
USRP Episode 1 :
Smoke Gets in Your Eyes
Présenté par Sébastien Dudek (FlUxIuS)

- USRP??? Sorry, I don't speak chinese

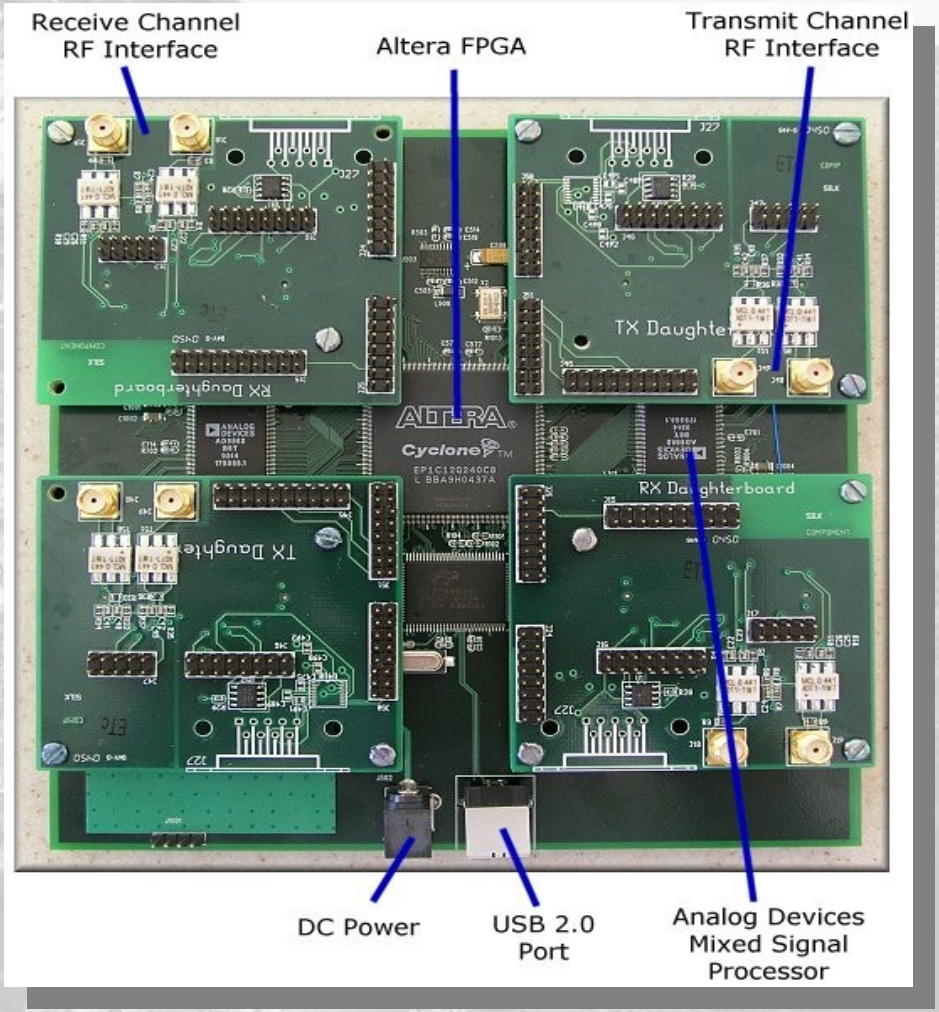
USRP : Culture générale

- USRP => Universal Software Radio Peripheral
- Se prononce : usurp (en)
- Crée par Matt Ettus (d'où ettus.com)
- Périphérique flexible et « peu » coûteux pour de la radio logicielle (SDR : « Software-defined Radio »)
- Se compose : convertisseurs ADC/DAC, un FPGA, un contrôleur USB 2.0 ou Ethernet GB et une large gamme de circuits frontaux RF.

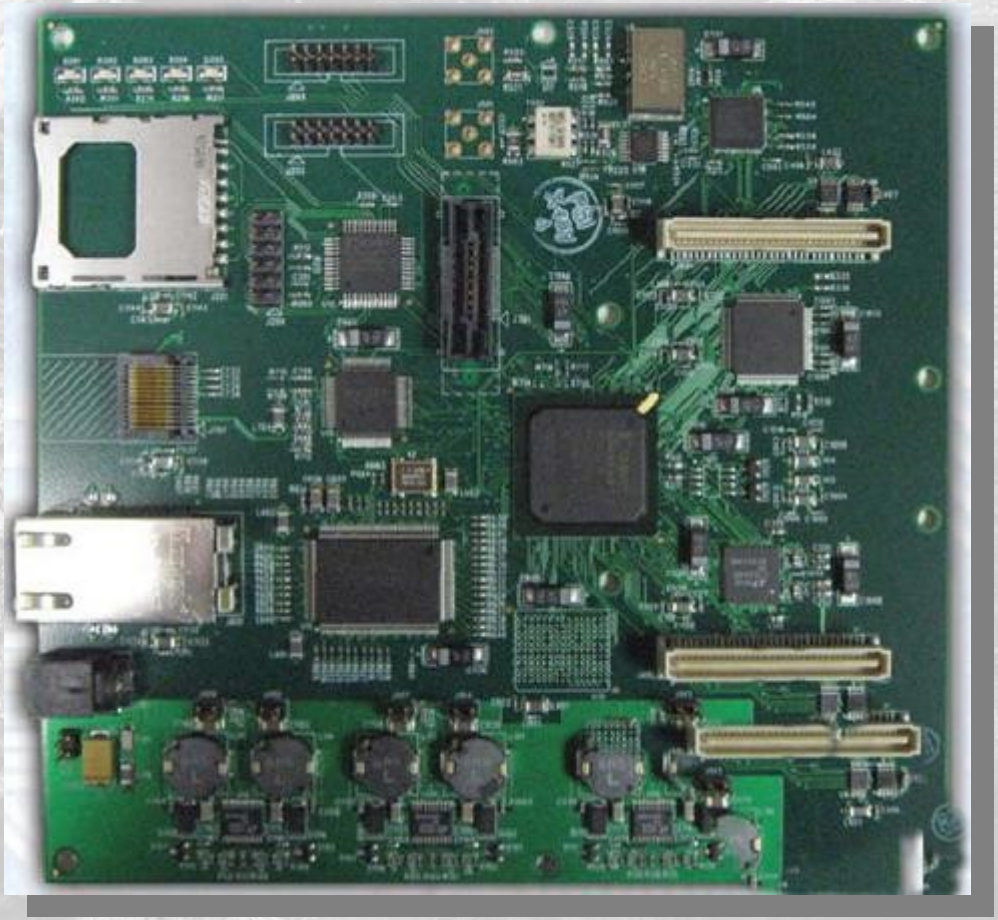
USRP : Les différents modèles

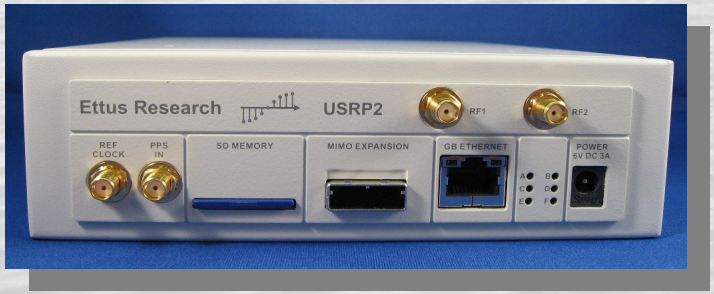
	USRP1	USRP2
Interface	USB 2.0 (32 MB/s half duplex)	Gigabit Ethernet (1000 MBit/s)
FPGA	Altera EP1C12	Xilinx Spartan 3 2000
Bande passante RF vers/depuis l'hôte	8 MHz @ 16bits	25 MHz @ 16bits
Coût	$700	$1400
Échantillonnages ADC	12-bit, 64 MS/s	14-bit, 100 MS/s
Échantillonnages DAC	14-bit, 128 MS/s	16-bit, 400 MS/s
Capacité pour les cartes filles	2 TX, 2 RX	1 TX, 1 RX
SRAM	NON	1 Megabyte
Énergie	6V, 3A	6V, 3A

USRP : Version 1

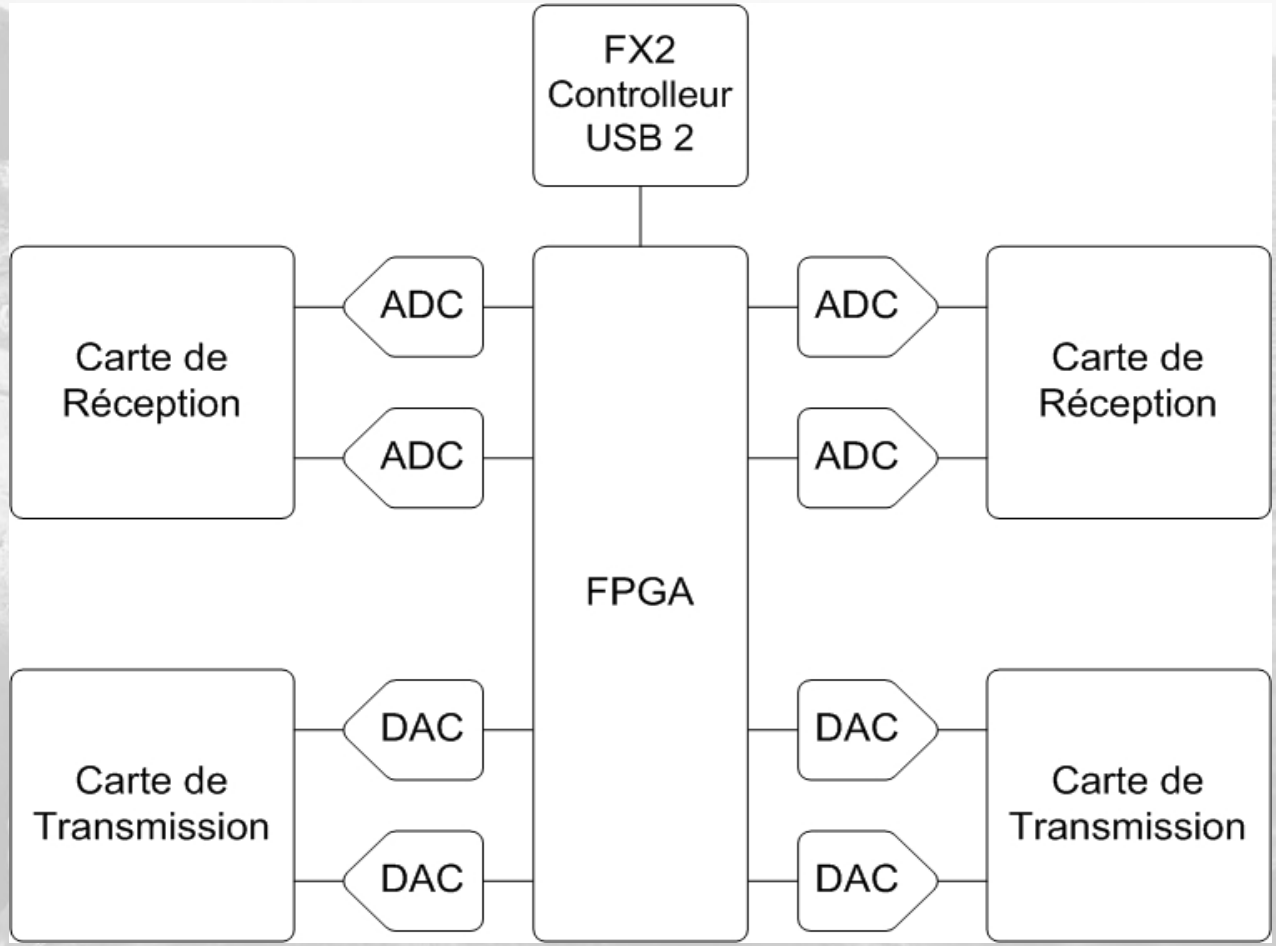


USRP : Version 2



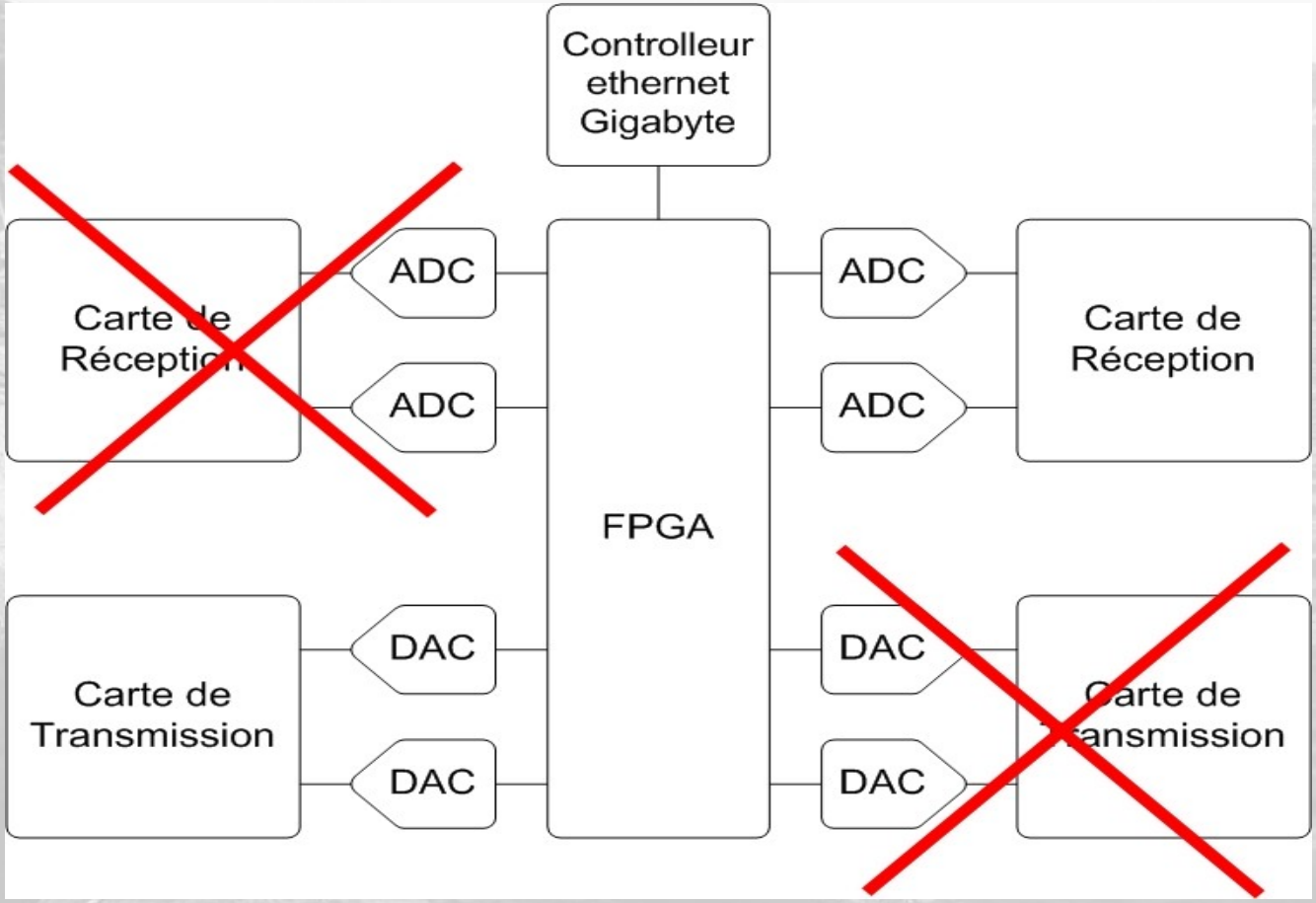


USRP : Schéma Bloc (1)



- Pour l'USRP1

USRP : Schéma Bloc (2)



- Pour l'USRP2

USRP : Le FPGA

- FPGA : field-programmable gate array
- Traite le signal.
- Réduit le débit de donnée en quelque chose de gérable.
- Inclut des DDCs

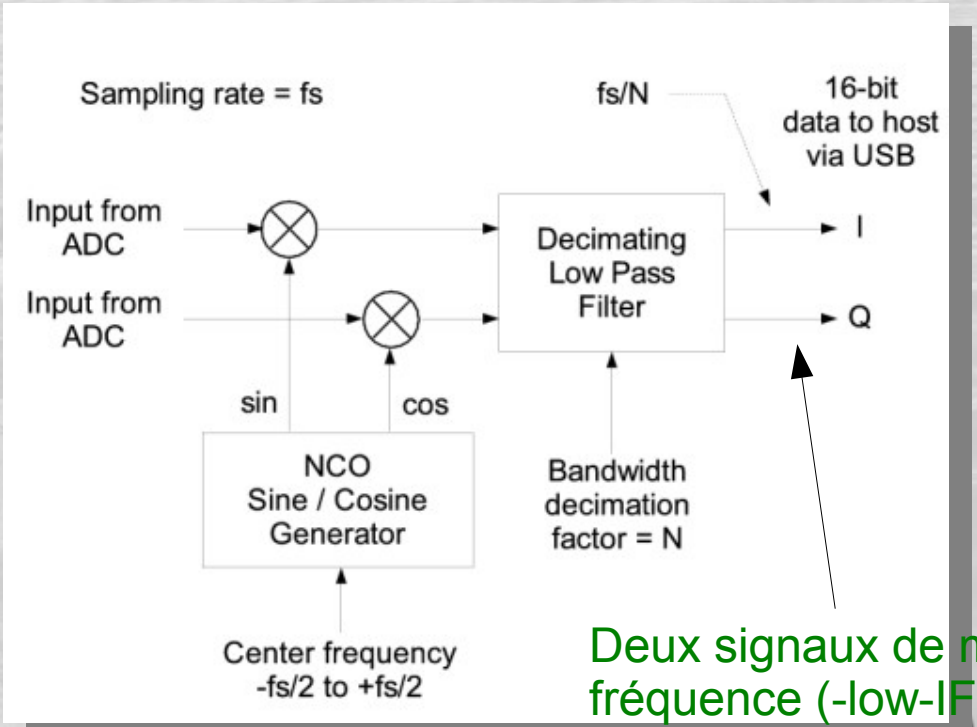
  (Digital Down-Converter)

  Dans l'autre sens on

  a exactement l'inverse

  => DUC

Utile pour le transfert USB /!\

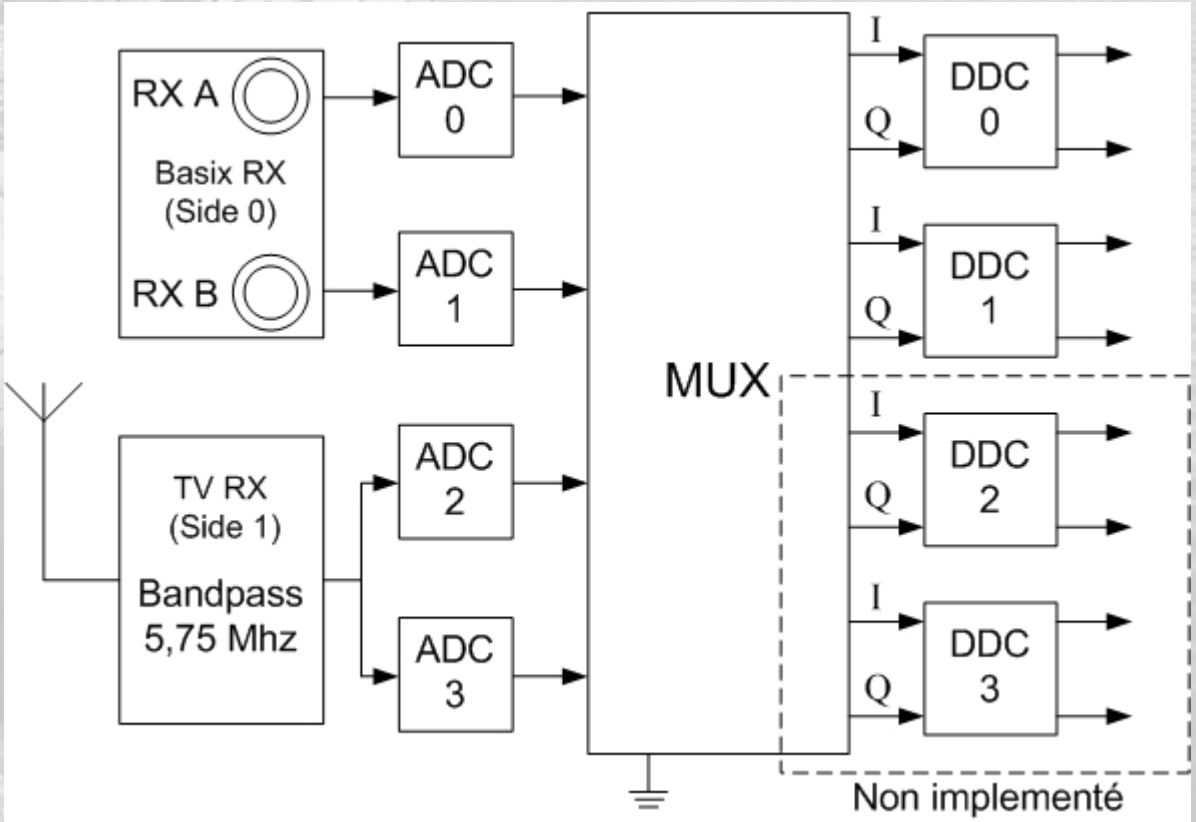


Deux signaux de moyenne fréquence (-low-IF)

USRP : MUX

- Entre les ADC et DDC on a un multiplexeur

  Contrôlable avec la fonction (set_mux() avec Gnuradio)



En savoir plus

- Les cartes filles :

  http://gnuradio.org/redmine/wiki/gnuradio/UsrpFAQDBoards

- Code du FPGA :

  http://code.ettus.com/redmine/ettus/projects/show/fpga

Très courte introduction au DSP

DSP : Culture générale

- DSP : Digital Signal Processing
- Sert à manipuler des signaux (vibrations sismiques, images, sons, ...).
- Ensemble : Mathématiques, Algorithmes et techniques.
- Utilisé dans de nombreux domaines aujourd'hui

DSP : Domaines d'utilisation

Domaine	Utilisation
Espace	Amélioration photographique, Compression des données, analyses sensorielles intelligentes par sondes spatiales
Médecine	Diagnostique des images (CT, MRI, ultra-sons, etc...), électrocardiogramme, stockage des images.
Commercial	Compression des sons et images pour les présentations, Effets, Vidéos conférences.
Téléphone	Compression de la voix et des données, réduction des échos, multiplexage du signal, Filtrage.
Militaire	Radars, Sonars, Communication « sécurisées ».
Industriel	Prospection pétrolière et minérale, surveillance et contrôle des processus, essais non-destructif, etc...
Scientifique	Enregistrement de données sismiques, Analyse de spectre, etc...

DSP : ADC/DAC (CAN et CNA)

- La plupart des signaux étudiés → Continus.
- Information digitale → échantillonnée et quantifiée.
- On décide : L'information à retenir et celle qu'on peut perdre.
- Cela exige : la sélection de la fréquence d'échantillonnage, nombre de bits et le type de filtre entre le domaine analogique et numérique.

DSP : Rappels Analogique->Numérique

- La conversion analogique vers numérique :
- Entrée principale :

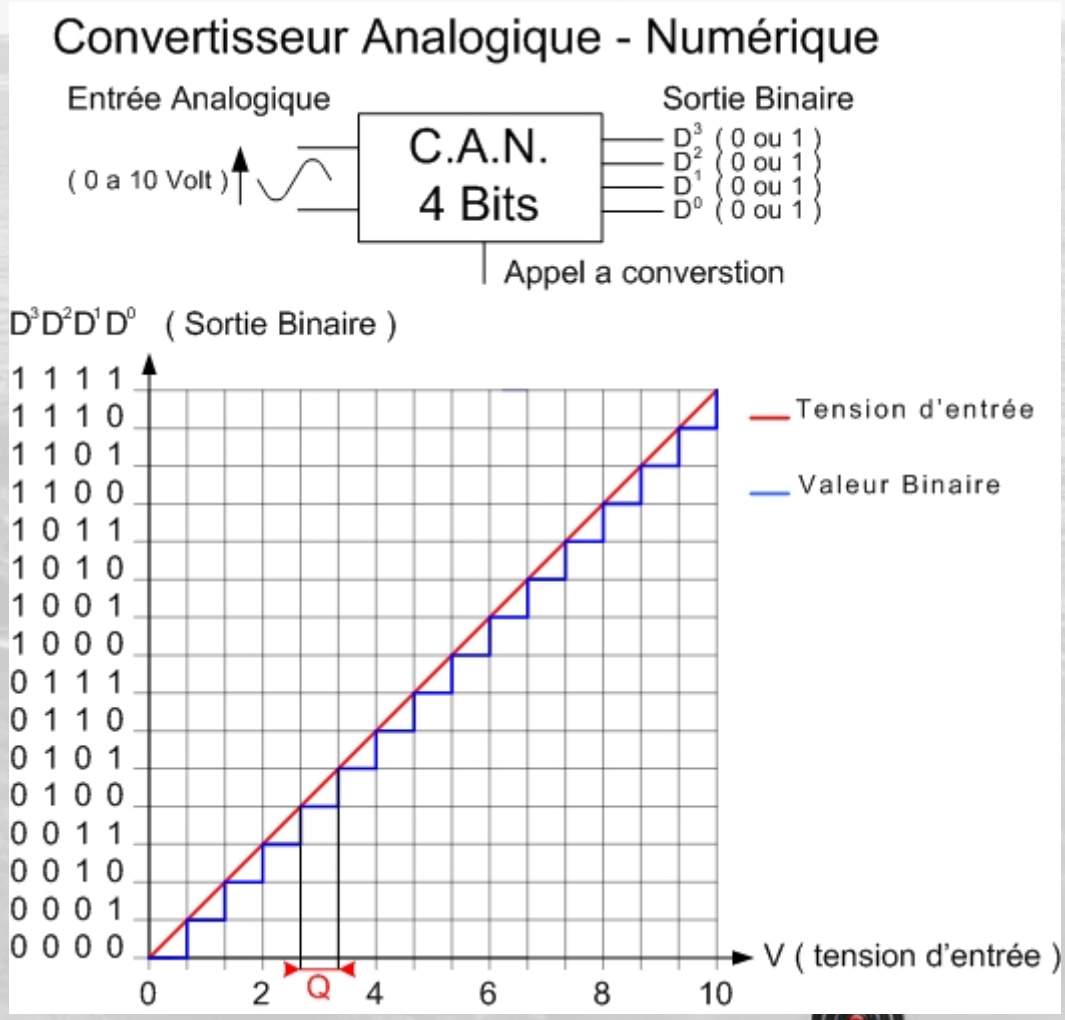
  Signal analogique
- Sortie :

  Résultat N

  (bus de donnée n bits)
- Quantum(LSB) : $q = V_1 - V_0$

  ou $q = \Delta V_{max} / 2^n$
- Fonction de transfert :

  $N = V_e / q$



DSP : Rappels Numérique->Analogique

- La conversion numérique vers analogique:
- Entrée principale :

  Entrée num. N
- Sortie :

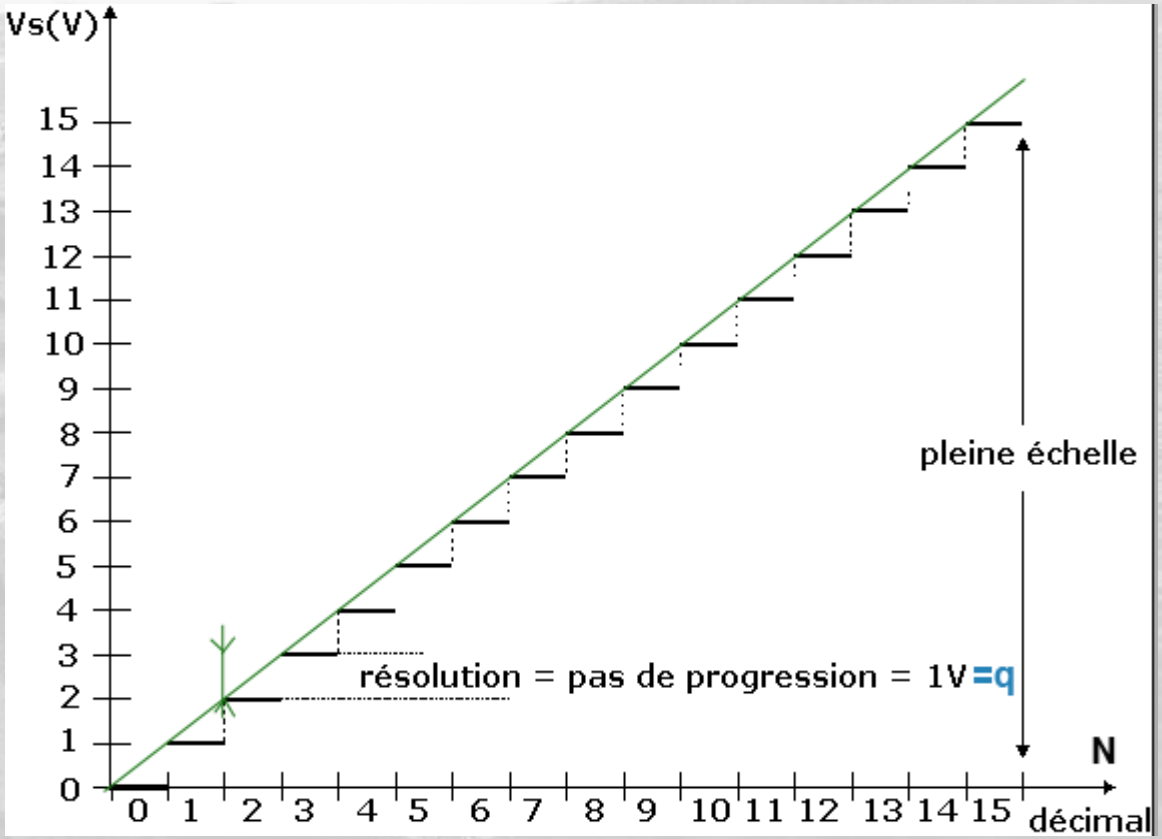
  Résultat V

  $V = N*q = N*V_{ref}/2^n$
- Loi de variation :

  $V = V_0 + (D_{n-1}.2^{(n-1)}q+$

  $D_{n-2}.2^{(n-2)}q+ \ldots + D_1.2q$

  $+ D_0.q)$; Ou alors : $V= V_0 + N.q$

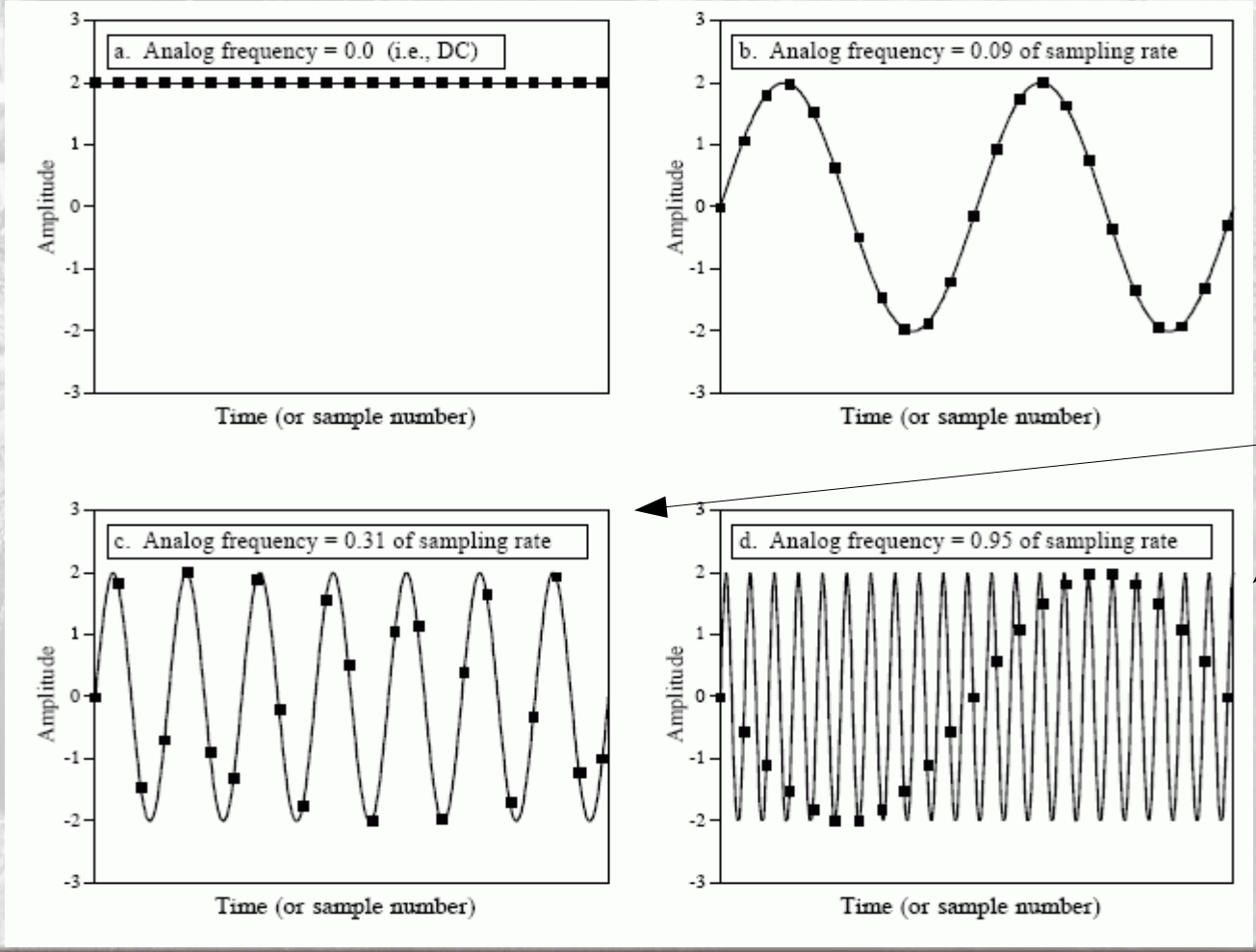


DSP : Transmission Analogique et Numérique (Culture)

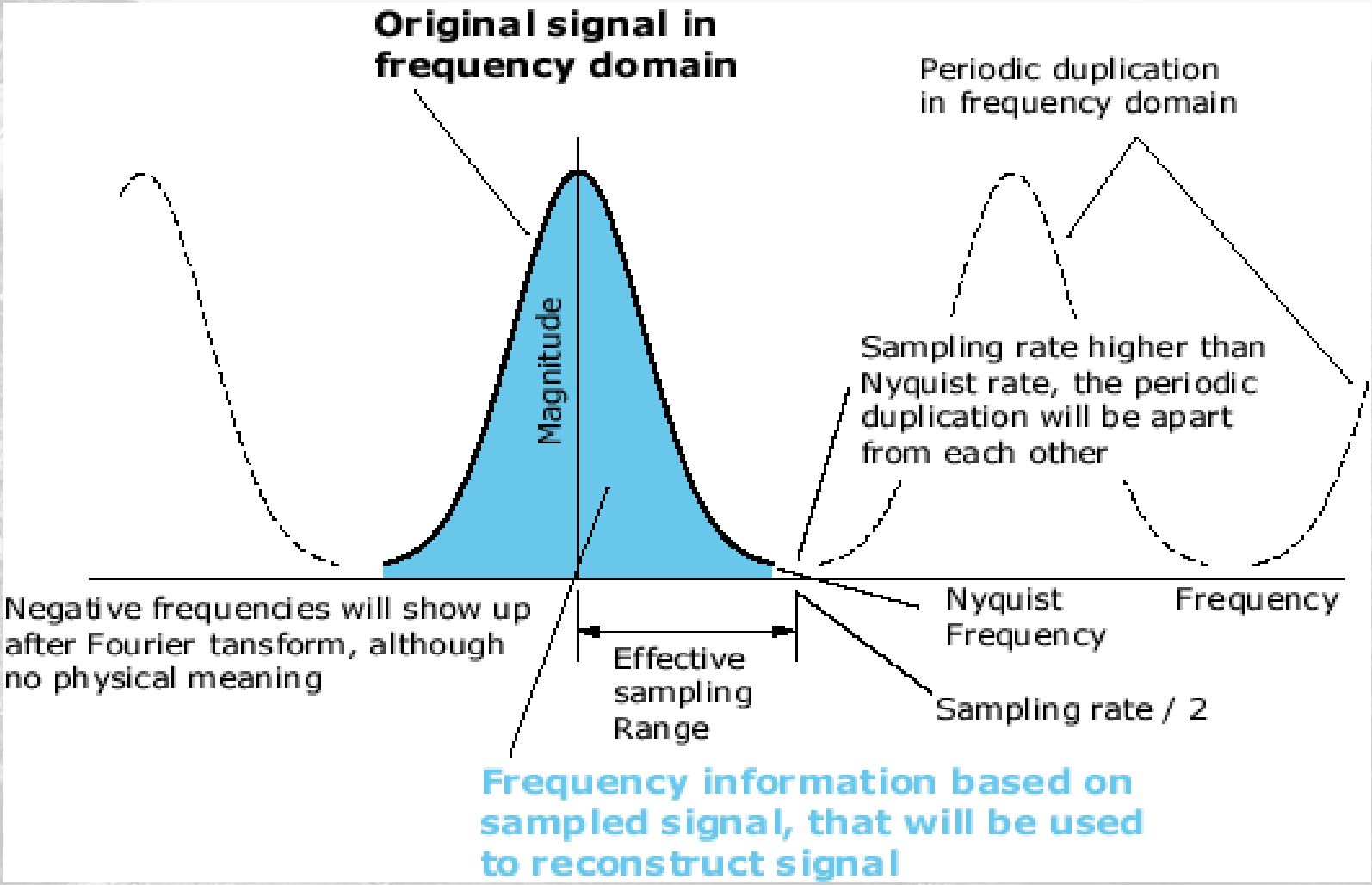
- Signal analogique : pas de propriétés particulières
- Signal numérique : Transition de niveau, conservation de valeur constante.
- Donc → besoins d'estimation lors de A/D :
  - Repérage des transitions du signal
  - Regénérer le signal d'horloge
  - Échantillonnage de la valeur reçue
  - Comparaison et application d'une valeurs
  - Reconstruction du signal.

DSP : Théorème de Nyquist/Shannon

- Fréquence d'échantillonnement > 2 * Fréquence maximum

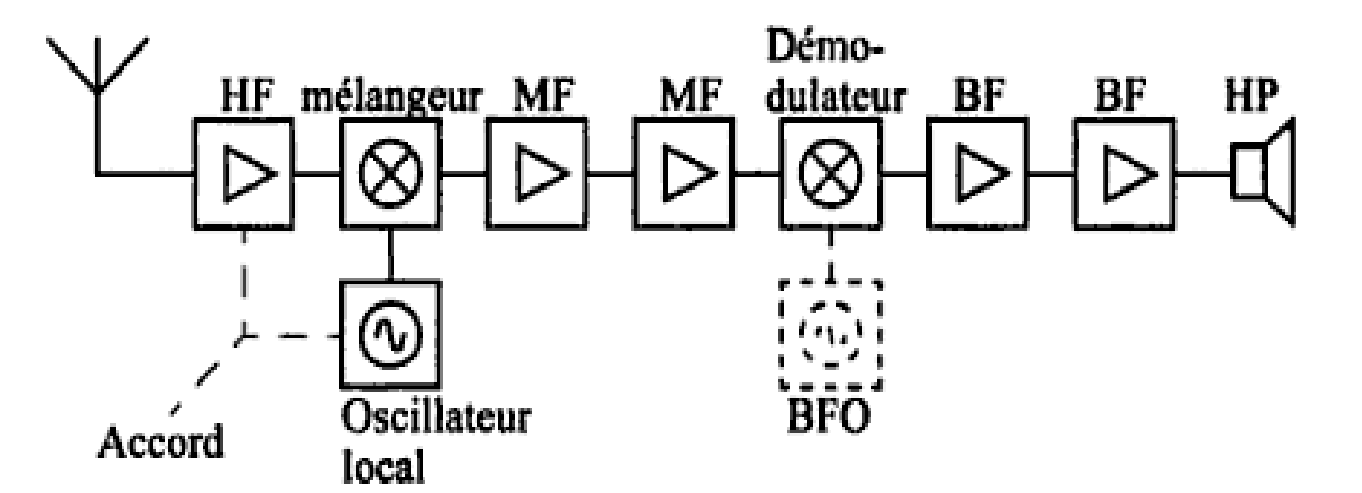


DSP : Fréquence de Nysquit



DSP : La superhétérodyne

- <u>Objectif</u> : Convertir la fréquence reçue en fréquence intermédiaire (plus facile à gérer).

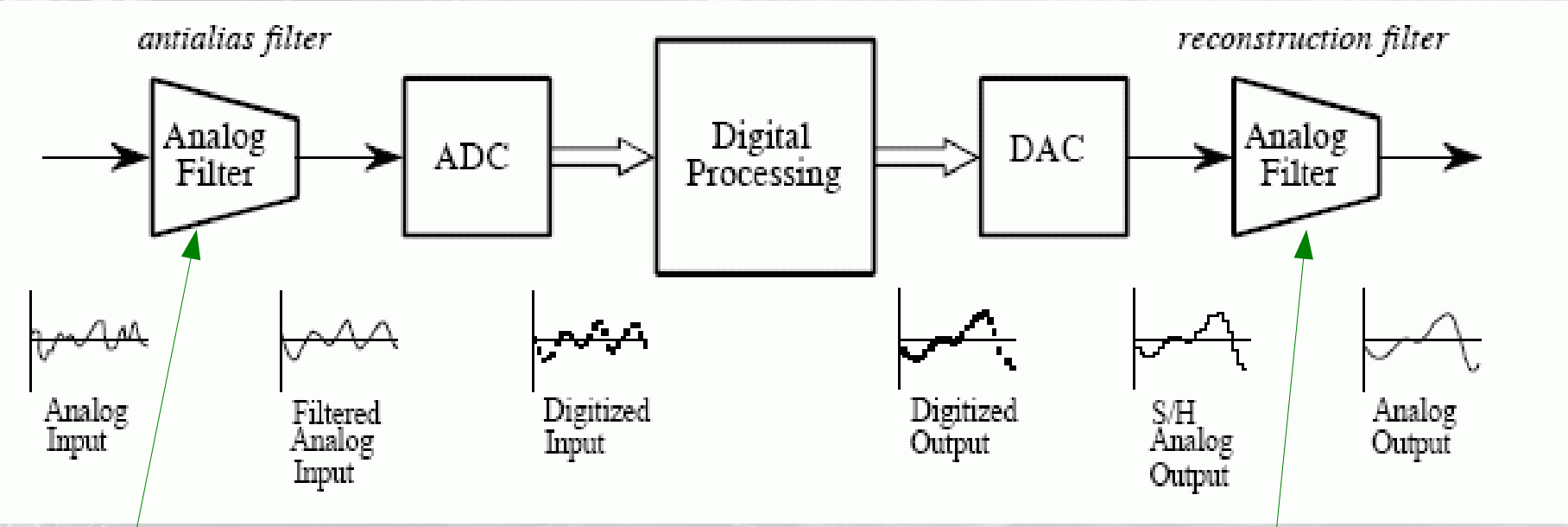


Pour un amplificateur moyenne fréquence sur F_{mo} et un signal reçu f_o → Oscillateur local $F_{LO} = f_o + F_{mo}$

On a donc en sortie du mélangeur : f_o et F_{LO} mais aussi $|f_o - F_{LO}|$ et $f_o + F_{LO}$

Ce qui nous intéresse

DSP : Schéma ADC/DAC



Dans la fréquence de Nyquist

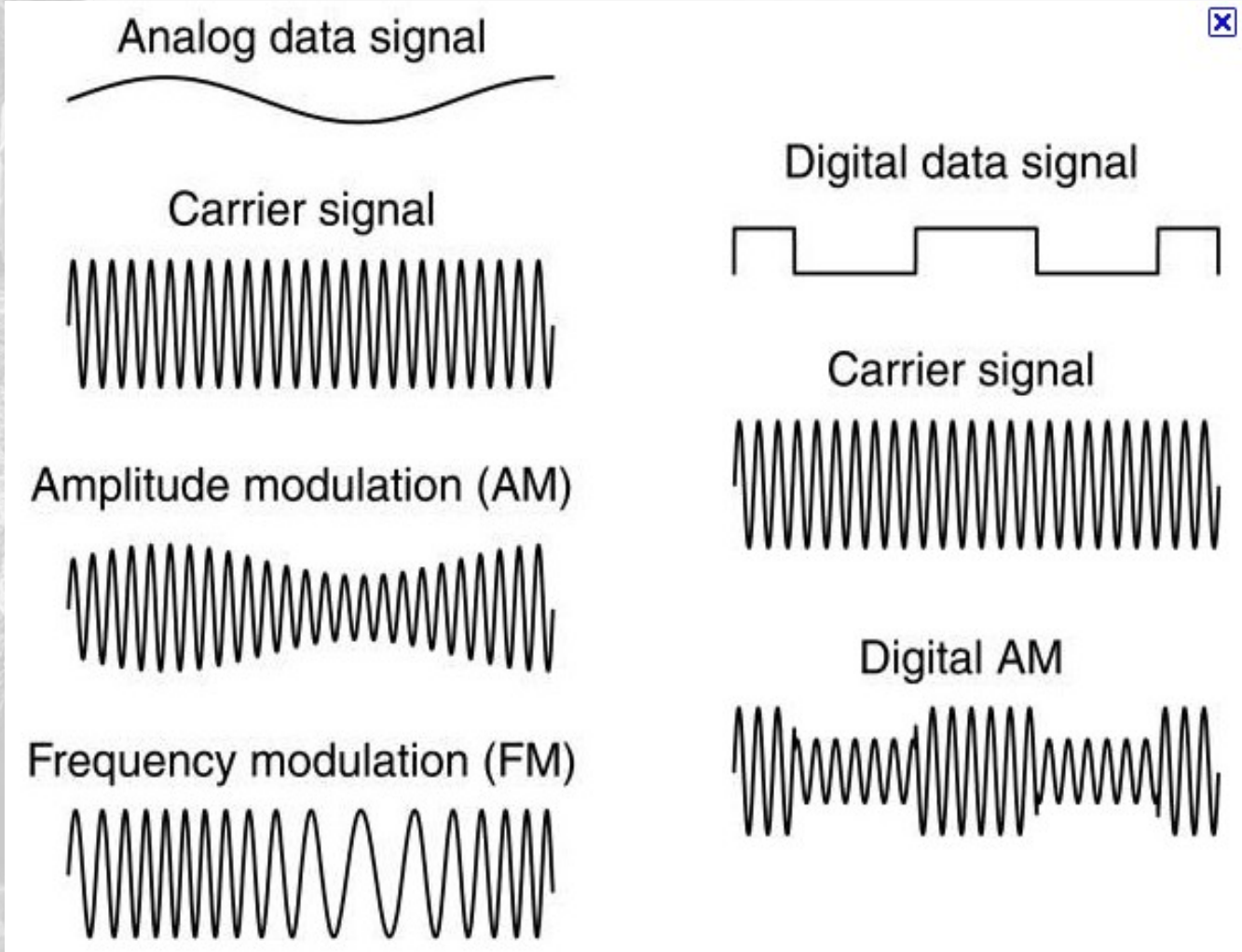
Reconstitution du Signal analogique

DSP : En savoir beaucoup plus

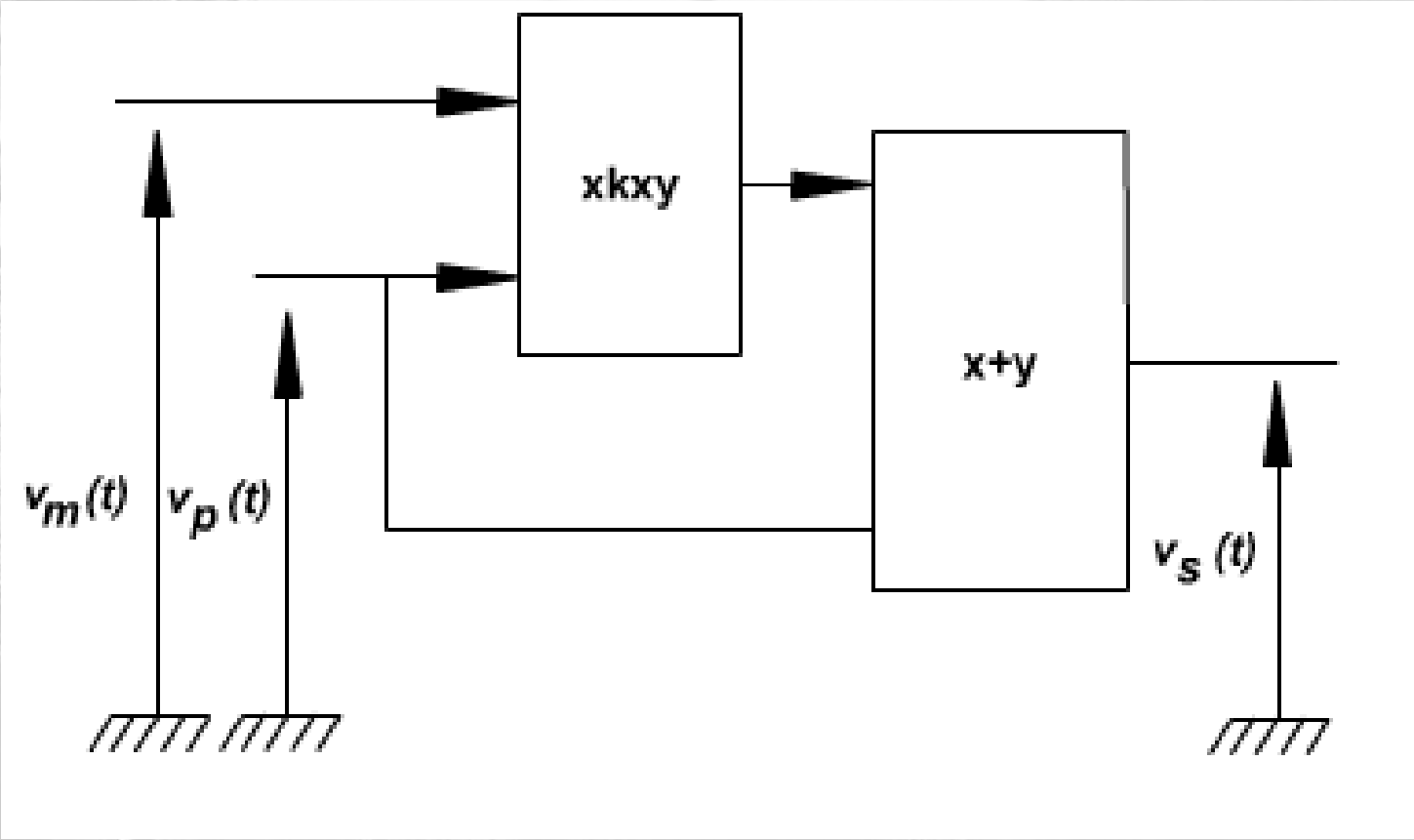
- Partie sur les récepteurs (Le Radio amateur)
- The Digital Signal Processing guide (de 800 pages)
- Chapitre 4.2 L'entropie dans la théorie de l'information – L'héritage de Kolmogorov

Communication numérique : La modulation AM et FM

La modulation AM et FM



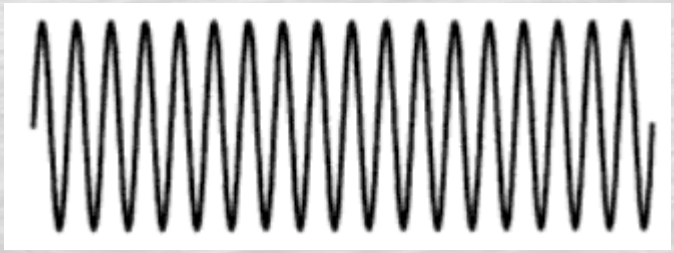
Modulation d'amplitude (1)



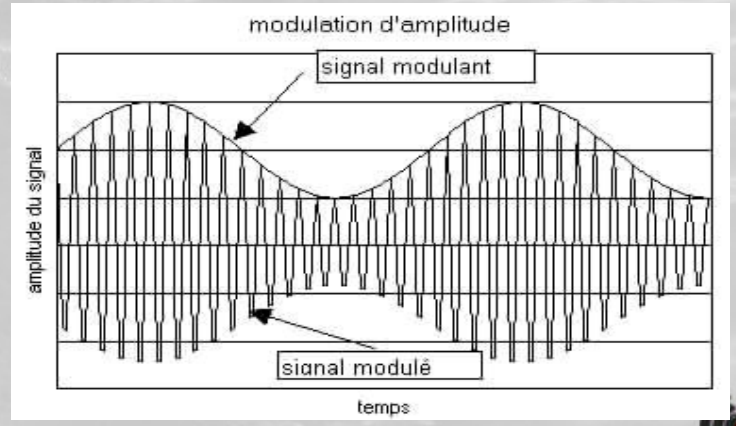
Modulation d'amplitude (2)

- Signal BF à émettre :

- Signal HF porteuse :



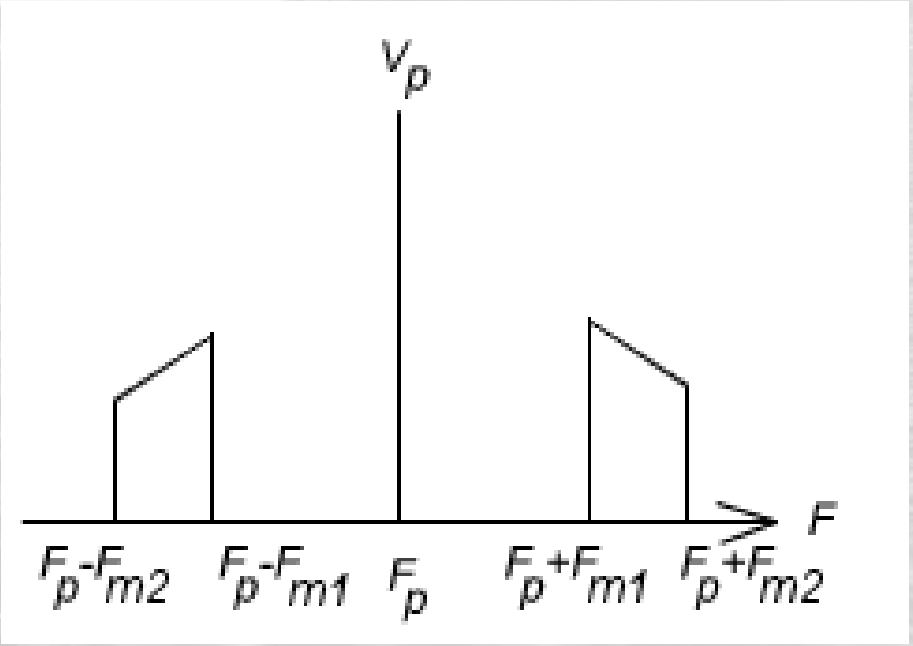
- Signal modulé :



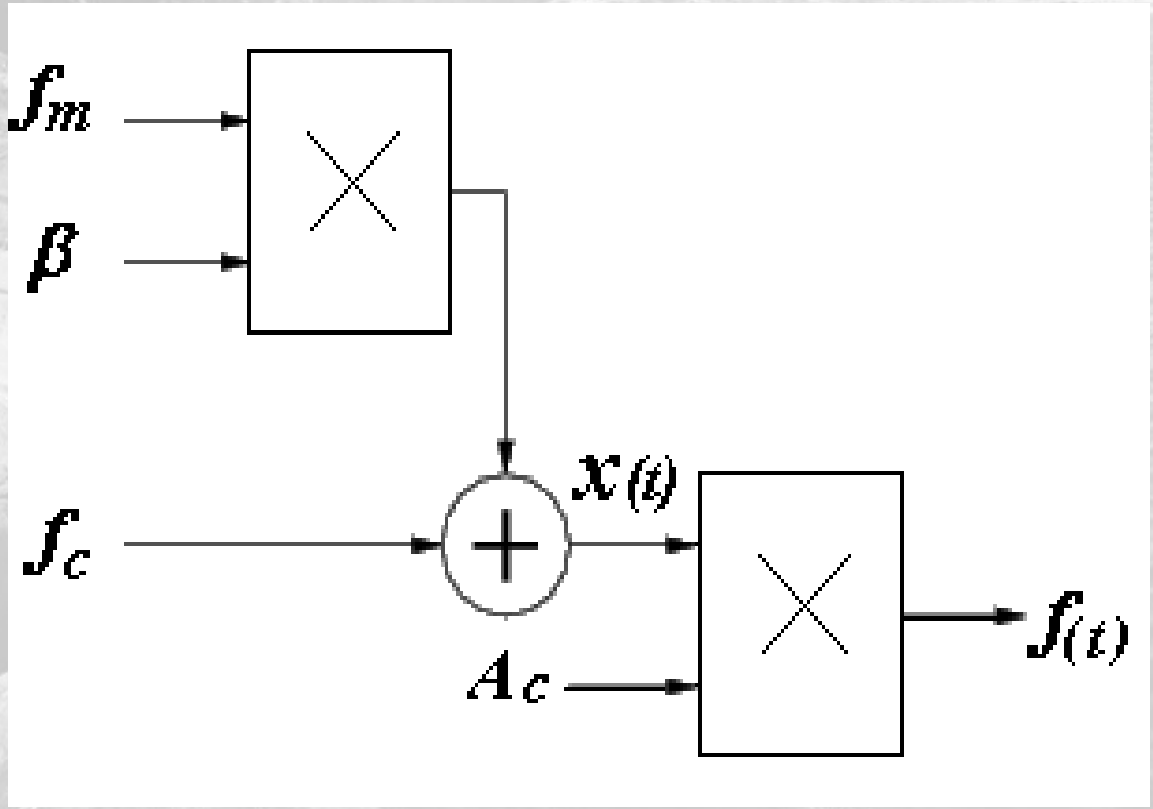
Modulation d'amplitude (3)

- <u>Largeur de la bande</u> : $2*fmod_{max}$

- 2/3 Puissance → Porteuse
- 1/6 → Dans chaque bande latérale



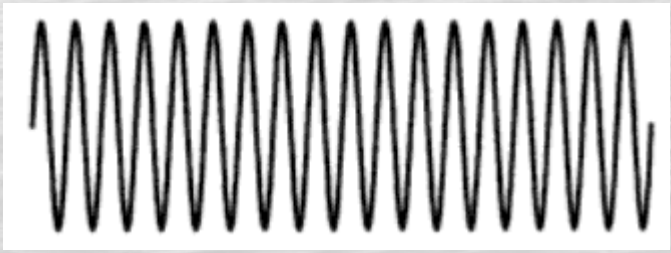
Modulation de fréquence (1)



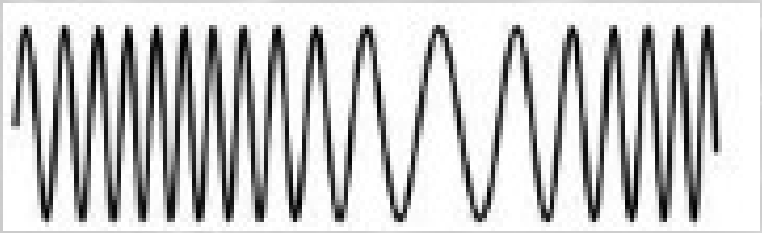
Modulation de fréquence (2)

- Signal BF à émettre f_m :

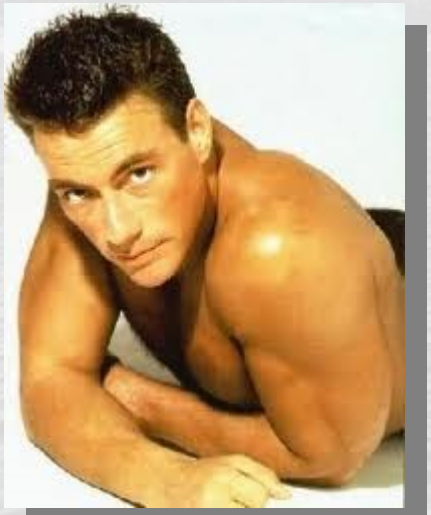
- Signal HF porteuse f_c :



- Signal modulé f(t) :



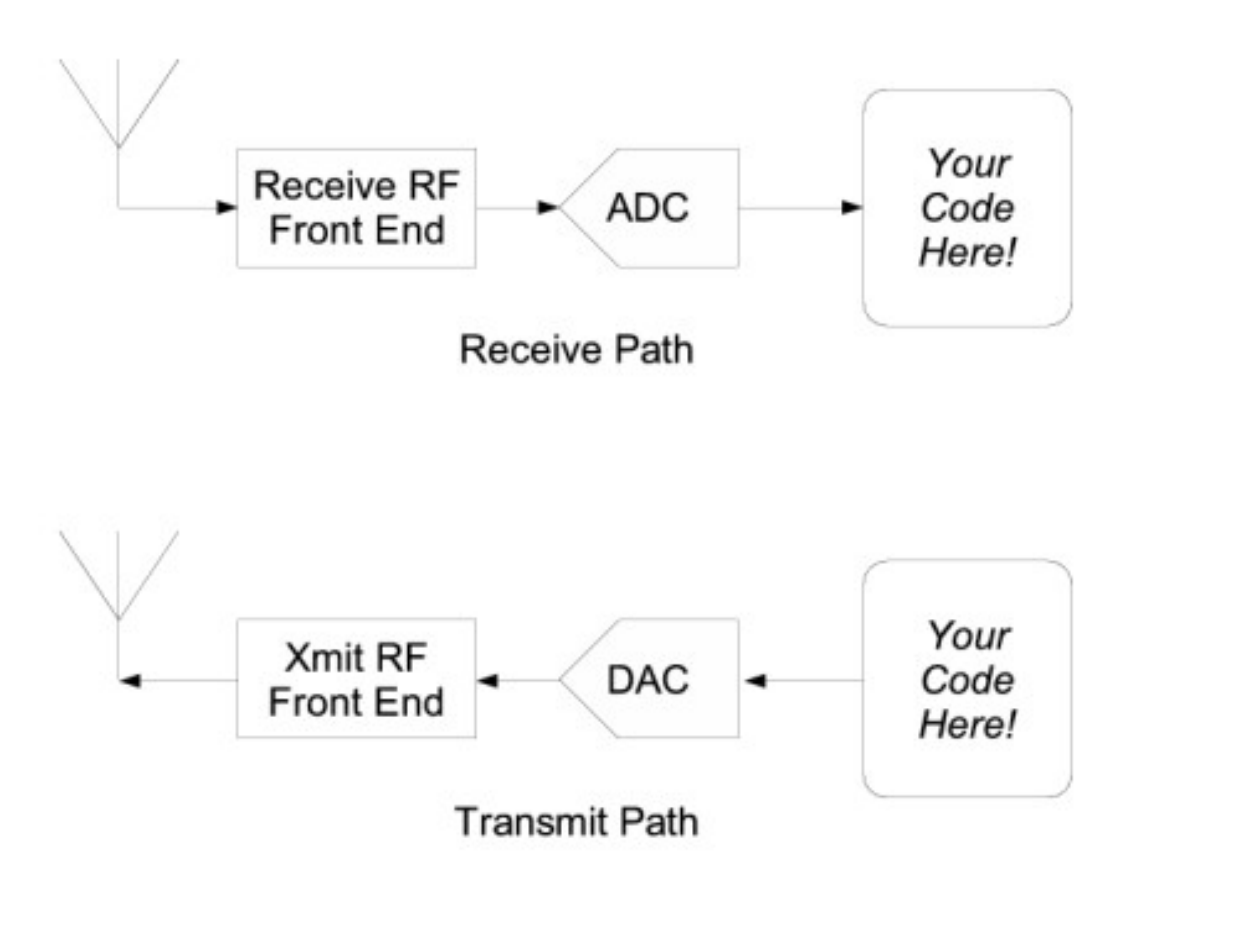
Sois SDR, sois AWARE!



Software-Defined Radio : GnuRadio

- Radio logicielle : Tout reconfigurer « On the fly » (À la volée)
- Permet modulation/démodulation, filtrages → par programme dans un DSP (notre PC).
- Écrit en C++ et blocs implémentés pour Python
- GRC « GnuRadio Companion » : Outils de modélisation et génération de code python.
- Disponible sur Linux, Mac OS X, Windows (émulé).
- Bon support (lists, wiki, chan, ...)

SDR : GnuRadio Diagramme



GnuRadio : Les préliminaires

- Les bases sur l'USRP
  - (FPGA, ADC/DAC, Gates...)
- Programmer :
  - Python
  - C++
- Notions de Digital Signal Processing
- Notions de communication numérique
- Savoir chercher dans l'API : http://gnuradio.org/doc/doxygen/index.html

GnuRadio : Installation

```
$ git clone git://gnuradio.org/gnuradio
$ cd gnuradio
$ export LD_LIBRARY_PATH=$BOOST_PREFIX/lib

$ ./bootstrap
$ ./configure –with-boost=$BOOST_PREFIX # exemple : /opt/boostX
$ make
…
# make install
```

N'oubliez pas d'installer la dernière version de boost en précisant le « --prefix » pour savoir directement ou chercher la librairie

GnuRadio : Démarrage

- Si tout est bien installé et connecté (USB2 ou Ethernet Gb au PC) :
  - Sur l'USRP1 : `$ ls -lR /dev/bus/usb | grep usrp`
  - Sur l'USRP2 : `$ ./find_usrps`
- L'USRP étant bien détecté, nous regardons si cela fonctionnement correctement :

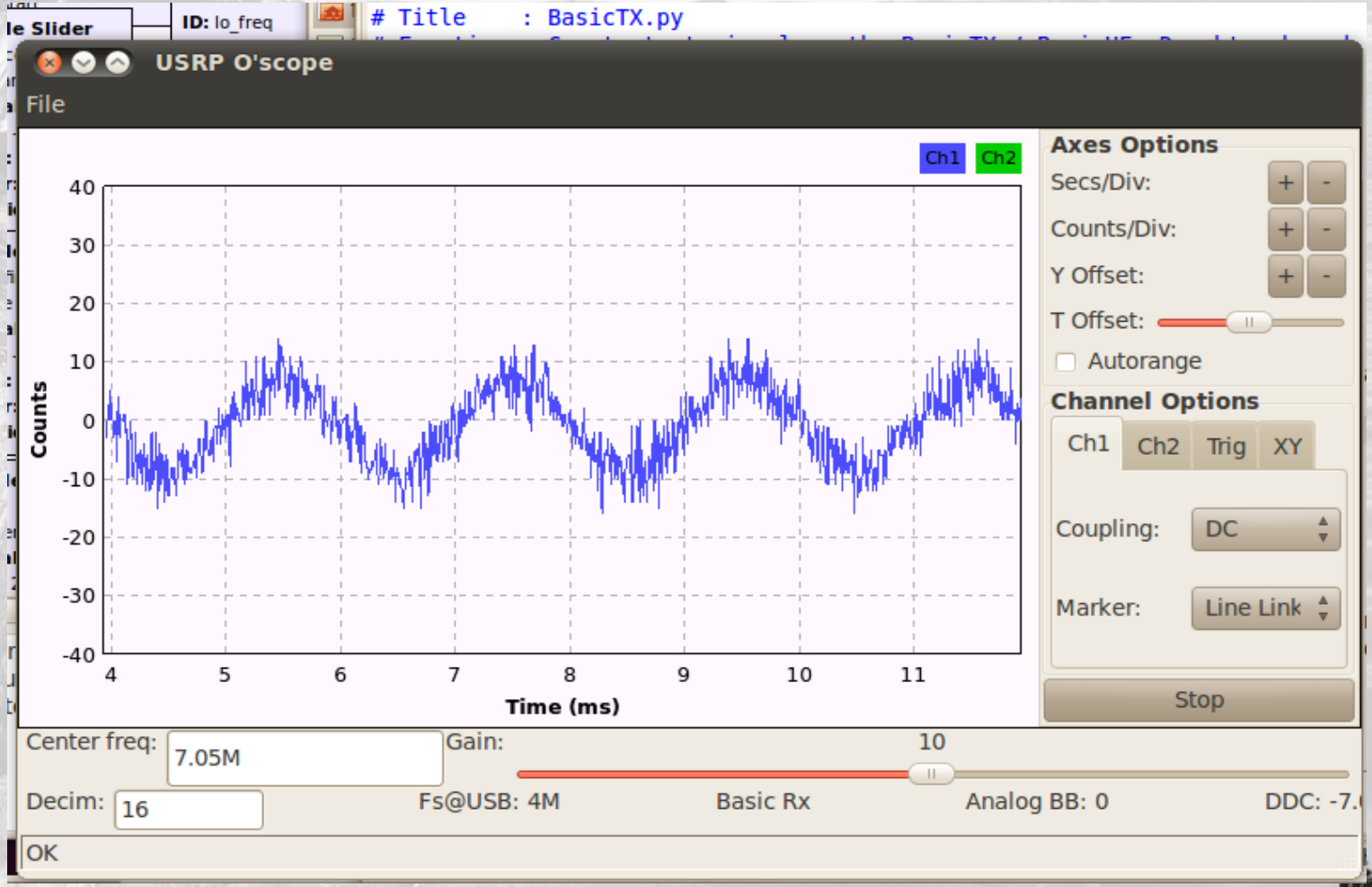
```
cd gnuradio-examples/python/usrp
./usrp_benchmark_usb.py
```

Jeu de transmission et réception

- Pour les premiers tests en transmission : basicTx.py
- Nous allons le modifier pour transmettre un signal à 7,05Mhz.
- Nous connectons les cartes BasicTx et BasicRx.
- Puis on lance le script : usrp_oscope.py
- Puis...

Jeu de transmission et réception

- Voilà ! (démo. en live, les antennes ne sont pas adaptées!)

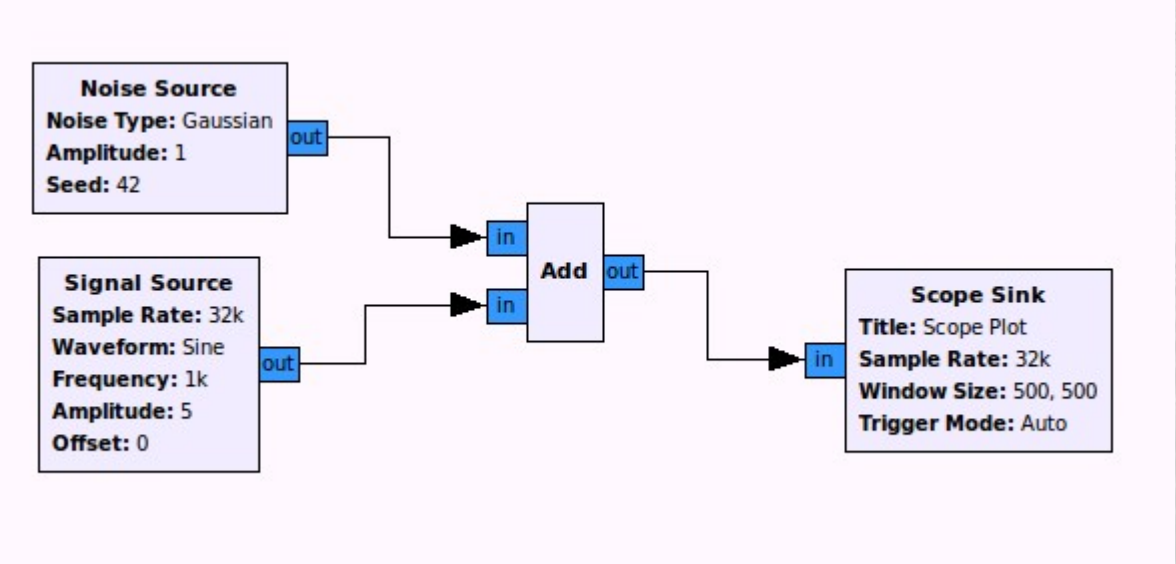


Gnuradio Companion

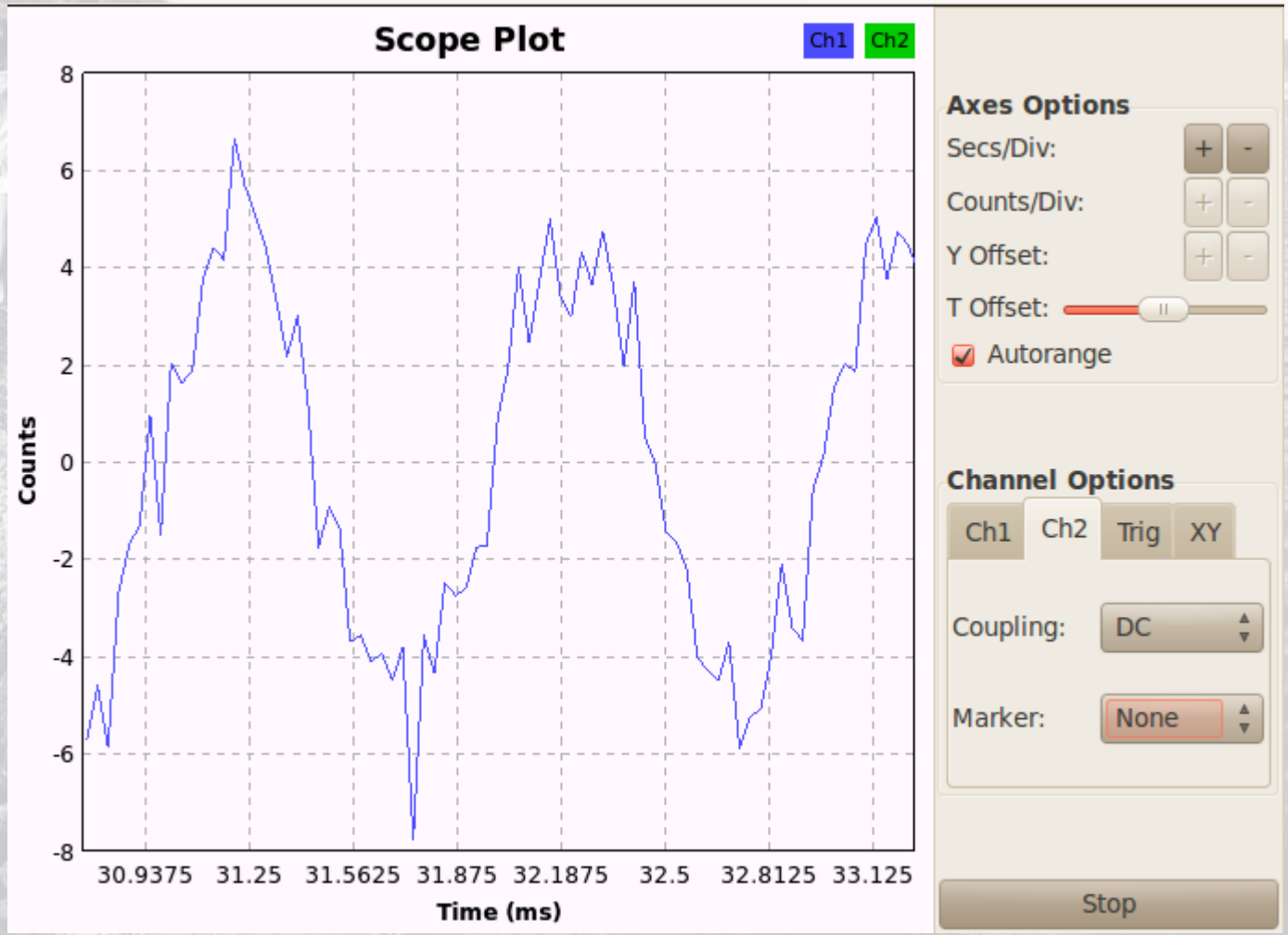
- GRC : Outils de modélisation graphique
- Avantages :
  - Permet d'avoir une vue de ce qu'on fait
  - Génère le code automatiquement si tout est correct
  - Plus besoin de programmer... (Pauvre Python)
- Il est possible aussi de simuler en SDR sans USRP, juste avec GRC.

Simulation banale (1)

- Nous allons simuler avec un signal sin(t) additionné avec du bruit gaussien.

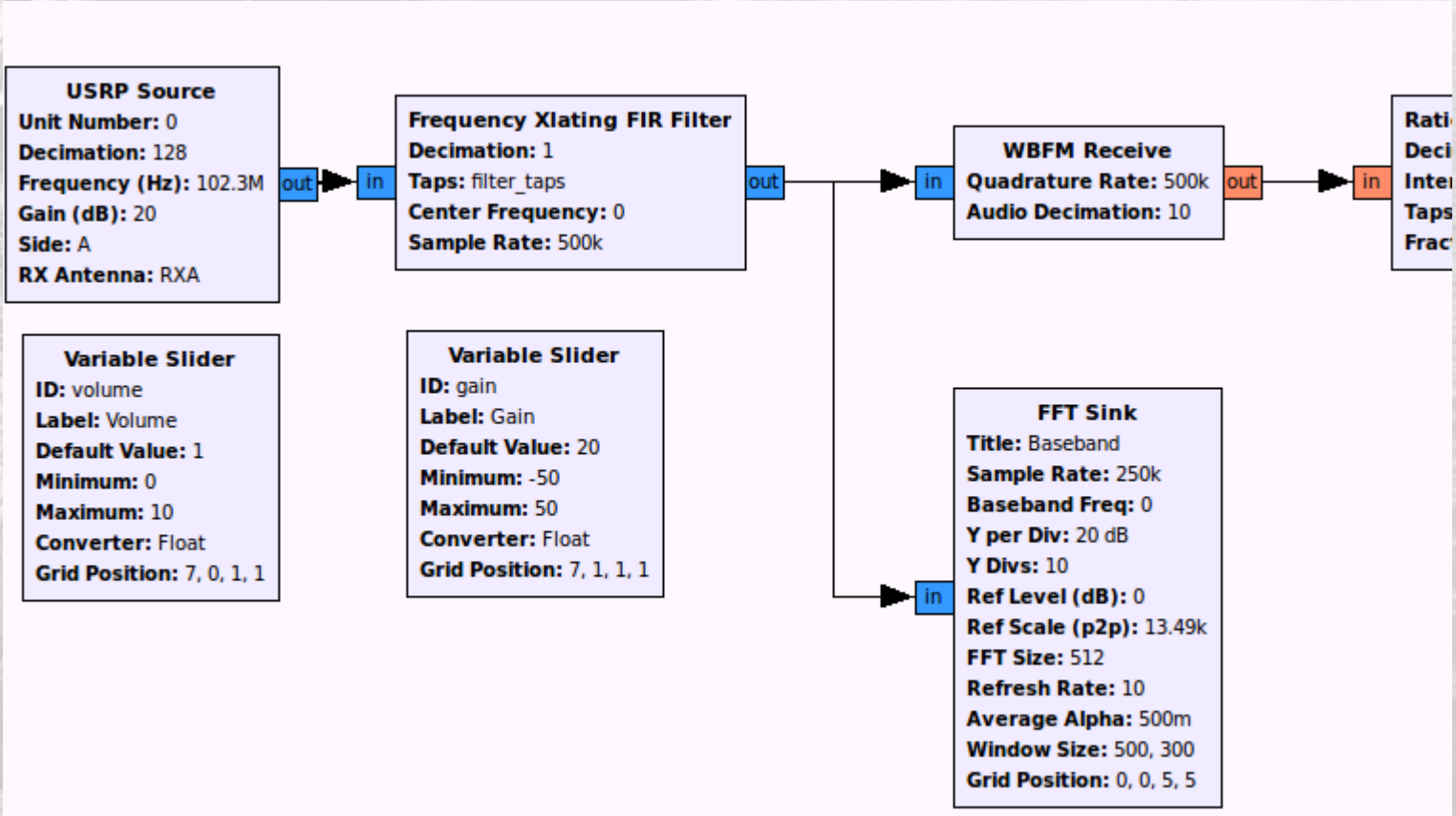


Simulation banale (2)



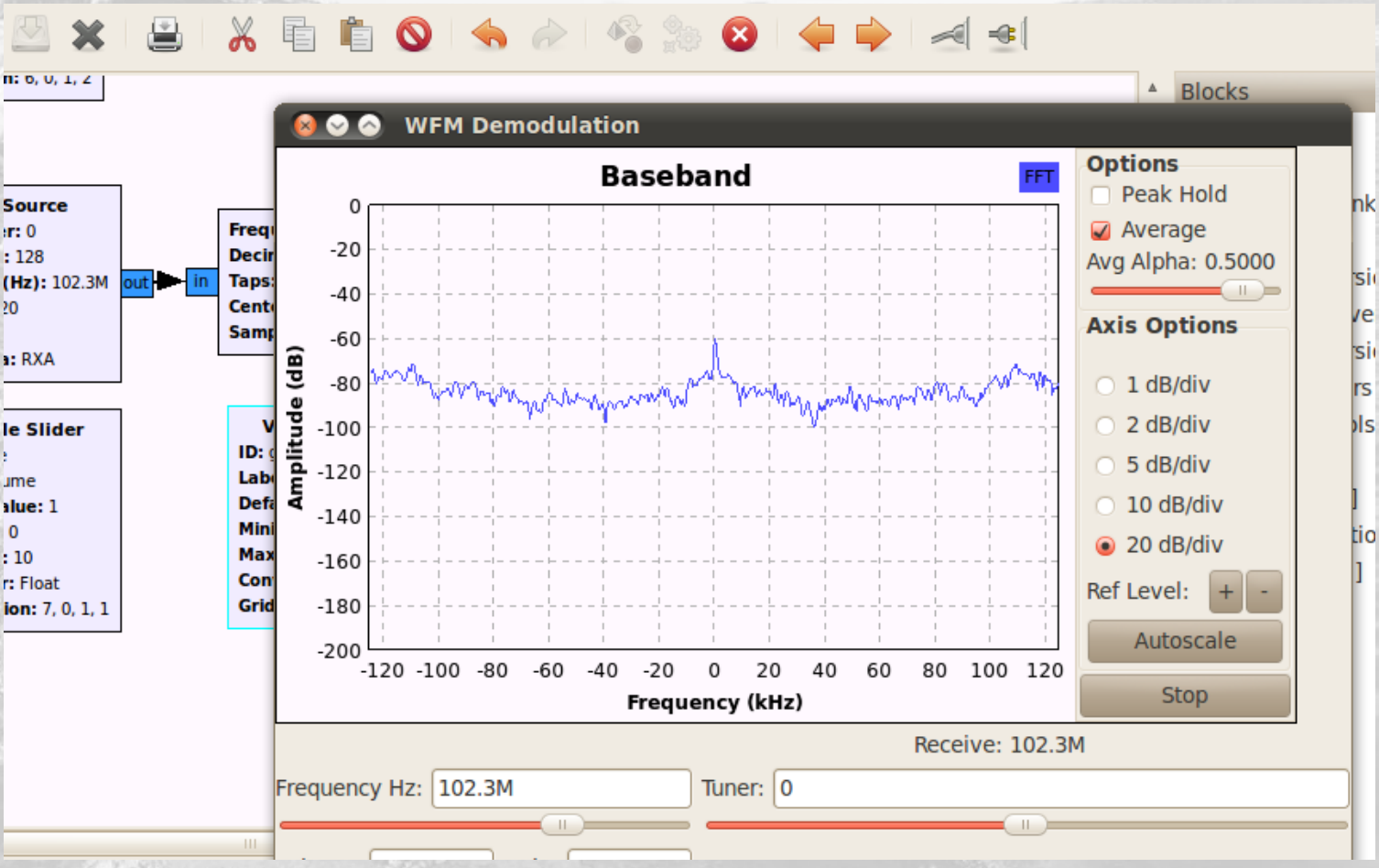
Réception de la Radio FM (1)

- Utilisation de BasicRx en WBFM



Réception de la Radio FM (2)

- (Toujours avec des antennes non adaptées)



En NBFM

- Démo live + Vidéo avec réception TVRx et parabole

2012 : Fin du monde? - Mouai! La mort de la radio FM! - « NOOonn! »

- Sera remplacé par une modulation : NWO
- Qualité : pas génial
- Faible taux de recouvrement
- Encodage audio : 64 kbps
- Codage numérique passé dans le canal : T-DMB

Le projet Invihertz

L'équipe

- Motivations :
  - Passion pour la radio et radio-amateurs
  - Influence de la CCC sur les attaques GSM
  - La recherche
- Personnalités :
  - cde
  - Tr00ps
  - Mescal
  - Free_MaN
  - Trance
  - FlUxIuS

Objectifs

- Travailler sur la radio logicielle.
- Étudier les différents protocoles et les défier.
- Proposer des solutions simples et embarquées.
- Et toujours apprendre!

Invihertz disponible au publique

- Adresse : http://invihertz.handgrep.se
- Contenu :
  - Wiki (Tutoriaux pris depuis le début, en cours de rédaction)
  - Code et schéma (à venir)
  - Projets (Dans l'embarqué et softs à venir)
- Le chan : #usrp sur WoldNet

Devenir membre

- « Pour devenir membre, il faut avant tout pouvoir le faire marcher sur ça propre machine » - Philosophie de xdbg par cde.
- Autres critères aux choix : être curieux, contribuer, communiquer, participer financièrement, traducteur, design …
- Pour tout autres questions → venez sur le chan !

Conclusion

- L'usrp → investissement en radio
- Les possibilités sont presque infinies
- Demande des notions particulières
- Le matériel n'est pas simple → demande du temps d'adaptation
- Si vous êtes passionnés vous aussi rejoignez Invihertz pour la contribution des projets à venir

Remerciements

- Avant tout → cde
- Mescal
- Free_maN
- Tr00ps
- Et pour finir au publique bien sur! ;)

Et moi alors ???!!