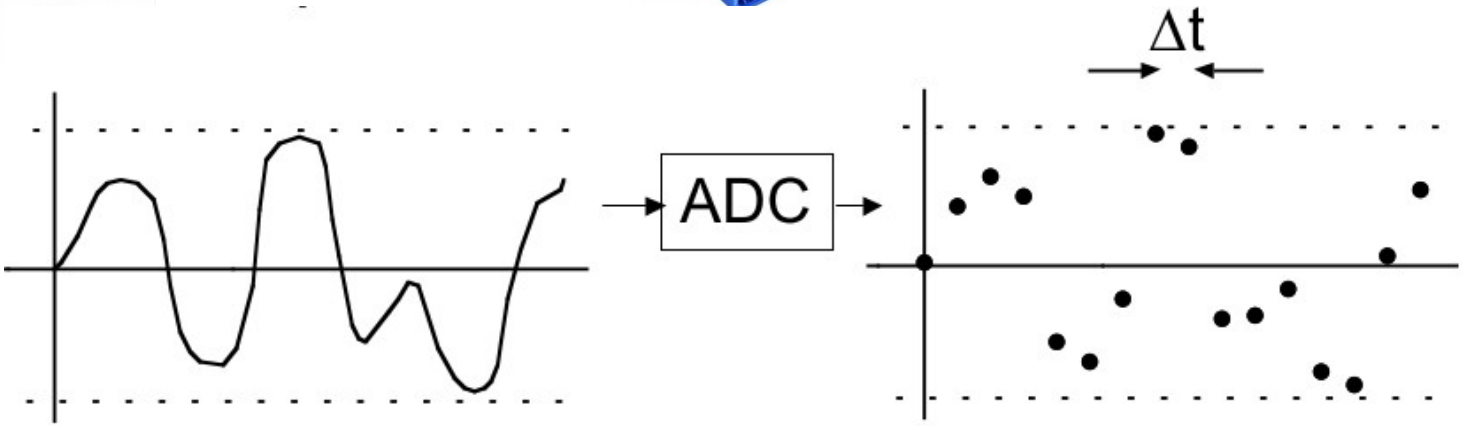


Radio logiciel
ADC caractéristiques

HackRF One
- Résolution 8 bits
- Fe = 20Msps

RTL-SDR
- Résolution 8 bits
- Fe = 2.4Msps

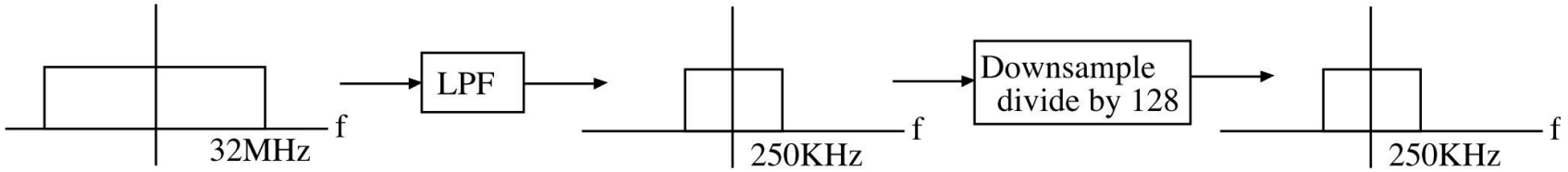
USRP Ettus
- Résolution 12 bits
- Fe = 64Msps



Radio logiciel

Décimation – interpolation : rôle

- La décimation permet de diminuer le nombre d'échantillon par seconde
- L'interpolation permet d'augmenter le nb d'échantillon par seconde



f_s = 64Msamp/sec f_s = 64Msamp/sec f_s = 500Ksamp/sec

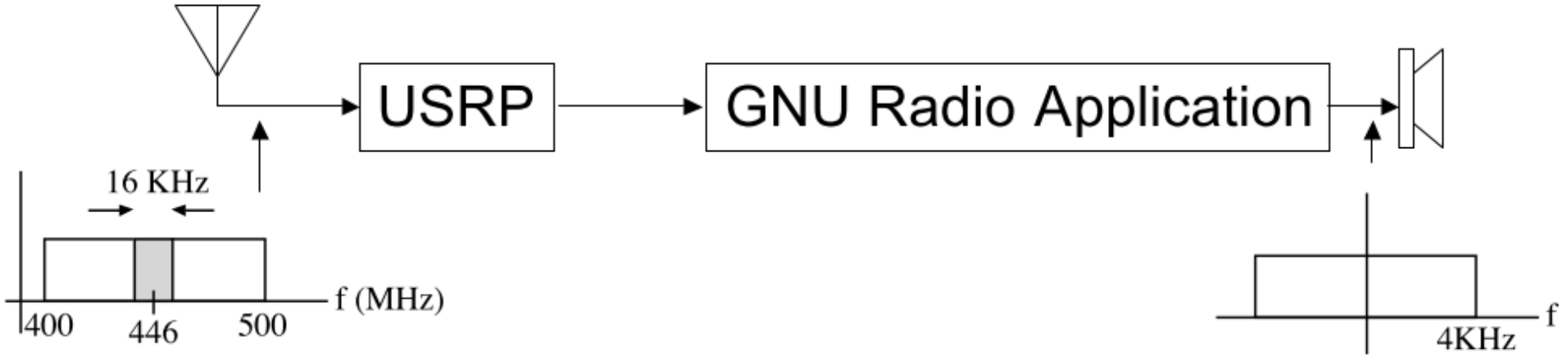
$$\frac{64M}{500K} = 128$$

- C'est le FPGA qui réalise ces calculs

Radio logiciel
Schéma d'un récepteur

- USRP : réglage fréquence oscillateur local, gain de l'ampli HF, facteur décimation…
- GNU Radio : utilise Python pour spécifier et connecter les blocs qui démodulent et décodent.

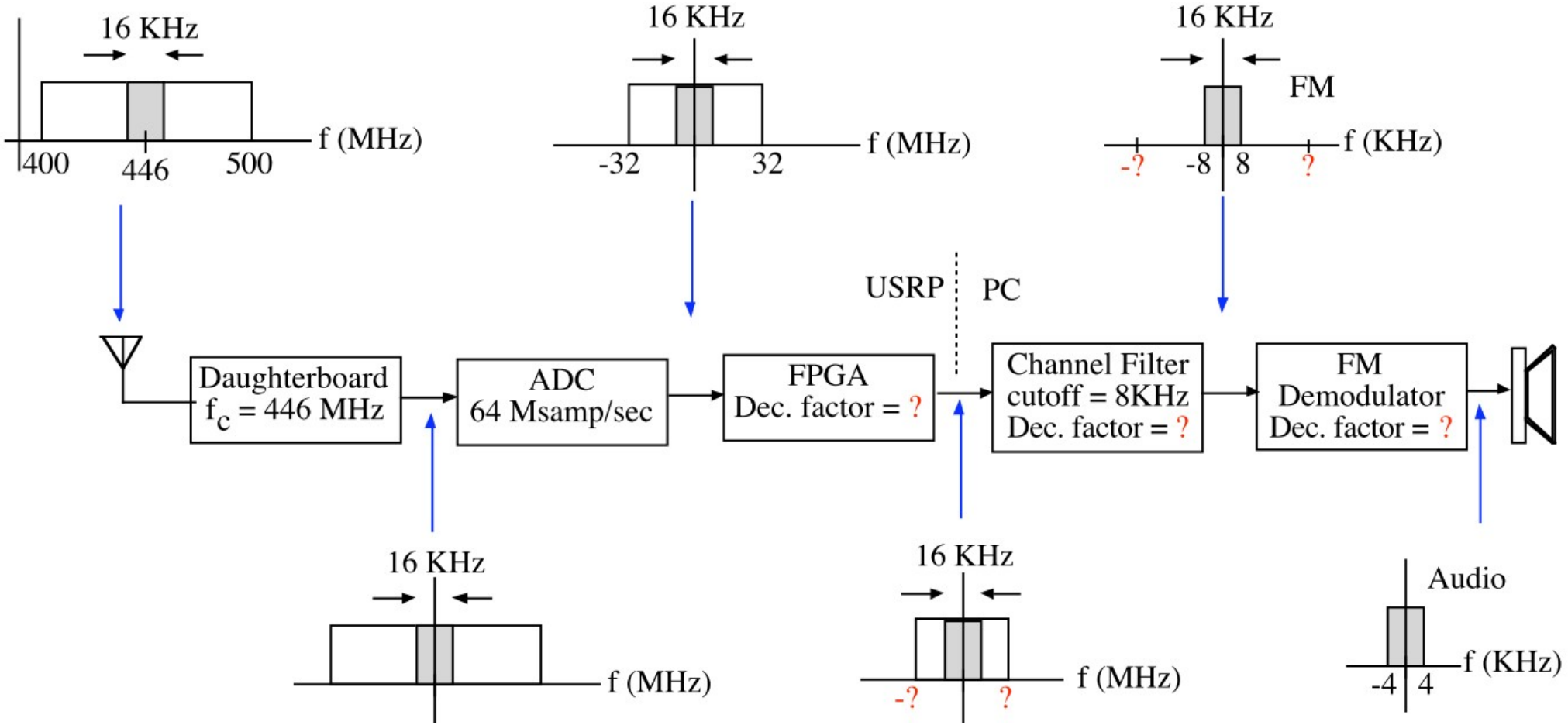
Radio logiciel
Exemple d'un récepteur NBFM 400-500 MHz



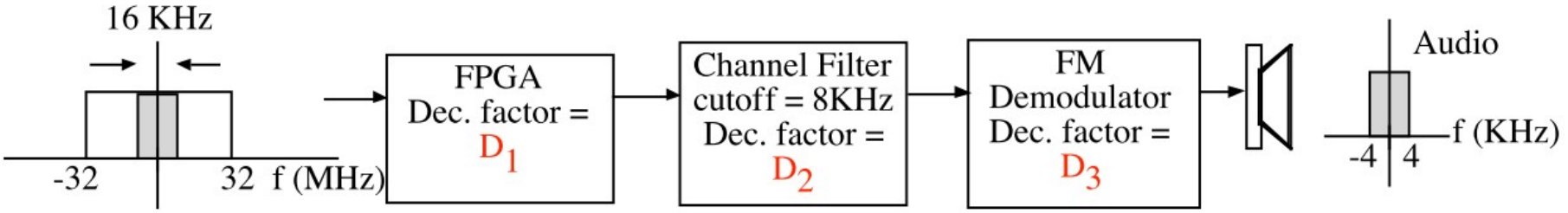
Problématique :

- recevoir un signal audio de 4KHz
- transmis en FM à 446MHz
- avec une bande étroite de 16KHz

Radio logiciel
Exemple d'un récepteur NBFM 400-500 MHz



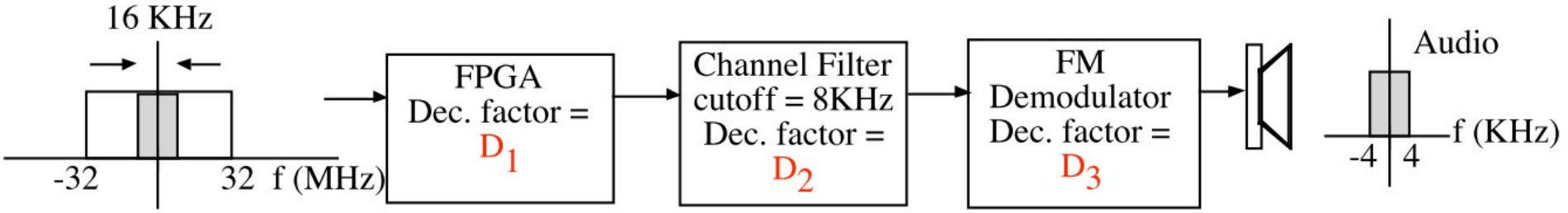
Radio logiciel
Exemple d'un récepteur NBFM 400-500 MHz



Total Decimation factor = 8000 = $D_1 D_2 D_3$

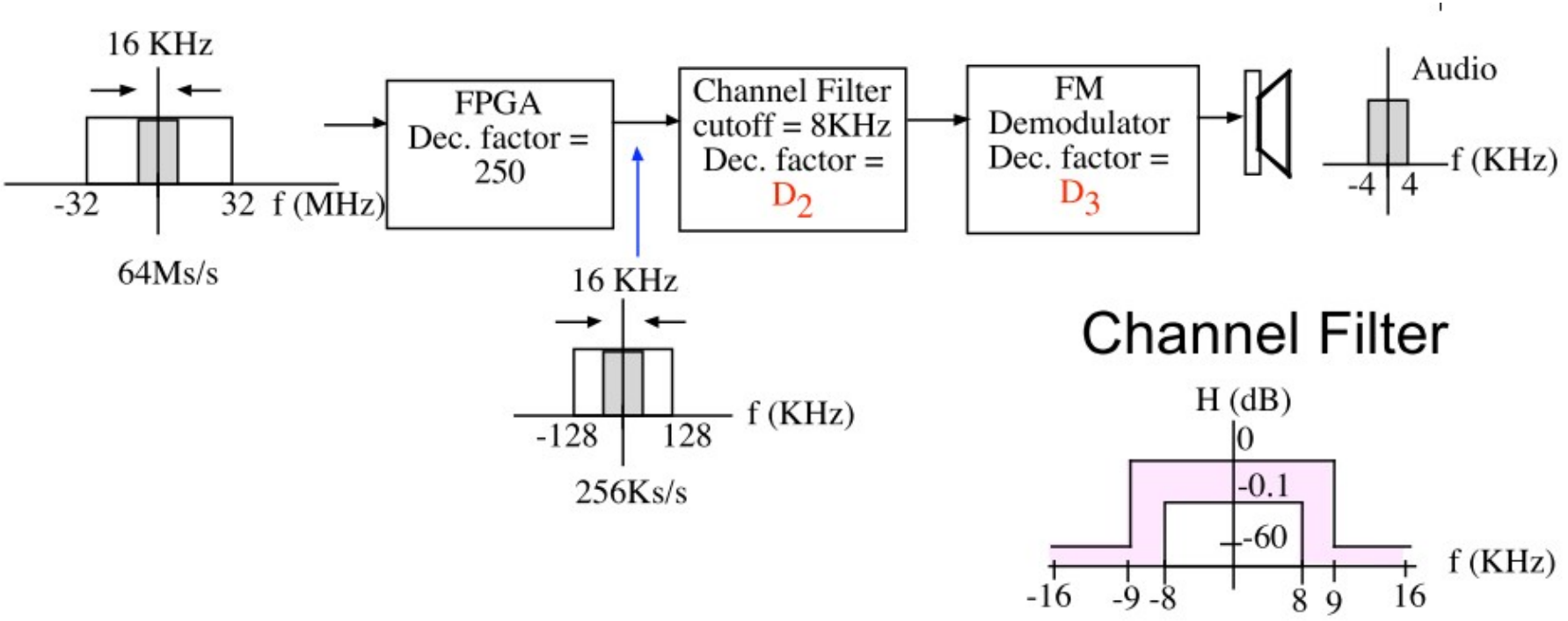
64Msamp/sec ⟶ 8Ksamp/sec

Radio logiciel
Exemple d'un récepteur NBFM 400-500 MHz



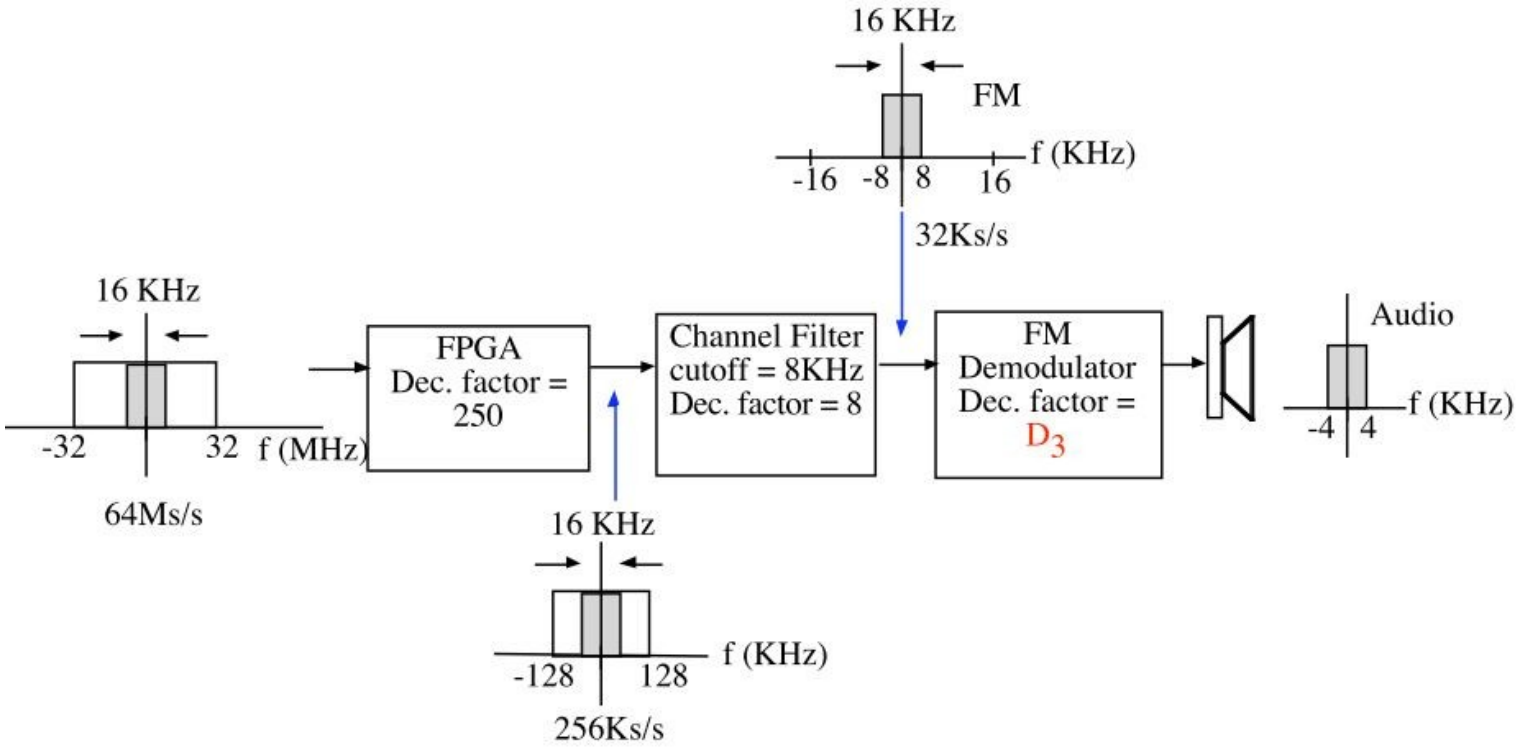
- Total Decimation factor = 8000 = $D_1 D_2 D_3$
- Maximize the decimation in FPGA
- Maximum decimation factor in FPGA = 256
- Select D_1 = 250 (factor of 8000)
- Output sample rate = 64Ms/s / 250 = 256Ks/s

Radio logiciel
Exemple : filtre spécification



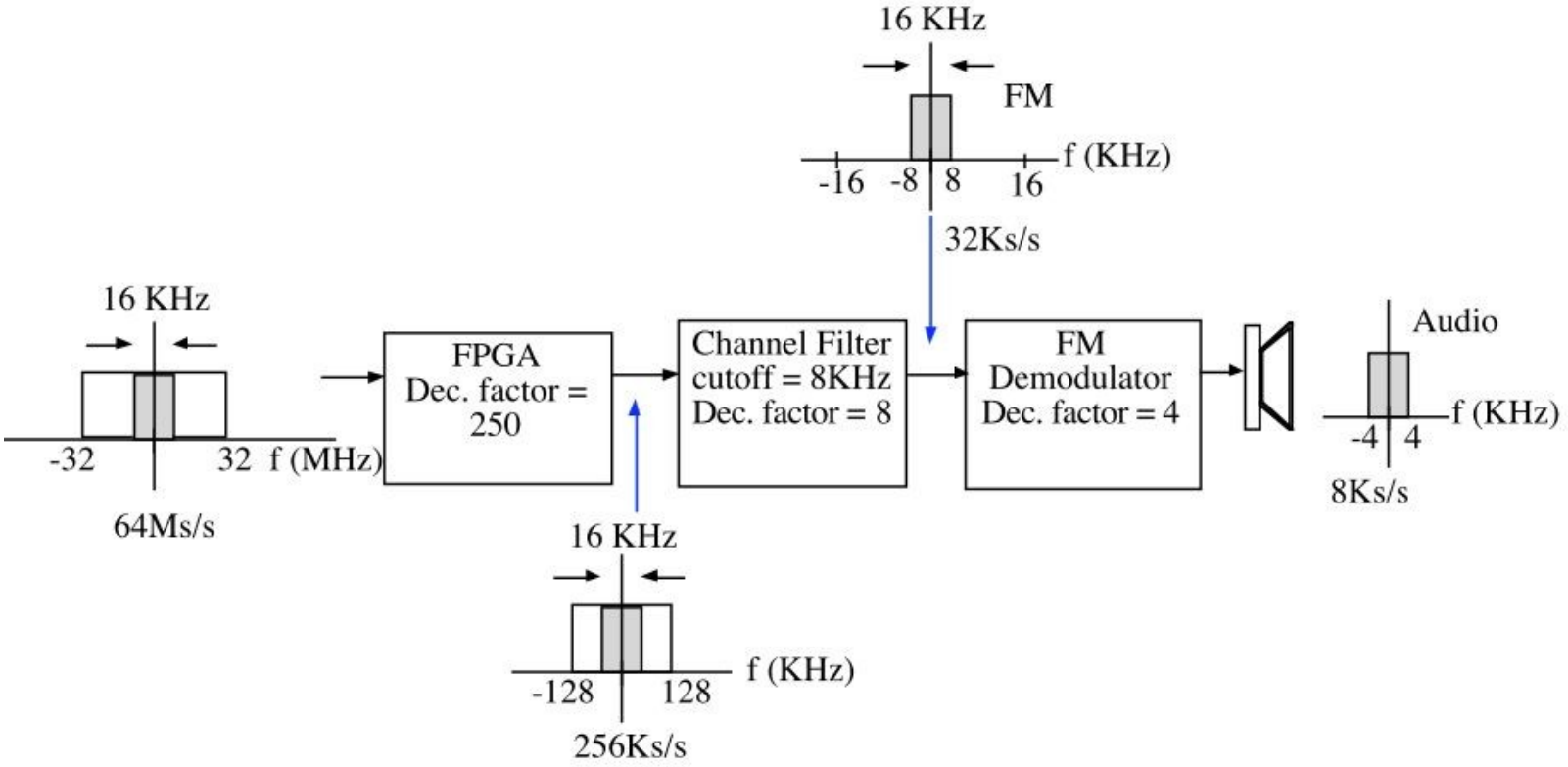
- Maximum frequency = 16 KHz → Reduce sample rate to 32 Ks/s
- 256Ks/s / 32Ks/s → $D_2 = 8$

Radio logiciel
Exemple : FM démodulation



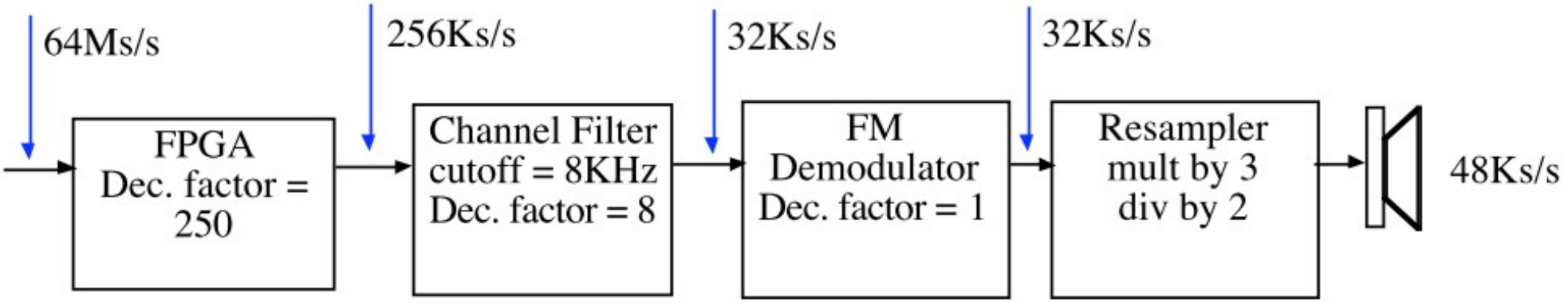
- Maximum frequency = 4 KHz → Reduce sample rate to 8 Ks/s
- 32Ks/s / 8Ks/s ⟶ $D_3 = 4$
- FM Demodulator block “extracts” audio signal from FM waveform by operating on I and Q

Radio logiciel
Exemple : schéma final



- Total decimation ratio = 250*8*4 = 8000
- Problem: The audio card requires an input sample rate ≥ 44.1 Ks/s
- Solution: Use a Resampler to increase the output sample rate

Radio logiciel
Exemple d'un récepteur NBFM 400-500 MHz complet

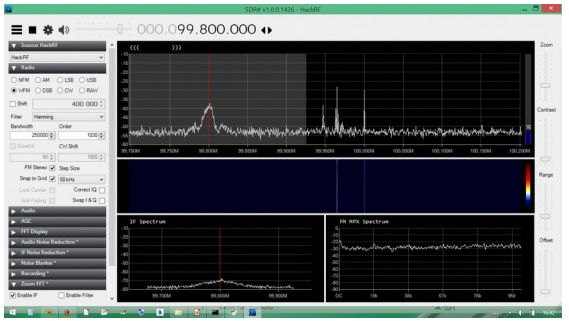


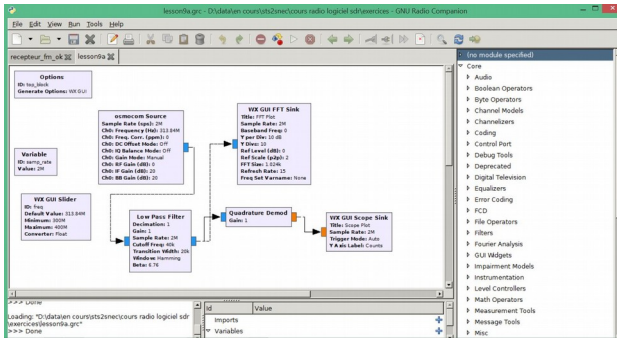
- Audio Card requires a sample rate ≥ 44.1 Ks/sec. Use 48 Ks/sec.
- Modify FM Demodulator to have a decimation factor of 1 (no change)
- Increase the sample rate to 48 Ks/sec with Resampler (x 3/2)

Radio logiciel
Installation

- Installation DRIVERS Hardware :
  - ✔ Clé TNT RTLSDR
  - ✔ HackRFOne

- Installation des logiciels :
  - ✔ SDRSharp
  - ✔ GNU Radio





GNU Radio
Installation

- Télécharger le fichier "gnuradio3.7.10.1_win64.msi" sur stssnsb.free.fr
- Installer le logiciel
- Installer le driver pour la clé USB et/ou HackRF One

OU

- Live DVD directement sous Linux sans installation

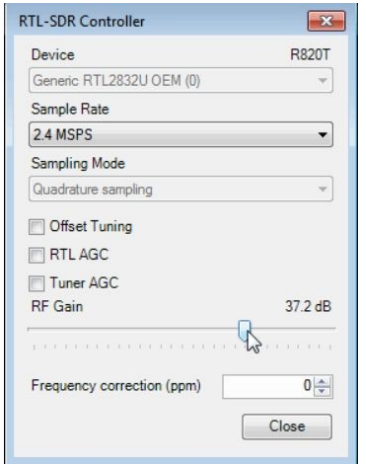
Radio logiciel
Drivers Windows

- Télécharger Zadig.exe sur http://zadig.akeo.ie(http://www.rtl-sdr.com/tag/zadig/)

- Lancer l'exécutable
- Cocher "List all devices"
- Choisir la clé TNT ou HackRFone
- Vérification : Gestionnaire de périphérique (aucun pb doit apparaitre)

Radio logiciel
Test avec SDRSharp

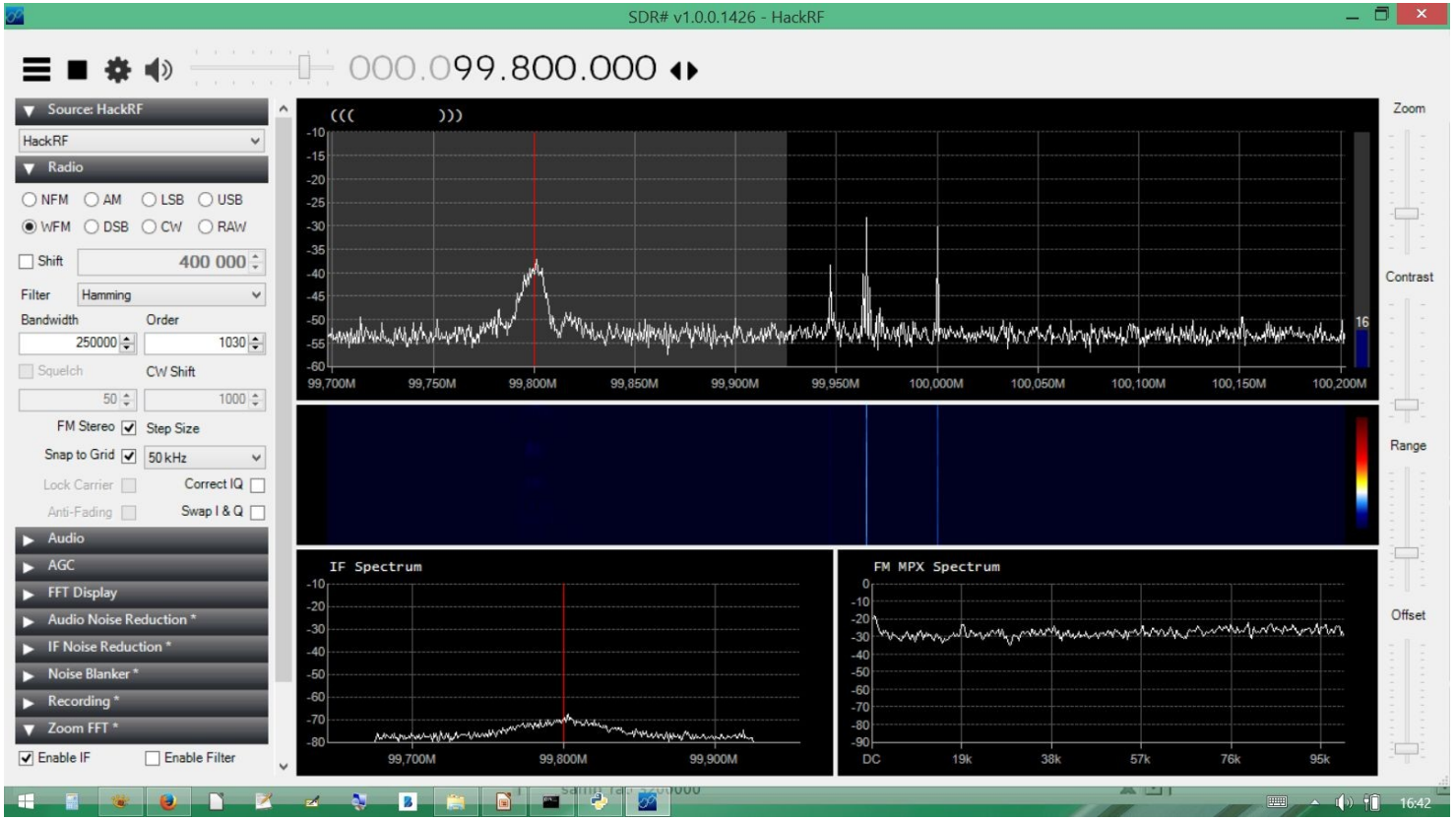
- Télécharger SDRSharp
- Choisir le hard à tester : RTL-SDR ou HackRF
- Choisir le "Device"
- Choisir la largeur de bande à analyser
- Régler le gain
- Lancer la lecture...



Radio logiciel
Test avec SDRSharp

Si test OK **alors** continuer

Sinon voir driver



GNU Radio
Premier diagramme

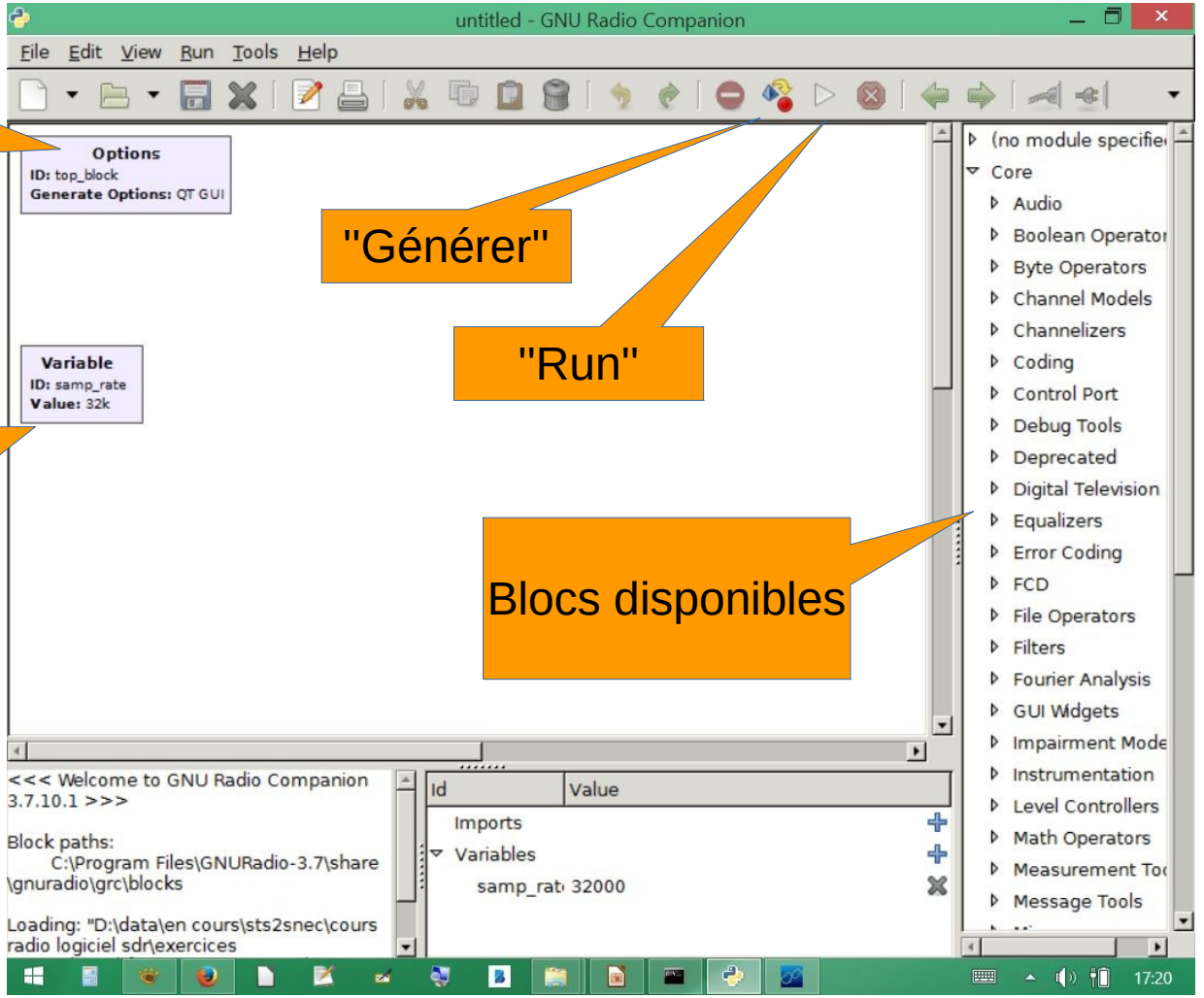
Type de widget : Qt ou Wt

Fréquence échantillonnage

"Générer"

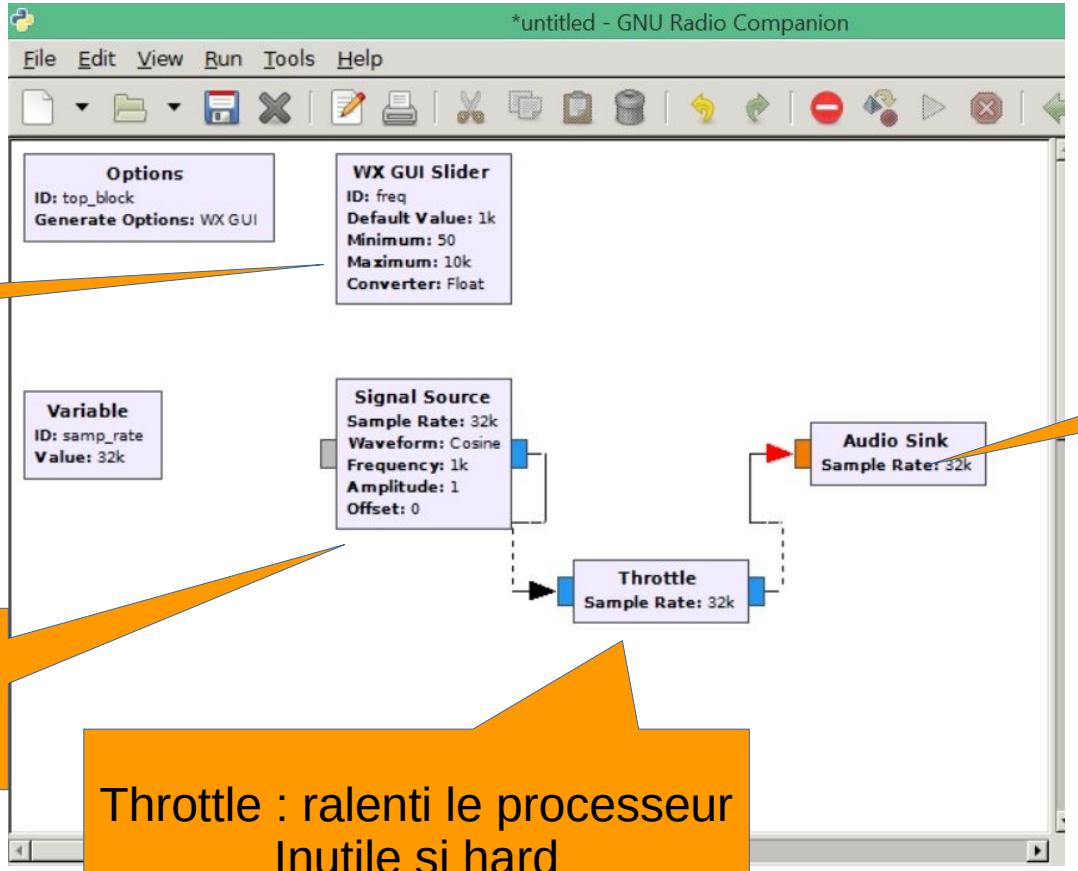
"Run"

Blocs disponibles



GNU Radio
Premier diagramme

- Générateur de signaux audio :



Slider : curseur

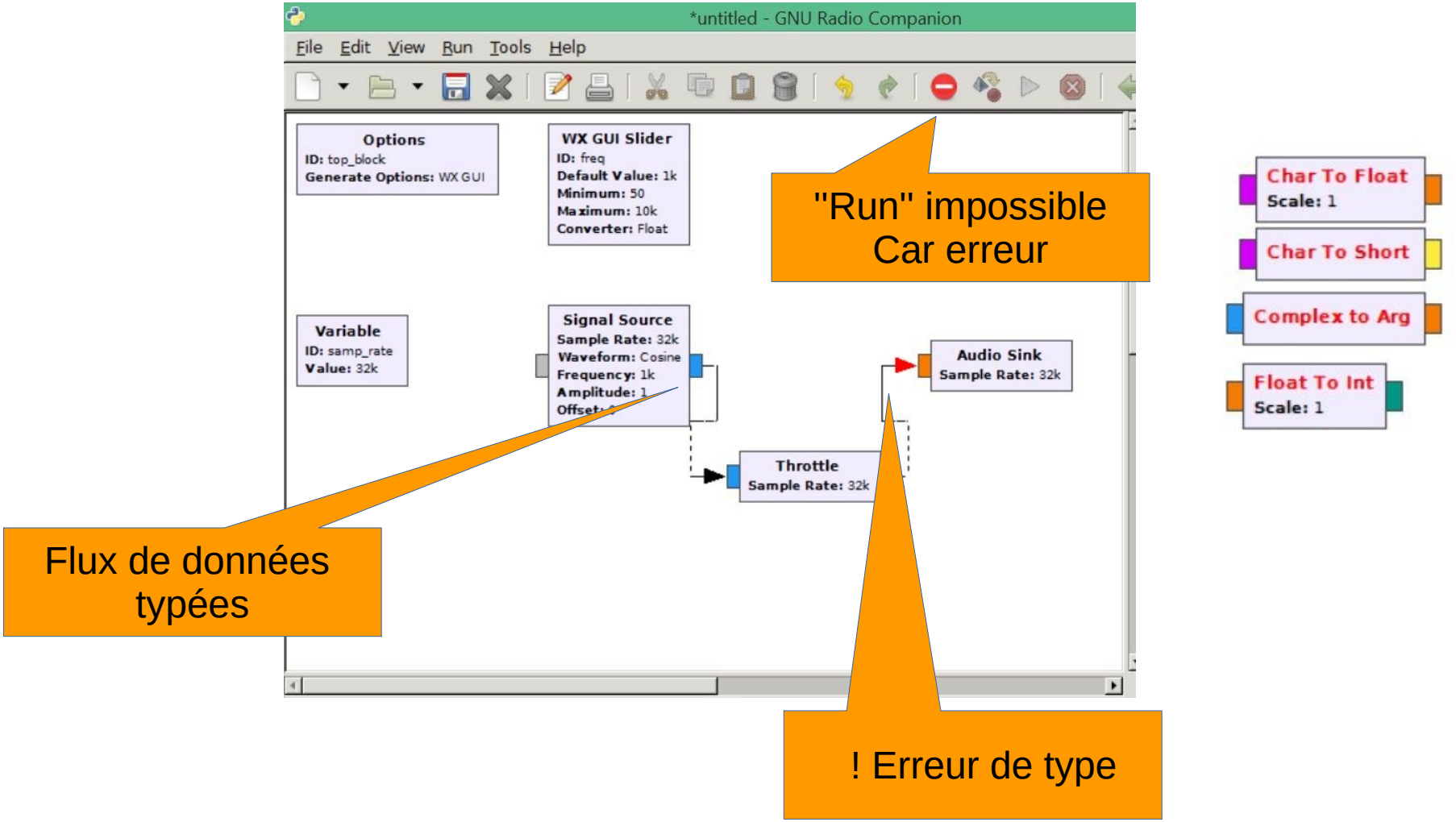
Sortie

GBF :
Contrôlé par slider
Sur frequency

Throttle : ralenti le processeur
Inutile si hard
de branché

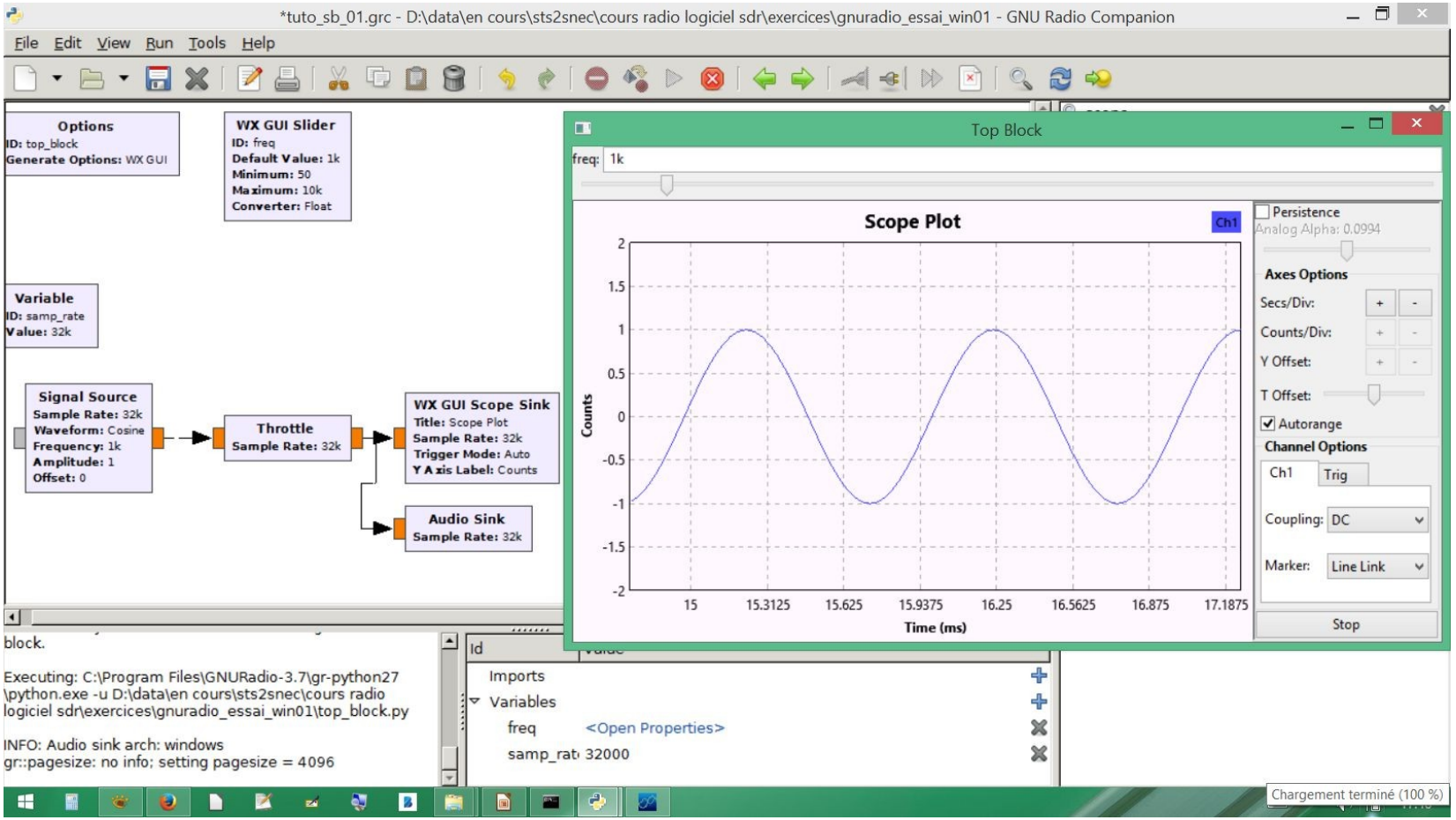
GNU Radio
Premier diagramme

- Générateur de signaux audio :



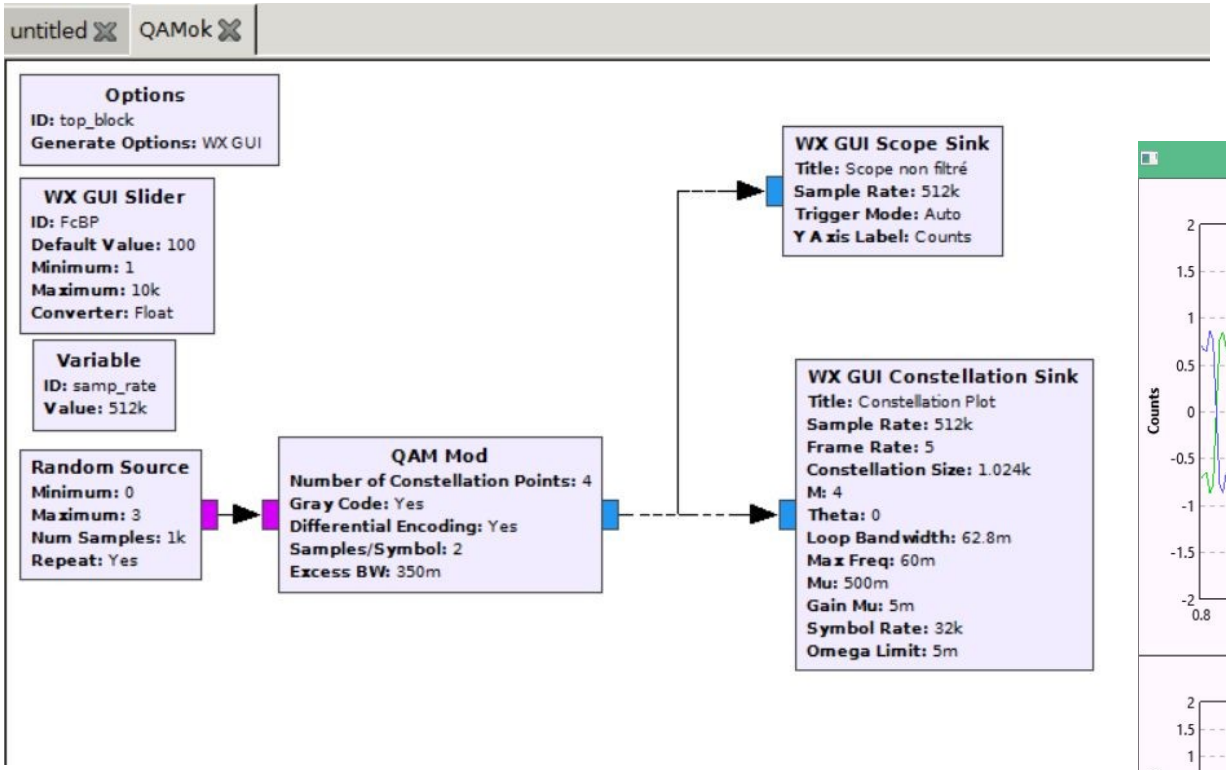
GNU Radio
Premier diagramme

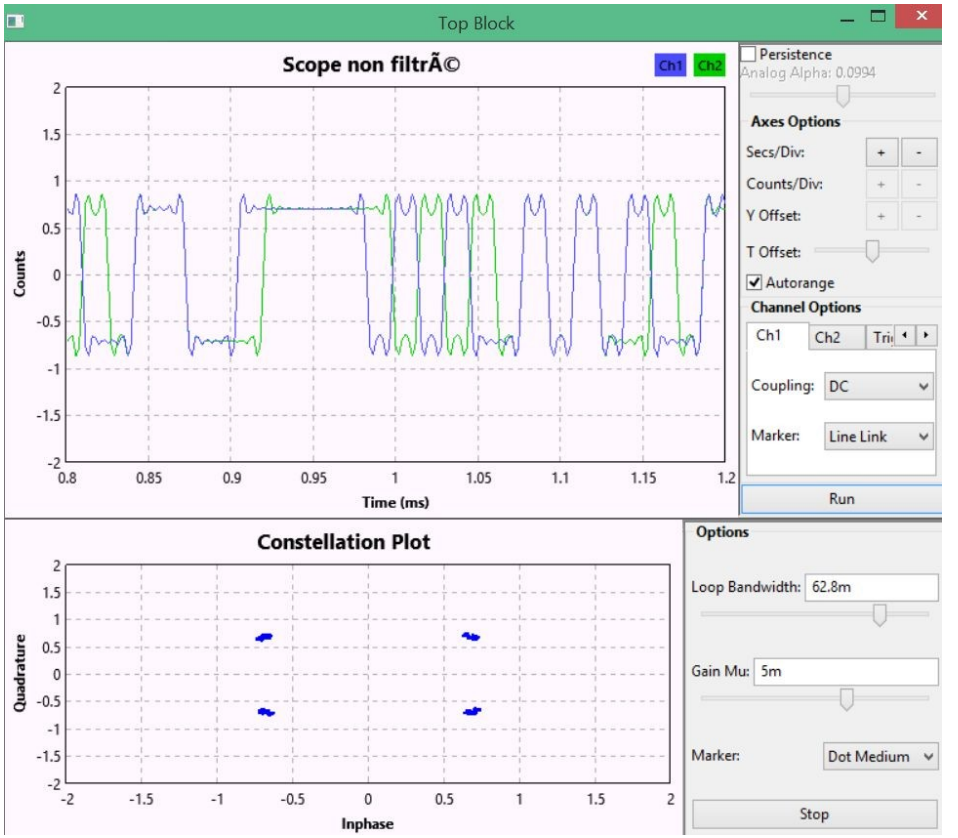
- Générateur de signaux audio :



GNU Radio
QAM simple

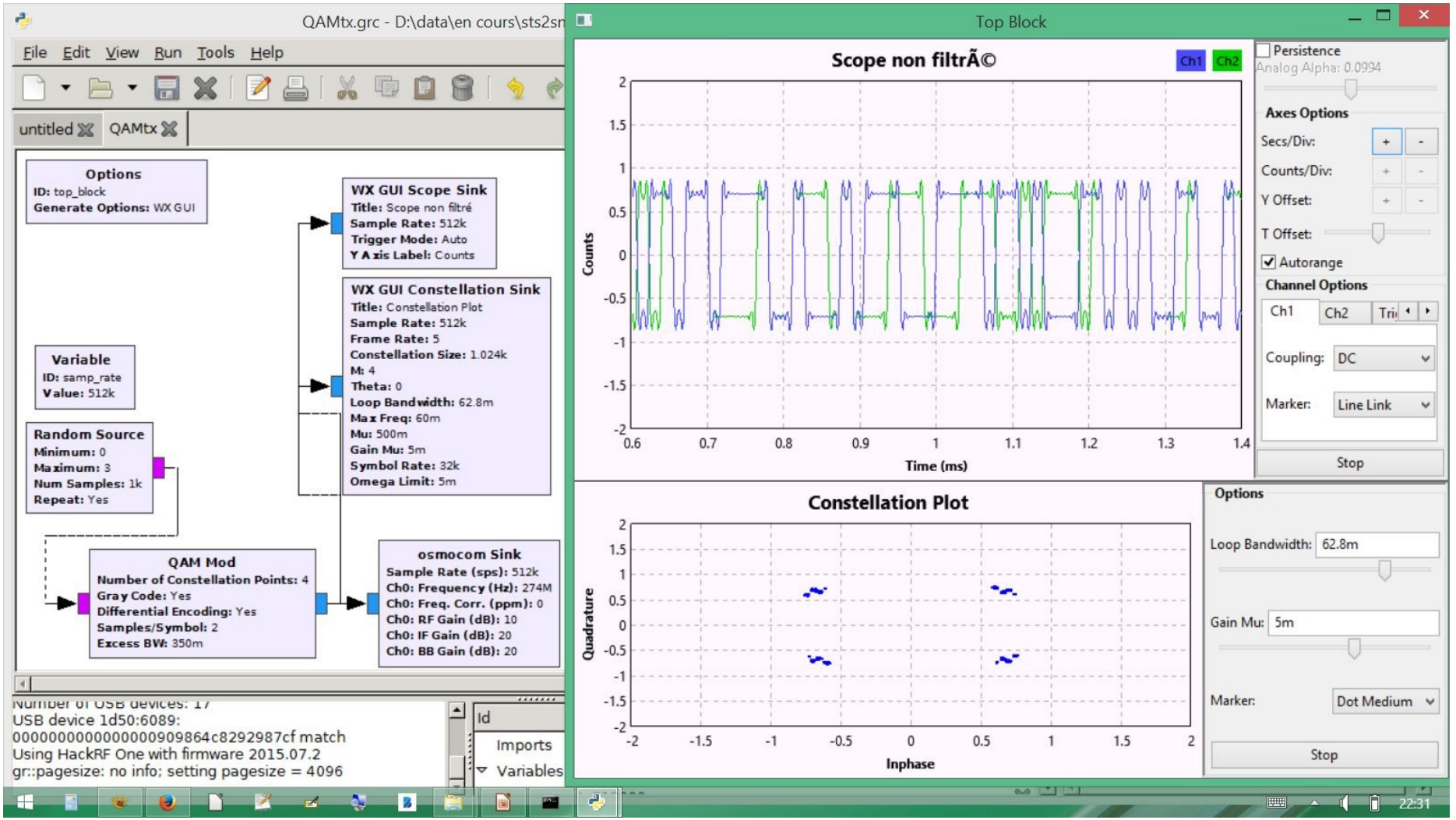
- Simulation :





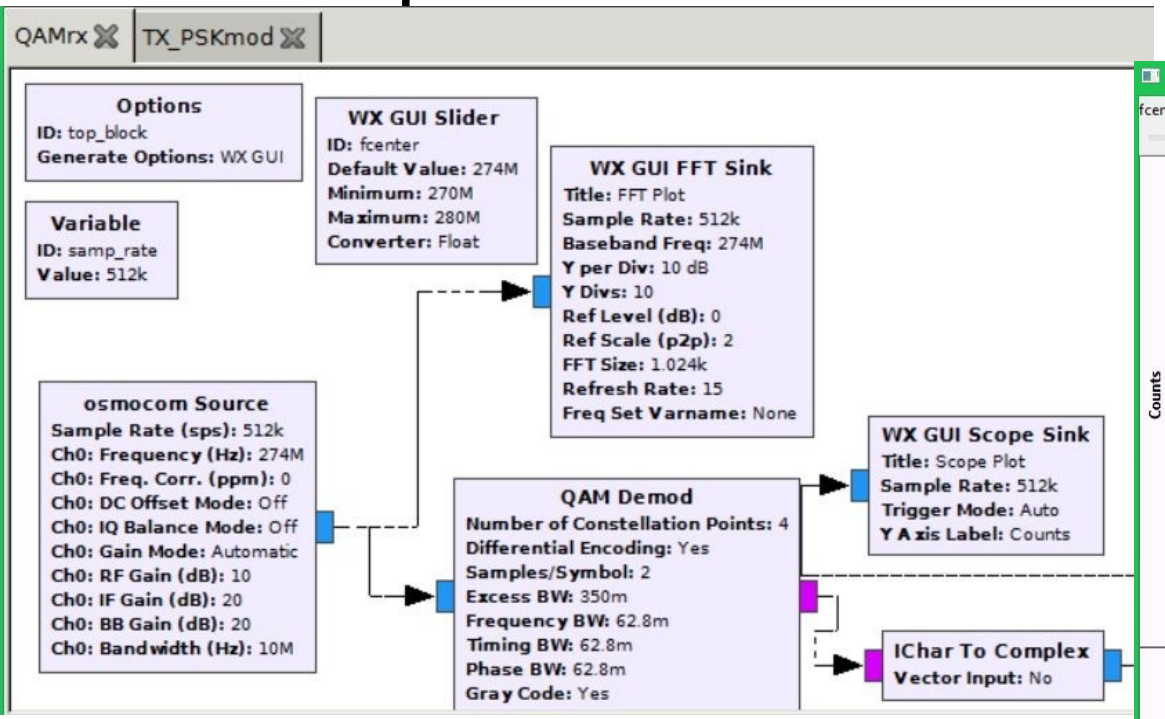
GNU Radio
QAM Tx + HackRF

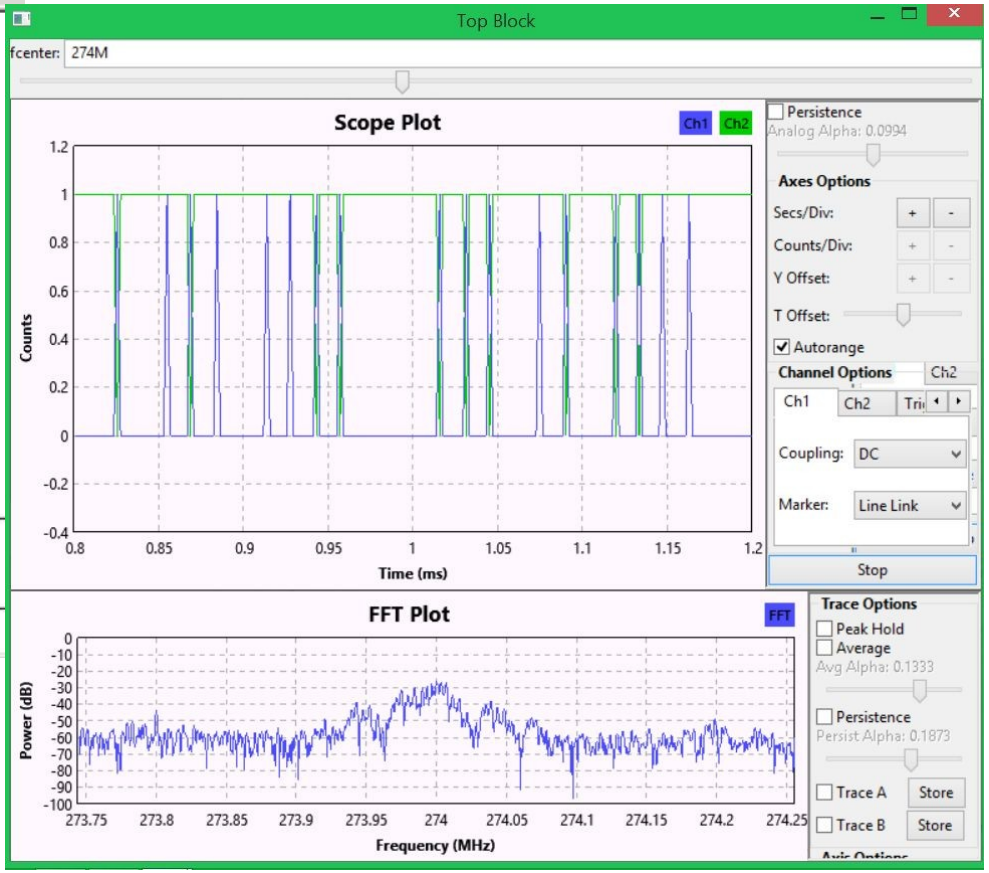
- Émission de données en QAM :



GNU Radio
QAM RX + RTL-SDR

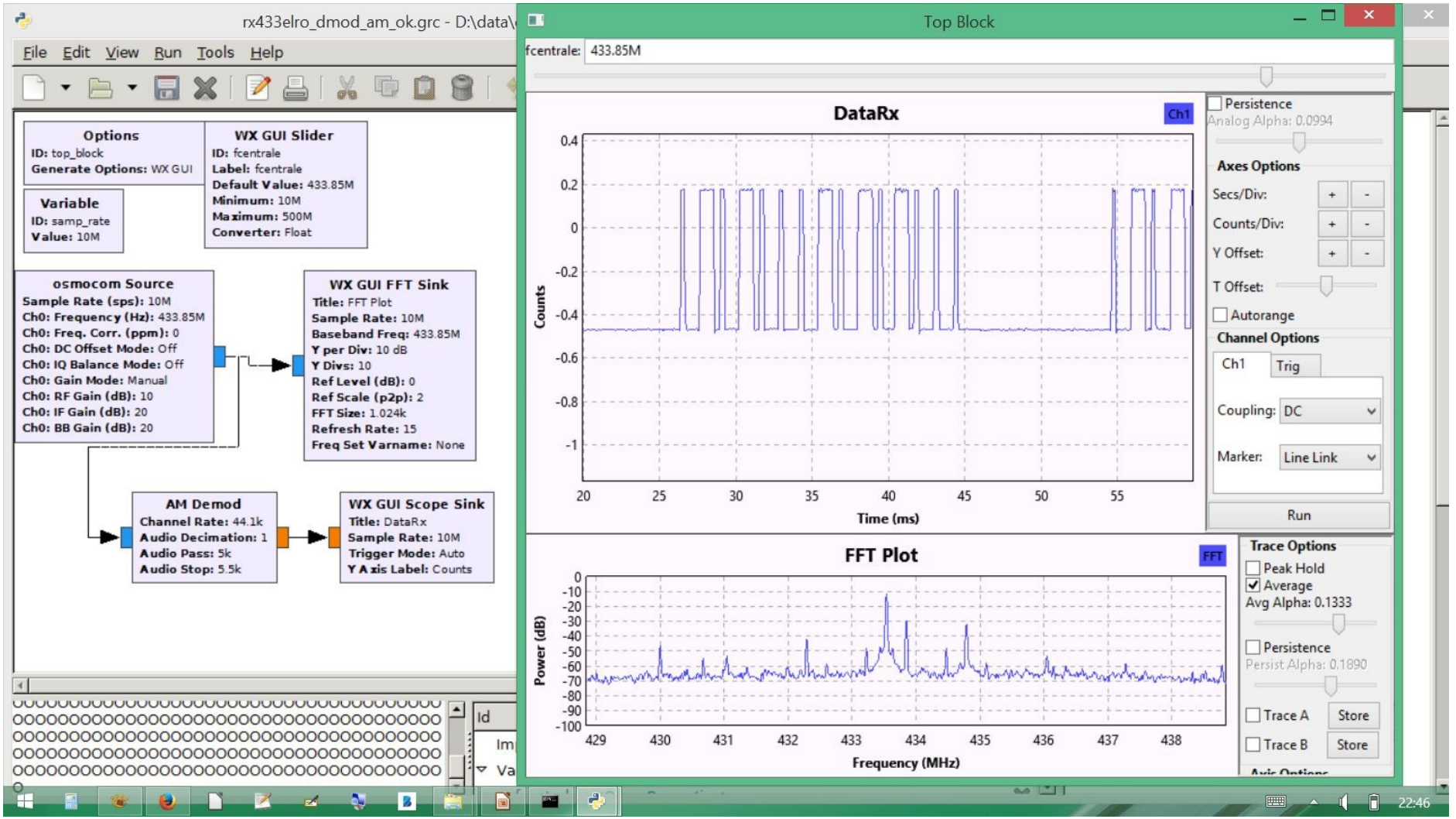
- Réception de données en QAM :





GNU Radio
Récepteur AM

- Réception du ELRO 433MHz en AM avec HACKRF



Radio logiciel
Références...

- Stssnsb.free.fr – rubrique : télécharger/sn2/sdr
- https://wiki.gnuradio.org/index.php/Main_Page
- https://wiki.gnuradio.org/index.php/GNURadioCompanion
- http://www.rtl-sdr.com/tag/zadig/
- http://www.rtl-sdr.com/rtl-sdr-quick-start-guide/
- https://greatscottgadgets.com/hackrf/
- https://www.passion-radio.com/fr/
- http://www.ece.uvic.ca/~elec350/grc_doc/ar01s12s07.html
- https://www.csun.edu/~skatz/katzpage/sdr_project/sdr/

MERCI