Introduction to Software Defined Radio

Introduction

The “Software Defined Radio” refers to a type of radiofrequency transceiver in which most of the processing is done digitally, whether it is the transmission or the reception of data. For a reception, the received signal is digitized through an ADC, then processed (for instance: filtered, decimated, demodulated, decoded, etc.). For the transmission, the data to transmit are processed (for instance: coded, modulated, etc.), then converted through a DAC and sent to the RF front-end.

First part: Presentation of the acquisition device: In-phase/Quadrature Software-Defined Radio transceiver

We will use a National Instruments USRP-2900 software defined radio transceiver which is connected to a computer via a Gigabit Ethernet connection (Fig. 1).

Fig. 1: National Instruments USRP-2900 transceiver

At the end of the reception chain, the computer stores the flow of samples in a file that we use during the following exercises under GNURadio.

1 - We have :

$$r'_R(t) = S_{RF}(t) * cos(2\pi fct) \qquad r'_I(t) = S_{RF}(t) * sin(2\pi fct)$$

Therefore we find r_R like so :

$$\begin{aligned} r'_R(t) &= (s_R(t)cos(2\pi f0t) - s_I(t)sin(2\pi f0t)) * cos(2\pi fct) \\ &= s_R(t)cos(2\pi f0t)cos(2\pi fct) - s_I(t)sin(2\pi f0t)cos(2\pi fct) \\ &= s_R(t)(\frac{cos(2\pi f0t - 2\pi fct) + cos(2\pi f0t + 2\pi fct)}{2}) - s_I(t)(\frac{sin(2\pi f0t + 2\pi fct) + sin(2\pi f0t - 2\pi fct)}{2}) \\ &= \frac{s_R(t)}{2}(cos(2\pi t(f0 - fc)) + cos(2\pi t(f0 + fc))) - \frac{s_I(t)}{2}sin(2\pi t(f0 + fc)) + sin(2\pi t(f0 - fc))) \end{aligned}$$

Following the same procedure, we find $r_I(t)$ like so :

$$\begin{aligned} r'_I(t) &= (s_R(t)cos(2\pi f0t) - s_I(t)sin(2\pi f0t)) * sin(2\pi fct) \\ &= s_R(t)cos(2\pi f0t)sin(2\pi fct) - s_I(t)sin(2\pi f0t)sin(2\pi fct) \\ &= s_I(t)(\frac{cos(2\pi f0t - 2\pi fct) - cos(2\pi f0t + 2\pi fct)}{2}) - s_R(t)(\frac{sin(2\pi f0t + 2\pi fct) - sin(2\pi f0t - 2\pi fct)}{2}) \\ &= \frac{s_I(t)}{2}(cos(2\pi t(f0 - fc)) - cos(2\pi t(f0 + fc))) - \frac{s_R(t)}{2}sin(2\pi t(f0 + fc)) - sin(2\pi t(f0 - fc))) \end{aligned}$$

2 - If we employ a translation in the baseband by heterodyning by setting $f0 = fc$, we additionally need to set characteristics to the h filter in order to obtain the result : $r_R(t) = s_R(t)$ and $r_I(t) = s_I(t)$ knowing that $r_R(t) = r'_R(t) * h$, same for $r_I(t)$.

With $f0 = fc$ we have :

$$r'_R(t) = \frac{s_R(t)}{2}(1 + cos(4\pi tfc)) - \frac{s_I(t)}{2}sin(4\pi tfc)$$
$$r'_I(t) = \frac{s_I(t)}{2}(1 - cos(4\pi tfc)) - \frac{s_R(t)}{2}sin(4\pi tfc)$$

From here, to find a satisfactory answer, we must apply the Fourrier transform knowing that :

$$cos(2\pi f0t) = \frac{\delta(f-f0)+\delta(f+f0)}{2} \qquad sin(2\pi f0t) = \frac{\delta(f-f0)-\delta(f+f0)}{2} \qquad \delta(f) = 1$$

We therefore apply the Fourier transform :

$$R'_R(f) = F\{r'_R(t)\}$$
$$= s_R(f) * \frac{1}{2}\delta(f) * [\delta(f) + \frac{\delta(f+2f0)-\delta(f-2f0)}{2}] - s_I(f) * \frac{1}{2}\delta(f) * [\frac{j}{2}\delta(f + 2f0) - \delta(f - 2f0)]$$
$$R'_I(f) = F\{r'_I(t)\}$$
$$= s_I(f) * \frac{1}{2}\delta(f) * [\delta(f) - \frac{\delta(f+2f0)-\delta(f-2f0)}{2}] - s_R(f) * \frac{1}{2}\delta(f) * [\frac{j}{2}\delta(f + 2f0) - \delta(f - 2f0)]$$

Through a series of operations using properties such as $x(f) * \delta(f - f\tau) = x(f - f\tau)$ we are able to obtain the following result :

$$R'_R(f) = \frac{1}{4}[2s_R(f) + s_R(f + 2f0) - s_R(f - 2f0) + js_I(f - 2f0) - js_I(f + 2f0)]$$
$$R'_I(f) = \frac{1}{4}[2s_I(f) - s_I(f + 2f0) - s_I(f - 2f0) + js_R(f - 2f0) - js_I(f + 2f0)]$$

We can see that these results for real and imaginary parts are very similar. In fact they are identical, except for a phase shift of π between each result. Following the explanation given in class, we have the following results for the h filter :

- It needs to be a low pass filter with $\frac{B}{2} < fc < 2f0 - \frac{B}{2}$ with $f0 > \frac{B}{2}$ to avoid overlapping
- $|H(f)| = 2 \; if \frac{B}{2} < f < 2f0 - \frac{B}{2}$ or $|H(f)| = 0 \; if \; f > 2f0 - \frac{B}{2}$

3 - The receiver cannot work with a large band signal, that is to say for a signal with : ($|f_{max}| > 2 \cdot f0 - \frac{B}{2}$). Indeed this can cause a spectral overlap which makes it impossible to isolate and demodulate into a clean signal.

4 - In order to recover $r_R(t)$ from $r_R(k \cdot Te)$ we must abide by the golden rule of signal sampling : we must respect the Shannon-Nyquist criterion, which gives us : $fe > 2 f_{max}$ which gives us in turn $\frac{1}{Te} > B$ and therefore $Te < \frac{1}{B}$.

5 - Interchanging the ADC component and the frequency transposition element, though theoretically possible, would have the effect of making the sampling rate of the ADC very high. As engineers, it is necessary to consider the cost of our solutions, and therefore, interchanging these elements has no real benefit for the cost hike it introduces into our system.

6 - We wish to express in the frequential domain and in the temporal domain the analytical signal and the complex envelope depending on fo of the following :

$$s_{RF}(f) = A(t)cos(2\pi f0t + \phi(t)) = s_R(t)cos(2\pi f0t) - s_I(t)sin(2\pi f0t)$$

Using the Fourier transform we obtain :

$$s_{RF}(f) = \frac{s_R(f)}{2} * [\delta(f - f_0) + \delta(f + f_0)] - \frac{s_I(f)}{2j} * [\delta(f - f_0) - \delta(f + f_0)]$$

$$s_R(f) * \delta(f - f_0) = s_R(f - f_0) \; and \; -\frac{s_I(f).j}{2j^2} * \delta(f - f_0) = js_I(f - f_0)$$

$$s_R(f) = \frac{1}{2}[s_R(f - f_0) + s_R(f + f_0) + js_I(f - f_0) - js_I(f + f_0)]$$

We find that the analytical signal is therefore :

$$sa(f) = s_{RF}(f) + j(-j * sgn(f)s_{RF}(f))$$
$$= s_{RF}(f) + sgn(f)sRF(f) = 2s_{RF}(f) \; because \; sgn(f) = 1$$
$$= s_{RF}(ff0) + js_I(f - f0) = [s_R(f) + js_I(f)] * \delta(f - f0)$$

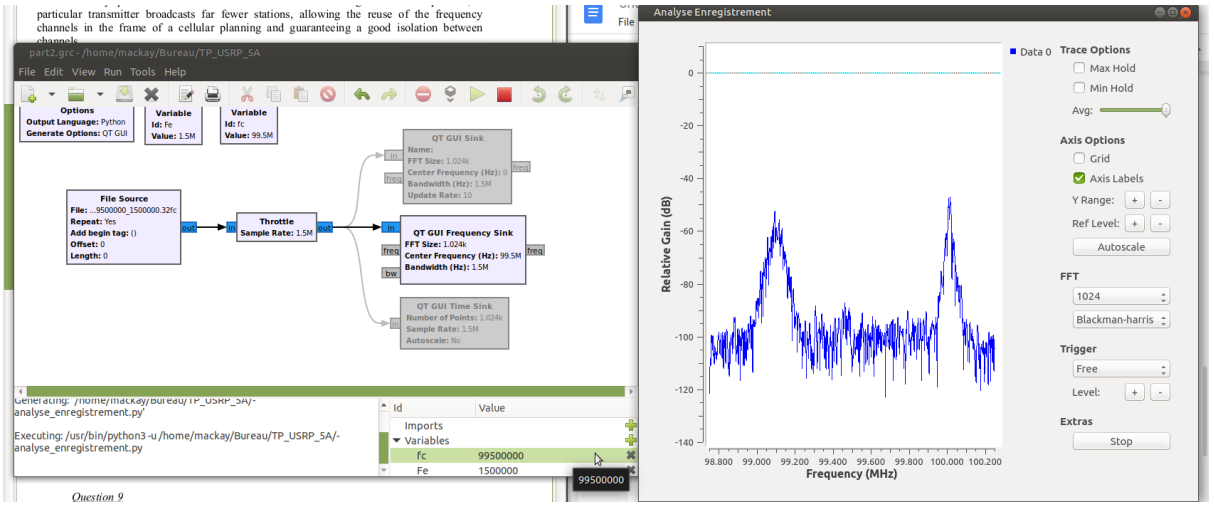
Therefore : $sa(t) = [s_R(t) + js_I(t)] \; exp\{j2\pi f0t\}$ in the time domain thanks to the inverse Fourier transform.

Second part: Reception of frequency modulation (FM) broadcasting

We focus on the sub-band between 87.5 MHz and 108 MHz, which is now dedicated to FM broadcasting. The different channels are spaced by at least 100 kHz so that it is theoretically possible to have 203 simultaneous broadcasting stations. The FM broadcasting recording file fm_99500000_1500000.32fc has been obtained thanks to the acquisition system introduced in the first part. It was recorded at Toulouse in 2015, using a center frequency of fc = 99.5 MHz and a sampling frequency of Fe = 1.5 MHz.

1. Frequency analysis of the recording

We will use GNURadio to retrieve information from this sample :

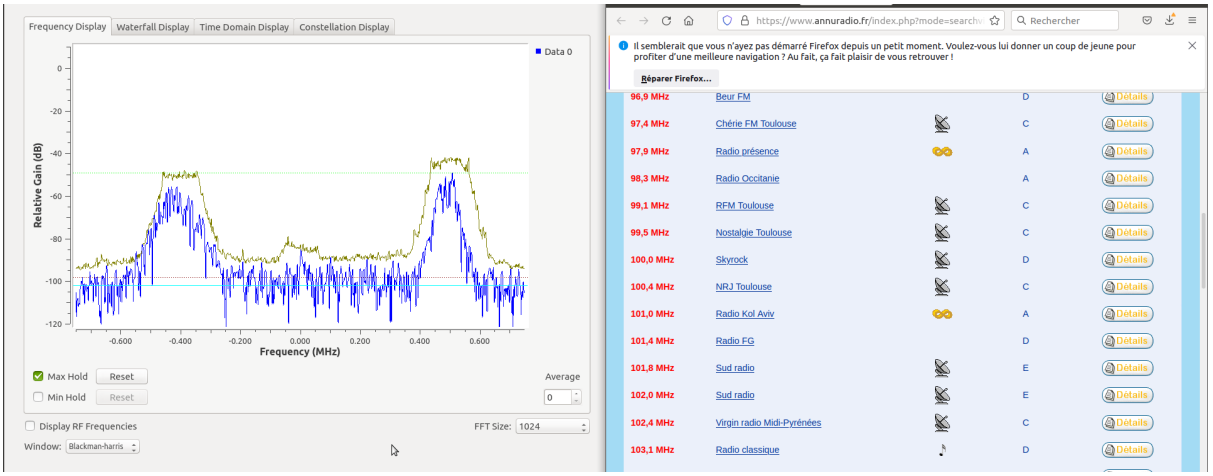


With the following block functions :
Option Block : Autogenerated option block containing different metadata and postscript to run. The options block sets special parameters for the flow graph. Only one option block is allowed per flow graph. Title, author, and description parameters are for identification purposes. The generate options control the type of code generated.
Variable : This block maps a value to a unique variable.
File Source : Reads raw data values in binary format from the specified file.
Throttle Block : Throttle (add an upper limit to the) flow of samples such that the average rate does not exceed the specific rate (in samples per second). If there is no throttle, the computer will not follow a “real time” mode of operation.
QT GUI Frequency Sink : Creates an FFT of our signal. A graphical sink to display multiple signals in frequency.
Possibility to use a time sink to see the waterfall representation of our signal.

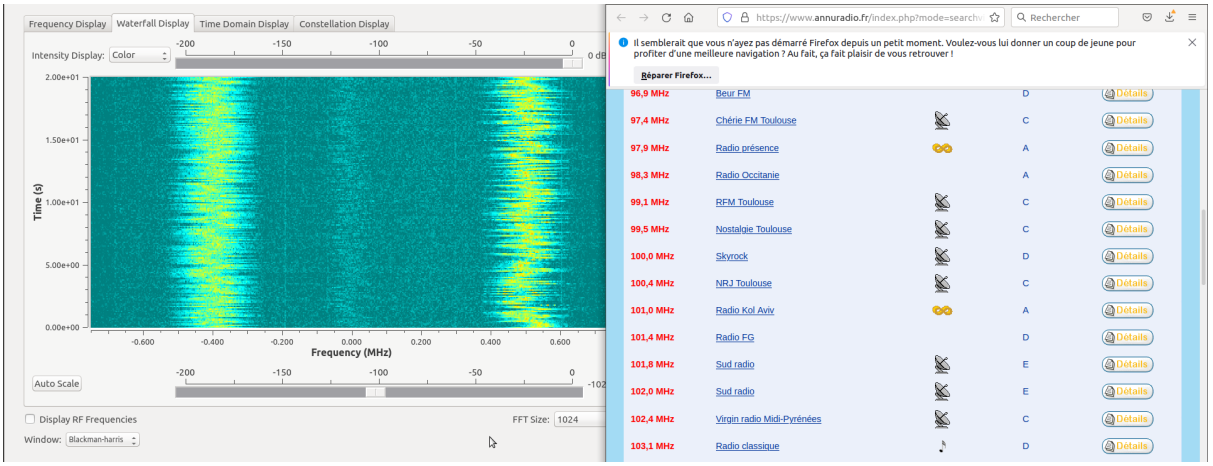
Question 8 :
Fe = 1.5MHz, fc=99.5MHz, Sample Rate = 1.5MHz, the bandwidth as shown in question 4 is Fe.

Question 9 :

We can observe frequency channels –to be noted L with L = 3.
According to the allocation of frequencies in the FM band near Toulouse (https://www.annuradio.fr/index.php?mode=searchville&choixville=TOULOUSE), We can observe RFM Toulouse, Skyrock and Nostalgie Toulouse thanks to frequency display with max hold activated.

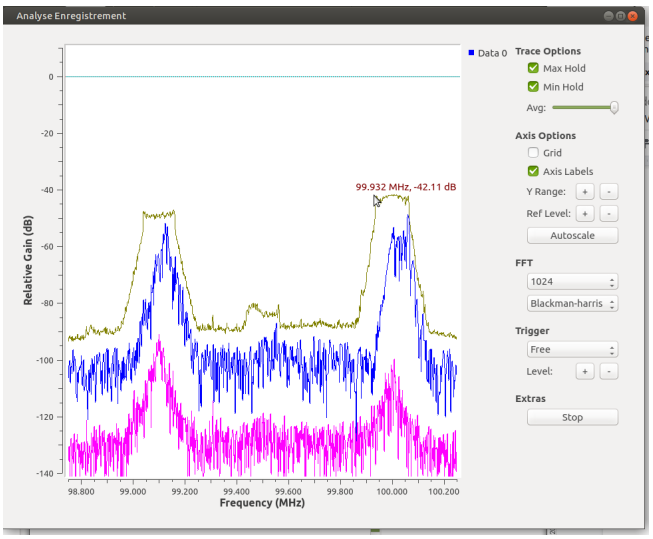


With the waterfall representation we can see a very faint amount of spectral information at 99.5MHz, for Nostalgie.



Question 10 :

From the frequency domain we compare the dB values of the peak of the signal and the peak of the noise : (SNR can be calculated in different but COHERENT ways)
SNR = (-46)-(-83) = 37dB pour RFM ou SNR = (-42)-(-86) =44dB pour Skyrock.
Depending on if we take the max, the min levels or the average, our results will vary!



Knowing that -3dB is already cutting our power in half, our SNR is pretty good. 40dB means a factor 10000 of our signal power over the noise : we will be able to demodulate. For Nostalgie, demodulating will be more of a challenge, the result won't be as clean.

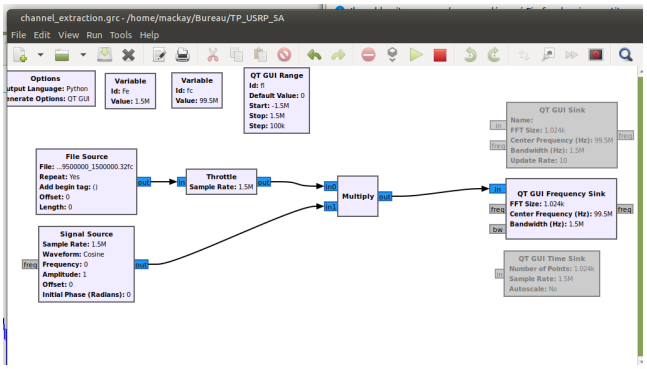
Question 11:

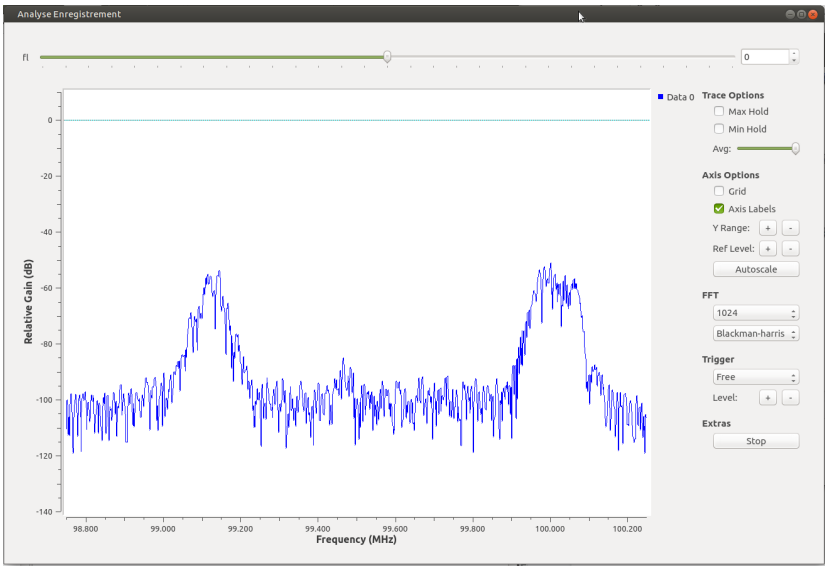
In waterfall we can deduce BW and the type of modulation. In a frequency domain, we can deduce the SNR and the BW. The BW is found by taking an instantaneous snapshot of our signal and by looking at the width of the signal at different dB values either side of the peak. We obtain with the instantaneous snapshot : 0.22MHz. By using the max values, our bandwidth seems larger, but this is the influence of using the max hold function. Here we obtain BW = 0.26MHz.

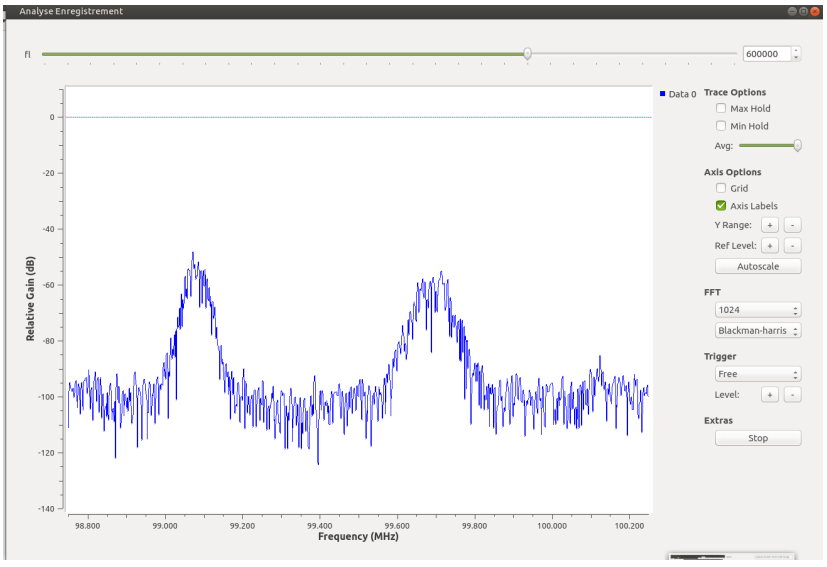
2. Channel extraction by frequency transposition and low-pass filtering

We want to receive each one separately by the use of a new processing chain based on the previous one. We can save it as "channel_extraction.grc". To do this, we propose a two stage reception described as:
1- frequency transposition in order to center the useful signal in terms of frequencies
2- low-pass filtering in order to attenuate the out-of-band noise.
We use three new blocks:
QT GUI Range : This block creates a variable that can be changed dynamically with a choice of widgets.
Multiply : Multiply across all input streams.
Signal Source : Signal generator: generates a variety of waveforms. Supports an output of type complex, float, int, and short. Block to create a complex exponential that will be used to transpose the signal.
We come back to baseband frequenciesusing offsets, centered at 0hz. For each channel, we use a different offsets to center the channel at 0Hz
BW = 1.5MZ so our treatment will be from -0.75MHz to 0.75MHz.

We use the following flow :







When we use a positive fl, the spectre shifts to the right, with a negative fl, the spectre shifts to the left.

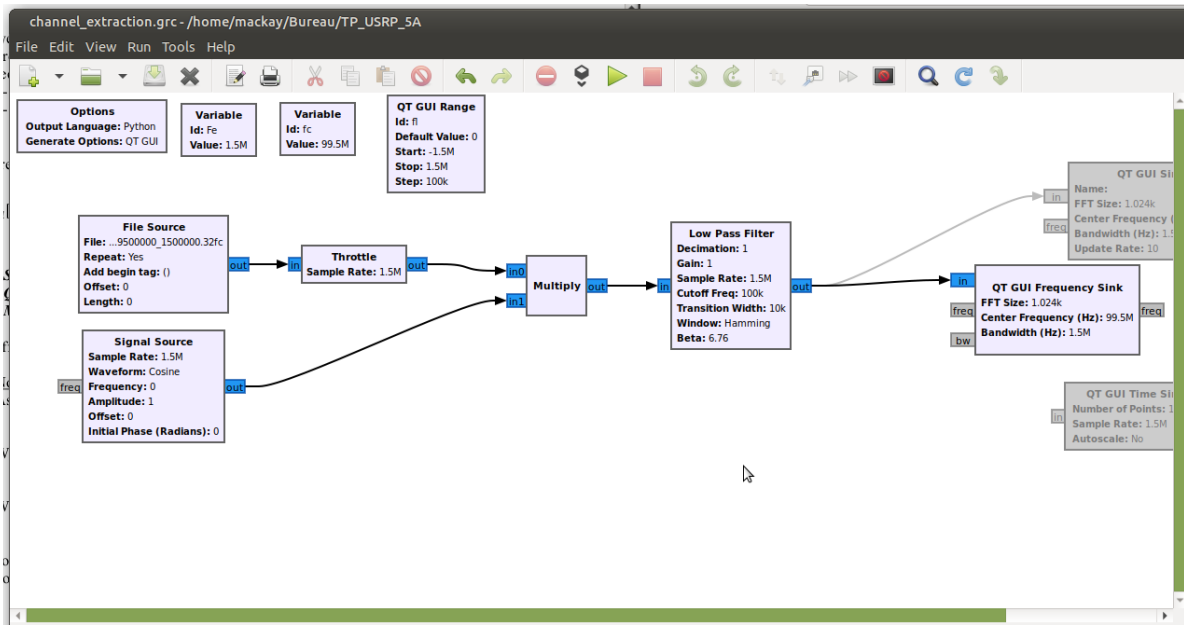
Question 12 :

The frequency offsets needed to center each channel is of : -99,1 MHz and -100Mhz.

Question 13 :

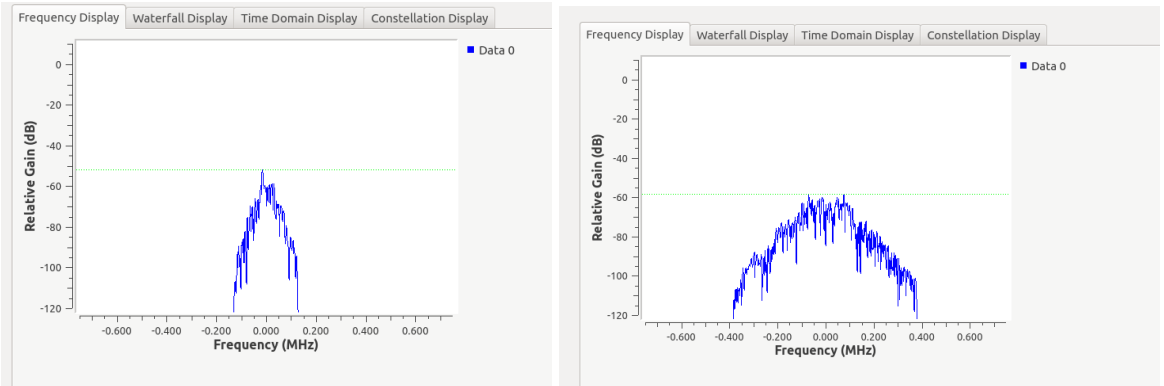
If the frequency offset is higher than the sampling frequency, i.e. fl is no longer bounded by +Fe and -Fe : the spectre repeats itself cyclically. We have a circular permutation linked to a periodicity (the exponential is 2pi periodic).

We must now implement a low-pass filter by considering the channel bandwidth from 2.1. To do this we implement the following flow into GNU Radio :



The cutoff frequency should be half of our channel bandwidth (measured to be around 0.22MHz) because we have centered the signal to baseband frequencies. Our transition frequency is chosen to have a reasonably steep incline of cutoff. Decimation is the fact that we keep one sample out of a total of 6 (opposite of interpolation). Less samples to treat. Decimation eliminates highest and lowest frequencies. Here we are centered at 0 and with a BW of 250kHz of interesting data but with a full spectrum BW of 1.5MHz.
By decimating by 6 we reduce the BW of the full spectrum by a factor of 6 starting at the extremities : ie reducing the BW = 1.5MHz/6=250kHz. We therefore only keep the information we want and eliminate useless info.

1 Decimation 3 decimation



Question 14 :

Low pass filter cutoff is at radio channel bw/2 = 125kHz, transition is 125e2Hz.

Question 15 :

Carson rule:

$$B = 2(\beta + 1)f_m$$

avec

- B: largeur de bande,
- β: indice de modulation,
- f_m: fréquence du signal modulant sinusoïdal

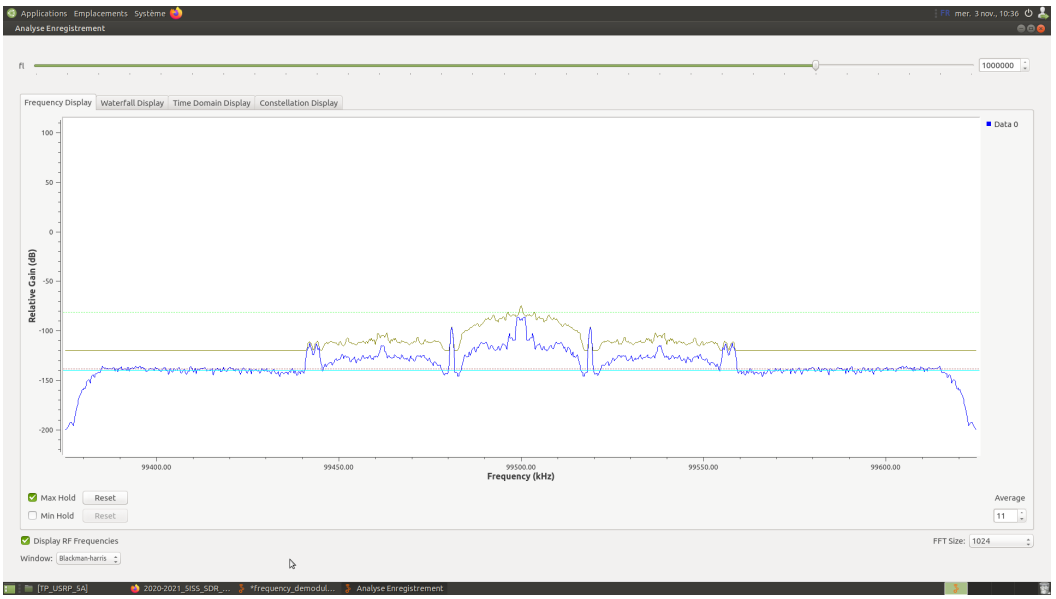
$BFM \approx 2(\Delta f + fM) \approx 2(75000 + 57000)$
$BFM \approx 264000$
We know that FM channels have a bandwidth of 256 kHz which means the theory is confirmed.

Question 16 :

Question unanswered due to lack of time. As engineers, it is necessary to take into account the time constraints of a project and prioritise the tasks that need to be accomplished. This question is very theoretical and the more practical questions of this lab seem to have more learning value.

Question 17 :

RDS uses a digital BPSK (two peaks) which allows us to send data about artists and song names.



We can see different peaks, which correspond to carrier information. They correspond to the Am/2 from figure 6.

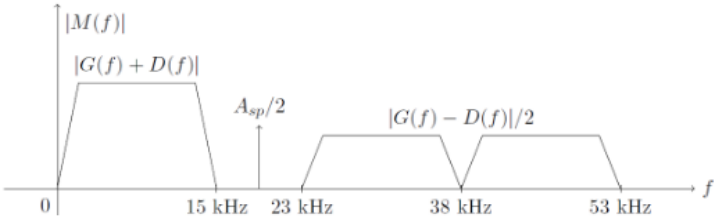
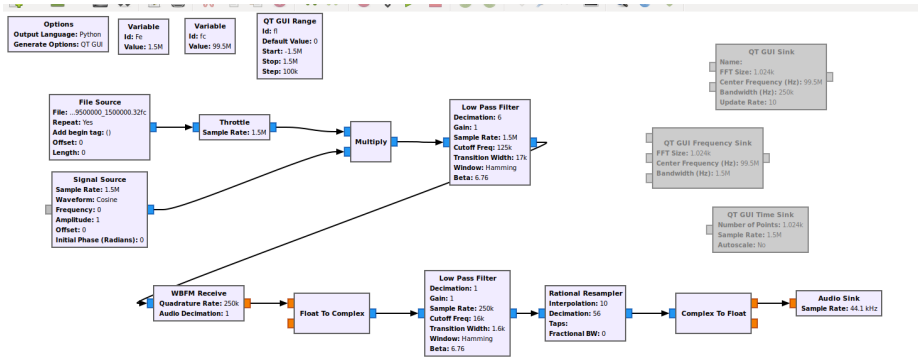


Fig. 6: Stereophonic composite signal before the frequency modulation

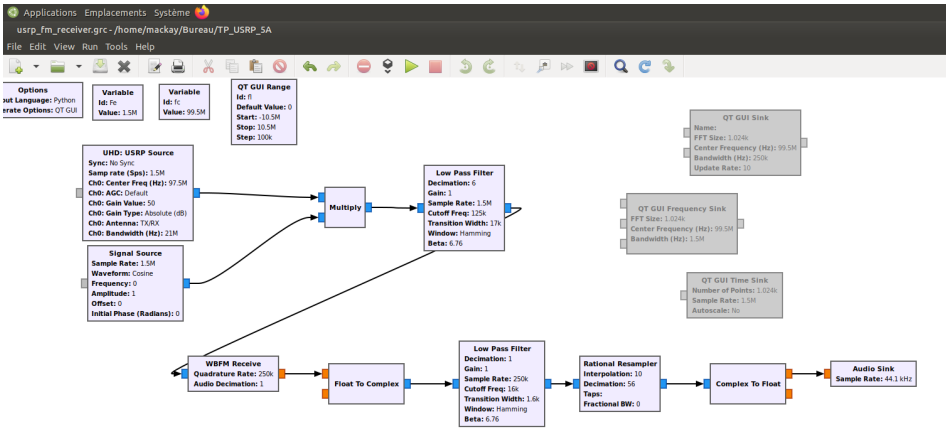
To demodulate our sample into a mono audio :



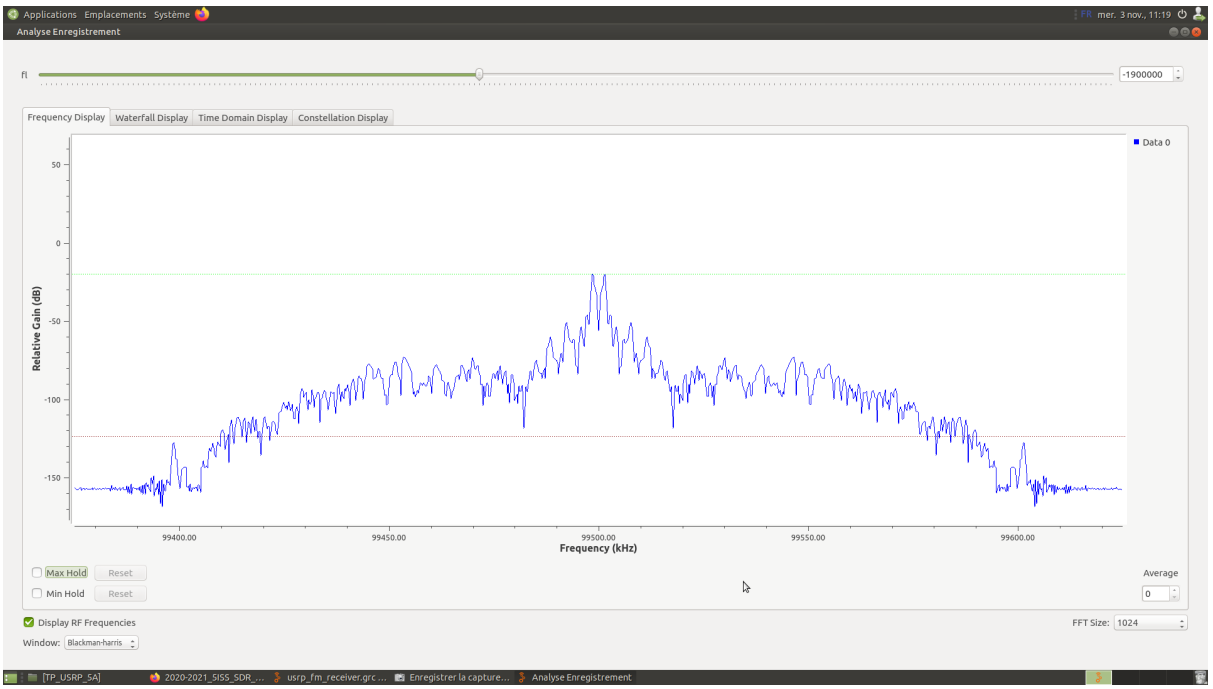
Our audio sink is of 44.1kHz, as it corresponds to the specs of our audio card.
On RFM, Jordy won a 50e check and the Sam Smith album. On nostalgie, YMCA by the village people is playing, and Counting Stars by One Republic is playing on Skyrock.

We will now attempt real-time demodulation.

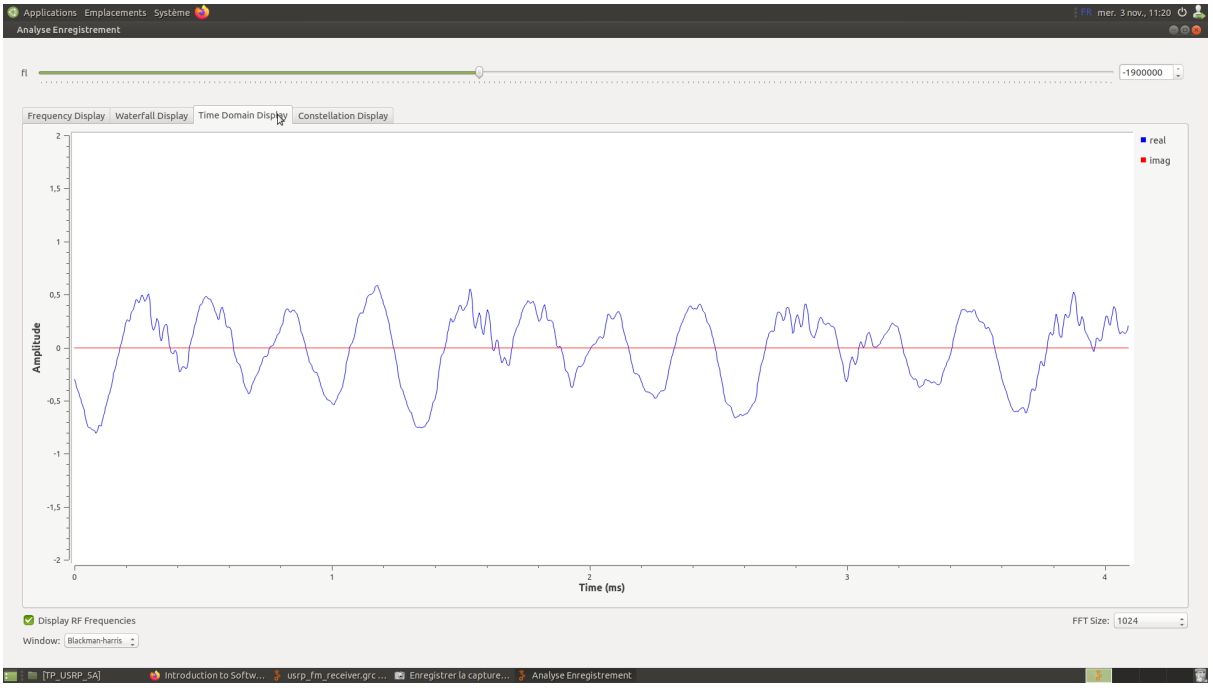
4. Real time implementation with an USRP receiver

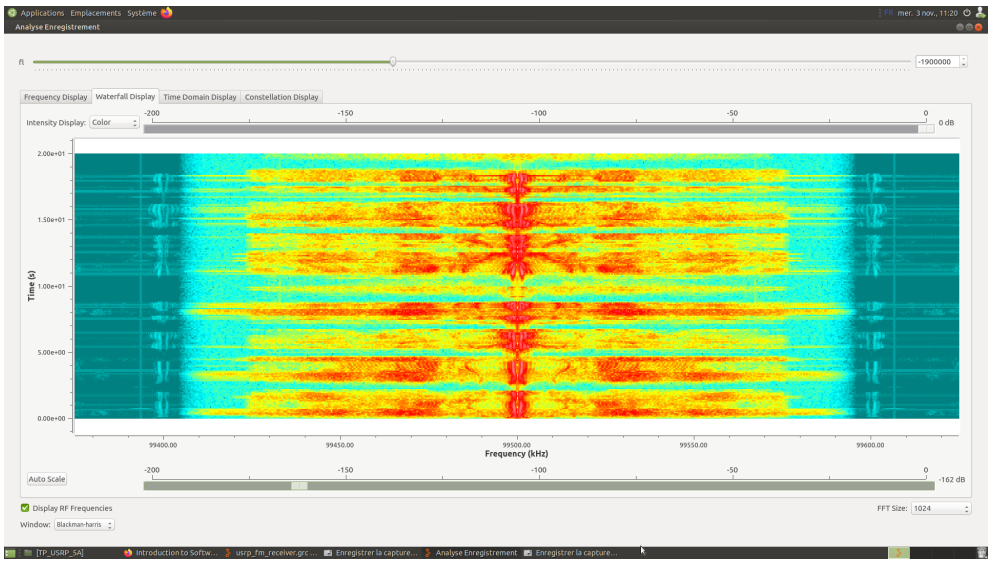


This system works as a means to demodulate in real time the FM radio from the 87MHz to 108MHz frequencies (using a slider), using an USRP. We are centered at the 97.5MHz.

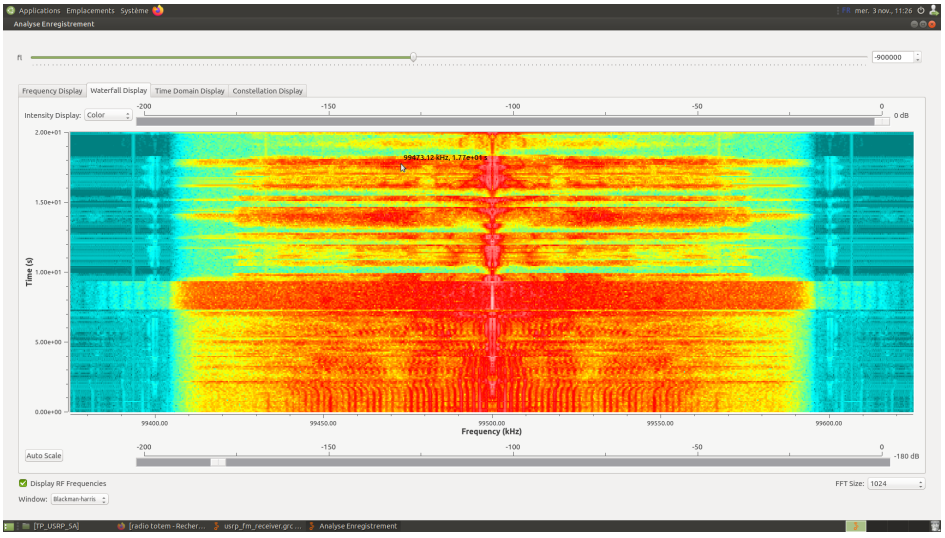


With the corresponding time domain display :





To the ear, the signal is already relatively clear. We can see in the different above visualisations that we receive clear frequencies, without too much residual noise. Below we switch to a frequency that is not a station and we can see the background noise.



Stereo :

The objective of stereo is to shift a component of the signal to extract relevant information to obtain two signals, to be emitted out of two speakers. By looking at the figure below, the objective is to shift the 38kHz to 0kHz and then filter it.

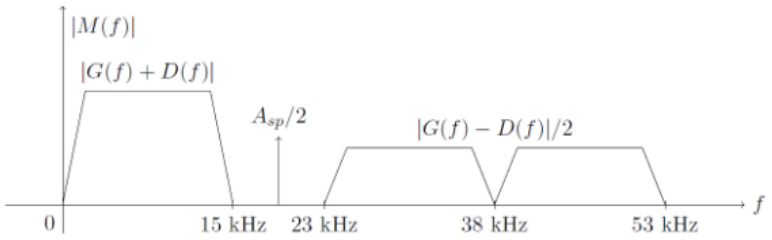
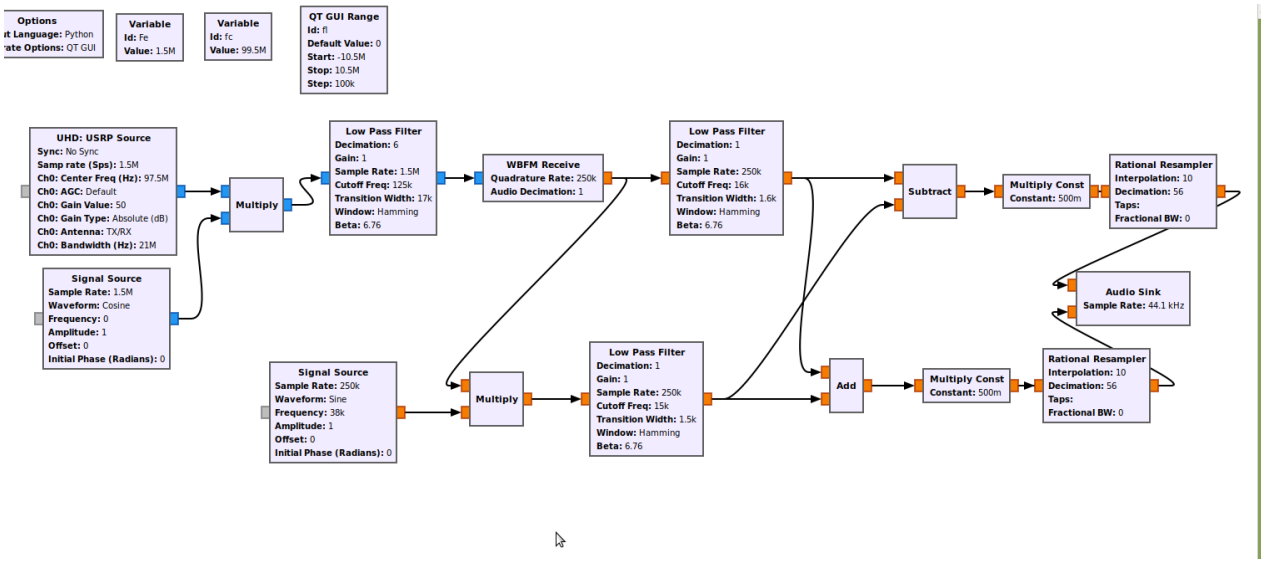
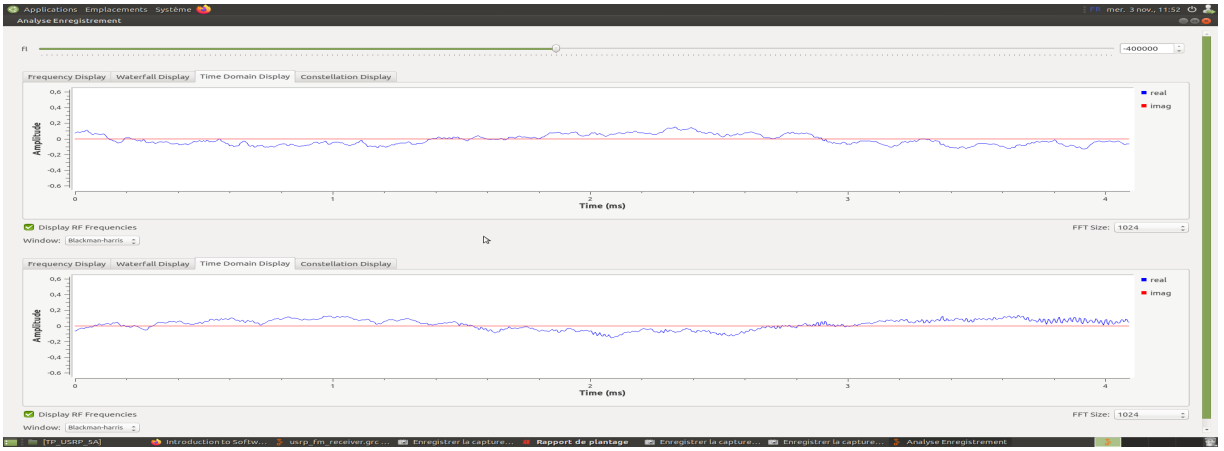


Fig. 6: Stereophonic composite signal before the frequency modulation

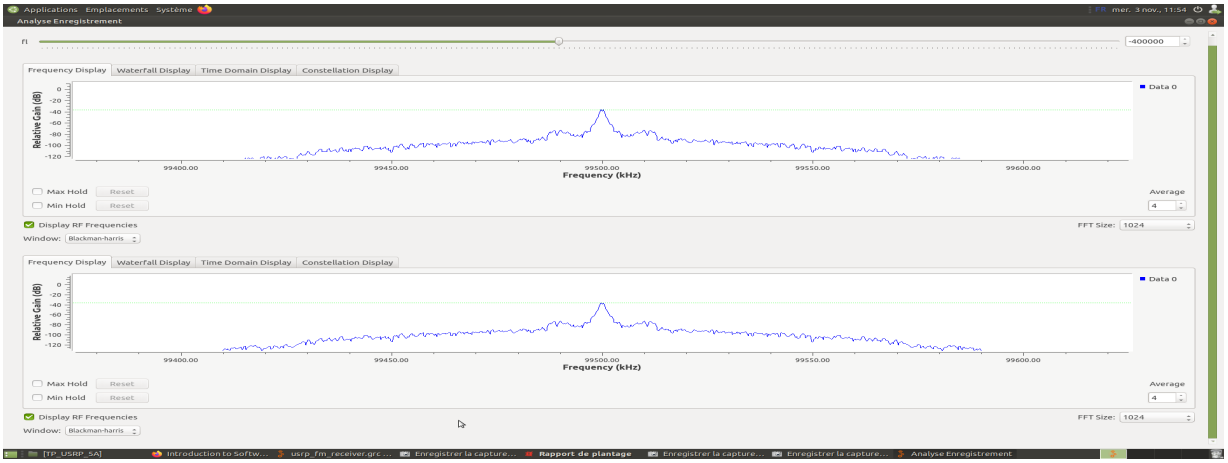
On GNURadio, this process can be put into place with the following flow :



Indeed, we double up our signal and affect these two signals by adding or subtracting each with each other in order to obtain two different signals that create a stereo effect. Offsetting our low pass filters in terms of cutoff frequency and therefore transition frequency is necessary to create the signal difference before adding/subbing.



We can see that our signals are not identical. The sound is clear with minimal background noise.



Conclusion :
This lab was effective in introducing the concept of Software Defined Radio as well as the different tools associated with this field such as GNU Radio. We were able to demodulate a real life FM radio signal to the point where listening to it was pleasant and in stereo. Further study is necessary to implement a real-time application with a connected USRP maybe.