

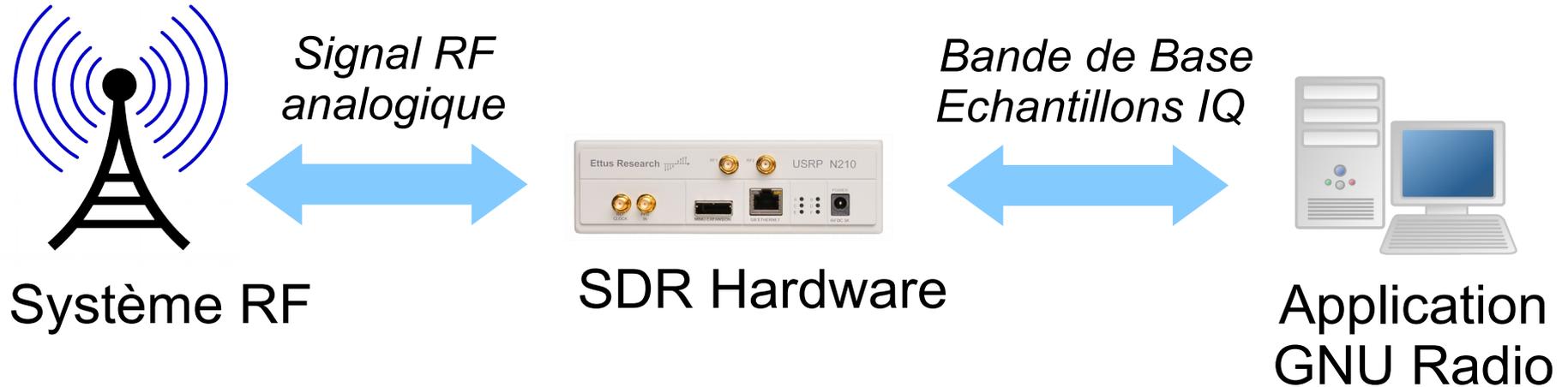
La radio logicielle

Mise en œuvre

Équipe pédagogique de la LP MHR
IUT de Ville d'Avray

21 Mars 2019

<https://cva-geii.parisnanterre.fr/>



- Antenne
- Filtre RX/LNA
- PA Tx
- Duplexeur

- CAN/CNA
- Conversion Up/Down
- Filtrage
- Ré-échantillonnage

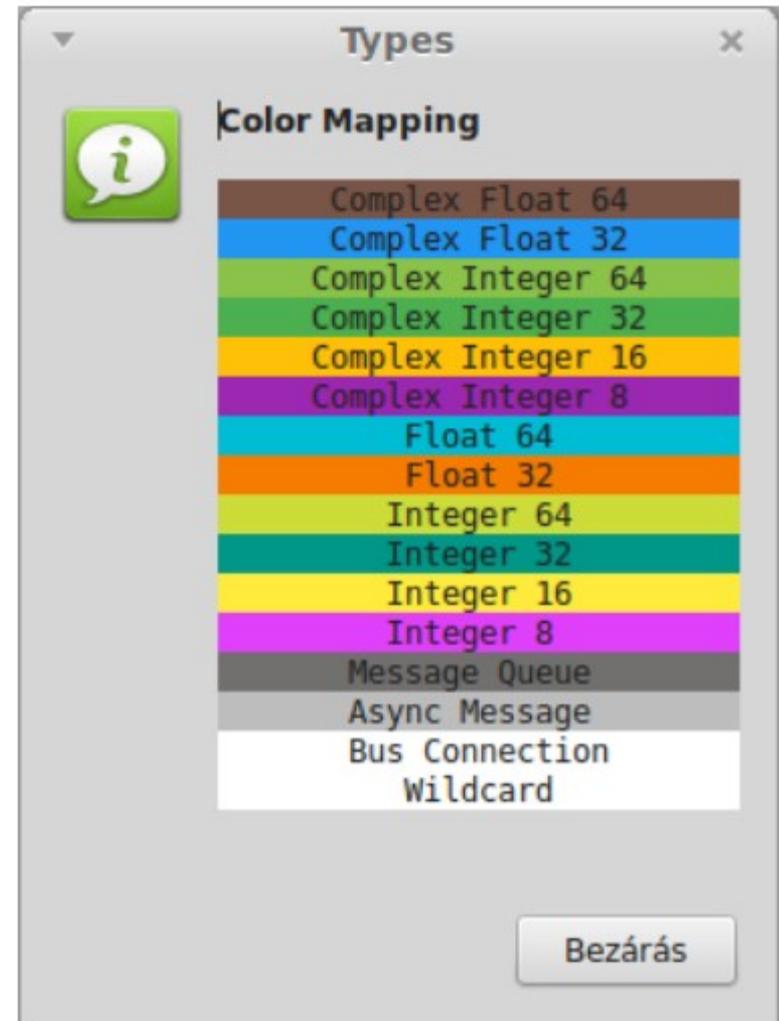
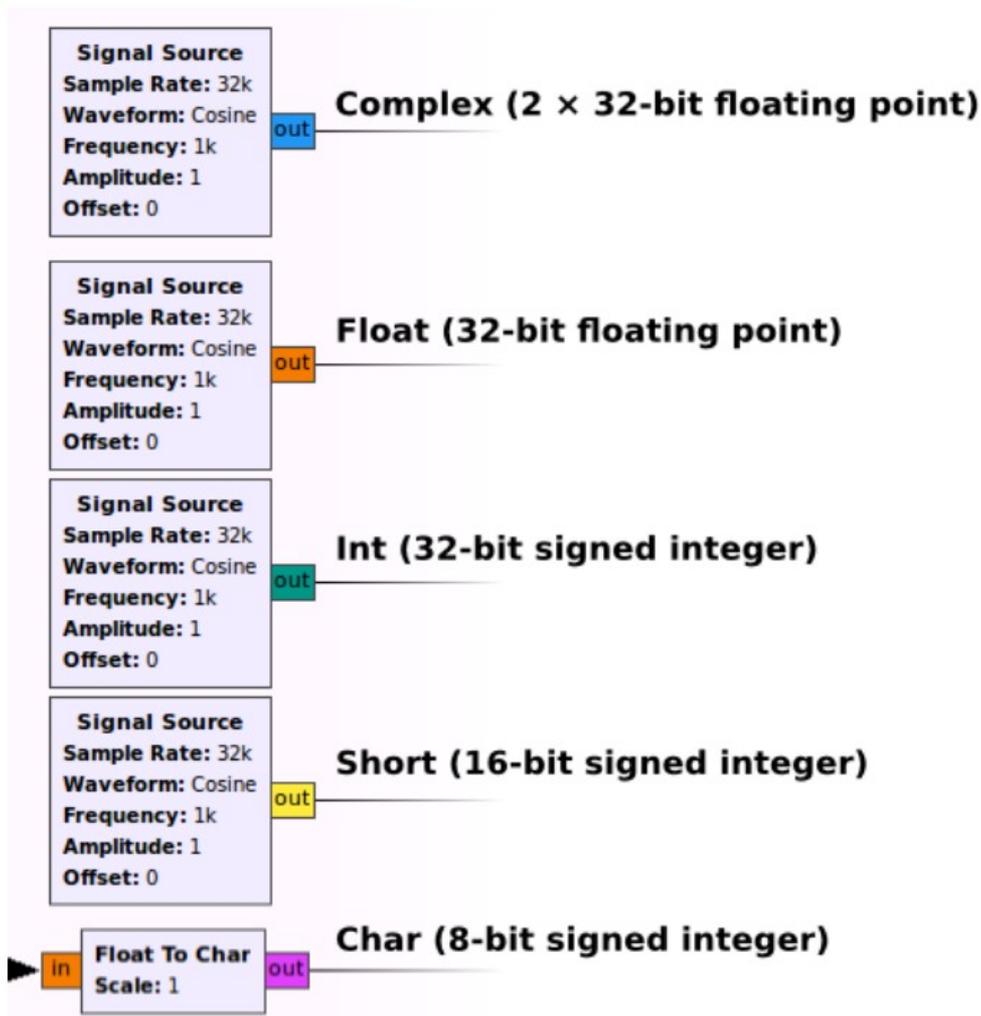
**Votre
application**

- **GNU Radio: composants logiciels de traitement de signaux en temps réel**
 - Ecrit en C++ (performance) et connectés en python
 - « Lego » de la radio
 - Entrée/Sortie: USRP, RTL-SDR, ADALM PLUTO, fichier, carte son, etc
 - Quelques fonctions de traitement de signal GNU Radio:
 - **Modulations analogiques: AM, FM, PM**
 - **Modulations numériques: PSK, QAM, GMSK, OFDM,...**
 - **Mais aussi: Filtres, Resampling, FFT, ...**

- 
- E1 : Prise en main
 - Du logiciel GNU Radio Companion (GRC)
 - Du dongle RTL-SDR
 - E2 : Réalisation d'un récepteur AM
 - Filtrage, démodulation, gain audio
 - Changement de station
 - E3 : Réalisation d'un récepteur FM
 - E4 : Application – votre premier récepteur ADS-B

- A la fin de ce tutoriel, vous serez capable de :
 - comprendre les concepts de radio logicielle
 - d'utiliser le logiciel libre GNU Radio companion
 - de réaliser un récepteur radio AM (modulation d'amplitude)
 - de réaliser un/des récepteur(s) FM
 - de recevoir les trames ADSB des avions

• Les types de données



The screenshot shows the GNU Radio Companion (GRC) interface. The main workspace contains a flowchart with three blocks connected in a line:

- Signal Source**: Sample Rate: 32k, Waveform: Cosine, Frequency: 1k, Amplitude: 1, Offset: 0.
- Throttle**: Sample Rate: 32k.
- QT GUI Time Sink**: Number of Points: 1.024k, Sample Rate: 32k, Autoscale: No.

On the left, there are two panels: **Options** (ID: top_block, Generate Options: QT GUI) and **Variable** (ID: samp_rate, Value: 32k).

On the right, there is a search bar labeled "QT" and a list of components. The list is expanded to show the following items:

- (no module specified)
- Instrumentation
 - QT
 - QT fosphor sink
- Core
 - Instrumentation
 - QT
 - QT GUI Bercurve Sink
 - QT GUI Constellation Sink
 - QT GUI Frequency Sink
 - QT GUI Histogram Sink
 - QT GUI Number Sink
 - QT GUI Sink
 - QT GUI Time Raster Sink
 - QT GUI Time Sink
 - QT GUI Vector Sink
 - QT GUI Waterfall Sink
- GUI Widgets
 - QT
 - QT GUI Check Box
 - QT GUI Chooser
 - QT GUI Message Edit Box
 - QT GUI Entry
 - QT GUI Label
 - QT GUI Push Button
 - QT GUI Range
 - QT GUI Tab Widget

At the bottom left, there is a console window showing the following output:

```

Skipping it
WARNING: Config file '/home/fdaout/.gnuradio/config.conf' failed to parse:
std::exception
Skipping it
WARNING: Config file '/home/fdaout/.gnuradio/config.conf' failed to parse:
std::exception
Skipping it
>>> Done
    
```

At the bottom right, there is a table with the following data:

Id	Value
Imports	
Variables	
samp_rate	32000

Exercice : réalisez le schéma (flowchart) ci-dessus.

Attention, pour modifier la couleur du port d'un bloc (type de donnée), il faut l'ouvrir et choisir un type float.

- **Quel est le type de signal généré (sinus, créneaux...)** ?
- **Quelles sont ces caractéristiques : fréquence, période, amplitude**





Exercice : générer un signal carré de période 1 ms, d'amplitude crête à crête de 0.5 V et de valeur moyenne nulle.

- **Modifier le flowchart précédent pour répondre à ces spécifications**
- **Vérifier**

Exercice : générer un sinus de fréquence variable et d'amplitude 2 V

- **Le bloc QT GUI range permet de faire varier un paramètre, ici la fréquence. Dans ce bloc, donnez un nom à la variable (par exemple frequency), une valeur min (100) et max (30000), ainsi qu'une valeur par défaut. Dans le bloc Signal Source, à la place de la fréquence 1000, mettez la variable frequency. Observez ce que vous obtenez et commentez.**



Exercice : générer un sinus de fréquence variable entre 10 Hz et 20 kHz (1kHz par défaut) et d'amplitude 0.5V.

- **Ajoutez le bloc QT GUI Frequency Sink pour observer le spectre.**
- **Visualisez le signal temporel et son spectre. Modifier la valeur de la fréquence (bloc QT GUI range). Observez ce que vous obtenez sur le spectre et commentez.**
- **Ajoutez un bloc QT GUI range afin de changer l'amplitude de la sinusoïde. Observez ce que vous obtenez sur le spectre et commentez.**
- **Que se passe-t-il lorsque la fréquence de la sinusoïde est trop élevée ? Expliquez ce phénomène.**



Quelques précautions doivent être prises quand vous utilisez une carte d'acquisition RF (comme le dongle RTL-SDR) :

- Évitez de toucher l'antenne (ou pire le port d'entrée): vous pourriez, à cause de l'électricité statique, endommager la carte d'acquisition
- La puissance maximale que supporte la carte est de 0dBm (pour un gain nul). Pensez à prendre en compte la valeur du gain : $P_{max}(dBm) = P_{in}(dBm) + G(dB) < 0dBm$

Exercice : A l'aide du bloc RTL-SDR Source, recevez le signal de la bande FM autour de 100 MHz.

Observez ensuite la FFT du signal que vous obtenez en branchant le bloc QT GUI Frequency Sink. Paramétrez le bloc de la manière suivante :

- Échantillonnage à 2MHz
- Fréquence centrale : 100MHz
- Gain de la chaîne RF : 50dB

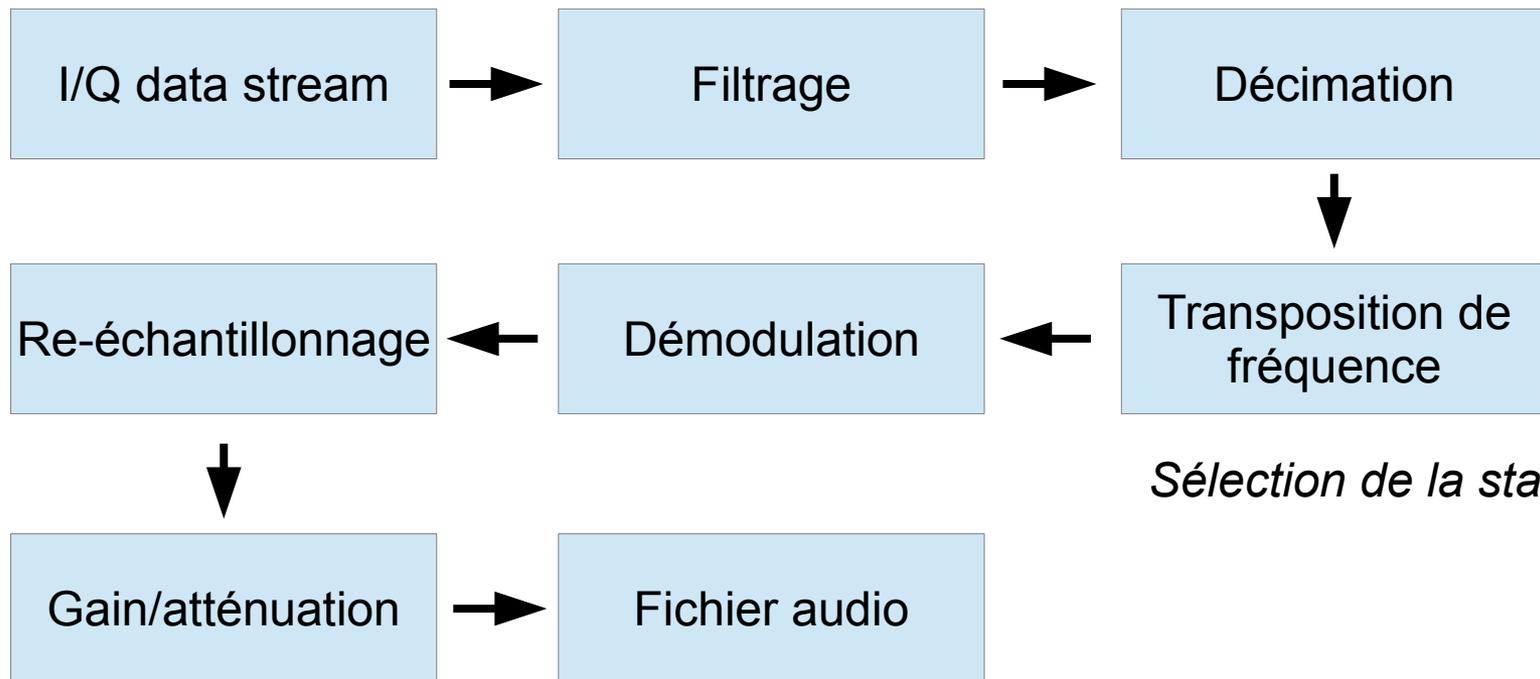
```

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Found Rafael Micro R820T Tuner
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T tuner
[RB2XX] PLL not locked!
Exact sample rate is: 2000000.052982 Hz
[RB2XX] PLL not locked!
>>> Done
  
```

- Structure du récepteur → logiciel

*Fichier ou carte
d'acquisition*

*Diminuer la bande du
signal*



Sélection de la station

Fréquence imposée

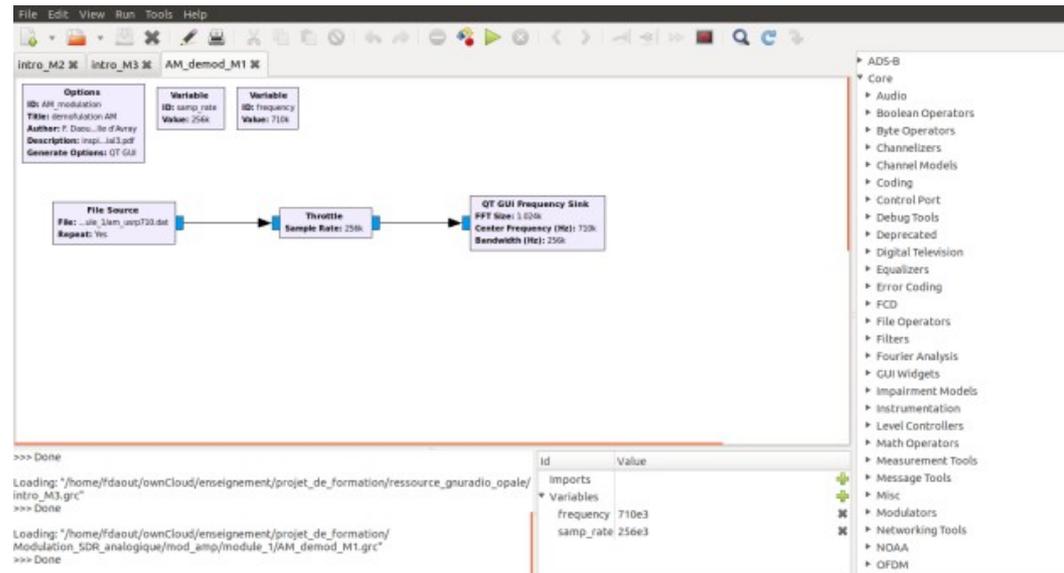
• Analyse du signal enregistré

Le fichier `am_usrp710.dat` a été acquis avec une carte USRP. Celui-ci n'est rien d'autre que le signal modulé reçu d'une radio commerciale américaine et enregistré. C'est ce signal que vous filtrerez, démodulerez et étudierez. Les caractéristiques de cette acquisition sont les suivantes :

- fréquence centrale $f_p = 710$ kHz
- $\text{samp_rate}_{fe} = 256$ kHz

Exercice : Construisez le flowchart ci-contre. Réglez la fréquence d'échantillonnage dans le bloc `samp_rate` sur 256 000 (fe).

Observez le spectre de ce signal



- **Sélection du canal désiré (710 kHz)**

Seule une bande de 10 kHz autour de la fréquence radio 710 kHz est nécessaire pour la démodulation. Il faut donc filtrer autour de cette bande.

Abaisser la fréquence d'échantillonnage, ou décimation, permet de réduire le coût des calculs et de ne garder que la bande utile,

Exercice : implantez un filtre passe-bas dont les caractéristiques sont :

- **Échantillonnage à 256 kHz**
- **Fréquence de coupure à 5 kHz**
- **Fréquence de transition de 500 Hz**
- **Décimation de 4 (diminution de la fréquence d'échantillonnage d'un facteur 4 pour arriver à la sortie du bloc à 64kHz)**
- **Fenêtre de Hamming**
- **Beta=6,76**



Observez le spectre du signal obtenu après cette manipulation.

Attention la fréquence d'échantillonnage (`samp_rate`) du bloc QT GUI frequency sink est maintenant de 64 kHz (256kHz/4). A la sortie de ce filtre, on ne doit conserver que la partie contenant la bande radio recherchée.

• Démodulation

Le signal obtenu en sortie du filtre passe-bas est le signal modulé en amplitude. Il existe plusieurs manières de démoduler un signal modulé en amplitude : démodulation cohérente ou détecteur d'enveloppe.

Dans notre cas, le signal est centré autour de la fréquence nulle. La méthode la plus simple dans ce cas consiste à prendre l'enveloppe du signal en utilisant le bloc Complex to Mag.

Exercice : mettez en œuvre la démodulation du signal à l'aide du bloc Complex to Mag.

Observer la forme temporelle du signal en utilisant le bloc QT GUI time sink (rappel : `samp_rate=64kHz`).



- Écouter le signal obtenu

Le signal audio possède une composante continue non nulle qu'il faut supprimer.

Exercice : ajouter le bloc *DC blocker* et observer la forme temporelle du signal en utilisant le bloc *QT GUI time sink* (`samp_rate=64kHz`)

Le signal audio ne possède pas forcément la bonne dynamique ($\pm 1V$), il est nécessaire de rajouter un bloc qui modifie l'amplitude du signal.

Exercice : rajouter le bloc *Multiply const* et régler la valeur de la constante pour avoir un signal, à la sortie du bloc, d'amplitude inférieure $\pm 1V$



- **Écouter le signal obtenu (suite)**

Le signal démodulé est échantillonné à 64 kHz.

La carte son de l'ordinateur possède une fréquence d'échantillonnage de 48 kHz.

Un traitement s'avère nécessaire pour ramener la fréquence d'échantillonnage du signal (64 kHz) à la valeur de 48 kHz et éviter ainsi de déformer le signal .

L'abaissement de la fréquence d'échantillonnage peut être réalisé avec un interpolateur.

Dans GNU Radio, le bloc Rational Resampler remplit cette fonction. Celui-ci prend deux arguments :

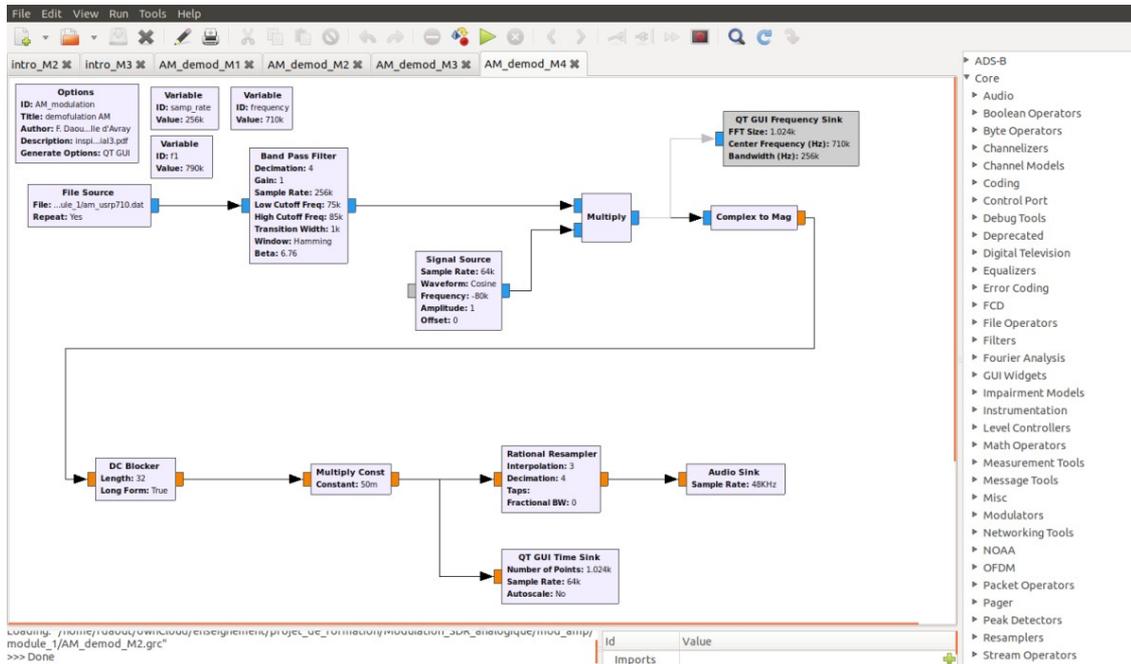
- ***Une décimation pour diviser la fréquence d'échantillonnage par ce nombre : 4 (64/4=16)***
- ***Une interpolation pour multiplier la fréquence d'échantillonnage par ce nombre : 3 (16*3=48)***

Exercice: ajoutez le bloc Rational Resampler et supprimer le bloc Throttle.

Observer le signal en branchant le bloc QI GUI time sink (samp_rate = 48kHz).

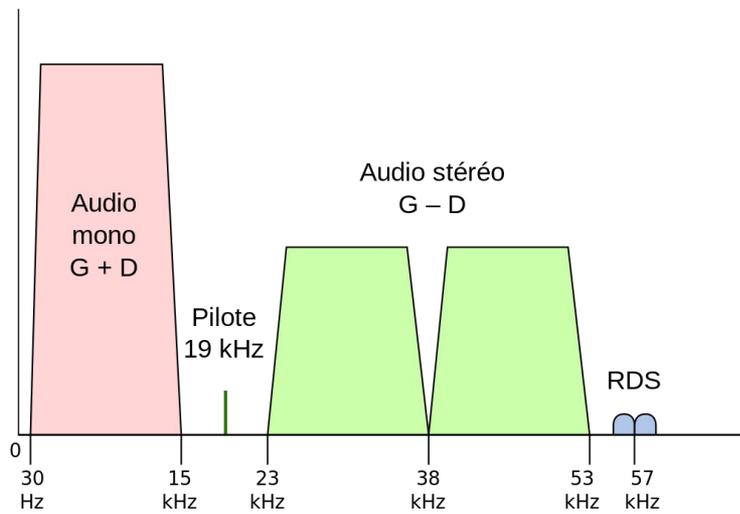


• Changement de station



L'objectif de cette partie est de choisir une station différente de 710 kHz. Lors de notre acquisition, la carte USRP était réglée sur la fréquence centrale de 710 kHz avec une fréquence d'échantillonnage de 256 kHz. Ce réglage nous permet d'observer les fréquences situées entre $710-128=582$ kHz et $710+128=838$ kHz. Dans cette bande de fréquence, d'autres stations sont visibles. Notre objectif est de choisir la station 790 kHz, de transposer cette station à la fréquence 0Hz et de démoduler cette station afin de pouvoir l'écouter.

- Les stations radios FM émettent en France dans la bande de radiodiffusion de 87,5 MHz à 108 MHz. Elles occupent une bande qui est généralement un multiple de 100 kHz. Elles émettent en modulation de fréquence. Le spectre du signal en bande de base, c'est-à-dire non modulé, est constitué de la somme des canaux gauche et droite pour l'écoute monophonique, suivi par la différence des canaux gauche et droite afin de reconstituer si possible une écoute stéréophonie. Entre les deux le signal pilote, qui est inaudible, aide à la reconstruction de la stéréo. Sa présence ou non informe l'émetteur d'une émission stéréophonique ou non. En bout du spectre se trouve les informations du Radio Data System (RDS) transmises sous forme numérique.



Spectre en bande de base d'une émission de radiodiffusion FM, avec stéréophonie et RDS.

GNU Radio companion associé avec une clef RTL-SDR va nous permettre de fabriquer un récepteur radio FM.

• Visualisation du spectre

La clef RTL-SDR permet de recevoir une bande de fréquence f_e autour d'une fréquence centrale f_p . Le signal ainsi acquis va être filtré, démodulé afin de capter la station radio de votre choix.

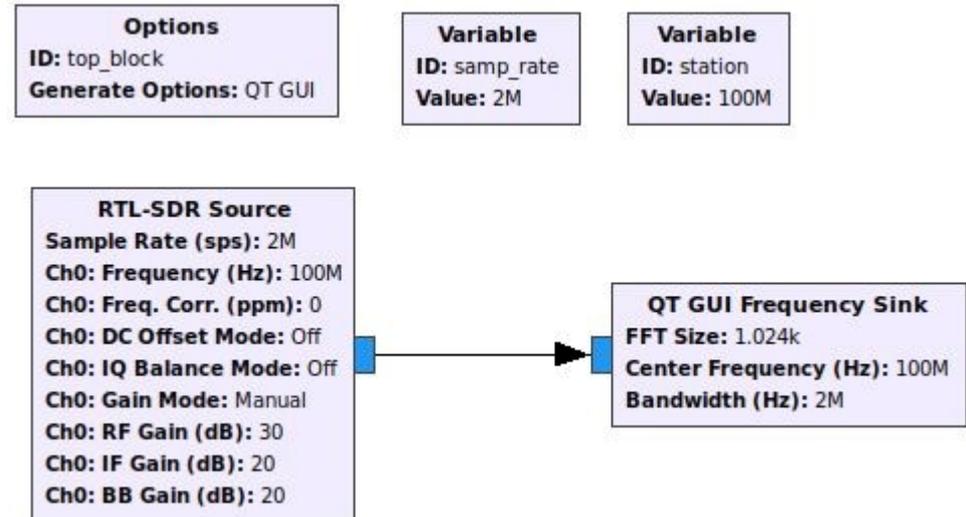
Les caractéristiques de cette acquisition sont les suivantes :

- fréquence centrale $f_p = \text{station} = 100 \text{ MHz}$
- $\text{samp_rate } f_e = 2 \text{ MHz}$



Construisez le flowchart illustré ci-contre, composé d'un RTL-SDR Source avec un RF Gain de 30dB, QT GUI Frequency Sink et de la Variable station. Réglez la fréquence d'échantillonnage dans le bloc samp_rate sur $2e6$ (f_e) et dans le bloc station sur $100e6$ (f_p).

Observez le spectre de ce signal et repérez la station la plus puissante.



- Sélection de la station de radio désirée

La bande occupée des stations radios est en moyenne de 100kHz autour de la fréquence de la radio (station). Il faut donc filtrer autour de cette bande. Nous allons aussi chercher à abaisser la fréquence d'échantillonnage pour passer de 2MHz à 48 kHz imposé par la carte son.

Remarque : $\frac{48 \text{ kHz}}{2 \text{ MHz}} = \frac{24}{1000} = \frac{3}{125} = \frac{3}{5 \times 5 \times 5}$



La fréquence de la station la plus puissante est mise dans le bloc station.

Implantez un filtre passe-bas à la sortie du bloc RTL-SDR Source dont les caractéristiques sont :

- ***Échantillonnage à 2MHz***
- ***Fréquence de coupure à 100kHz***
- ***Fréquence de transition de 10kHz***
- ***Décimation de 5 (diminution de la fréquence d'échantillonnage d'un facteur 5 pour arriver à la sortie du bloc à 400kHz)***
- ***Fenêtre de Hamming***
- ***Beta=6,76***

Observez le spectre du signal obtenu après cette manipulation. Attention la fréquence d'échantillonnage (samp_rate) du bloc QT GUI frequency sink est maintenant de 400kHz (2000kHz/5). A la sortie de ce filtre, on ne doit conserver que la partie contenant la bande radio recherchée.

- Interpolation et décimation

La fréquence d'échantillonnage imposée par la carte son est de 48kHz et la clef RTL-SDR échantillonne à $f_e=2\text{MHz}$. Un calcul rapide nous montre que $3*f_e/5/5/5 = 48\text{kHz}$. Avec le filtre nous venons de faire $f_e/5=400\text{kHz}$. Le bloc Rational Resampler permet de réaliser une interpolation et une décimation.

$$\text{Remarque : } \frac{48 \text{ kHz}}{2 \text{ MHz}} = \frac{24}{1000} = \frac{3}{125} = \frac{3}{5 \times 5 \times 5}$$



Implantez un bloc Rational Resampler à la sortie du filtre passe-bas dont les caractéristiques sont :

- **Une décimation pour diviser la fréquence d'échantillonnage par ce nombre : 5**
- **Une interpolation pour multiplier la fréquence d'échantillonnage par ce nombre : 3**

Observez le spectre du signal obtenu après cette manipulation. Attention la fréquence d'échantillonnage (`samp_rate`) du bloc QT GUI frequency sink est maintenant de 240kHz (2000kHz x3/5/5).

• Démodulation

Le signal obtenu est modulé en fréquence. Il existe dans GNURadio un bloc (WBFM Receive) qui permet de réaliser la démodulation de fréquence. De plus en profitera pour réaliser la dernière décimation par 5.



Rajoutez un bloc WBFM Receive à la sortie du bloc Rational Resampler dont les caractéristiques sont :

- **Quadrature Rate : 240e3 (3*fe/5/5)**
- **Audio Décimation : 5**

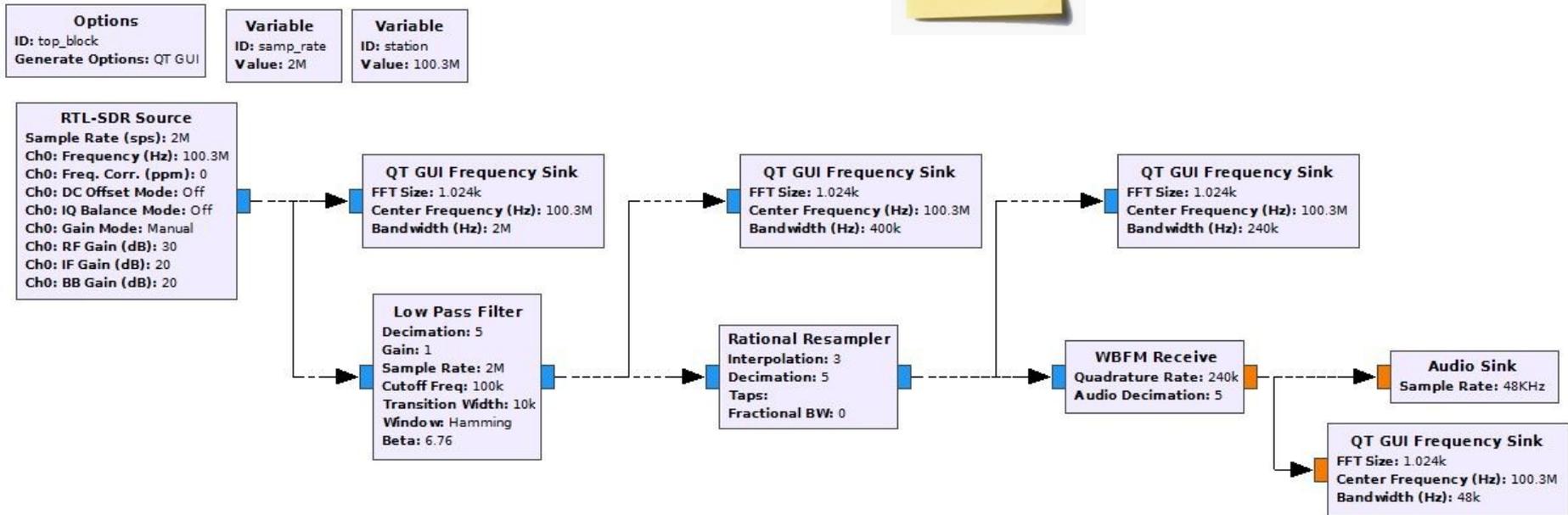
Observez le spectre du signal obtenu après cette manipulation. Attention à la compatibilité du type de données entre la sortie du bloc WBFM Receive et l'entrée du QT GUI frequency sink. Attention également à la fréquence d'échantillonnage (samp_rate) du bloc QT GUI frequency sink qui est maintenant de 48kHz (2000kHz x3/5/5/5).

$$\text{Remarque : } \frac{48 \text{ kHz}}{2 \text{ MHz}} = \frac{24}{1000} = \frac{3}{125} = \frac{3}{5 \times 5 \times 5}$$

- Écouter la radio

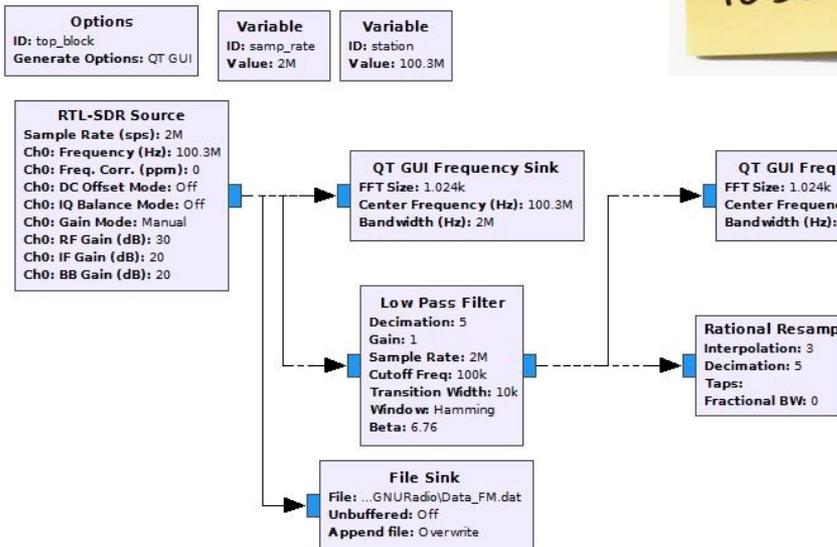
Rajoutez le bloc Audio Sink à la sortie du bloc WBFM Receiver dont la caractéristique est :

- Sample Rate : 48kHz**

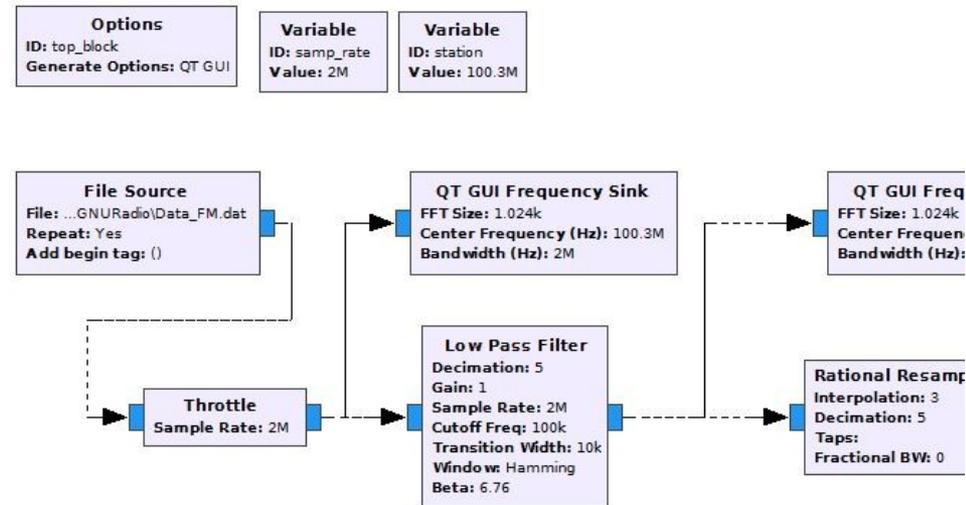


- Enregistrer un signal puis le rejouer pour l'étudier

A l'aide du bloc File Sink il est possible d'enregistrer un signal pour l'étudier plus tard. Dans le champ File saisir le nom de fichier ainsi que l'endroit où il sera sauvegardé.
Attention : le fichier devient très vite de taille volumineuse (960Mo pour une minute).

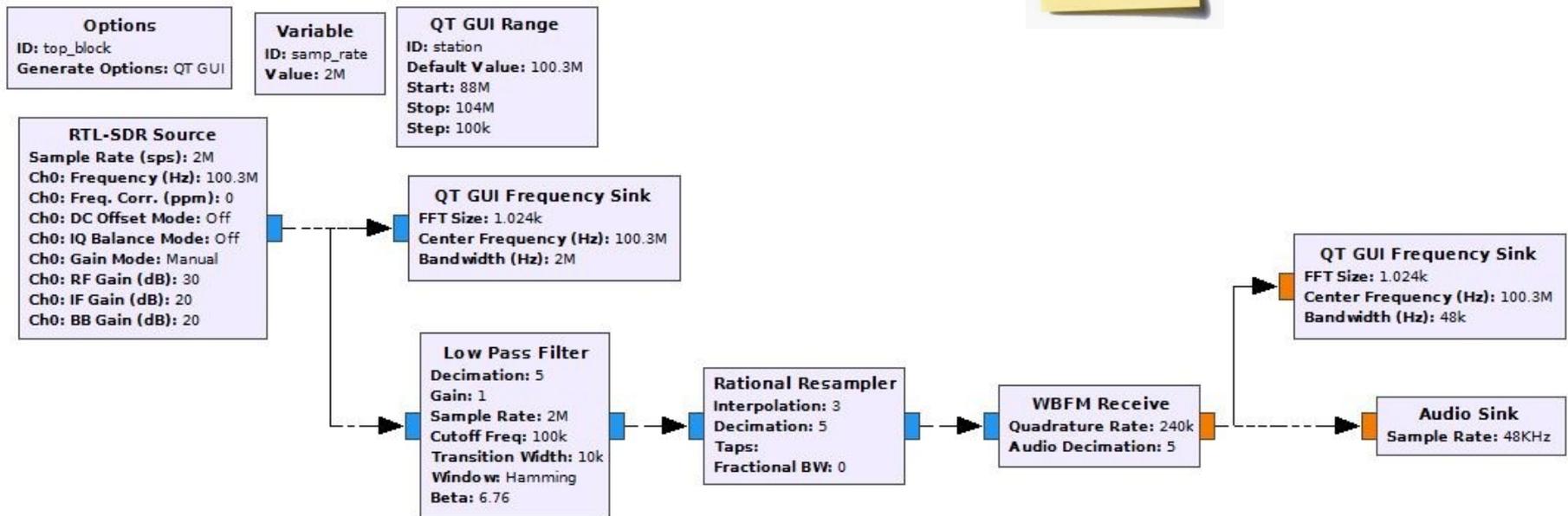


A l'aide du bloc File Source il est possible de rejouer à l'infinie si Repeat est à Yes un signal d'un fichier enregistré. Dans le champ File aller chercher le fichier voulu.
Attention : placer un bloc Throttle entre le bloc File Source et le bloc de premier traitement, afin d'imposer la fréquence d'échantillonnage.



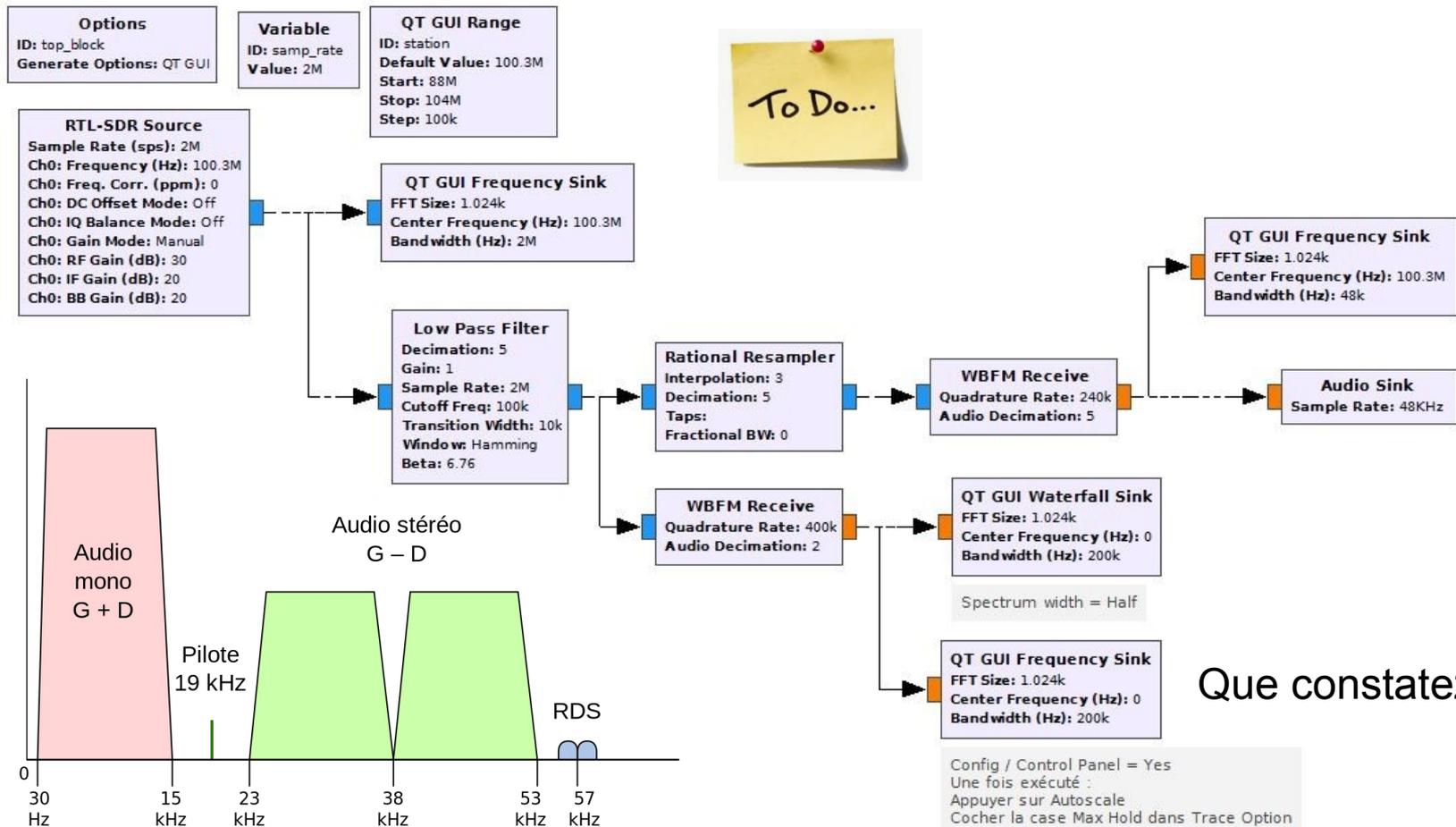
• Changement de station de radio

Pour le changer la station de radio, on peut modifier la variable station. Mais cela nécessite de relancer le programme à chaque fois. Une autre méthode consiste à remplacer la variable station par un bloc QT GUI Range et le paramétrer.



• Visualiser un spectrogramme

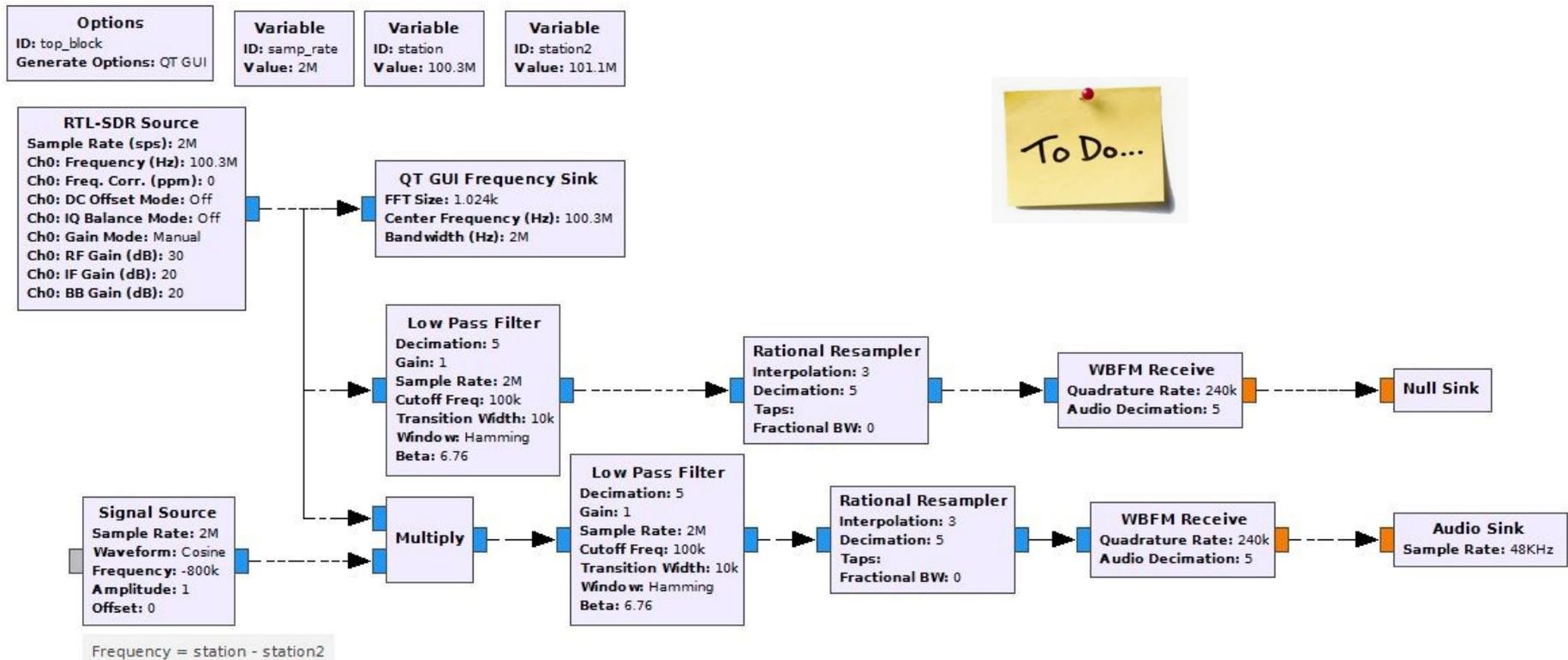
A l'aide du bloc QT GUI Waterfall Sink, on peut visualiser le spectrogramme d'un signal, c'est-à-dire tracer l'amplitude du signal en fonction du temps et de la fréquence (spectre du signal en fonction du temps).



Que constatez-vous ?

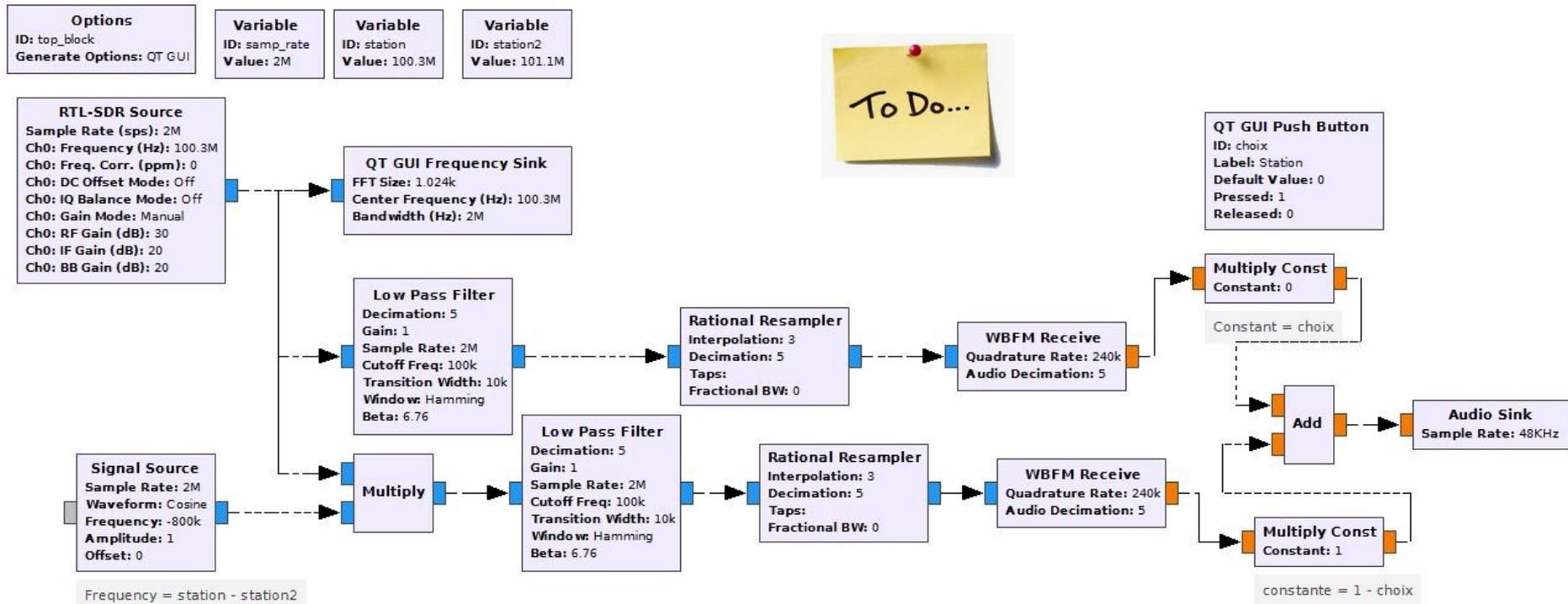
• Réception de deux stations radios

Il est possible d'écouter deux stations l'une après l'autre, ou en même temps si on aime la cacophonie, du moment qu'elles se trouvent dans la bande entre $f_p - f_e/2$ et $f_p + f_e/2$.



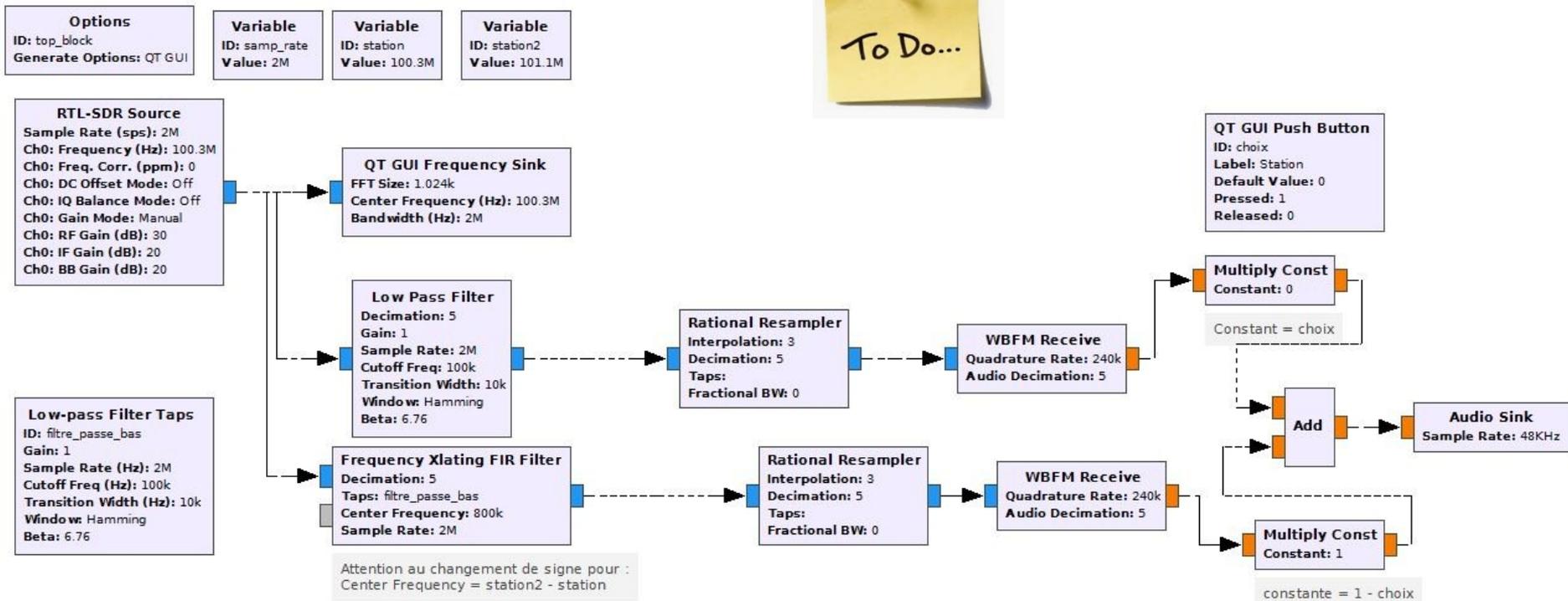
• Réception de deux stations radios (amélioration 1)

A l'aide du bloc QT GUI Push Button et du bloc Multiply Const, il est possible de basculer d'une station radio à une autre.



• Réception de deux stations radios (amélioration 2)

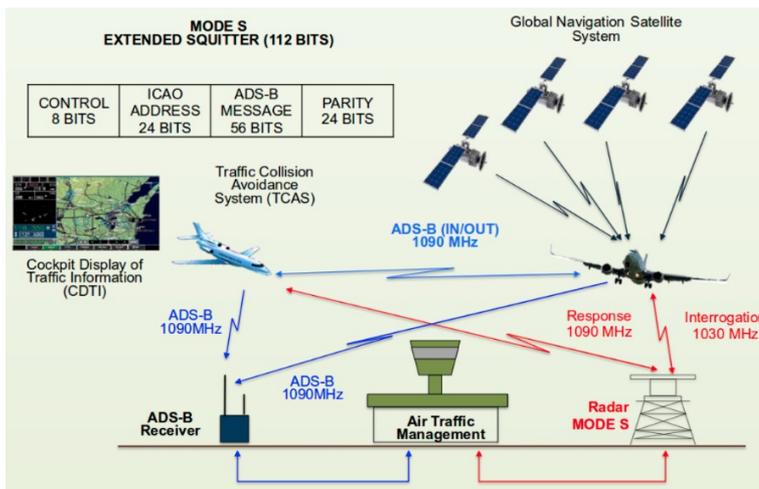
Le bloc Frequency Xlating FIR Filter permet de remplacer la multiplication (transposition en fréquence), le filtrage puis la décimation. Le filtrage est réalisé avec le bloc Low-pass Filter Taps. Dans le champ Taps du bloc Frequency Xlating FIR Filter on appelle le bloc Low-pass Filter Taps pour réaliser le filtrage.



• Objectif

Avec un dongle RTLSDR, il est possible de recevoir, décoder et suivre, soi-même, les infos de géolocalisation GPS envoyées en temps réel par les avions qui passent près de chez soit !

L'objectif de cette étape est de mettre en pratique ce que vous avez appris dans les parties précédentes. Il s'agit de faire un récepteur ADS-B à partir d'un dongle RTLSDR. Pour des raisons de temps, votre travail s'arrêtera à la visualisation temporelle des messages sans chercher à les décoder. Cette étape d'informatique industrielle pourra être réalisée dans un second temps.

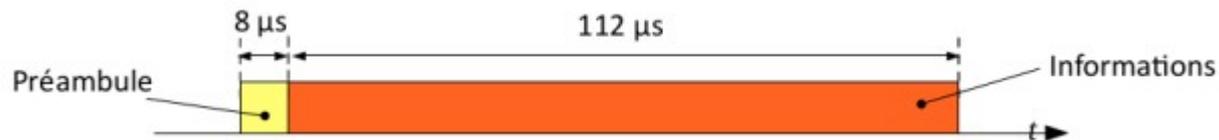


Un avion équipé de l'ADS-B détermine sa position par le système de positionnement satellite GPS et envoie périodiquement sa position, aux stations au sol (tour de contrôle) ainsi qu'aux autres avions évoluant dans la zone (figure ci-contre). La fréquence utilisée est le 1090 MHz.

• Présentation des signaux ADS-B (1/3)

Les signaux ADS-B (Automatic Dependent surveillance-Broadcast) sont émis par les avions, environ toutes les secondes, pour communiquer leur immatriculation (adresse OACI : Organisation de l'Aviation Civile Internationale), leur position (longitude, latitude, altitude), leur destination...

Ils utilisent une modulation particulière basée sur la modulation ASK et la modulation par position d'impulsion (PPM : « Pulse Position Modulation »). Toutes ces informations (immatriculation, position, destination...) sont transmises à la fréquence porteuse de 1090 MHz dans une trame d'une durée de 120 μ s. La trame ADS-B est constituée d'un préambule de 8 μ s suivie des informations codées en binaire.

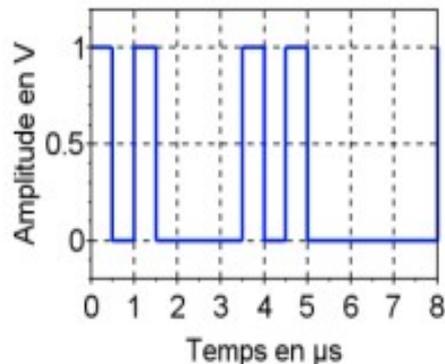


- Présentation des signaux ADS-B (2/3)

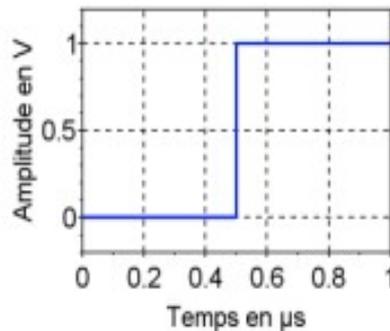
Le préambule, toujours identique, permet de synchroniser temporellement le signal reçu pour détecter les bits d'informations.

Un bit a une durée de $1 \mu\text{s}$, donc il y a 112 bits dans une trame ADS-B ayant un débit binaire de 1 Mbits/s.

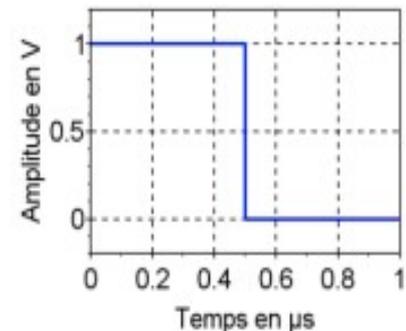
Le préambule et les bits sont codés à l'aide d'impulsion de largeur $0,5 \mu\text{s}$.



a. Préambule



b. Impulsion encodant le bit 0

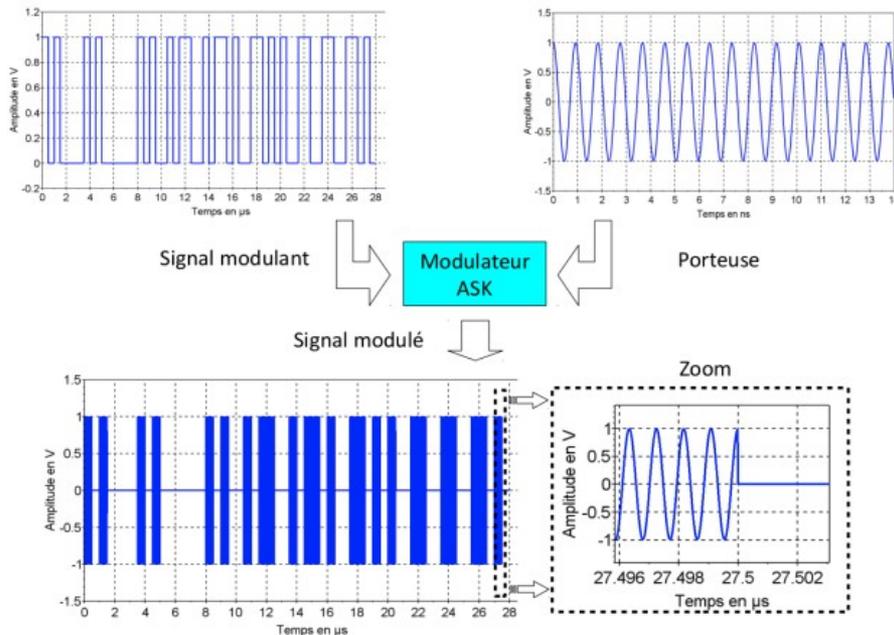


a. Impulsion encodant le bit 1

• Présentation des signaux ADS-B (3/3)

La trame ADS-B (le signal modulant) est multipliée par une sinusoïde à la fréquence de 1090 MHz (la porteuse) : c'est la modulation ASK.

A la sortie du multiplicateur, le signal modulé est amplifié puis émis par une antenne pour une transmission sans fil.



A la réception, afin de pouvoir décoder l'information, il faut :

- détecter l'enveloppe pour retrouver la trame,

- faire une synchronisation temporelle pour localiser le début de la trame (maximisation de la corrélation entre le préambule et la séquence binaire reçue),

- récupérer les bits et en extraire les informations.

- Capture du signal ADS-B



Exercice : à l'aide du bloc RTL-SDR Source, recevez le signal ADS-B. Pour observer le spectre du signal ajoutez le bloc QT GUI Frequency sink. Paramétrez le bloc RTL-SDR source de la manière suivante :

- Échantillonnage à 2 MHz
- Fréquence centrale : 1090 MHz
- RF Gain : 50 dB

Observez le spectre du signal obtenu.

Il peut être intéressant d'activer l'option Max hold (accessible dans le répertoire config via l'option control pannel du bloc QT GUI frequency sink).

Ce signal est généralement de niveau assez faible et peu visible.

- **Démodulation du signal ADS-B**

En sortie du bloc RTL-SDR source on mesure le signal modulé en amplitude, autour de la fréquence 0.

Ce signal est généralement de niveau assez faible et difficilement exploitable. Nous allons le remettre en forme pour obtenir un signal uniquement constitué de 0 et de 1.



***La démodulation est réalisée avec le bloc Complex to Mag.
La remise en forme est réalisée avec le bloc Threshold en agissant sur une valeur seuil afin d'obtenir un flux de données binaires. Le réglage du seuil se fait en observant la forme temporelle du signal démodulé (à la sortie du bloc Complex to Mag , en utilisant le bloc QT GUI time sink).***

Réaliser ce circuit et régler le seuil du bloc Threshold afin d'obtenir un flux de 0 et de 1.

L'étape suivante sera la synchronisation de la trame obtenue.

• Recherche du préambule et synchronisation

Afin d'identifier les trames ADS-B nous allons rechercher les trames dont le préambule correspond à l'entête ADS-B. En considérant 2 échantillons par symboles (c'est à dire une fréquence d'échantillonnage de 2 MHz), le préambule est constitué des valeurs binaire suivante : 10 10 00 01 01 00 00 00

Afin d'identifier le préambule, il faut transformer le type de signal float en unsigned char. Le bloc Float to Uchar permet de réaliser cette conversion. Le bloc Correlate Access Code- Tag permet de trouver le préambule. Pour cela, il faut renseigner les paramètres suivants :

- ***Access Code= 1010000101000000***
- ***Threshold=0***
- ***Tag Name :preamble***



Le bloc suivant (Uchar to float) permet de convertir le type unsigned char en type float. En utilisant le bloc QT GUI time sink, il est possible de synchroniser le trigger sur le début du préambule. Pour cela, il faut remplir le champ Trigger tag key avec la valeur preamble (le champs trigger est accessible dans l'onglet Trigger).

Réaliser ce flowchart, observez le résultat

std::exception
Skipping it

Id	Value
Imports	