SDR

PAR OÙ COMMENCER ?
A QUOI ÇA SERT ?
COMMENT ÇA FONCTIONNE ?
QUEL MATÉRIEL POUR QUEL BUDGET ?

PRÉSENTATION PAR SYLVAIN F4GKR

*SDR : software-defined radio - une radio logicielle

Objectifs de la présentation

- Un peu de théorie pour présenter les concepts et mieux comprendre comment ça fonctionne,

- Avec dans l'idée de vous aider à savoir de quoi vous avez besoin (=combien dépenser…)

- Un tour d'horizon de quelques matériels disponibles pour les amateurs


Introduction

Origines

- Première apparition du terme « Radio Logicielle » dans une publication du chercheur Joseph Mitola – 1991 «*Software Radio: Survey, Critical Analysis and Future Directions*»

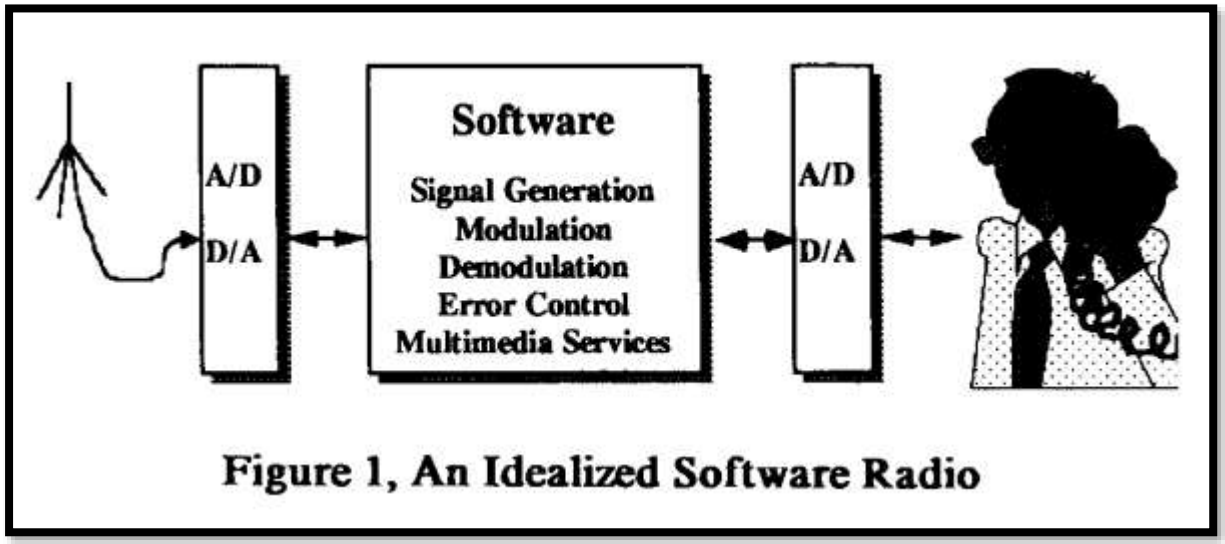
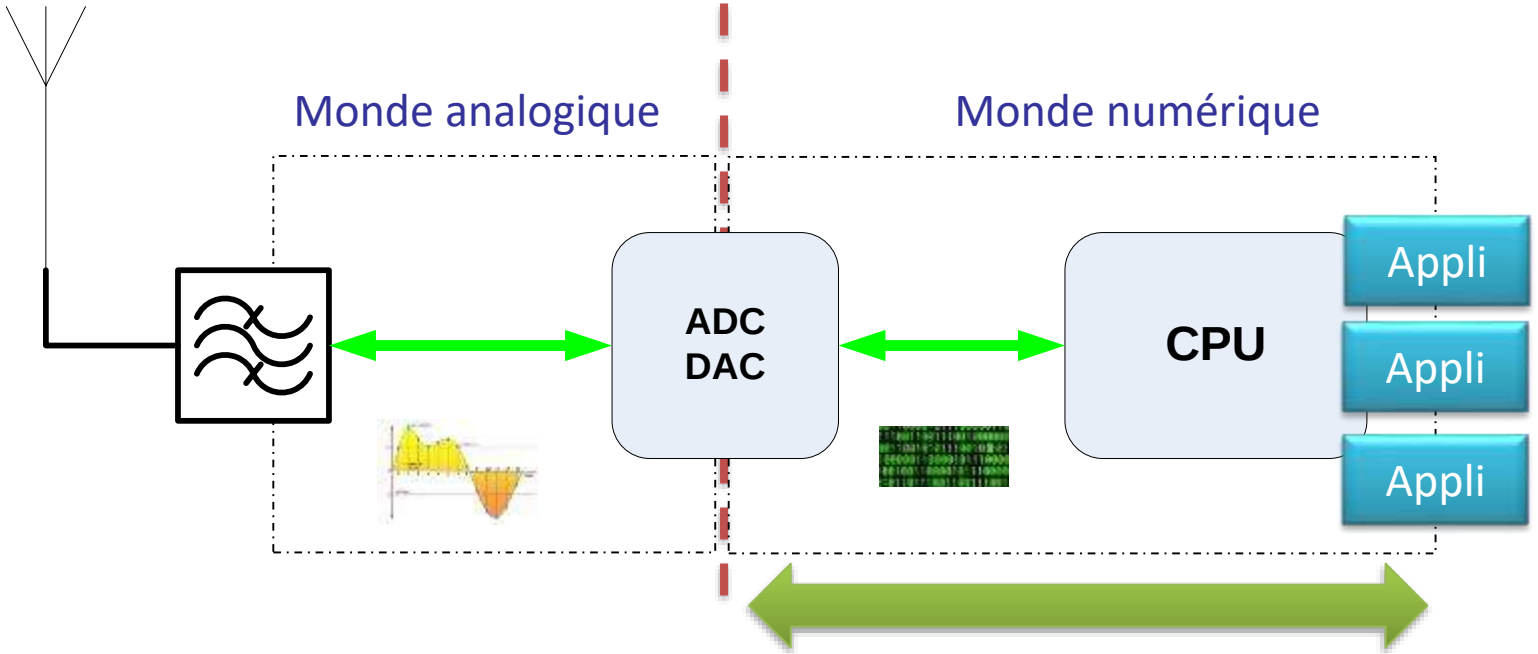


Figure 1, An Idealized Software Radio

SDR : Ce que l'on cherche à faire

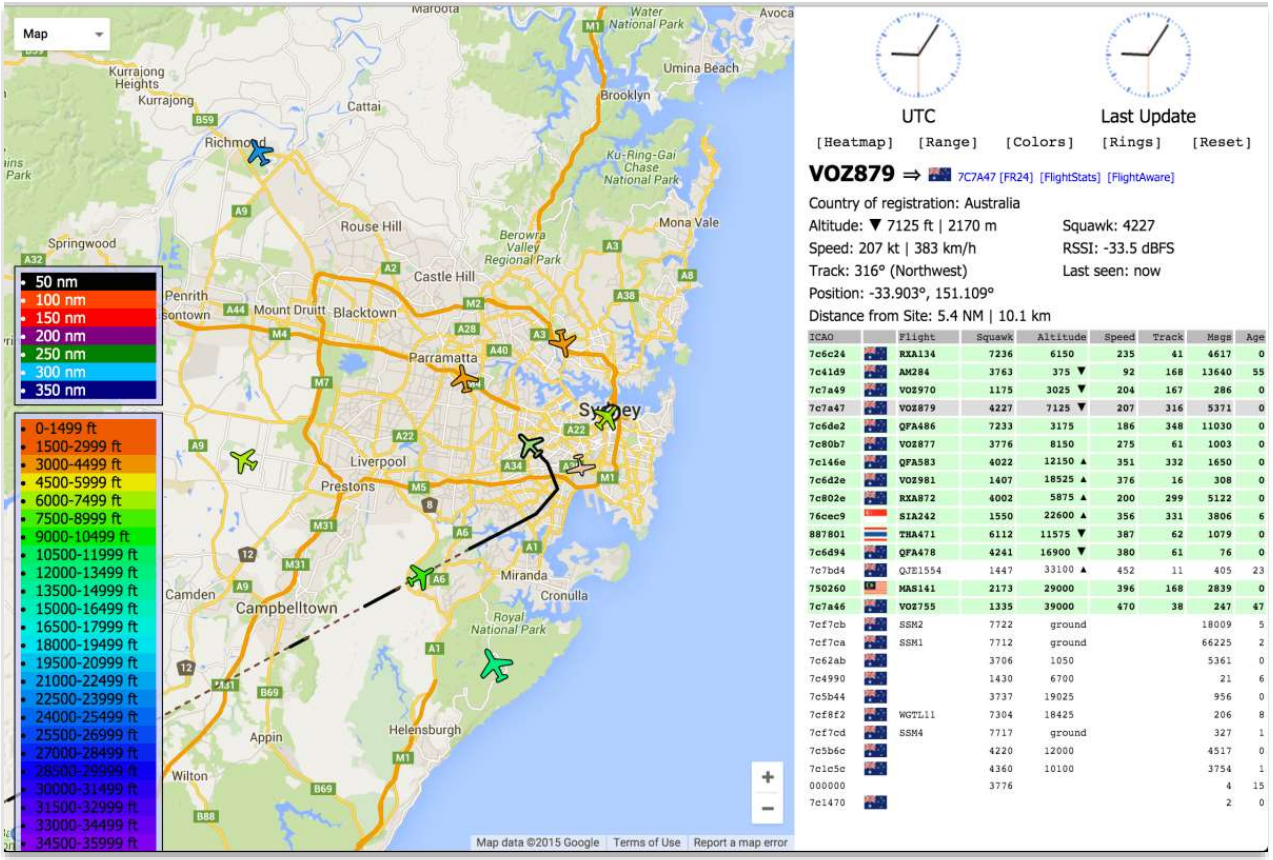


Les éléments qui composent le SDR idéal :

- **Une antenne et un peu de filtrage,**
- **Des circuits de conversion analogique/numérique,**
- **CPU + applications.**

Avec un « monde analogique » aussi réduit que possible

Quelques exemples



- Logiciel « dump1090 » pour réception de l’ADSB

Quelques exemples

- Logiciel « gnu-radio » et « gr-ais » pour réception de l'AIS

Quelques exemples

- Logiciel « rtl_433 » pour réception des données de stations météo

RTLSDR

A partir de 20 € !!!
(ajouter le prix du PC bien sûr)

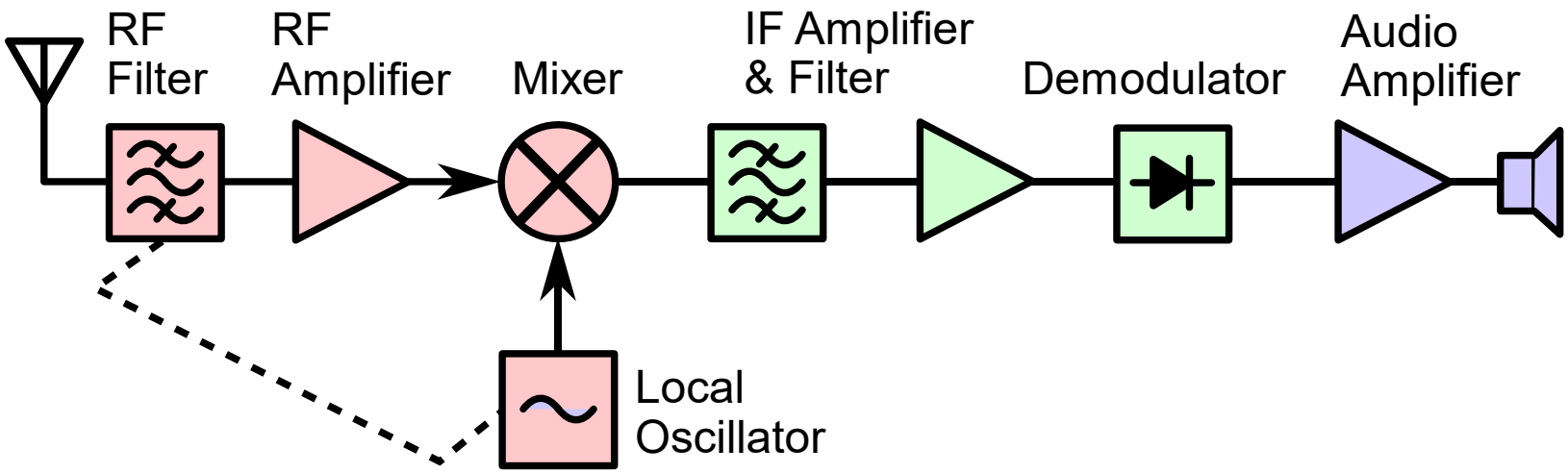
Pourquoi tant de différence de prix ?

SDR	Tune Low (MHz)	Tune Max (MHz)	RX Bandwidth (MHz)	ADC Resolution (Bits)	Transmit? (Yes/No)	Price ($USD)
RTL-SDR (R820T)	24	1766	3.2 / 2.56 Stable	8	No	~20
Funcube Pro+	0.15 410	260 2050	0.192	16	No	~200
Airspy	24	1800	10	12	No	199
SDRPlay	0.1	2000	8	12	No	149
HackRF	30	6000	20	8	Yes	299
BladeRF	300	3800	40	12	Yes	400 & 650
USRP 1	DC	6000	64	12	Yes	700

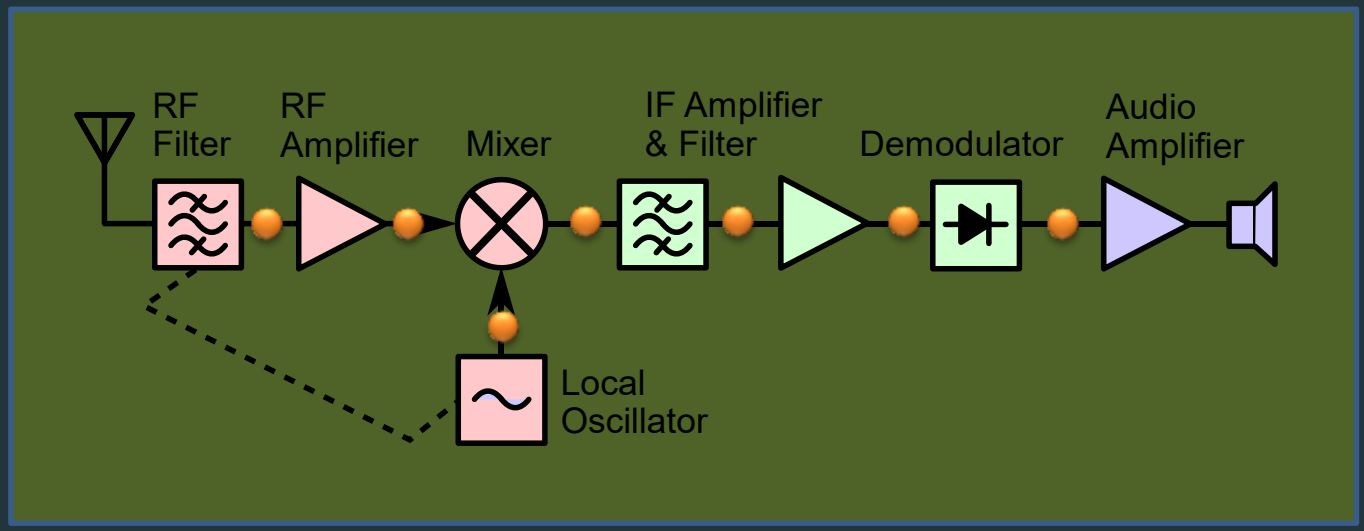
Première partie

Un peu de théorie pour comprendre d’où ça vient et comment ça fonctionne

Récepteur radio « conventionnel »

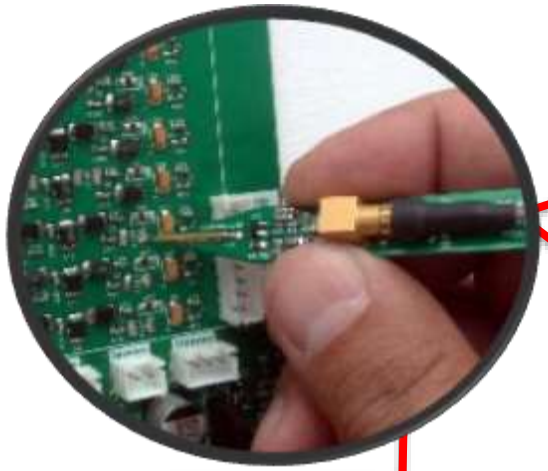


Un « récepteur SDR » va chercher à remplacer le maximum de ces blocs par des opérations réalisées dans un logiciel.



Quelle est l'information traitée ?

Ce qu'il y a dans « les fils »



Domaine fréquentiel (« spectre »)

Domaine Temporel (« instantané »)

Domaine Temporel (« moyenne»)

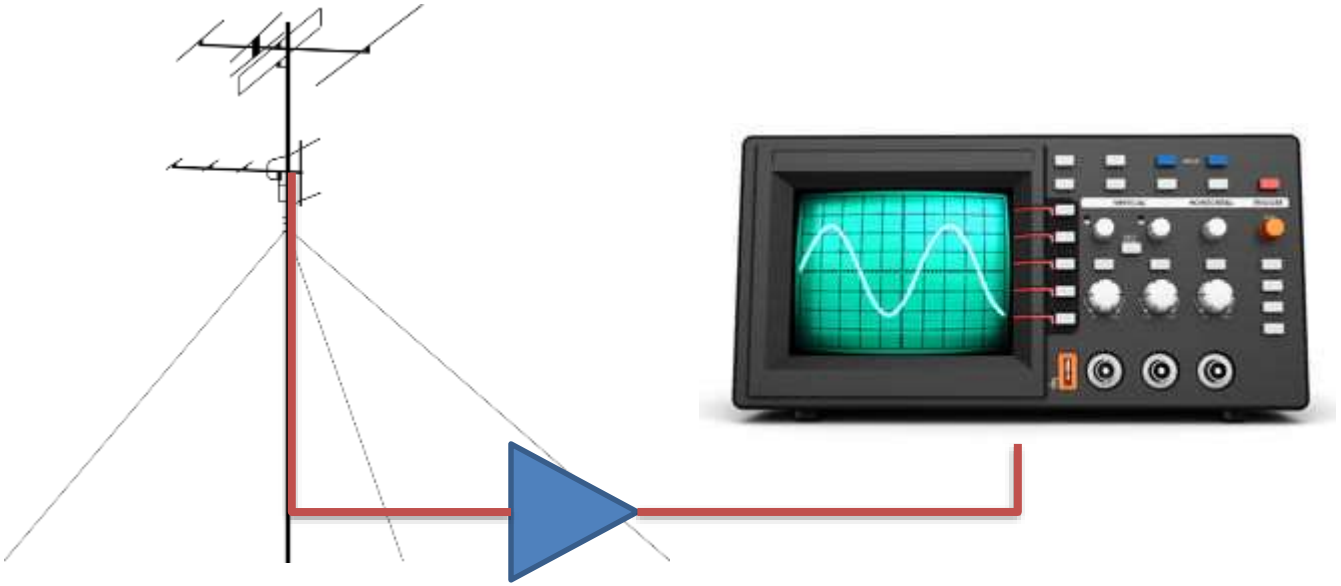
Signal continu

- Continu au sens « temporel » : Un signal est continu s'il est connu à chaque instant ***t***. (comprendre « à n'importe quel instant »)

- « quel que soit le niveau de zoom (temporel), je connais toujours la valeur du signal »
- « quel que soit l'endroit où je regarde »

Signal continu

- Exemples : la température qu'il fait, la vitesse du vent, … mais aussi bien sûr la tension (en volts) au bornes de l'antenne

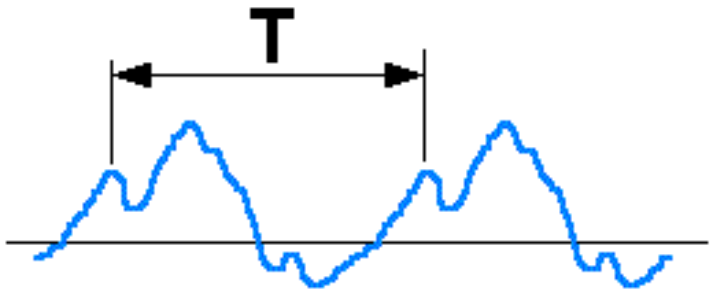


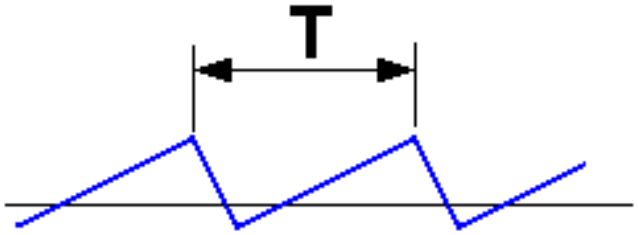
Signal discret

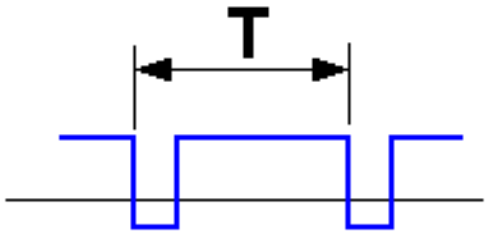
- **Quand mesurer** un signal continu ???
  - On estime la valeur « *à un instant* ***t*** » de la grandeur qui nous intéresse,
  - Que peut-on alors dire de la valeur du signal continu « avant » et « après » la mesure ???
  - **À quelle cadence (tous les combiens de temps) faut-il mesurer la grandeur qui nous intéresse ?**

Signal périodique

- Un signal est dit **périodique** si les variations de son amplitude se reproduisent régulièrement au bout d'une période ***T*** constante







On sait que le signal est périodique si on l'observe pendant « assez longtemps »

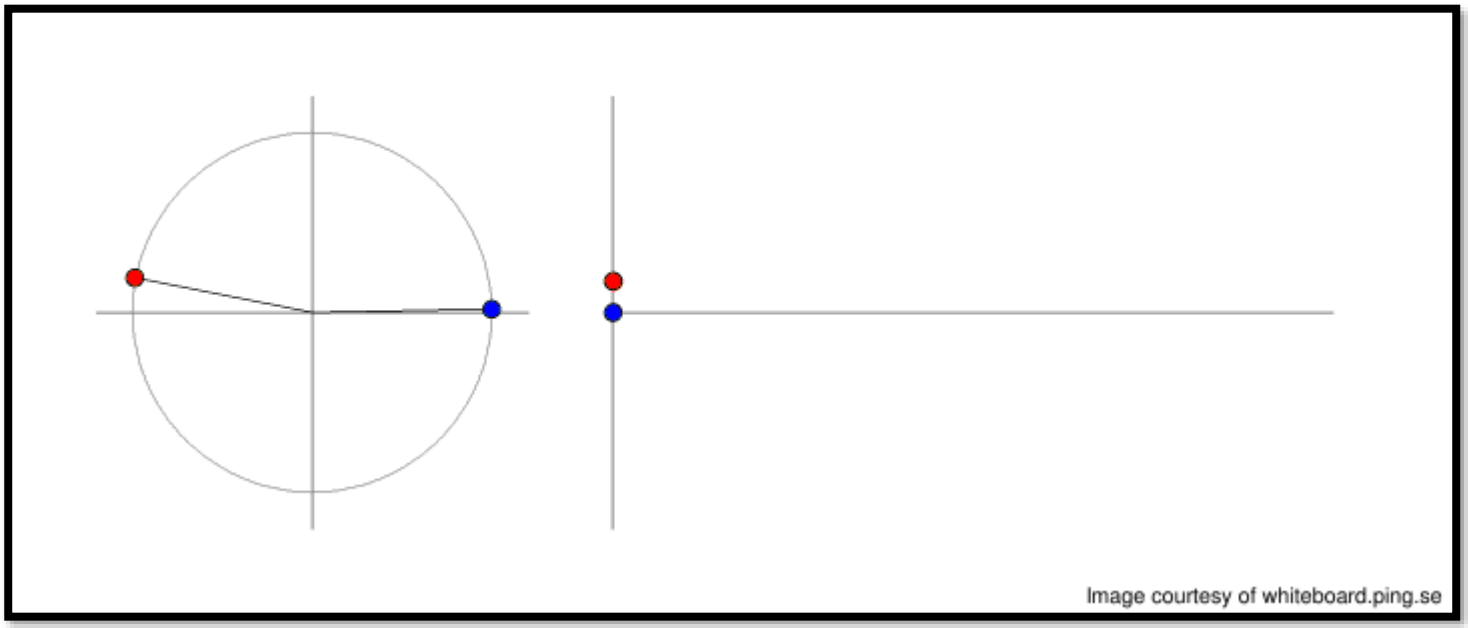
→Si on observe pendant T, on ne peux pas être sûr que le signal se répète après… Pour une observation de durée T, la plus grande période observable est donc T/2

Nyquist et Shannon

- On se fixe une durée de l'observation T
- Il suffit de **2 mesures pendant le temps T** pour connaître un signal
- Pour plus de détails sur la partie « théorique », voir la présentation faite à F8KCF en mars 2017 :

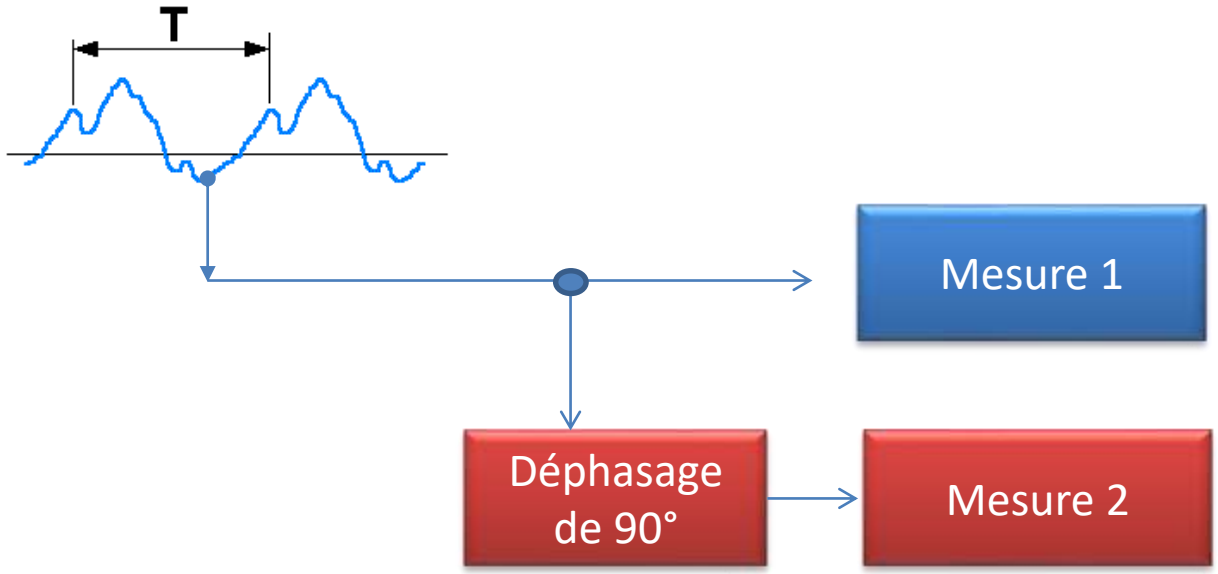
http://f8kcf.free.fr/index.php/the-news/115-conf-sdr-docs

2 mesures… pour quoi faire ?



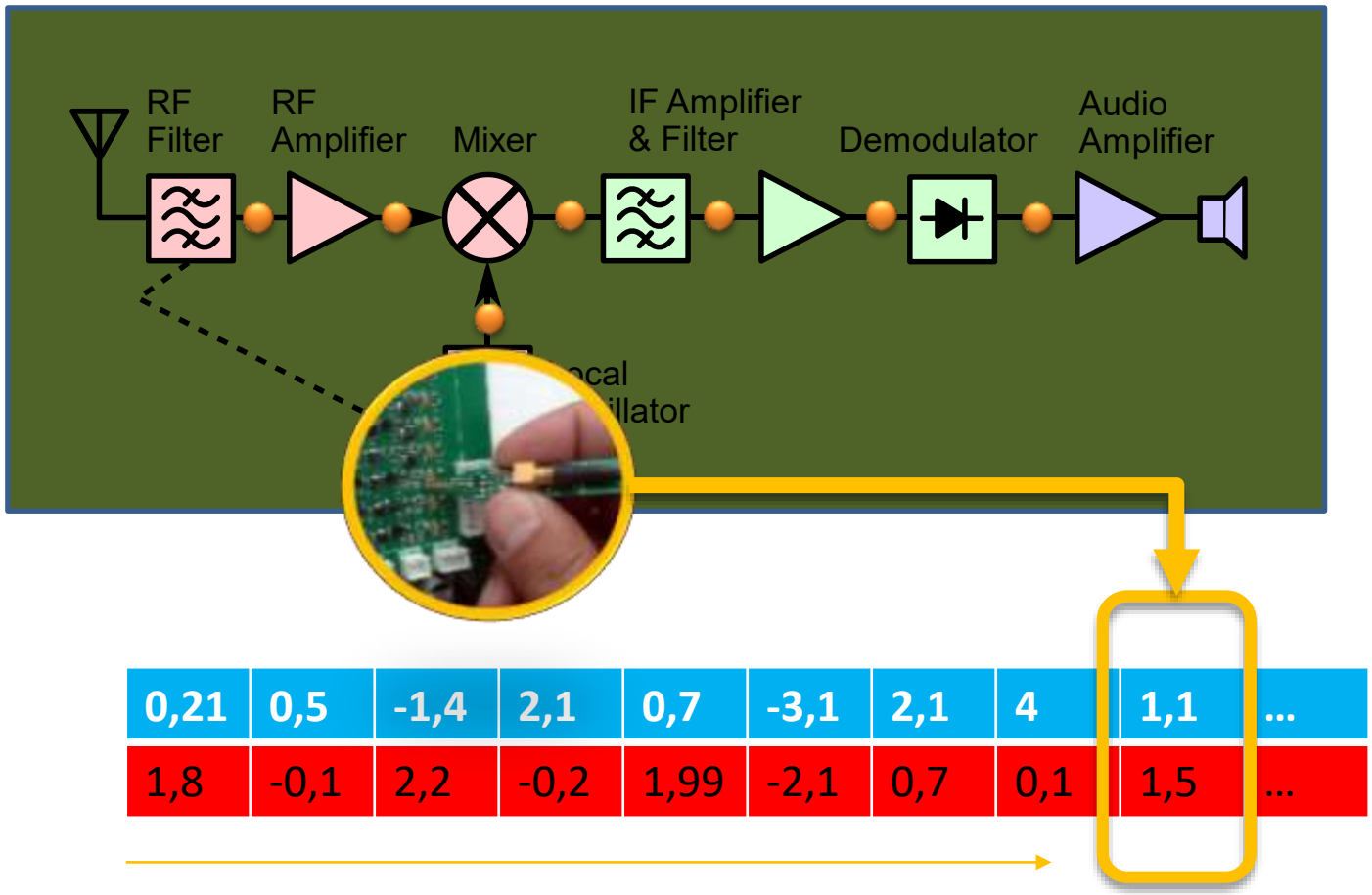
- Les deux signaux n'évoluent pas « dans le même sens » et pourtant leur représentation temporelle est identique (donc ça va pas)

En pratique on fait



- Le signal à quantifier est déphasé de 90° puis mesuré
- On parle de signaux « IQ » ou « Complexes »

Information = échantillons (volts)



On traite une série de valeurs mesurées (échantillons) et on applique des opérations sur ces valeurs

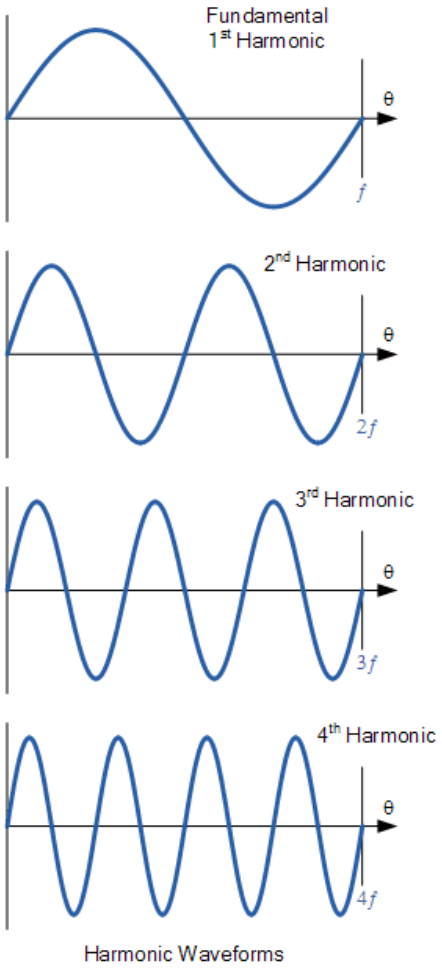
2 mesures : IQ

- Deux mesures permettent de savoir dans quel sens « ça tourne »
- Quand ça tourne dans le sens des aiguilles d'une montre : la fréquence est « positive »
- Quand ça tourne dans l'autre sens, la fréquence est « négative »

- Exemple : VFO sur 7.100 MHz:
  - Un signal à 7.150 est vu à « +50 kHz »
  - Un signal à 7.050 est vu à « -50 kHz »

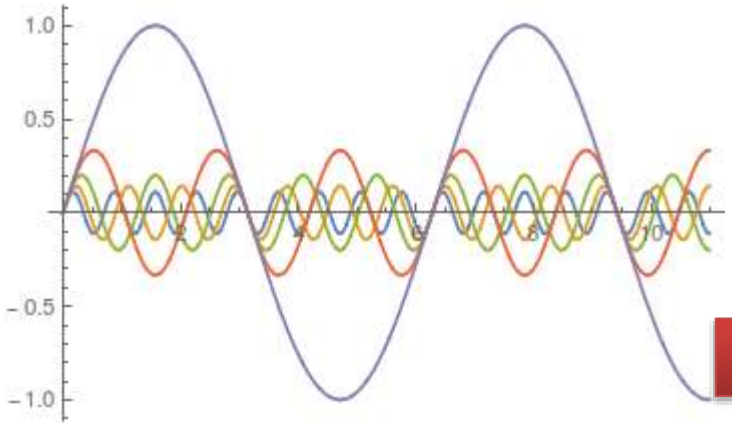
Harmoniques

Une harmonique d'un signal ***s*** est un signal dont la fréquence est un multiple **entier** de la fréquence de ***s***

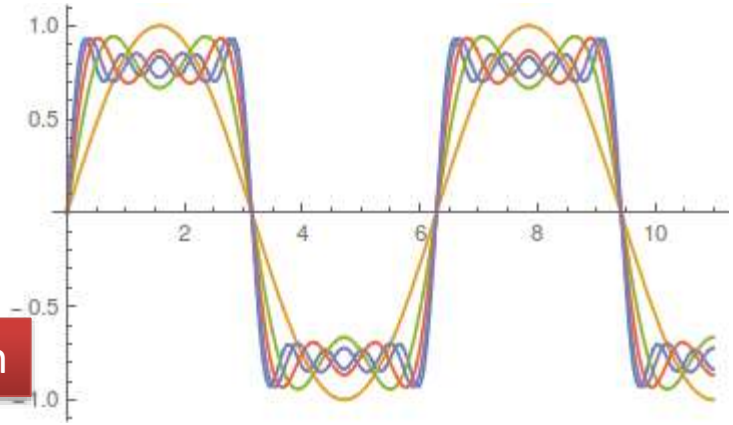


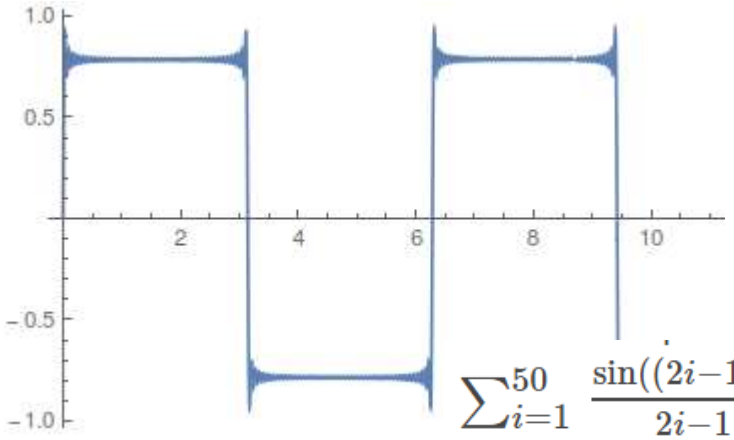
Harmonic Waveforms

Combinaison de signaux harmoniques

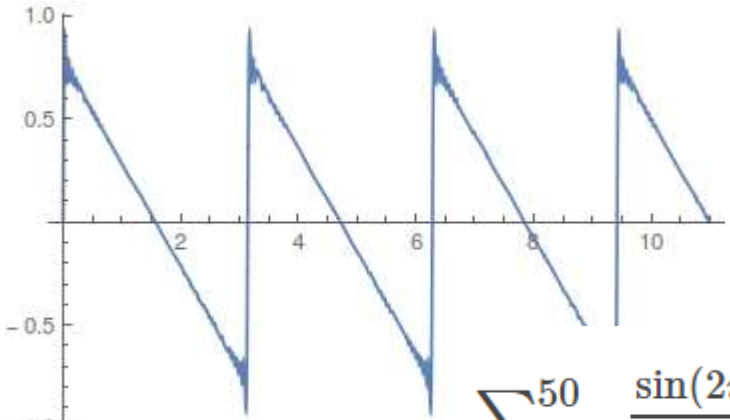


combinaison





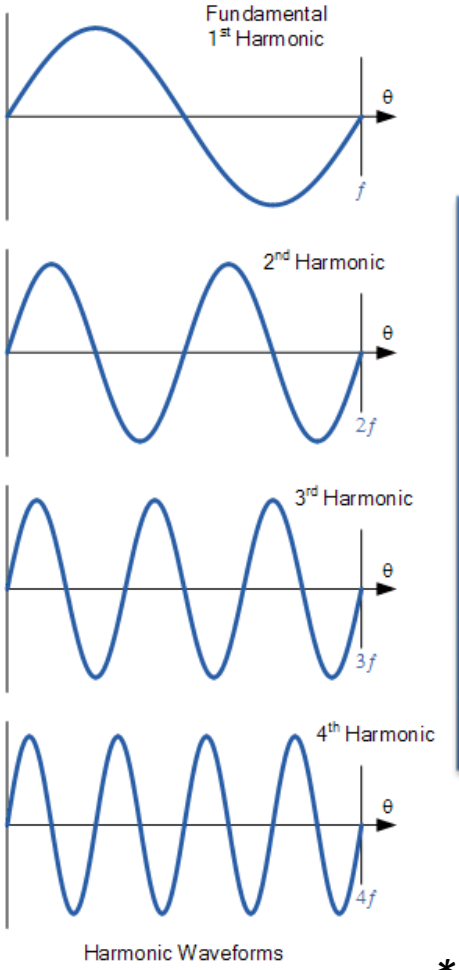
$$\sum_{i=1}^{50} \frac{\sin((2i-1)x)}{2i-1}$$



$$\sum_{i=1}^{50} \frac{\sin(2ix)}{2i}$$

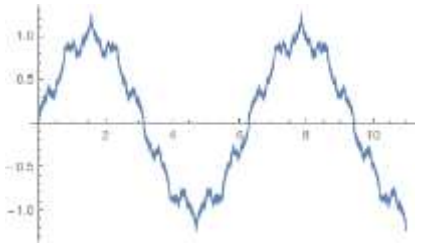
Source: http://hawaiireedlab.com/wpress/?p=1693

Synthétiser un signal



Harmonic Waveforms

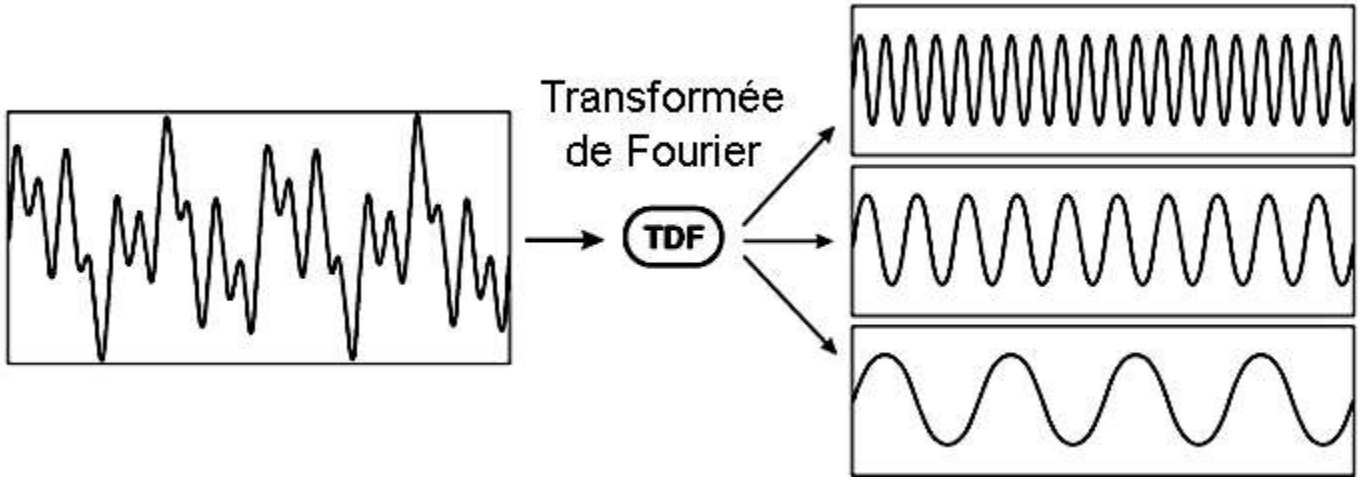
En combinant « comme il faut » des signaux sinusoïdaux* harmoniques, on peut « synthétiser » d'autres signaux périodiques (mais pas des sinusoïdes)



**Remarque : on peut aussi utiliser autre chose que des sinusoïdes…*

Décomposer un signal

A partir d'un signal d'entrée, retrouver les « ingrédients » qui ont servi à le générer
C'est le processus inverse de ce qui a été vu précédemment



La transformée de Fourier (joseph) permet de faire cette décomposition

Synthétiser/Décomposer

Représentation temporelle

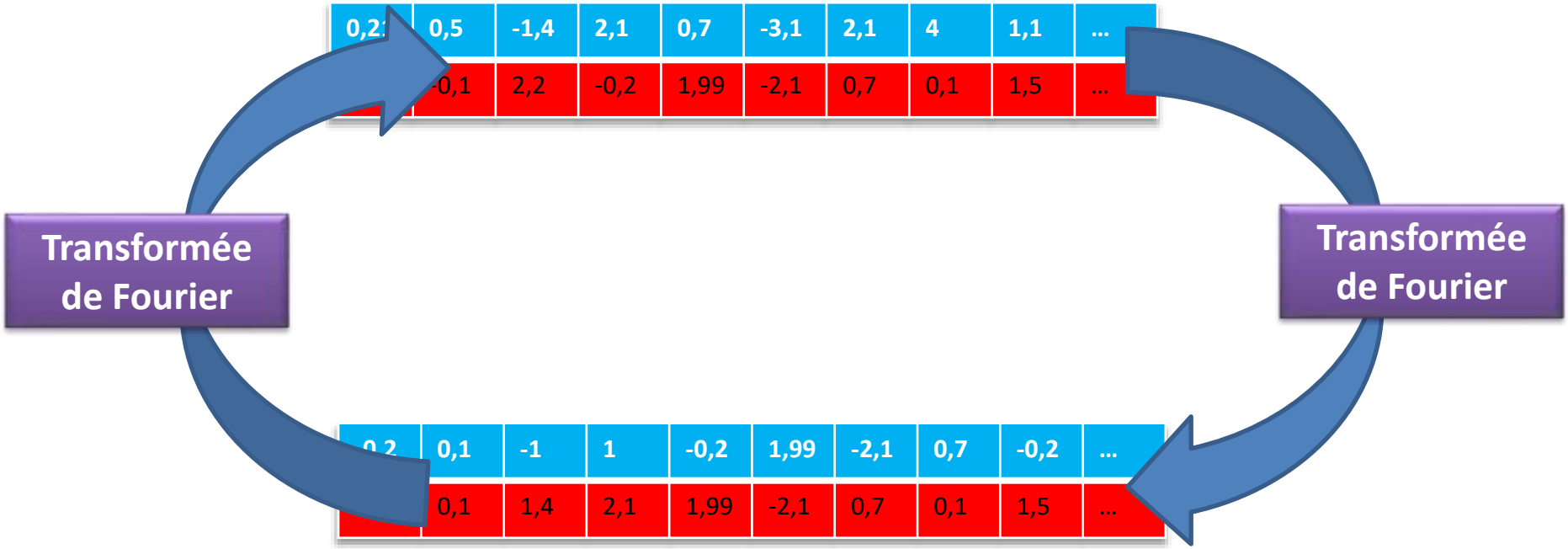
Transformée de Fourier

Transformée de Fourier

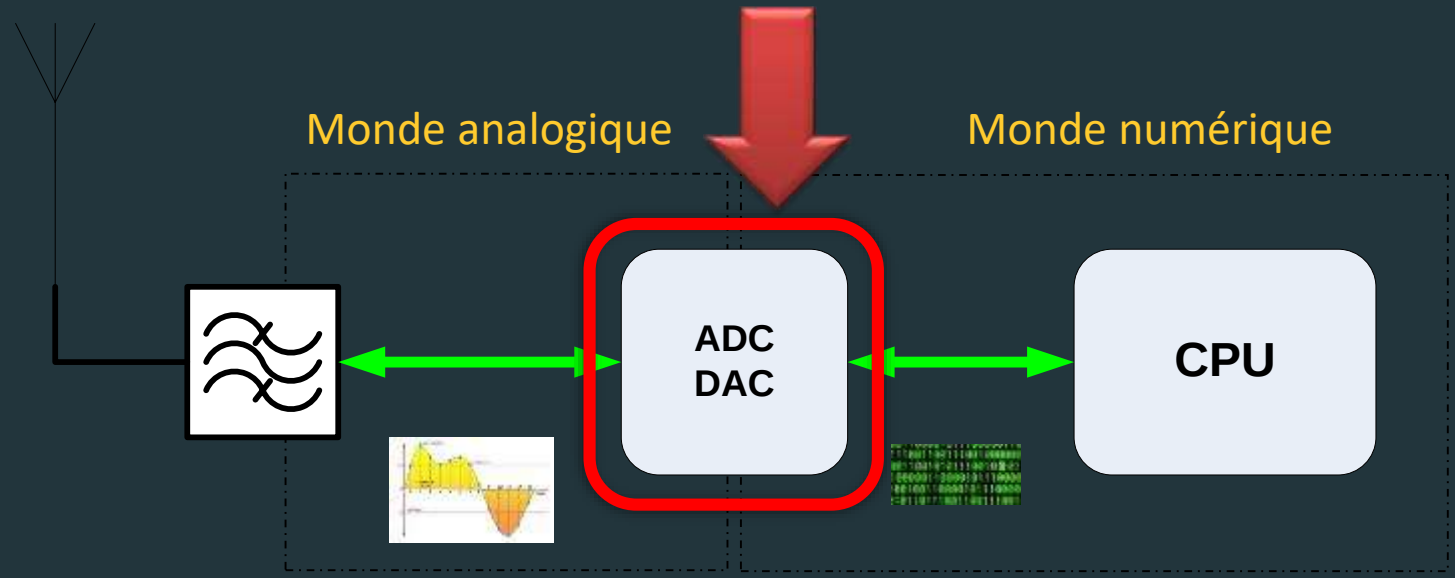
Représentation fréquentielle

Synthétiser/Décomposer

Domaine temporel : un échantillon = une mesure à un instant donné



Domaine fréquentiel : un échantillon = une « certaine intensité et phase » de signal à une « certaine fréquence »



De l'analogique au numérique

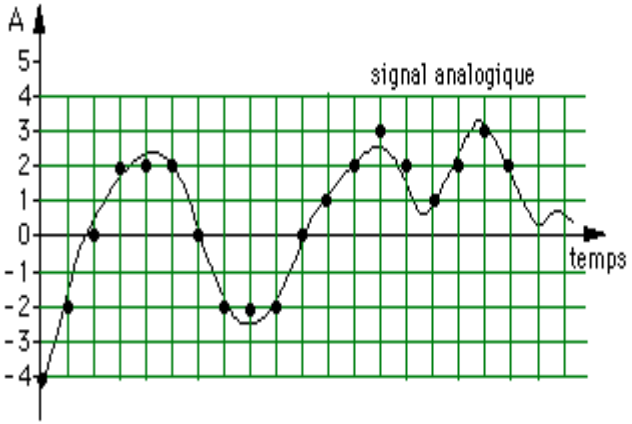
(*« du coax aux signaux »*)

Conversion analogique → numérique

- On a besoin d'une horloge d'échantillonnage régulière à une fréquence F_{ech}
- Comme il faut 2 mesures… :
  - Si on a 1 seul ADC, il faudra 2 échantillons qui permettront de numériser jusqu'à $F_{ech}/2$.
  - Que si on a 2 ADC et un déphaseur 90°, 1 seul échantillon **PAR ADC** permettra d'échantillonner jusqu'à F_{ech}

Dans les deux cas, on a $2*F_{ech}$ mesures

La conversion analogique / numérique

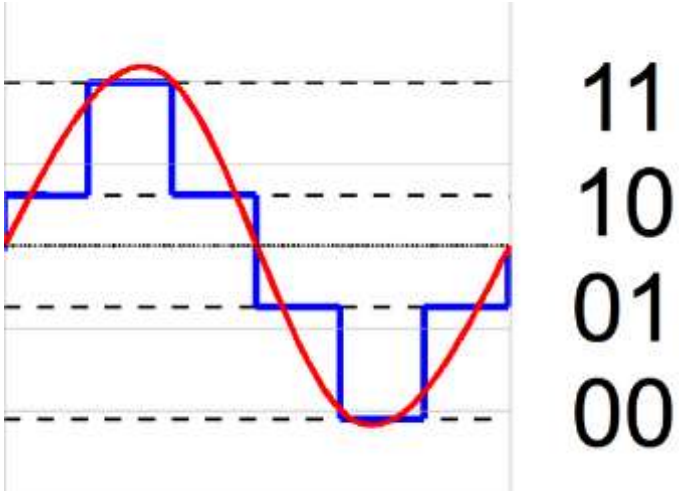


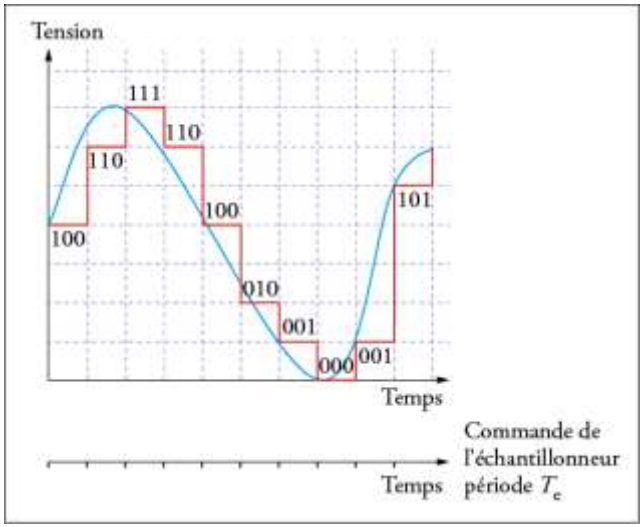
t	val
0	-4
1	-2
2	0
3	2
4	2
5	2
6	0
7	-2
8	-2
9	-2
	...

On va « mesurer » la tension entrante à des instants précis.

Les ADC qui comportent un étage d'échantillonnage blocage sont les bienvenus… la valeur à mesurer ne doit pas changer pendant la mesure.

La conversion analogique / numérique

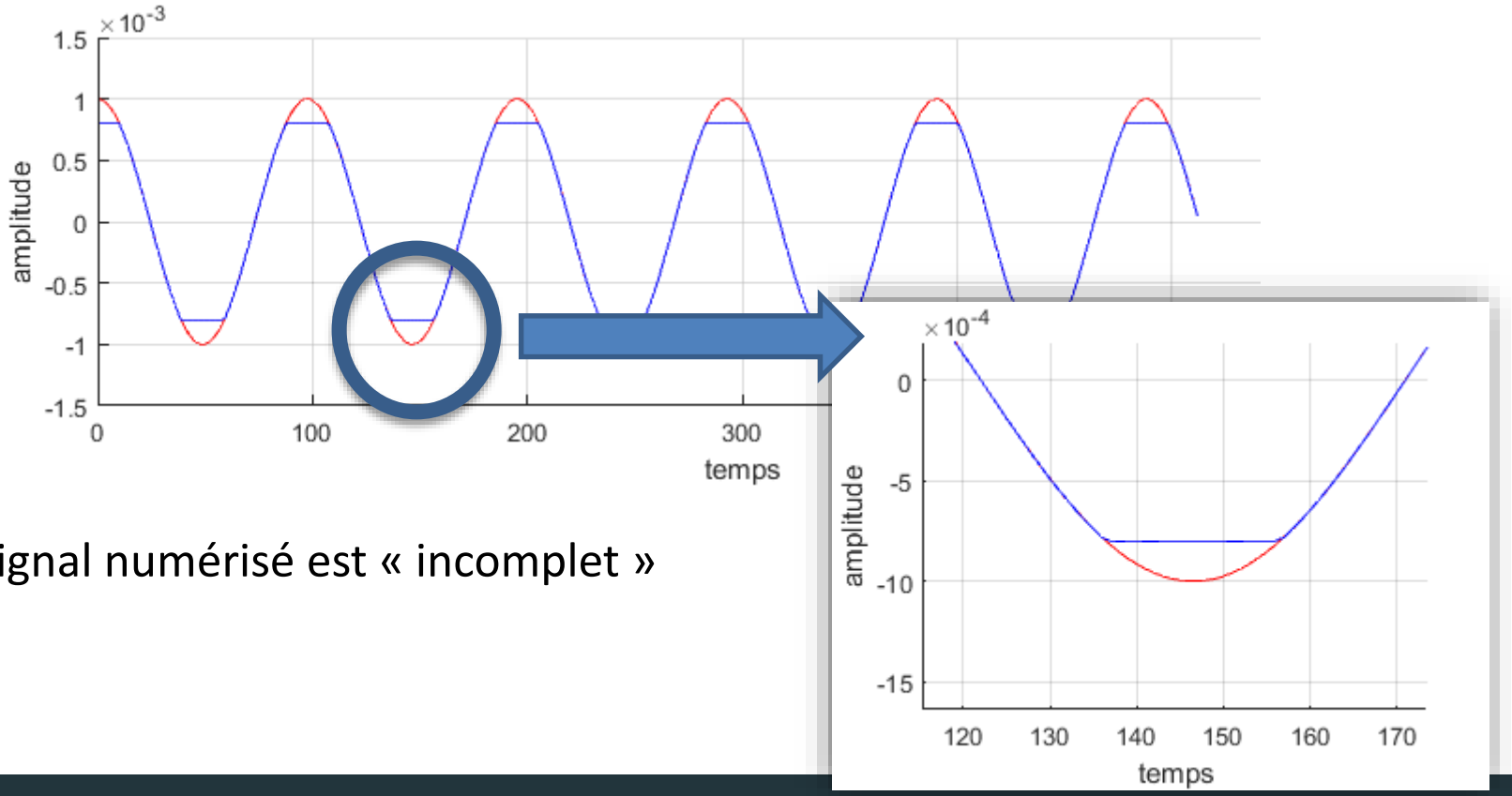




En fonction de la **résolution** du circuit (nombre de bits), on obtiendra une représentation plus ou moins « fidèle »

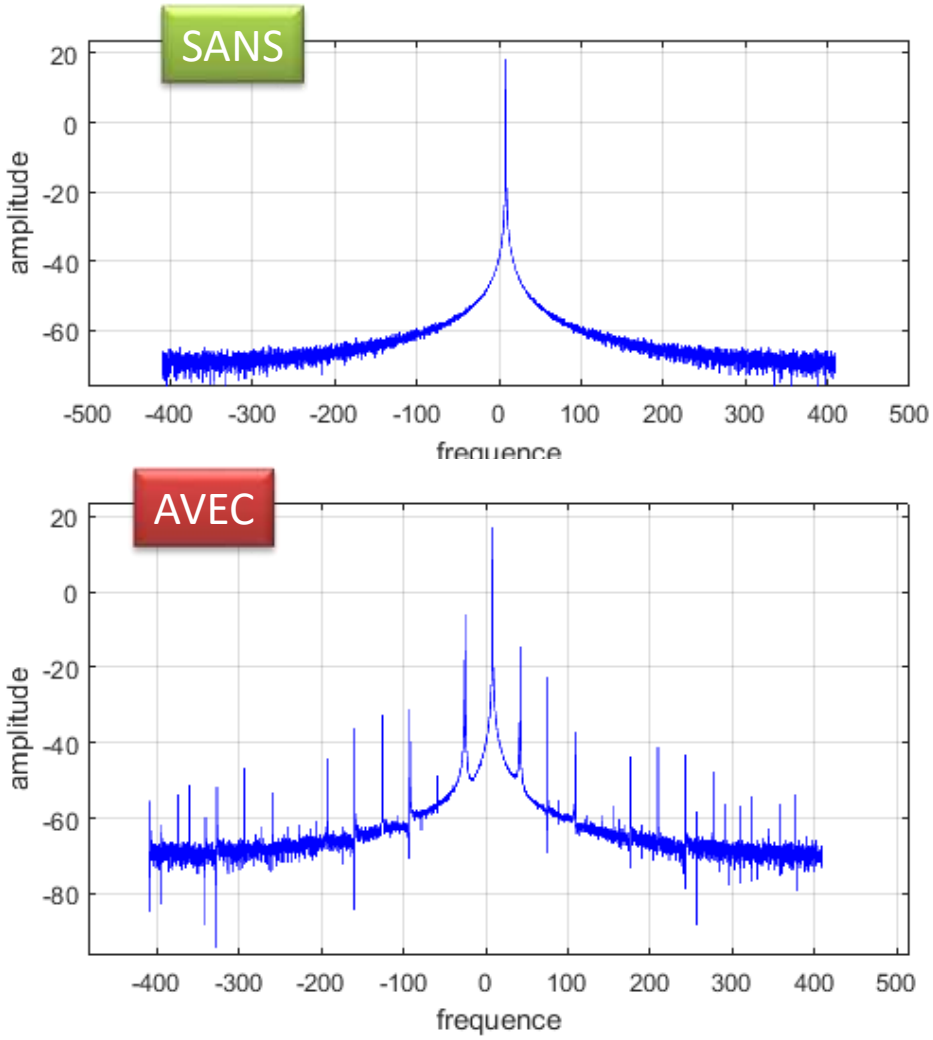
Saturation à l'entrée

Gain trop important ou signal entrant trop fort avec saturation de l'amplificateur, ou dépassement VREF adc :



Le signal numérisé est « incomplet »

Saturation



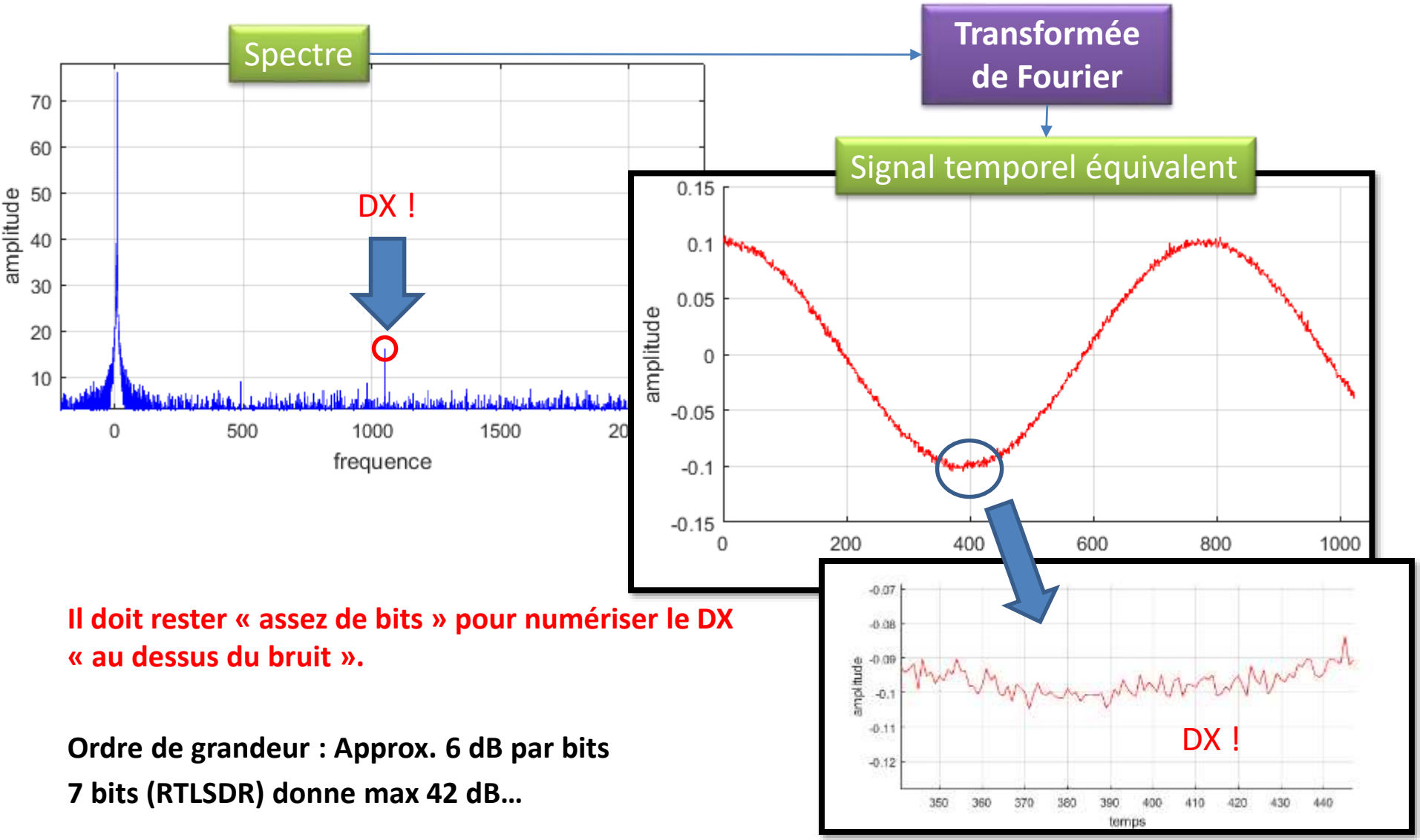
- Apparition de signaux « fantômes » sur le spectre,
- Apparition de porteuses audibles après démodulation

Dynamique

Contexte du problème:

- Deux signaux à l'entrée: un signal « utile » très faible et un « brouilleur » très fort (exemple: station DX européenne sur 7MHz à côté émission AM très forte)
- L'ADC doit avoir « assez de bits » pour numériser les deux

Dynamique

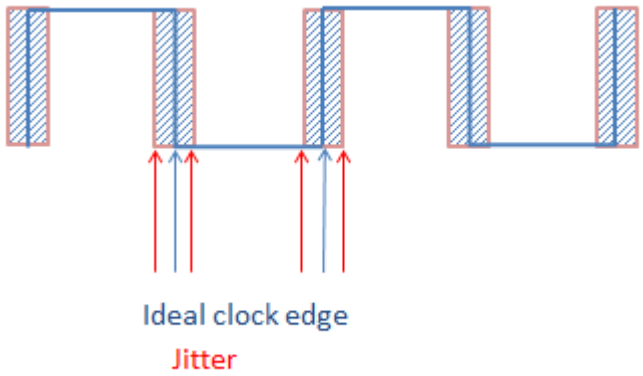


Il doit rester « assez de bits » pour numériser le DX « au dessus du bruit ».

Ordre de grandeur : Approx. 6 dB par bits

7 bits (RTLSDR) donne max 42 dB…

Horloge de numérisation : *Jitter*

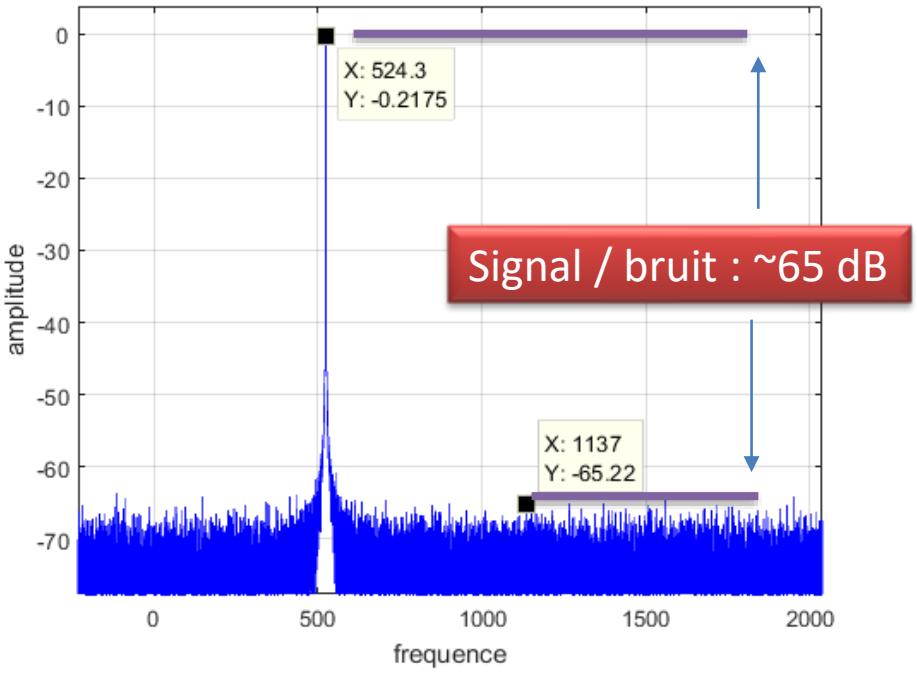


Le bruit d'horloge d'échantillonnage se traduira par une « prise de mesure » à un autre instant que celui qui était prévu.

Il vaut mieux une horloge stable qu'une horloge juste…

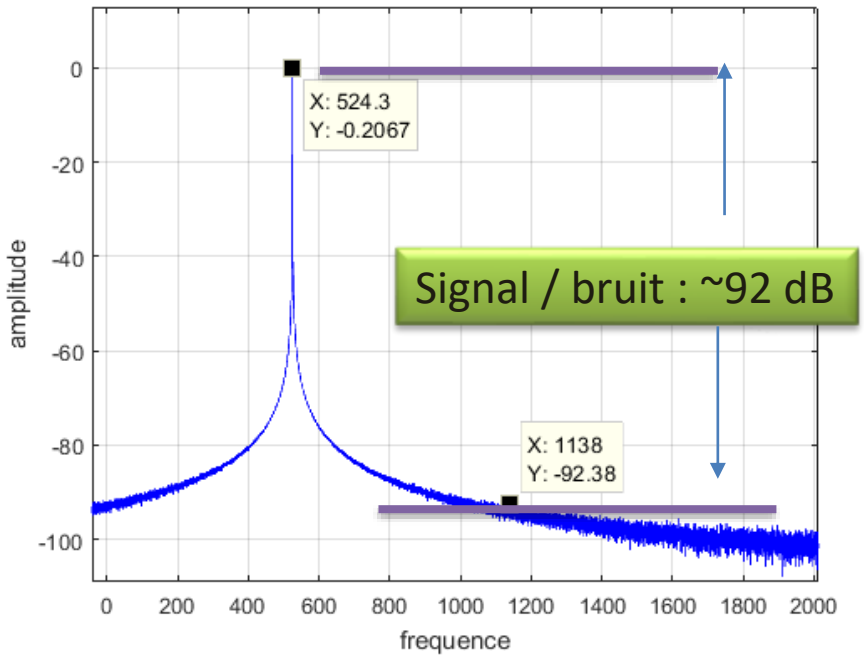
Horloge de numérisation : *Jitter*

±5 µs d'erreur sur Fclock=100 KHz :



Signal / bruit : ~65 dB

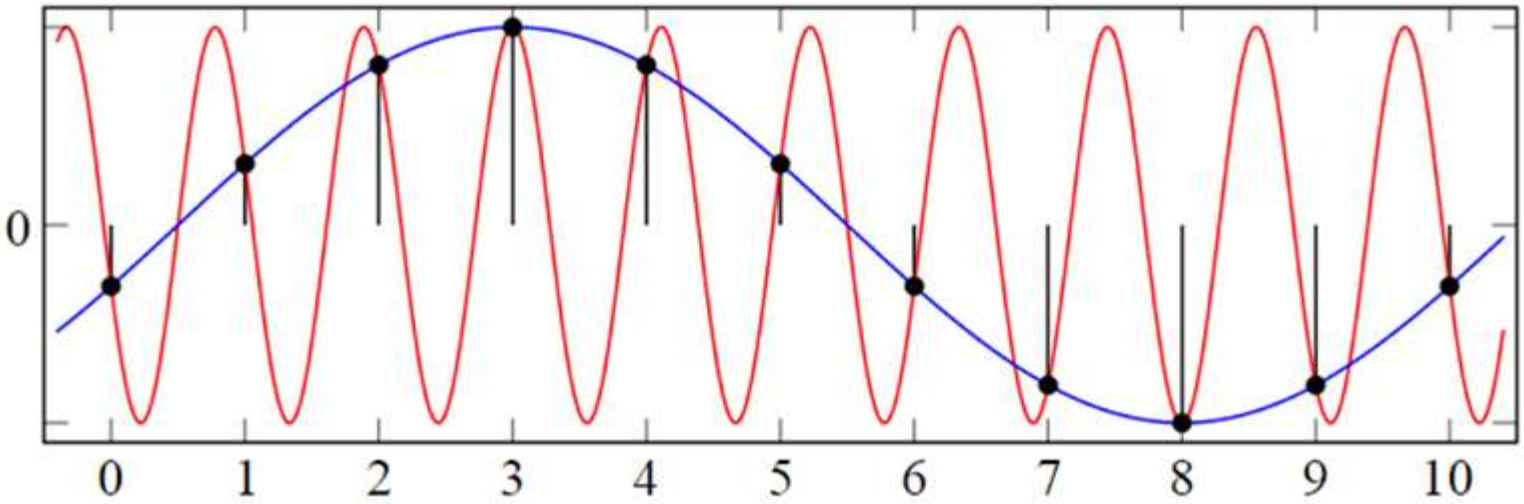
±5 ns d'erreur sur Fclock=100 KHz :



Signal / bruit : ~92 dB

Dans cette simulation, la différence est de l'ordre de 30 dB !

Aliasing

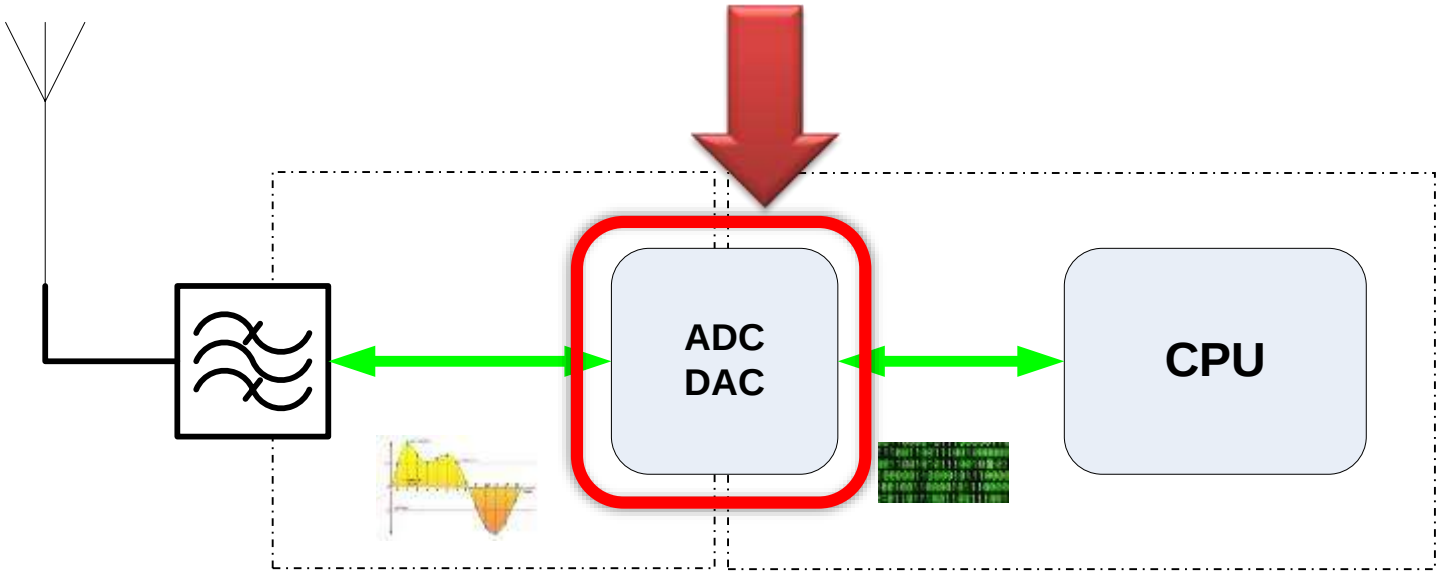


- S'il reste à l'entrée de l'ADC des signaux de fréquence supérieures à Nyquist, il seront numérisés et se retrouveront dans le spectre avec une fréquence apparente (alias) différente;

- D'où l'importance du filtre avant numérisation.

Synthèse sur les ADC

- **C’est le point critique du récepteur SDR**: signal mal échantillonné difficilement « rattrapable » par traitement ensuite,
- Compromis à trouver entre résolution, fréquence et d’échantillonnage,
- Soigner la stabilité de l’horloge d’échantillonnage



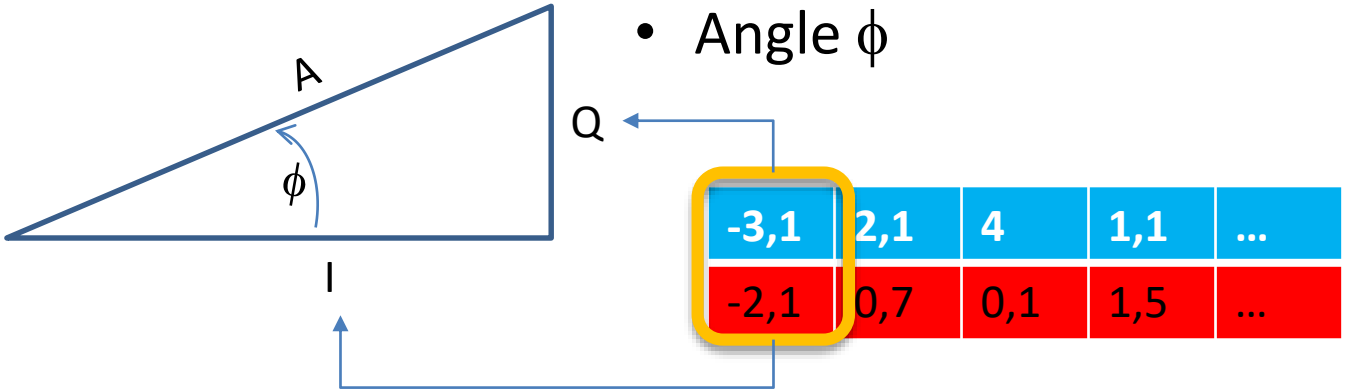
Pourquoi tant de différence de prix ?

SDR	Tune Low (MHz)	Tune Max (MHz)	RX Bandwidth (MHz)	ADC Resolution (Bits)	Transmit? (Yes/No)	Price ($USD)
RTL-SDR (R820T)	24	1766	3.2 / 2.56 Stable	8	No	~20
Funcube Pro+	0.15 410	260 2050	0.192	16	No	~200
Airspy	24	1800	10	12	No	199
SDRPlay	0.1	2000	8	12	No	149
HackRF	30	6000	20	8	Yes	299
BladeRF	300	3800	40	12	Yes	400 & 650
USRP 1	DC	6000	64	12	Yes	700


Seconde partie
Le logiciel

I & Q sont les ingrédients de base

- Amplitude = longueur de A
- Angle ϕ



- Un peu de trigonométrie (Pythagore):

$$\boldsymbol{A} = \sqrt{I^2 + Q^2} \qquad \boldsymbol{\phi} = atan\left(\frac{Q}{I}\right)$$

Et donc ? AM … FM …

- **AM** = Amplitude Modulation = Variation de l'amplitude au cours du temps
  - Démoduler l'AM = calculer ***A*** à chaque instant et envoyer à la sortie audio

- **FM** = Modulation de Fréquence (de Phase)
  - Démoduler la FM = calculer ϕ à chaque instant et envoyer *la variation* à la sortie audio

I et Q sont les ingrédients de base

-3,1	2,1	4	1,1	…
-2,1	0,7	0,1	1,5	…

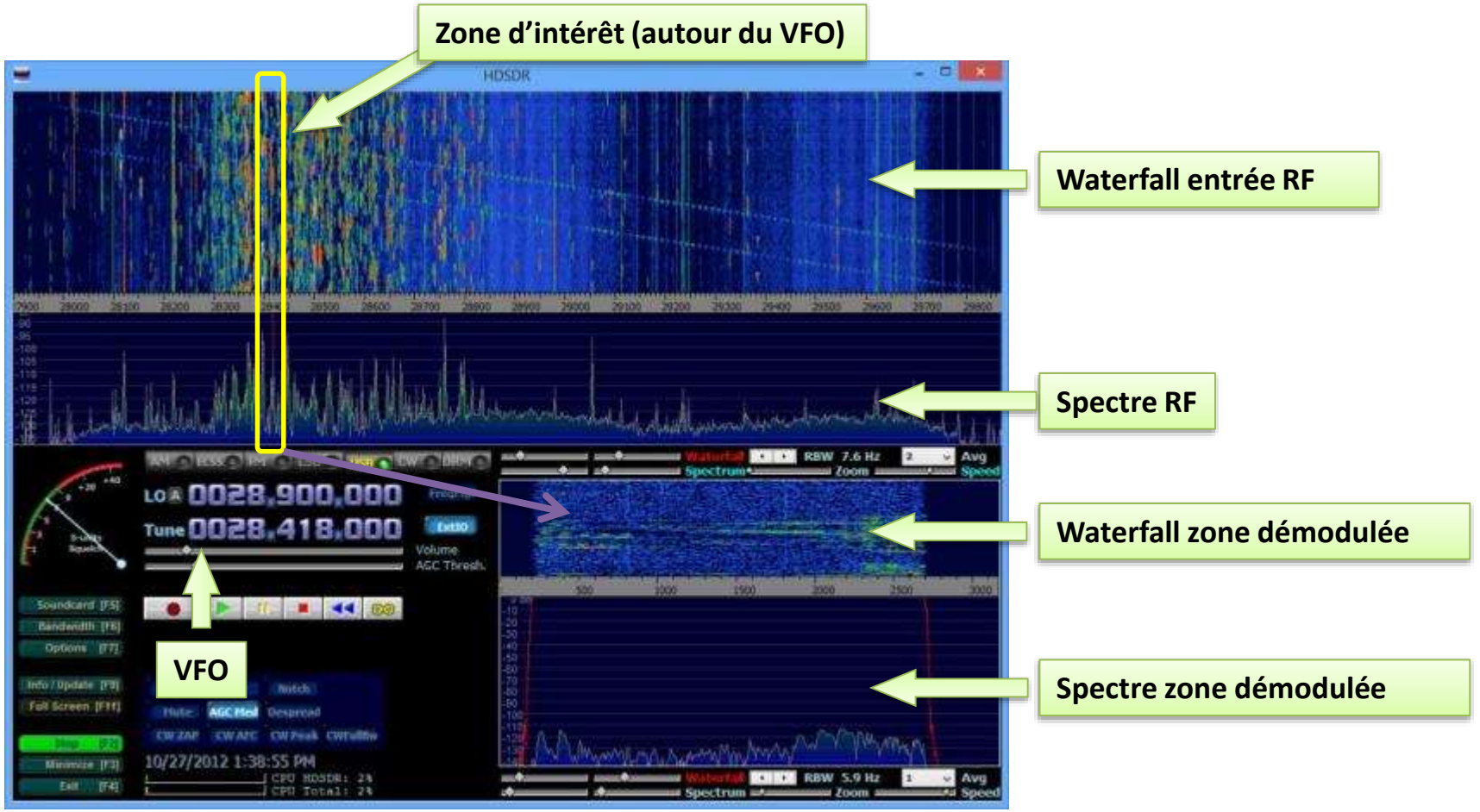
$$A = \sqrt{I^2 + Q^2}$$

$$\phi = atan\left(\frac{I}{Q}\right)$$

Audio

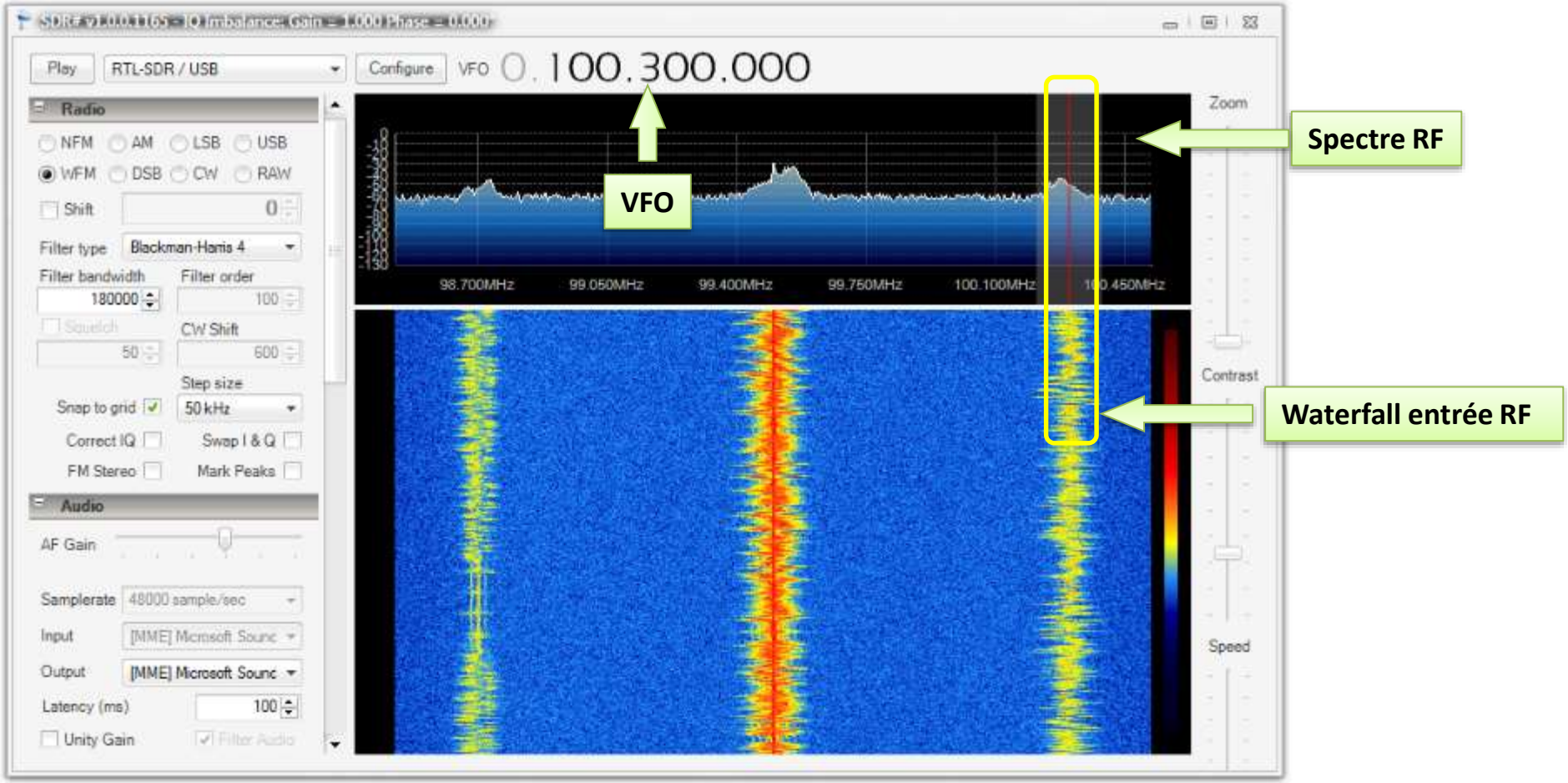
- Chaque modulation correspond à un sous-programme de calcul dédié
- En fonction du choix de l'utilisateur le flux de données IQ est envoyé vers le bloc correspondant

Logiciel SDR : exemple HDSDR

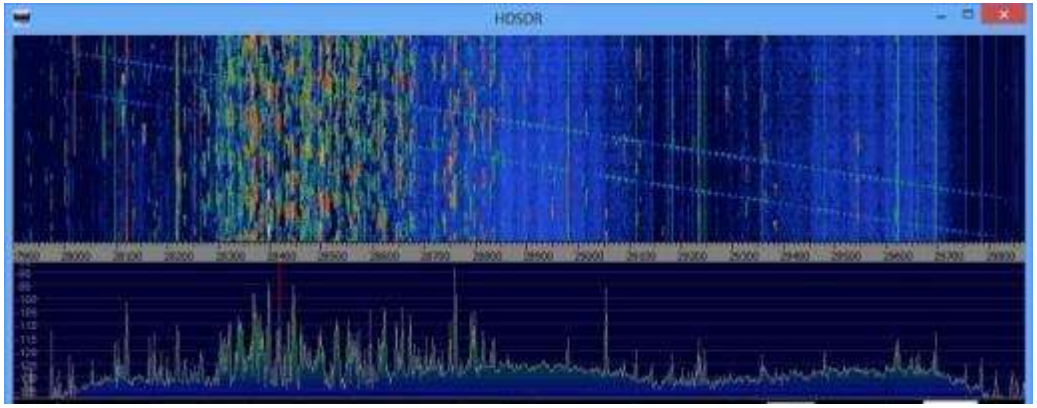


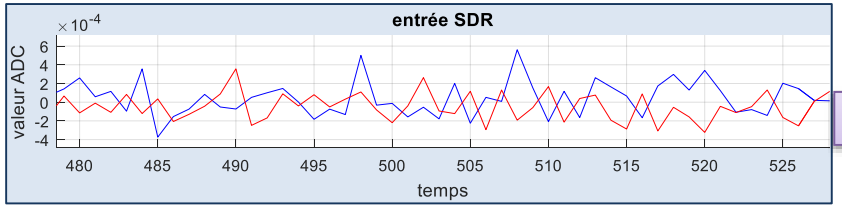
http://www.hdsdr.de/

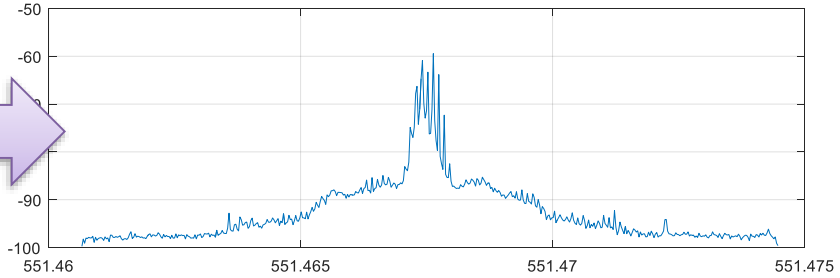
Logiciel SDR : exemple SDR#



Affichage du spectre



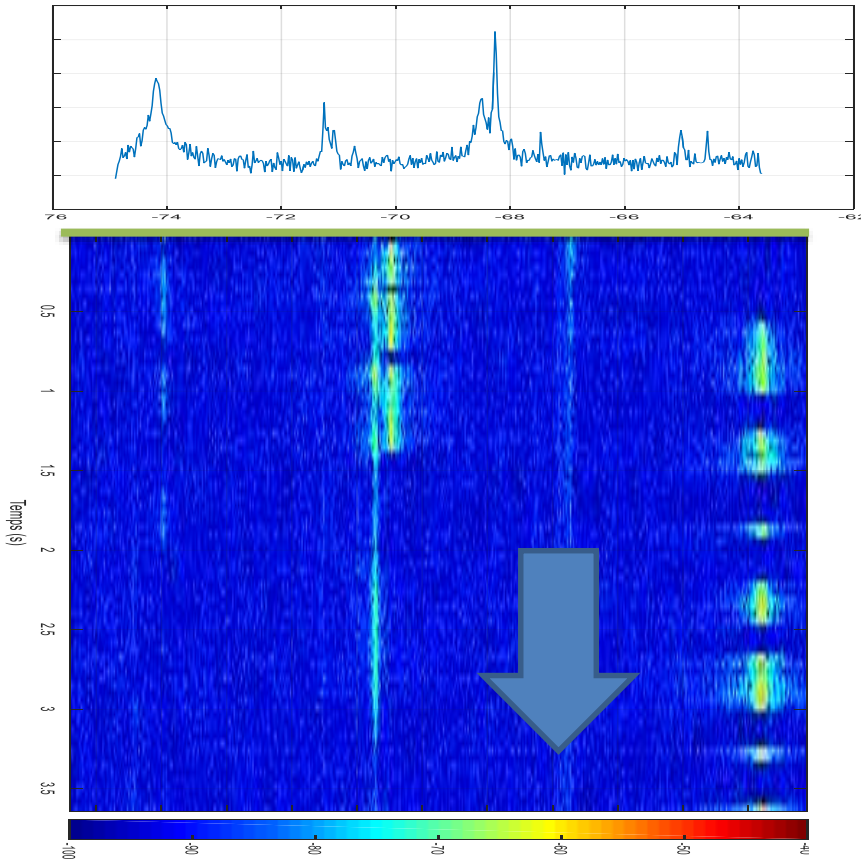




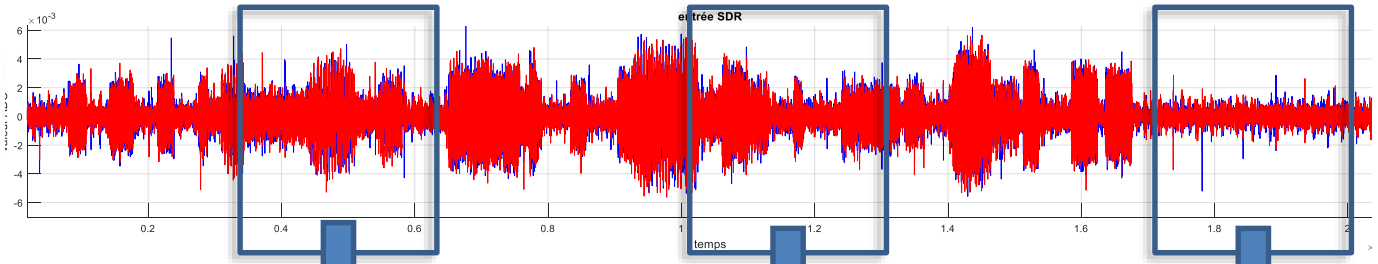
- Transformation « temporel » -> « fréquentiel »

Le Waterfall (chute d'eau)

- Juste un empilage de spectre…
- On calcule des FFT en continu en prenant des « morceaux » du signal qui arrive du récepteur
- On peut aussi moyenner plusieurs FFT pour faire « sortir » des signaux faibles



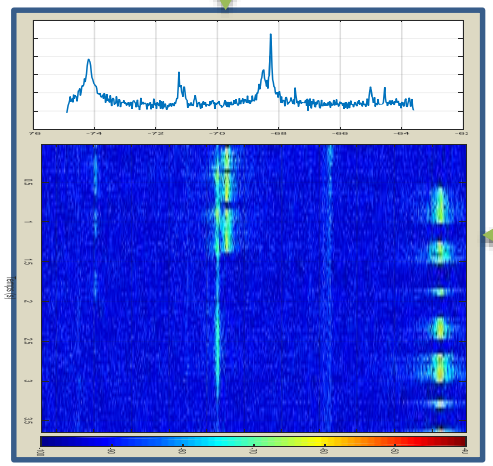
En résumé



FFT

FFT

FFT



La transformée de Fourier *rapide* (FFT)

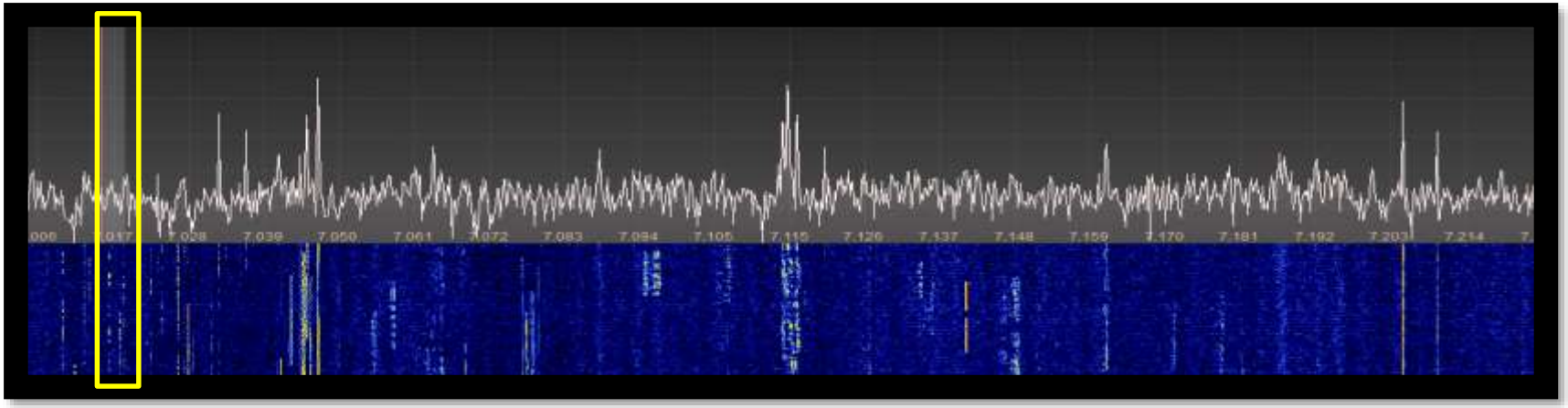
- Mesure de la complexité d'un algorithme : nombre de multiplications nécessaires pour faire le traitement demandé

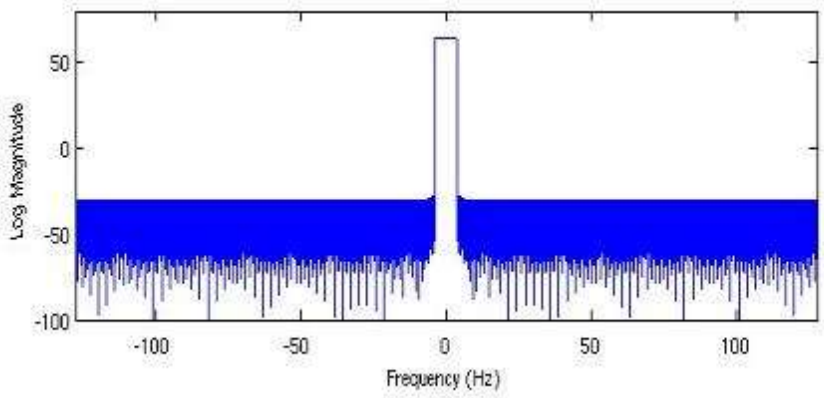
- FFT : Algorithme de calcul optimisé pour des suites de valeurs (des échantillons). Ne requiert « que » $n.\log_2(n)$ multiplications
- Optimisé pour des tailles puissances de 2 (..512,1024,2048 etc.)

- Exemples : SDR à 2 MHz
  - n=1024, $\log_2(n)=10$ il faut environ 10240 multiplications
  - n=4096, $\log_2(n)=14$ il faut environ **57344** multiplications

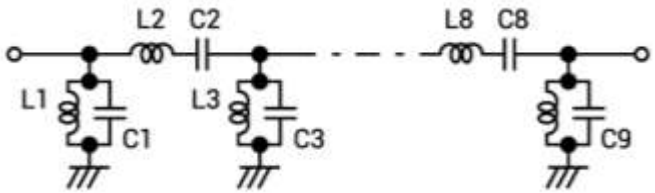
  - n=1024 donne une résolution de 2M/1024 = 1,95 KHz
  - n=4096 donne une résolution de 2M/4096 = 488 Hz

Extraction de la bande « utile »





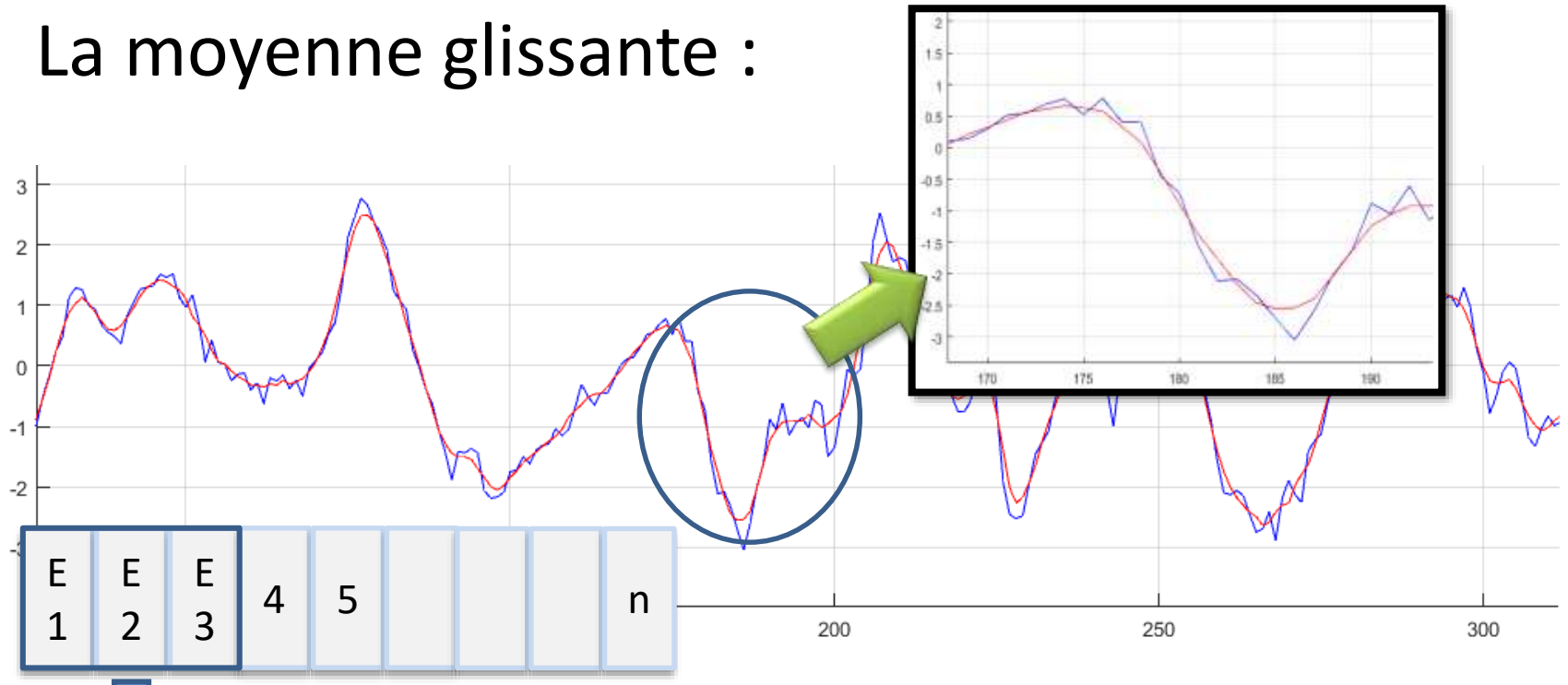
==



On a donc besoin d'un filtre qui va atténuer tout ce qui est « hors bande »

Exemple de FIR

- La moyenne glissante :



E 1	E 2	E 3	4	5				n

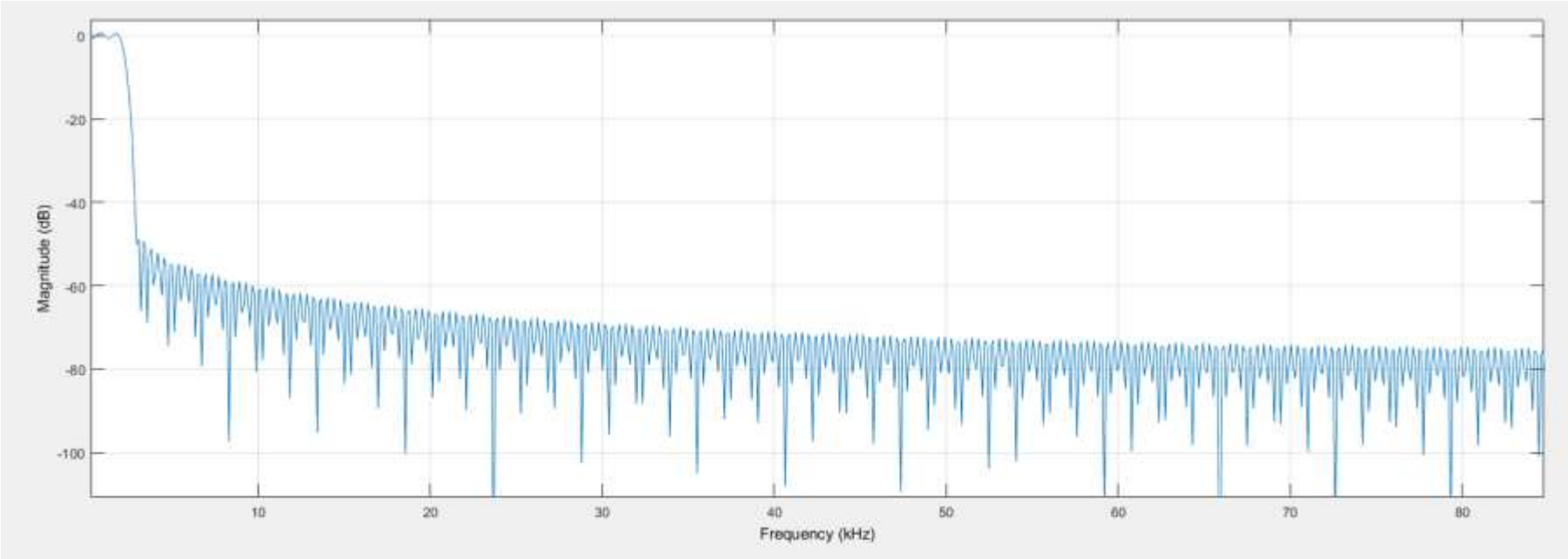
$$S_3 = \frac{E_1 + E_2 + E_3}{3}$$

S 1	S 2	S 3	S n

2 additions + 1 multiplication par échantillon

Avoir la bonne longueur…

- Combien de coefficients faut il ?



Exemple ici : 2MHz de bande en entrée, on veut 3KHz de bande

Il faut… 5060 coefficients

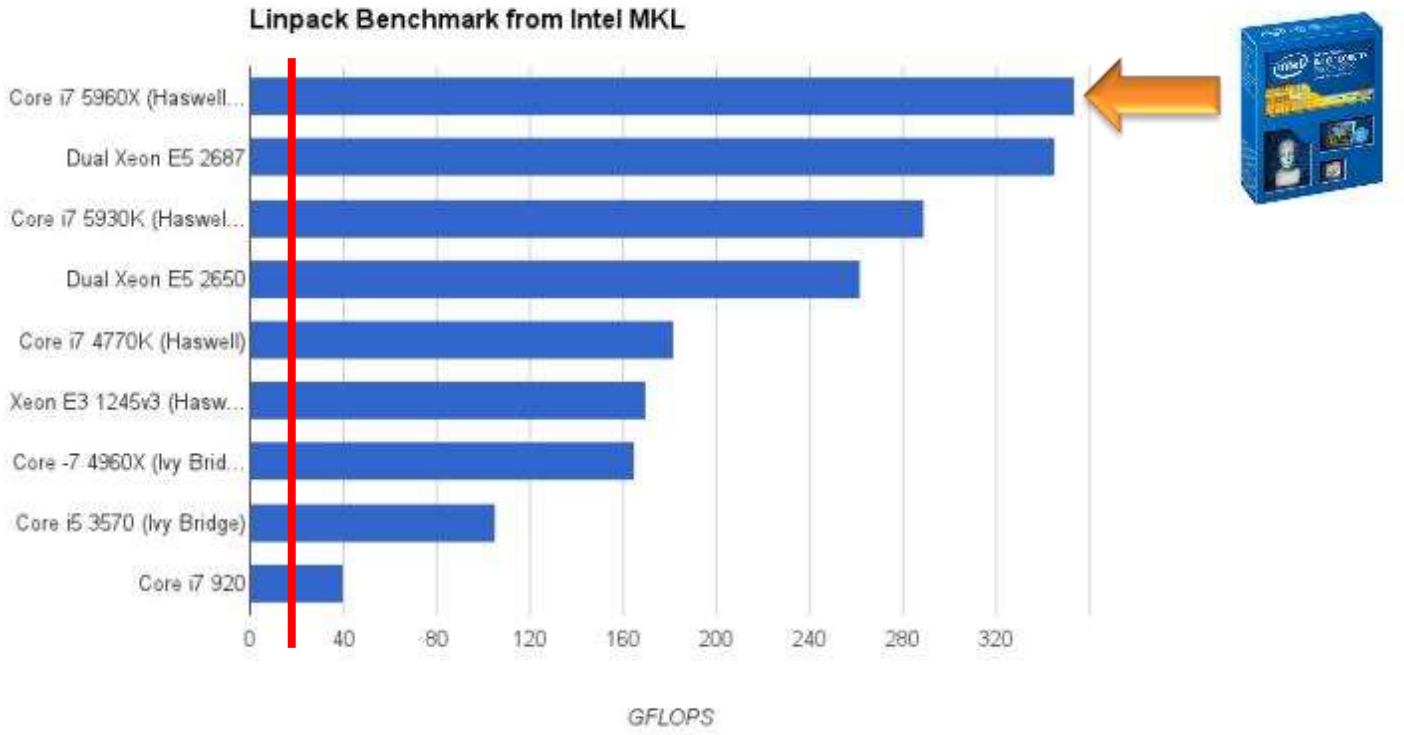
5060 coefficients, et alors ?

- Avec 2 millions d'échantillons par seconde (IQ…) , on a donc 4 millions de valeurs nouvelles par seconde à traiter
- Chaque valeur doit être multipliée par un filtre de longueur 5060
- Ça fait donc : (4 million)x(5060) = 20,2 Milliards de multiplication par seconde = 20 « gigaFlops »

Pour « à peine » 50 dB de réjection hors bande… avec tout juste 2MHz de bande en entrée…

20 Giga Flops pour filtrer ?

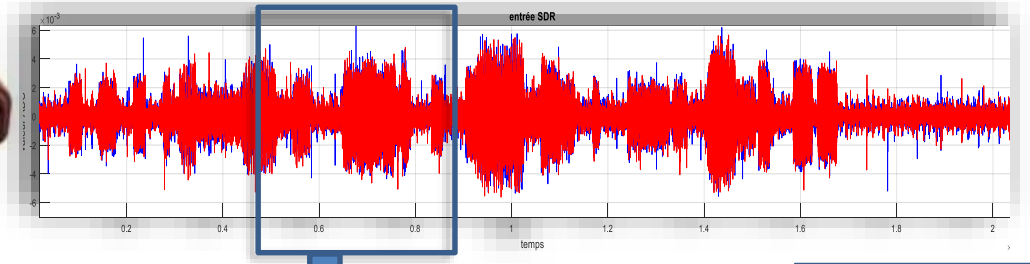
A part sur un processeur « haut de gamme », on consomme la majorité de la puissance



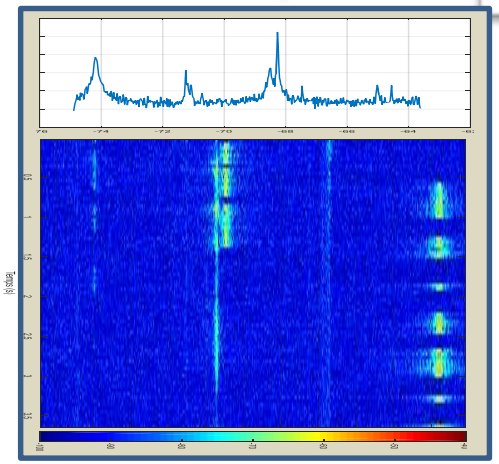
Conclusion… faut pas faire comme ça !

Il existe des algorithmes optimisés, plus compliqués

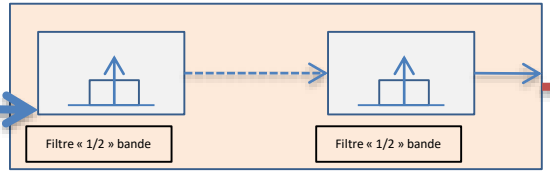
En résumé jusqu'à présent…

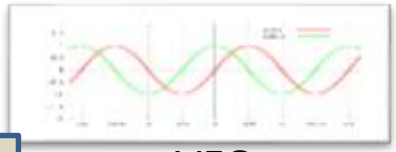


FFT



mix





VFO

$$A = \sqrt{I^2 + Q^2}$$

$$\phi = atan\left(\frac{I}{Q}\right)$$

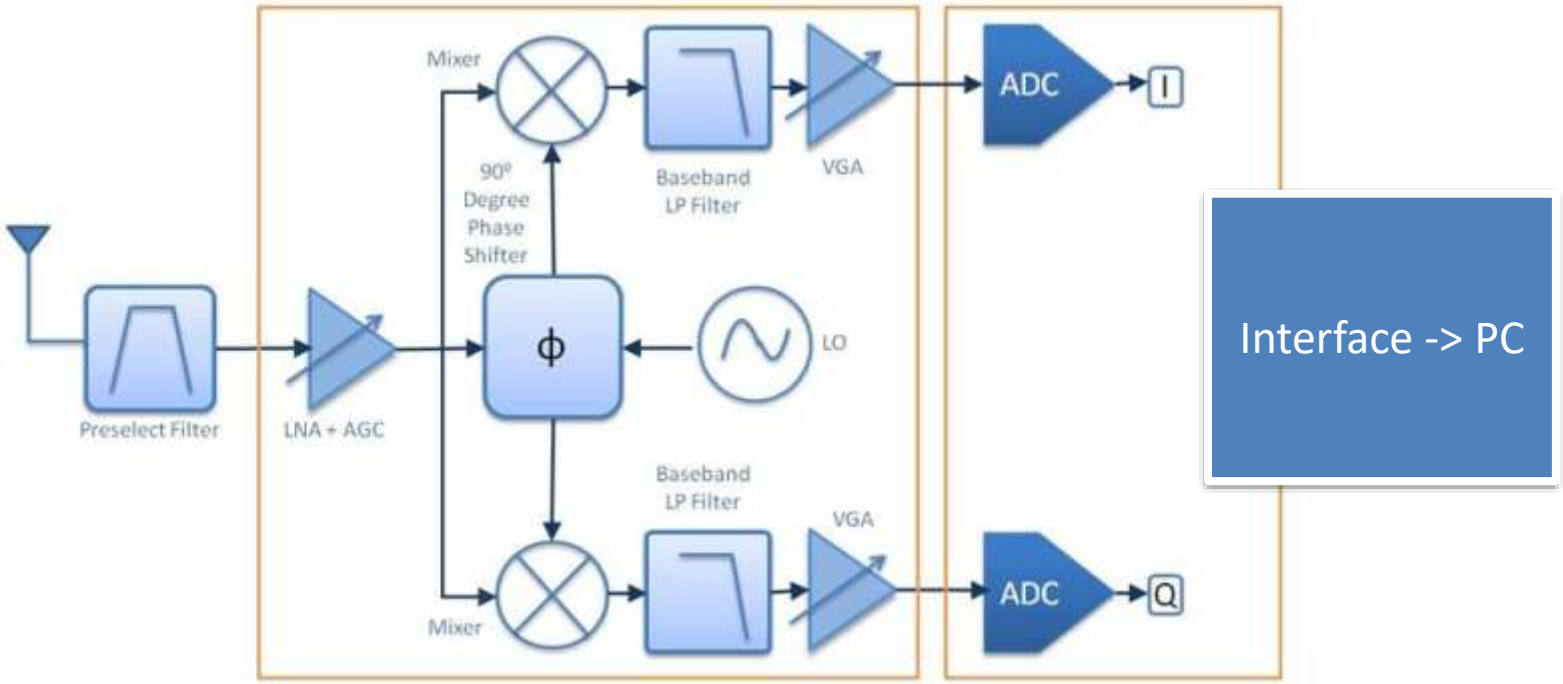
Démodulateurs

Audio

Troisième partie

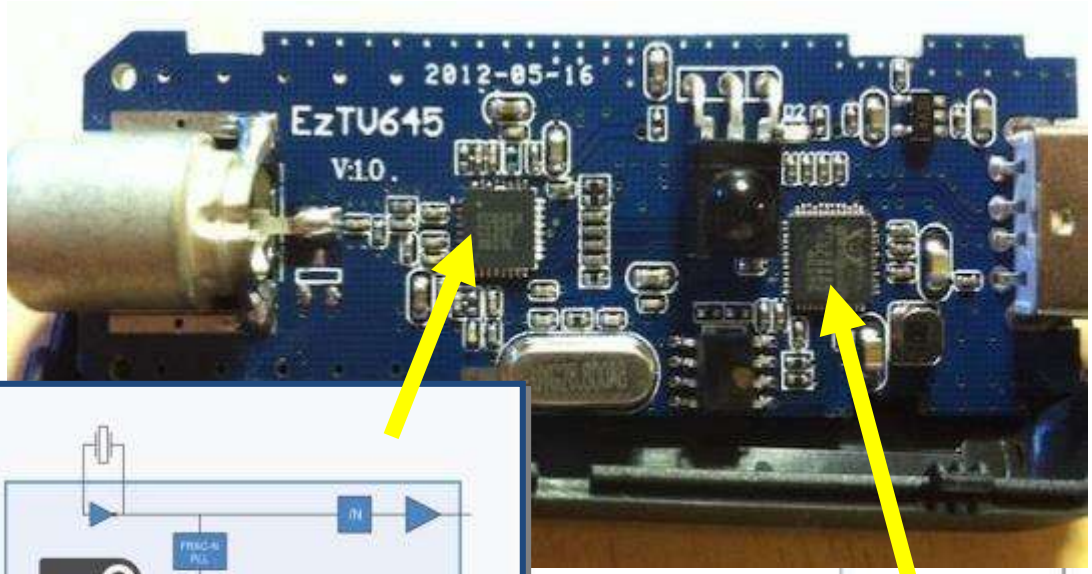
Architecture matérielle

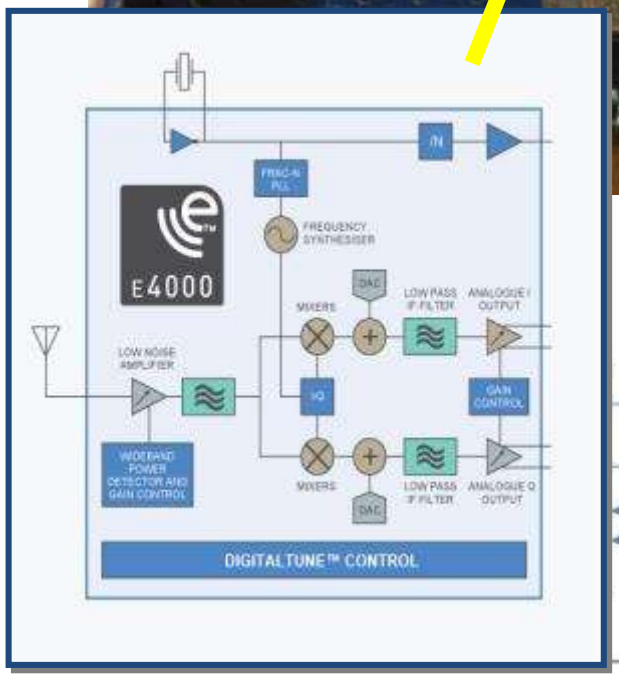
SDR à conversion directe

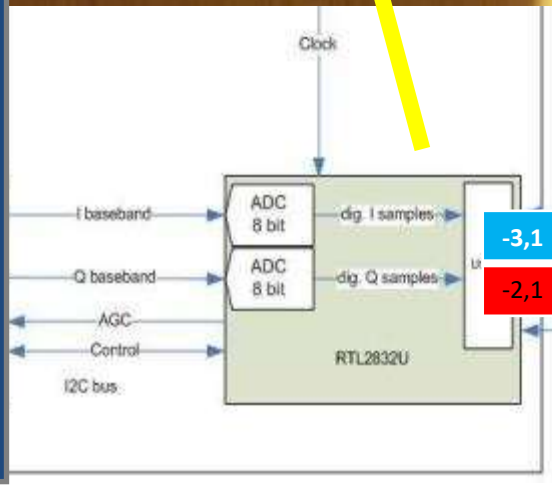


Un oscillateur local sélectionne la fréquence d'intérêt. Autour de cette fréquence, deux bandes sont « extraites » puis numérisées pour être envoyées vers l'ordinateur et le logiciel de traitement.

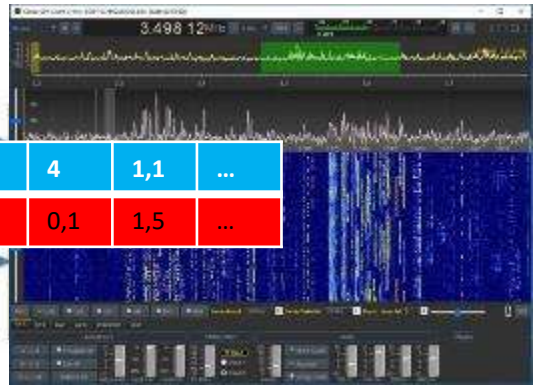
Exemple : clé USB RTLSDR



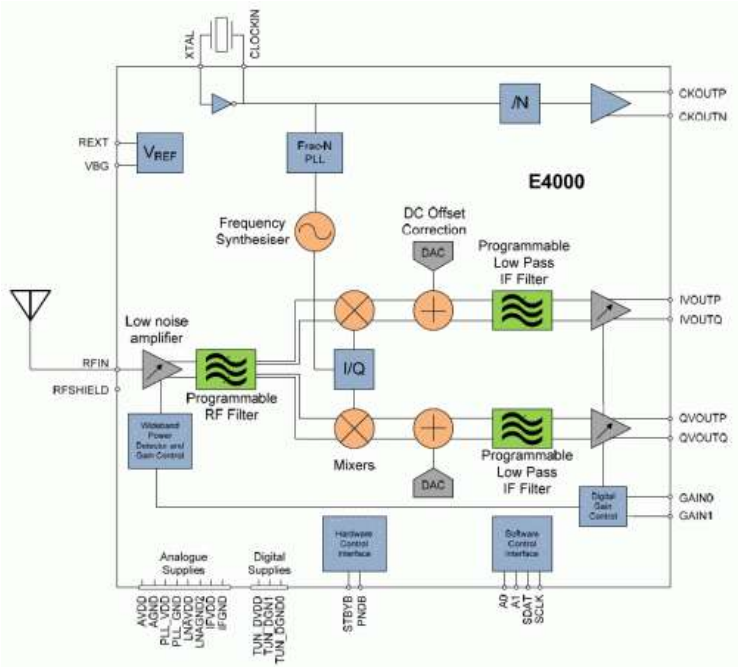


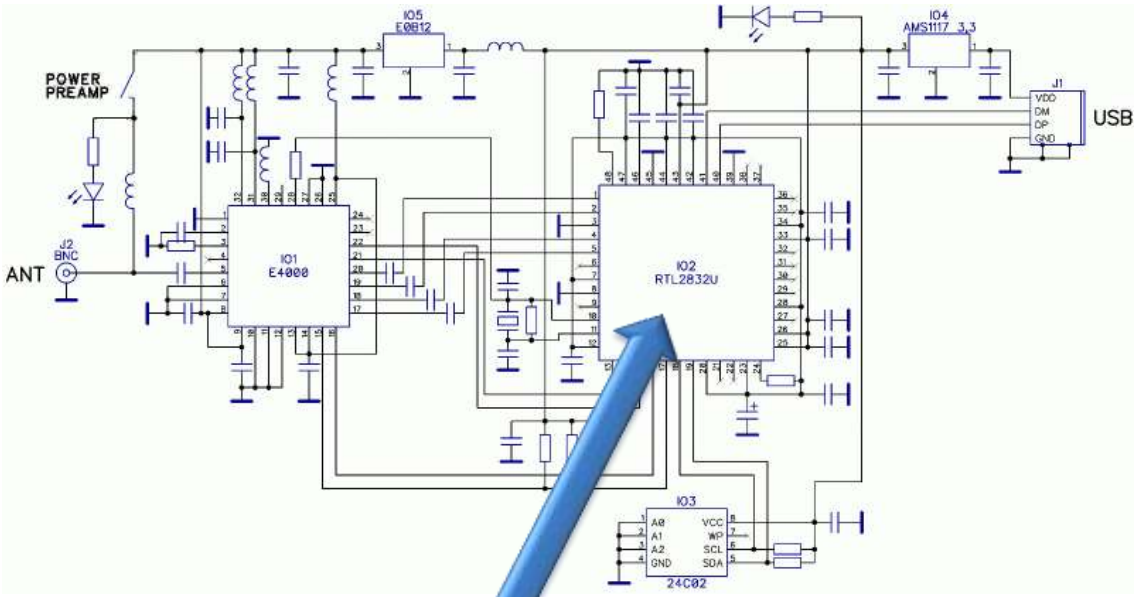


-3,1	2,1	4	1,1	...
-2,1	0,7	0,1	1,5	...



Clé RTLSDR





2 ADC « 8 » bits
Décodage DVBT (pas utile en mode SDR)
Lien USB
µProcesseur intégré

Points faibles de la conversion directe

- 2 voies = Peu probables qu'elles soient identiques…
- Fuite de l'oscillateur local

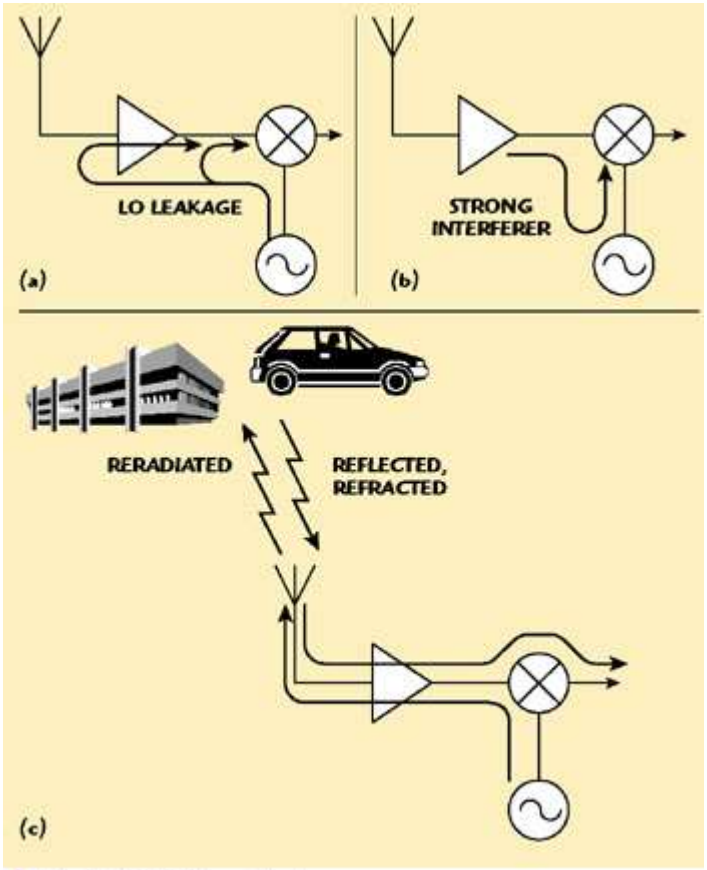
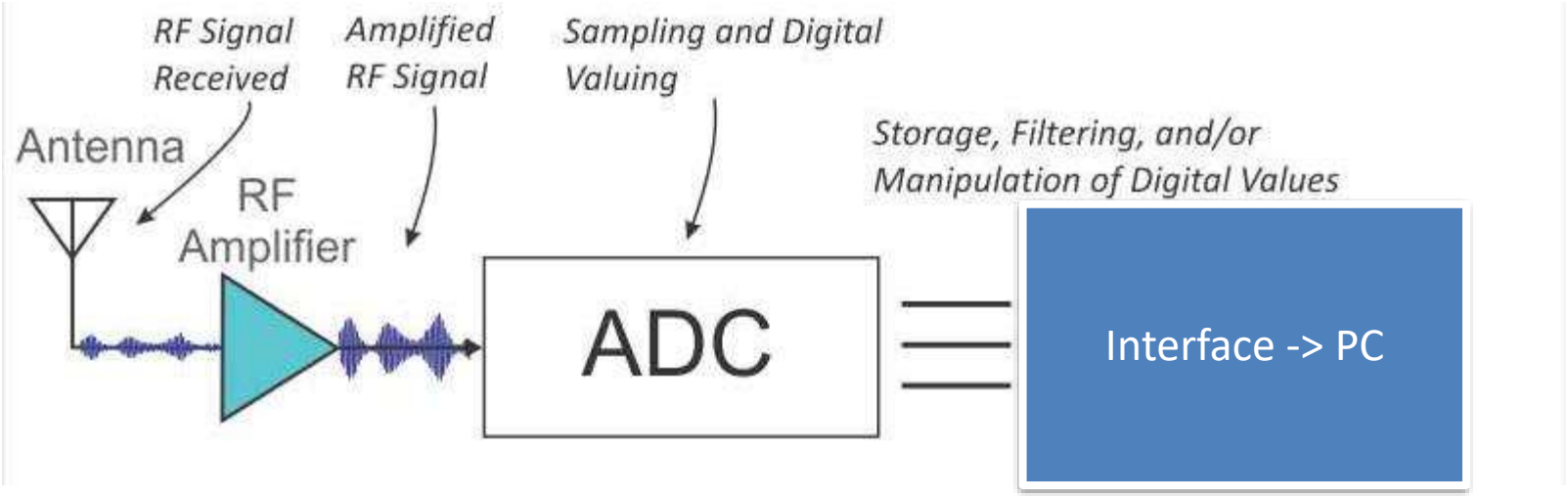


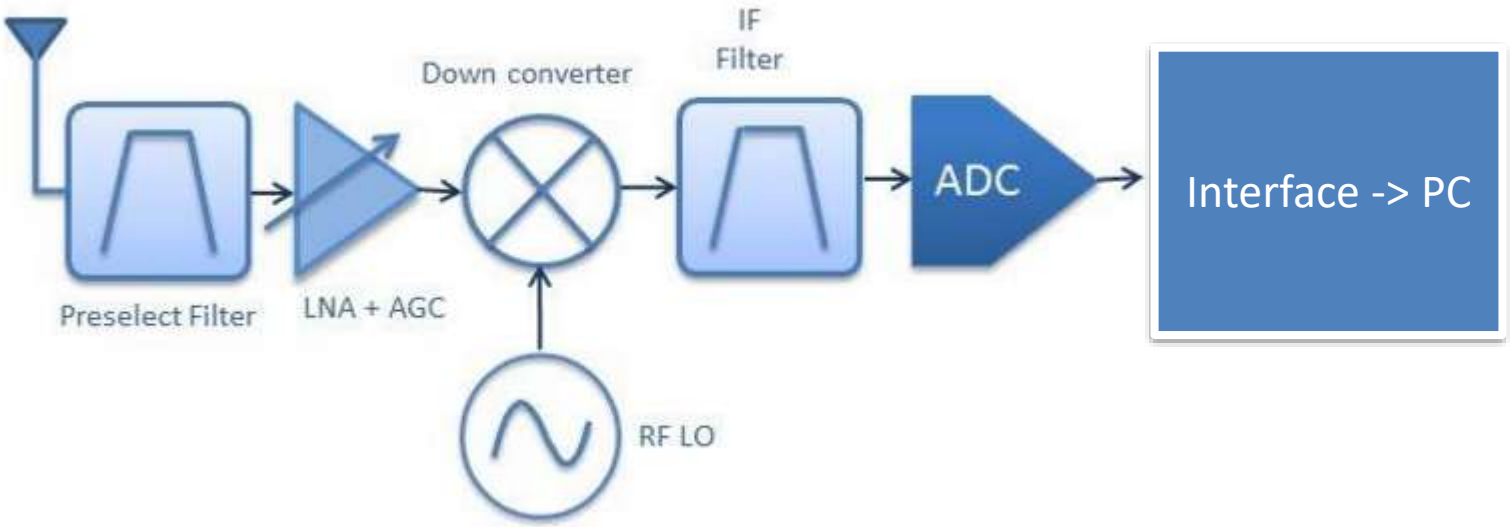
Fig. 8 DC offset mechanisms.

SDR à échantillonnage direct



Après amplification le signal résultant est numérisé pour être envoyé vers l'ordinateur et le logiciel de traitement.

SDR à échantillonnage direct



En pratique, bien souvent un oscillateur local transpose le signal reçu. Après filtrage, le signal résultant est numérisé.

Transférer les échantillons : USB

- L'USB est un **bus série** : 1 octet est transmis bit par bit entre le PC et le périphérique;
- Le débit théorique annoncé est en **bits/seconde**, mais il faut « enlever » toute la partie protocole / signalisation
- Le débit restant pour le SDR est donc plus réduit…

- Et on doit souvent transmettre au minimum 2 octets par échantillon (IQ sur 8 bits)

USB3.0 et USB3.1

	Version	Speed	Bits/sec	HD movie 25GB
	USB 1.1	Low speed (LS) Full speed (FS)	1.5 Mbps 12 Mbps	~9.25 hours
	USB 2.0	High speed (HS)	480 Mbps	~14 mins
	USB 3.0	SuperSpeed (SS)	~~5 Gbps~~	~70 sec
	USB 3.1	SuperSpeedPlus (SSP)	~~10 Gbps~~	~35 sec

USB2 :

- 480 Mbits = limite très très théorique, en pratique de l'ordre de 15 à 20 Mo/seconde
- 10 MHz de bande sous 8 bits passe… rarement plus

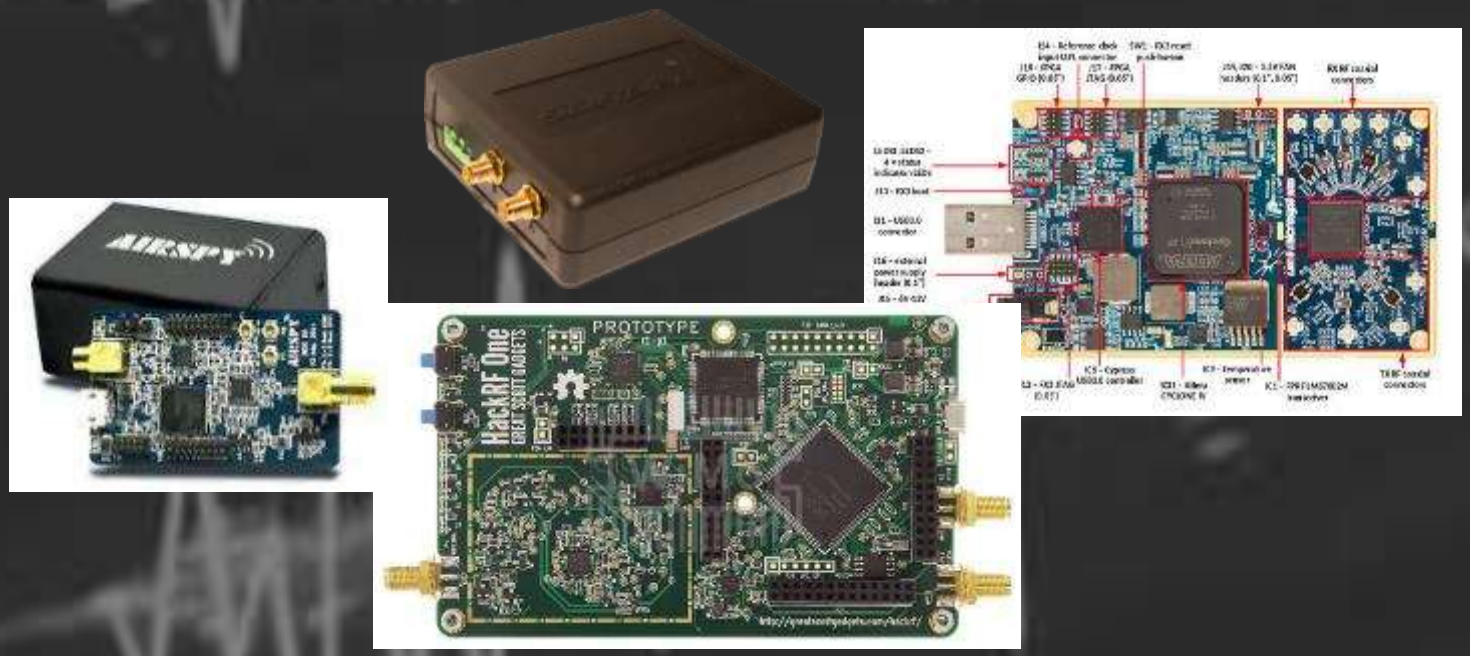
USB3

- 5 Gbps = 5 Giga bits per seconds
  = environ **300 millions de IQ @ 8 bits**
  = environ 150 millions de IQ @ 16 bits = 150 MHz avec une « super résolution »

- **Problème** : avoir le PC qui va assez vite pour les calculs qui restent à faire…

Logiciels SDR pour l'émission

- Beaucoup de logiciels pour le RX, mais pas grand-chose (encore) pour le TX « analogique » (SSB, FM etc.)
- A noter: **SDRAngel de F4EXB**
- Pour l'expérimentation: GNURadio, compatible avec 99% des transceiver SDR existants



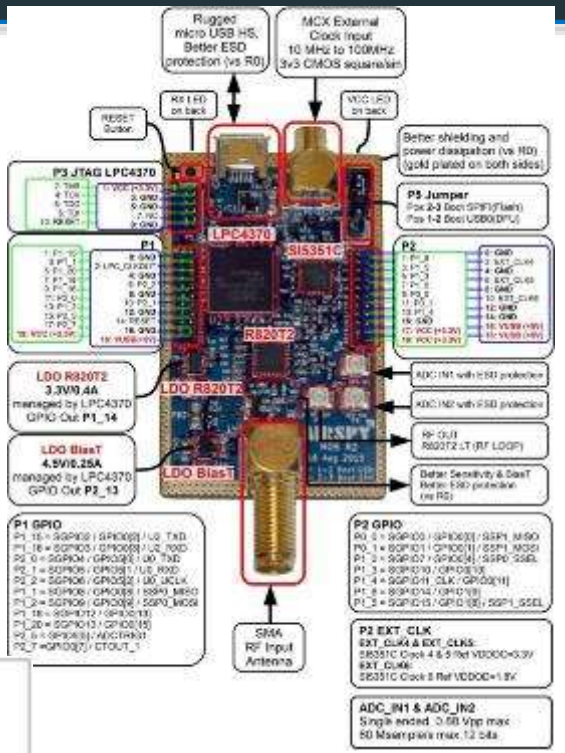
Quatrième partie

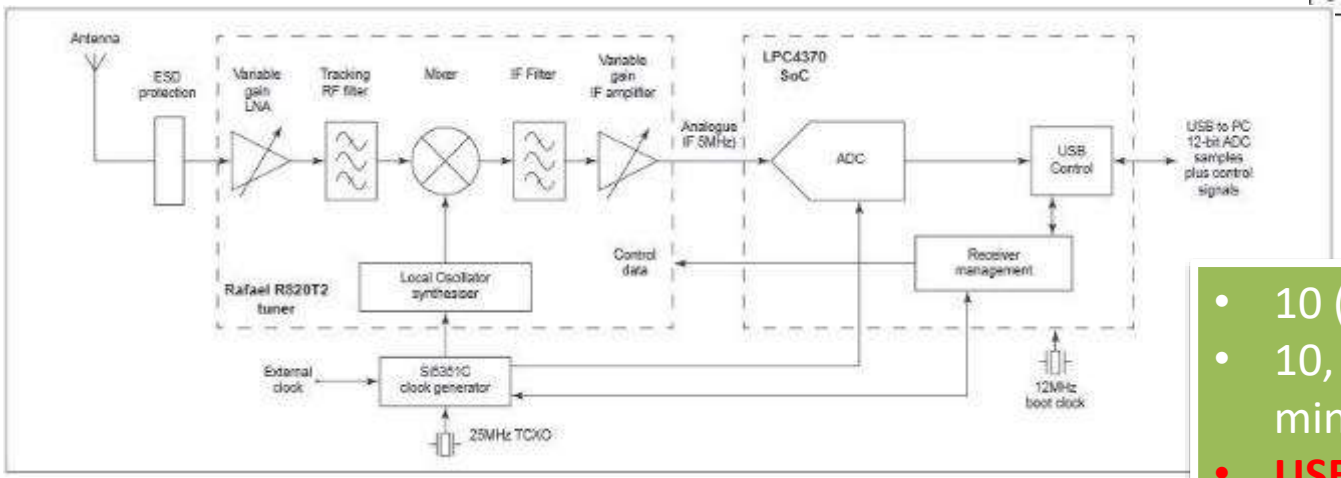
Pour votre prochaine liste de courses…

Récepteurs « AirSpy »

de 24 à 1800 MHz

- Utilise un tuner R820T2
- 1 Seul ADC 12 Bits, intégré au µP LPC4370,
- 10MHz externe possible sur le R2 uniquement
- ATTENTION: Prise USB fragile (voir site HB9FAO pour bonne idée de boitier)
- **Exige** un contrôleur USB performant à 10 MHz





- 10 (2.5) MSPS pour le R2
- 10, 6 and 3 MSPS pour le mini
- **USB 2**

Récepteurs « SDRPlay »

de 0,01 à 2000 MHz

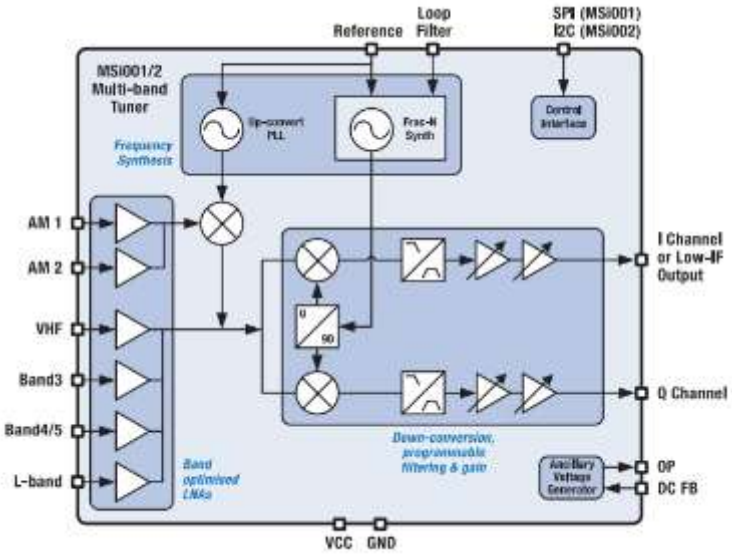
RSP1A

RSP2

RSP2 « pro »

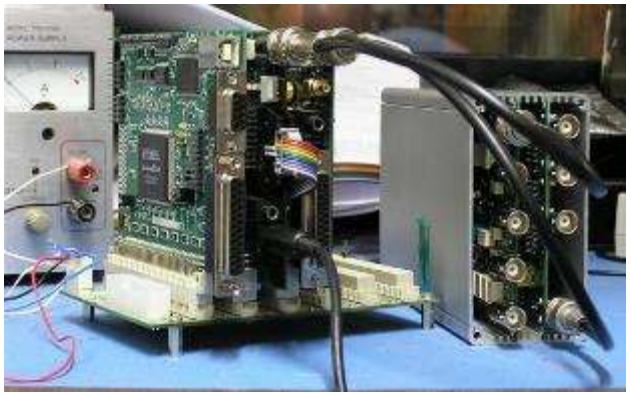
- De 0,2 à 10 MSPS pour le RSP2 et 2PRO
- Possibilité de REF externe 24 MHz (I/O), chaînage possible de plusieurs RX
- **USB 2**

2 ADC 12 Bits



Famille HPSDR *(High Performance SDR)*

- Projet Open-Source - https://openhpsdr.org/
- Couvre l'aspect matériel ET logiciel
- Disponibilité de kits complets ou sous-ensembles, ou encore de produit commercial prêts à l'emploi

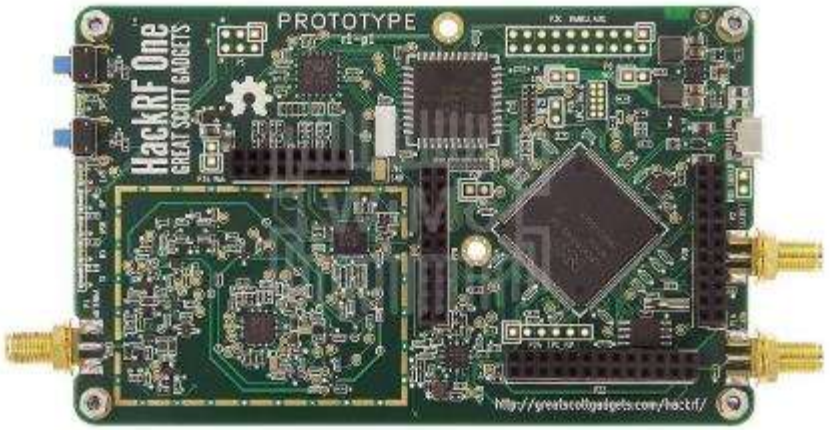


FlexRadio

- Probablement la solution la plus ancienne sur le marché
- Prix variables selon modèles et options, ~7500$

HackRF

TX et RX de 1 à 6000 MHz

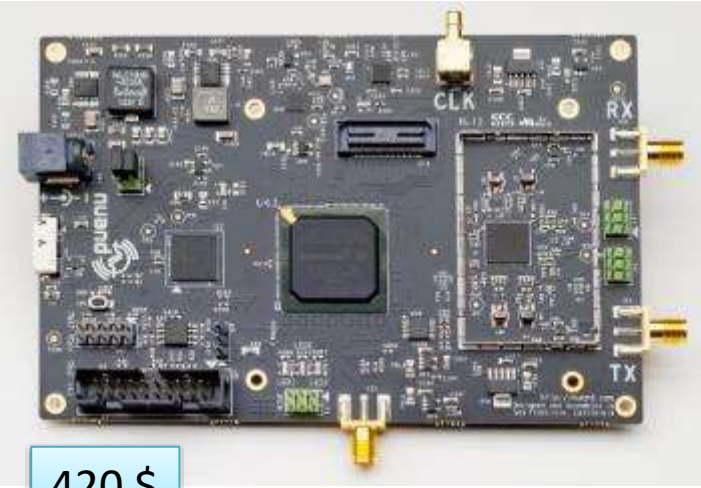


De 1 MHz à 6 GHz, Emetteur-récepteur en semi-duplex
Jusqu'à 20 Mhz, 8 bits en quadrature (8 bits I et Q 8-bit)
USB 2.0
Alimentation par USB

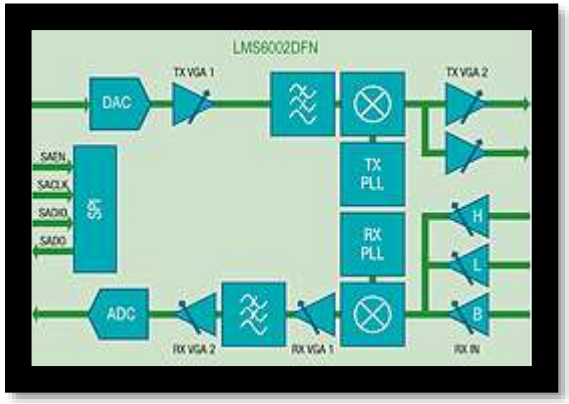
- Expérience personnelle: Préampli RX très fragile, accrochages en HF, émission « très moyenne » , équilibrage IQ moyen

BladeRF

TX et RX de 240 à 3800 MHz



420 $

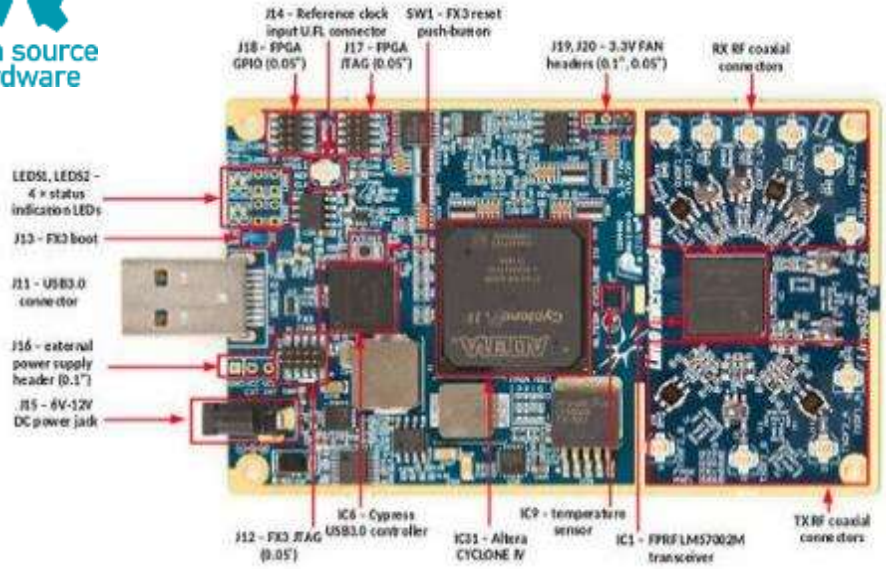


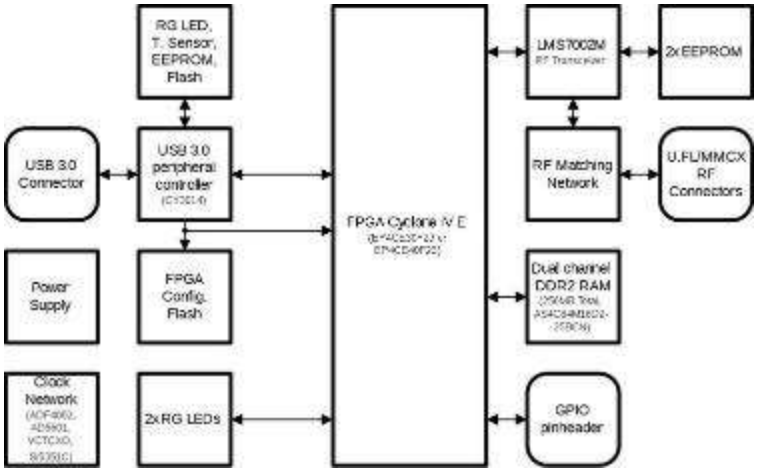
Chip LMS6002D de chez Lime

- Full duplex TX et RX, de 2 MSPS à 28MSPS
- Double ADC 12 bits (RX), Double DAC 12 Bits (TX)
- Horloge intégrée, possibilité synchro PPS externe (firmware specifique)
- USB3
- Test perso: Très bonne sensibilité RX

LimeSDR

open source hardware





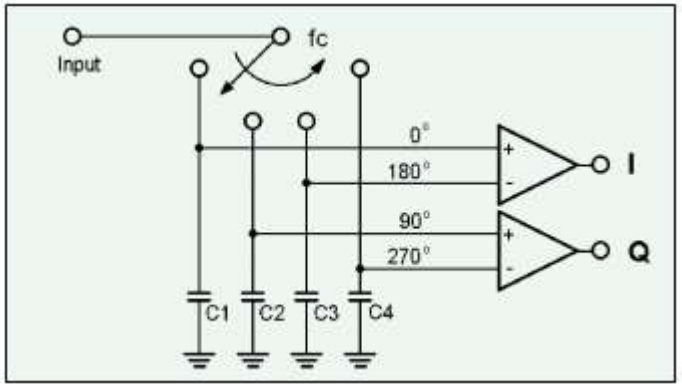
- Financement participatif, très gros succès
- Aspect logiciel… bâclé !

SoftRock

RX et TX – HF et VHF selon modèles

open source hardware

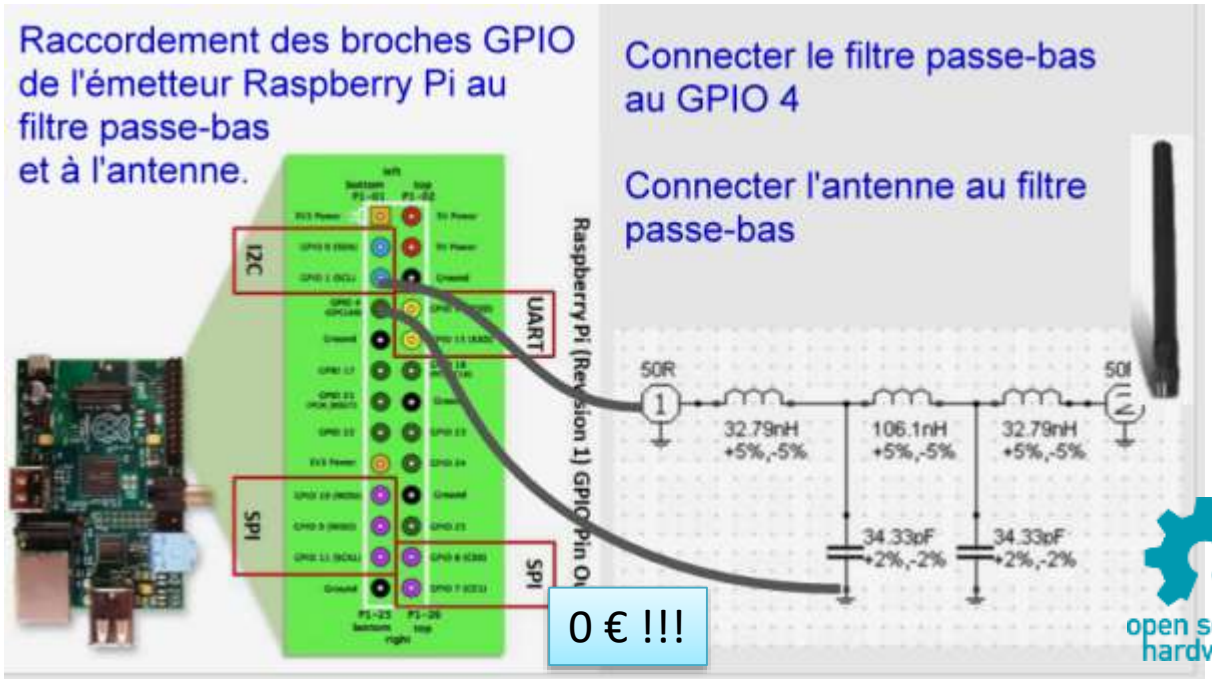
89$!



- Mélangeur à base de « QSD » (switches commutés à la fréquence de l'OL)
- Un peu semblable au Gilbert Cell mixer

- Vendu en kit
- Emission QRP
- Repose sur ADC et DAC audio du PC

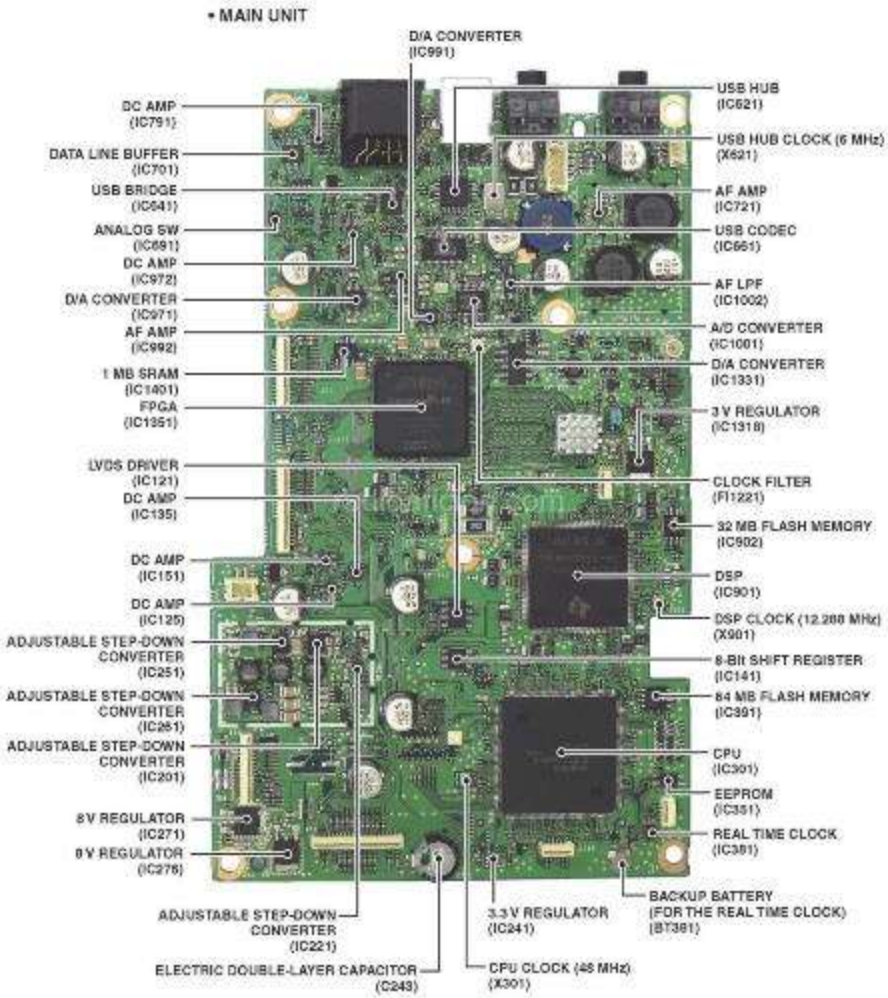
RPiTx



- La solution la plus « fun » de toutes, merci F5OEO
- Un Raspberry Pi + du fil (et éventuellement des filtres…)
- Logiciel spécifiques pour ATV, SSB, FM etc.

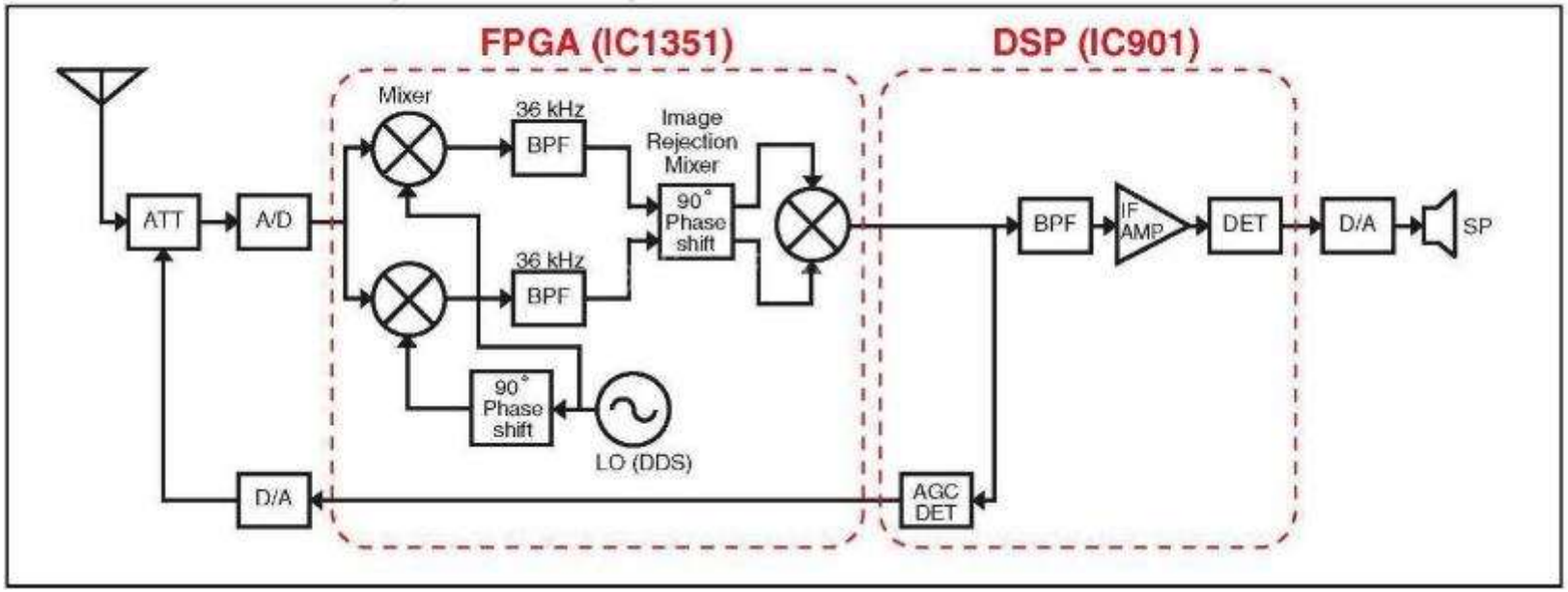
IC7300

Où sont les selfs, les quartz, les CV ?



Exemple de l'IC7300

• FPGA BLOCK DIAGRAM (Receive circuits)



- 1 seul ADC – échantillonnage direct
- Tout le traitement est fait en numérique dans un réseau logique programmable (FPGA)

IC7300 : l'approche idéale ?

- Un seul ADC à échantillonnage direct, mais:
  - Attention à la dynamique,
  - Attention au repliement éventuel (FM…) impose un filtre très sélectif,
  - Débit d'information (échantillons) important
- Traitement par FPGA uniquement:
  - Pas forcément très couteux, mais mise au point très très complexe,
  - Calculs en virgule fixe, avec souvent des problèmes d'arrondi (attention à l'OL)

Fin…

Merci à tous pour votre attention !
sylvain.azarian@gmail.com