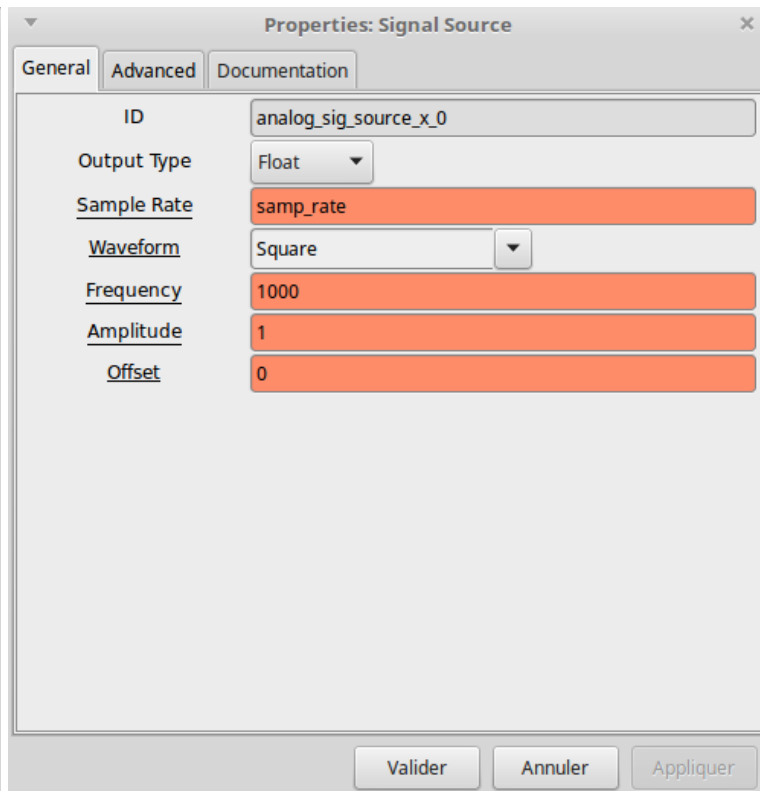
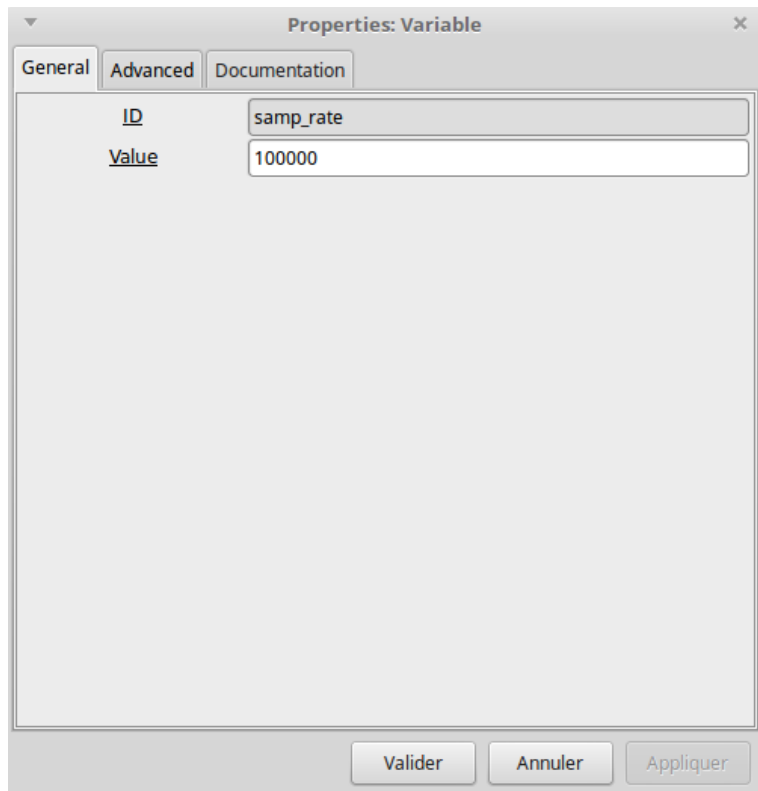


## Table des matières

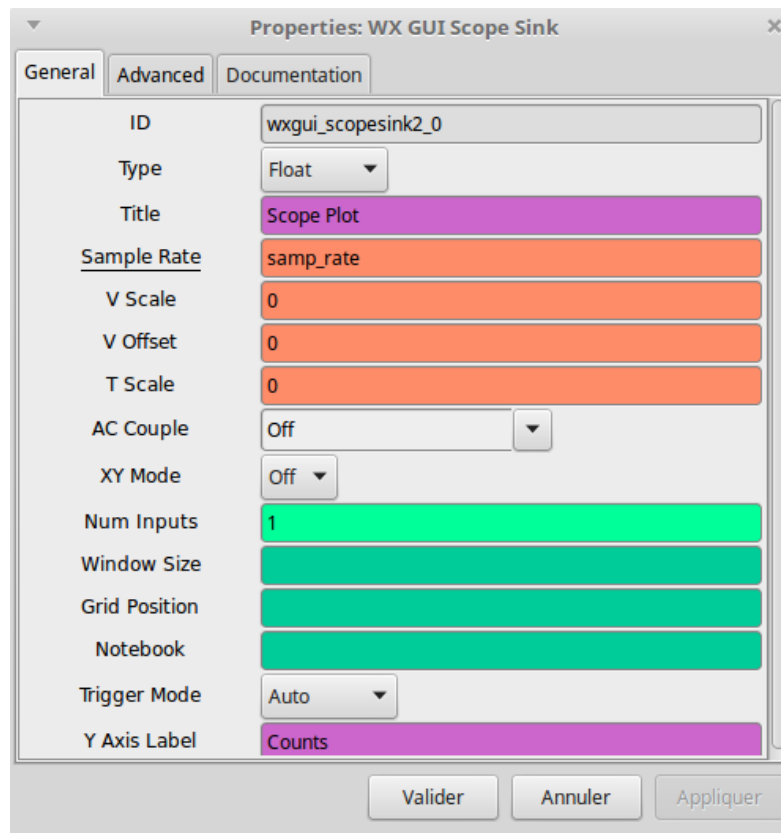
- 1) Etude de la transmission d'un signal rectangulaire.....1
- 2) Réalisation d'un synthétiseur universel de signaux.....9

### 1) Etude de la transmission d'un signal rectangulaire

1) Dans le bloc options je définie le generate option sur WX GUI. Je définie la fréquence d'échantillonnage globale à 100kHz en cliquant sur le bloc variable. Ensuite dans le plan de travail j'insère le bloc signal source. Je clique sur le bloc signal source et je défini l'output type sur float pour dire que c'est un réel. Je défini aussi sa waveform sur square pour dire sue c'est un signal rectangulaire.

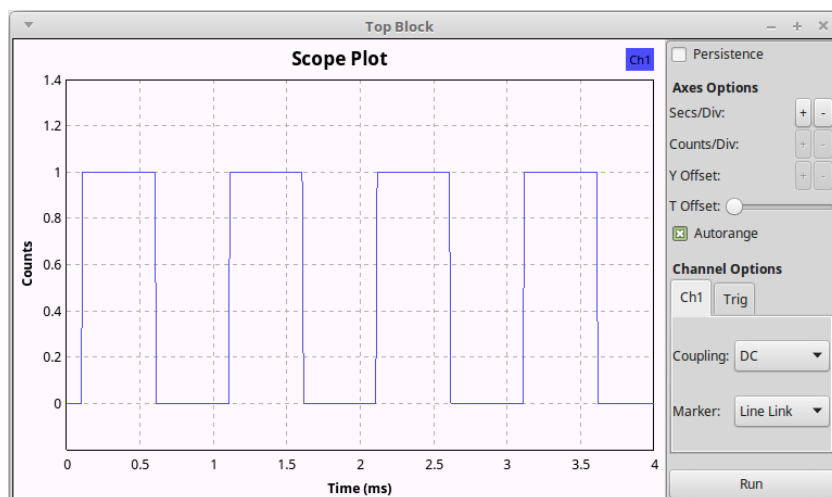


2) Pour pouvoir observer ce signal je place un bloc WX GUI Scope Sink qui va servir d'oscilloscope numérique. Je clique dessus et je défini aussi l'output type sur float.

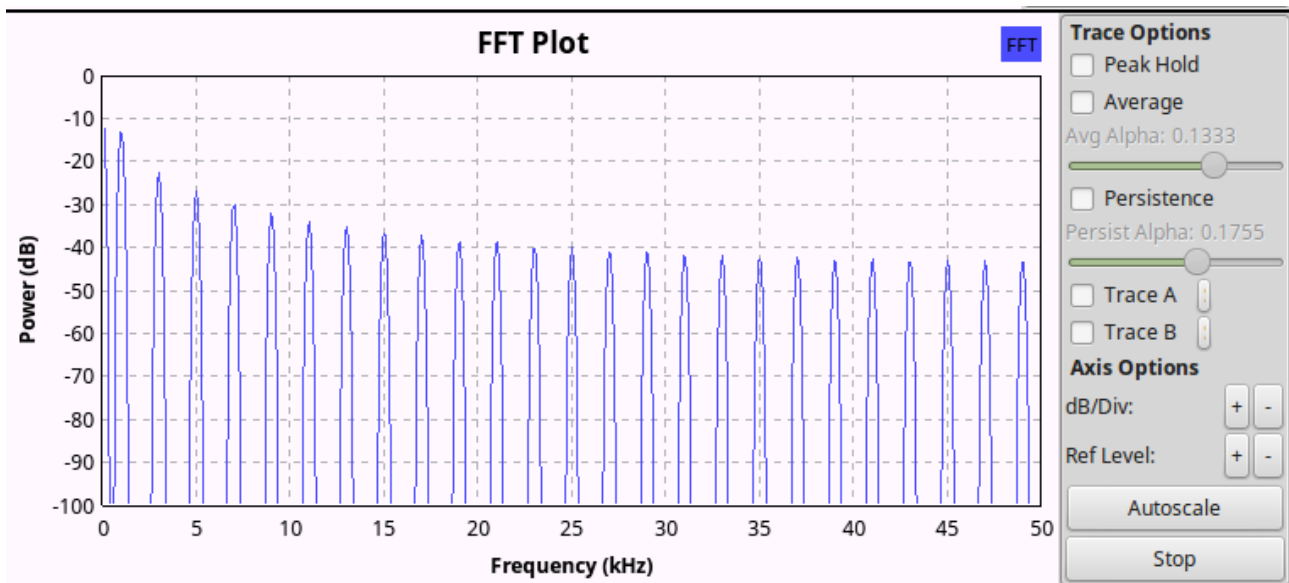


3) Entre les deux blocs j'insère le bloc Throttle afin d'éviter que le processeur et la RAM ne soit saturés à cause du logiciel. Le bloc Throttle placé entre les deux blocs permet d'éviter que le logiciel utilise toutes les performances données par le processeur et donc éviter de planter le pc.

Je sauvegarde et je lance la compilation pour observer le signal. Je règle l'oscilloscope de façon à bien visualiser le signal. Pour se faire je règle les Secs/Div et je clique sur stop.



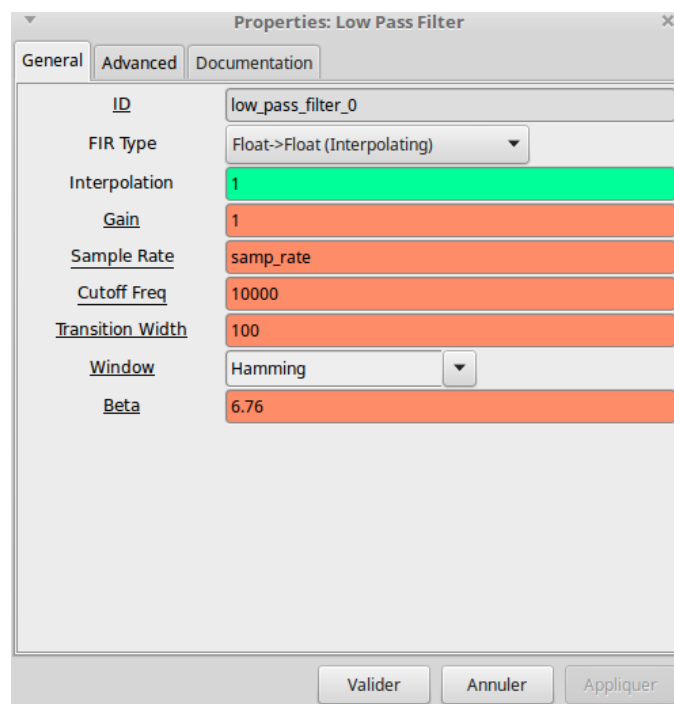
4) J'insère le bloc WX GUI fft sink à la sortie du bloc Throttle que je configure aussi en float. Je relance la simulation et je peux voir que le spectre d'amplitude du signal est apparu.



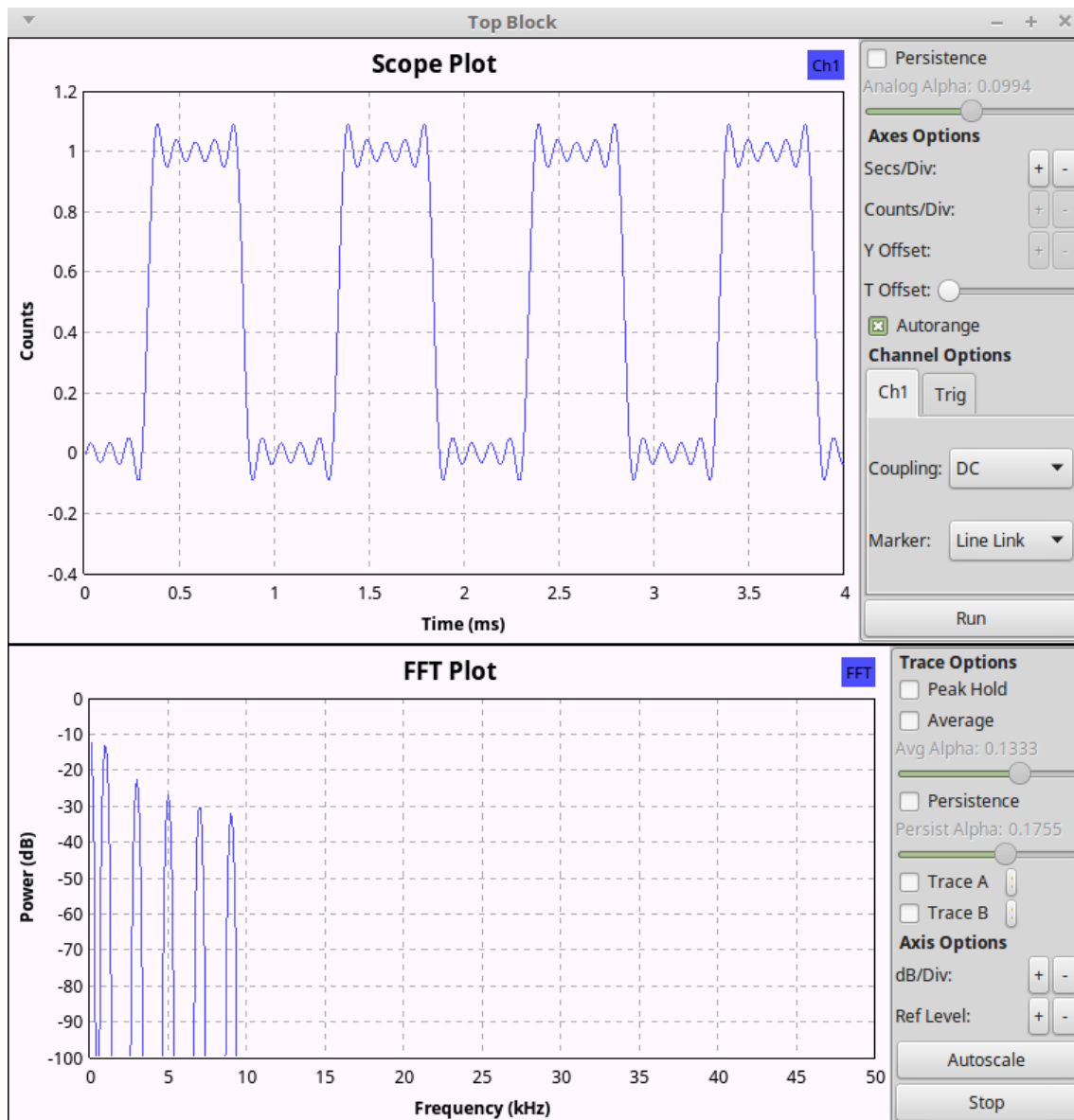
Je peux voir qu'à une fréquence de 1kHz on a un gain de -13dB. Plus la fréquence augmente plus le gain diminue.

5) L'écart fréquentiel entre deux raies du signal est de 2kHz. Ce résultat est conforme à la décomposition en série de Fourier car on a un signal pair et quand on a un signal pair les harmoniques sont égales à 0 une fois sur deux. Toutes les harmoniques pair sont nulles

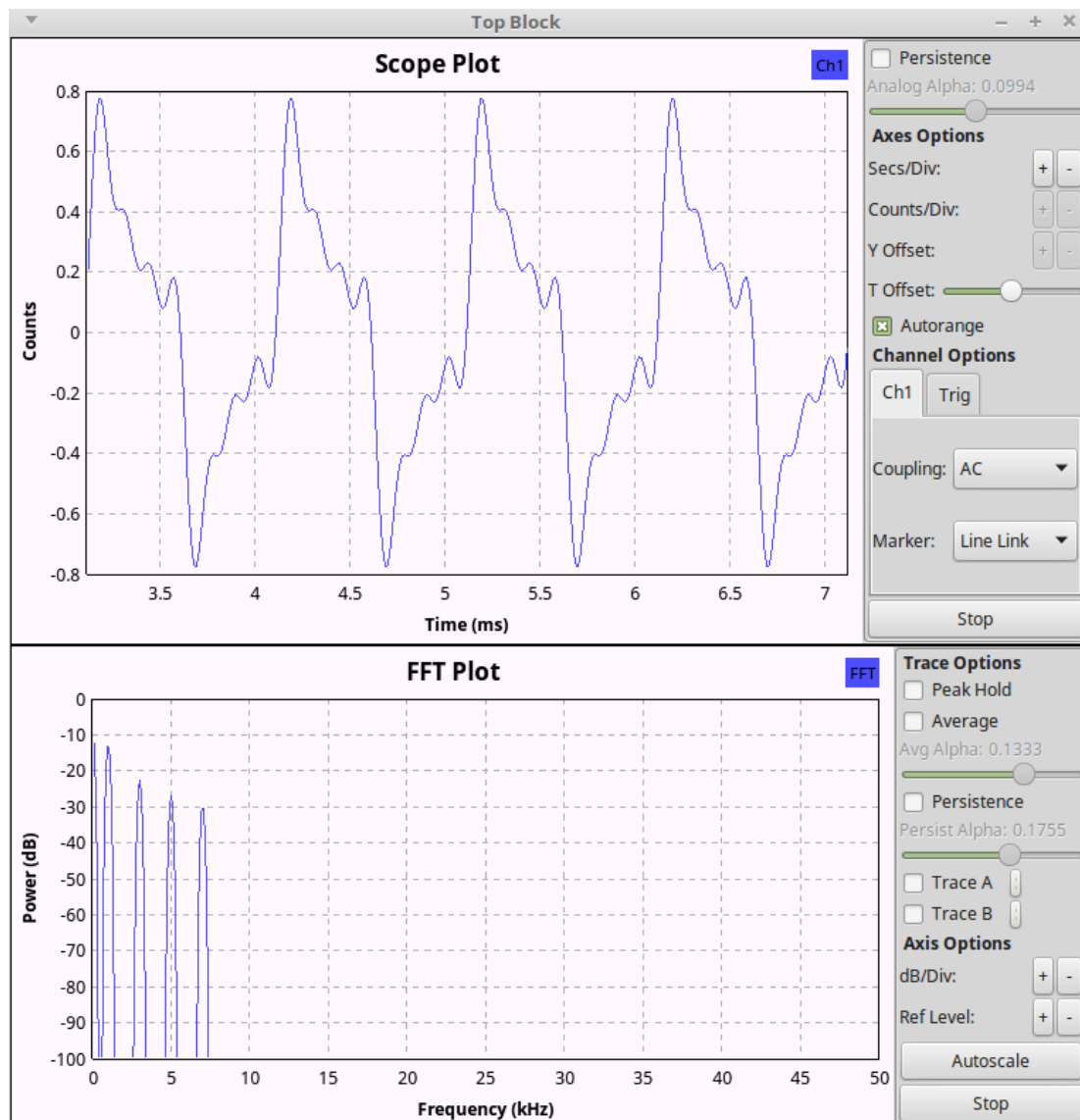
6) Je place un bloc low pass filter entre le bloc signal source et le bloc throttle. Je le configure de la façon suivante :



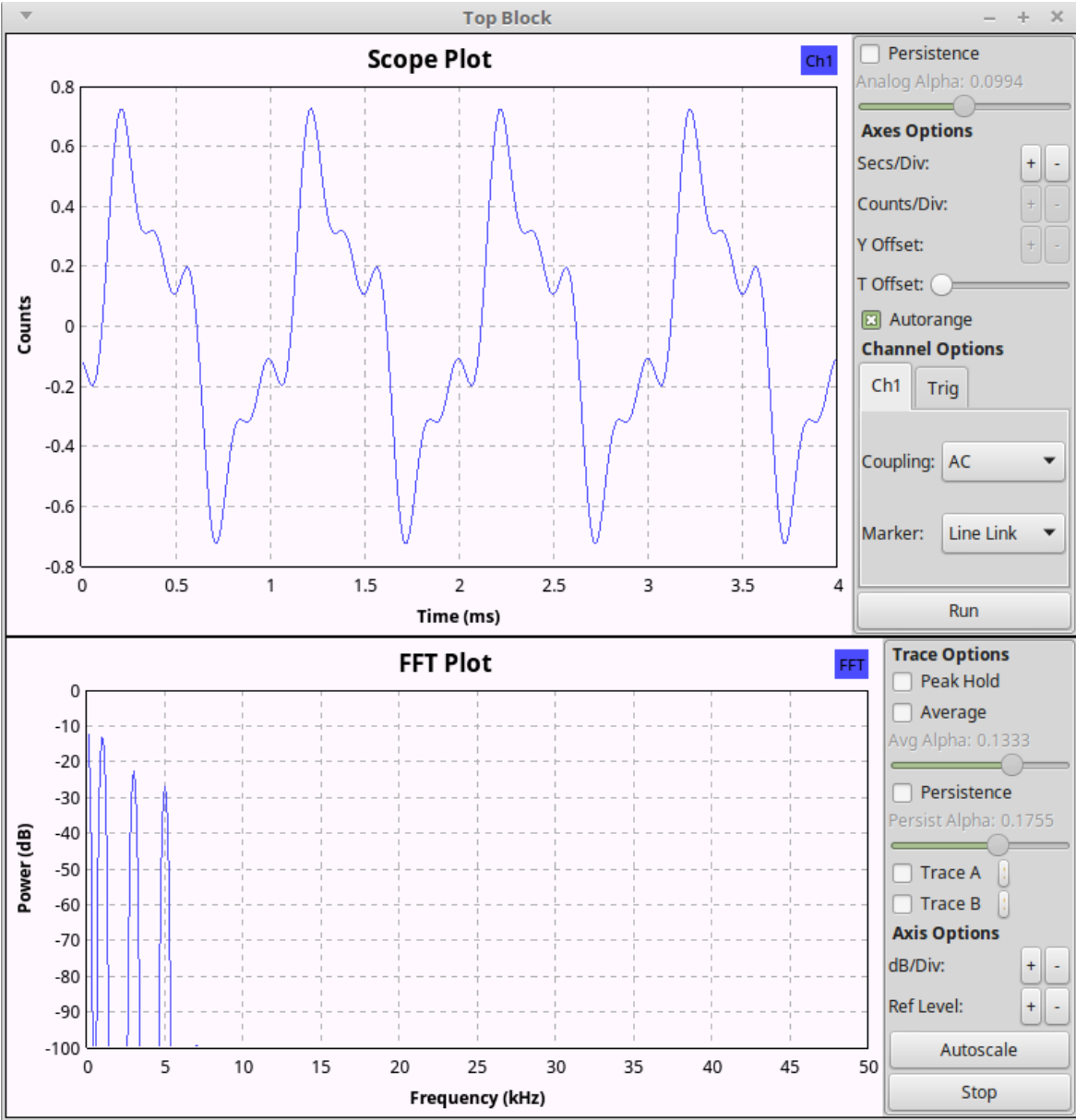
Une fois configuré je relance la simulation et je peux voir que toutes les fréquences supérieures à 10kHz sont atténuées.



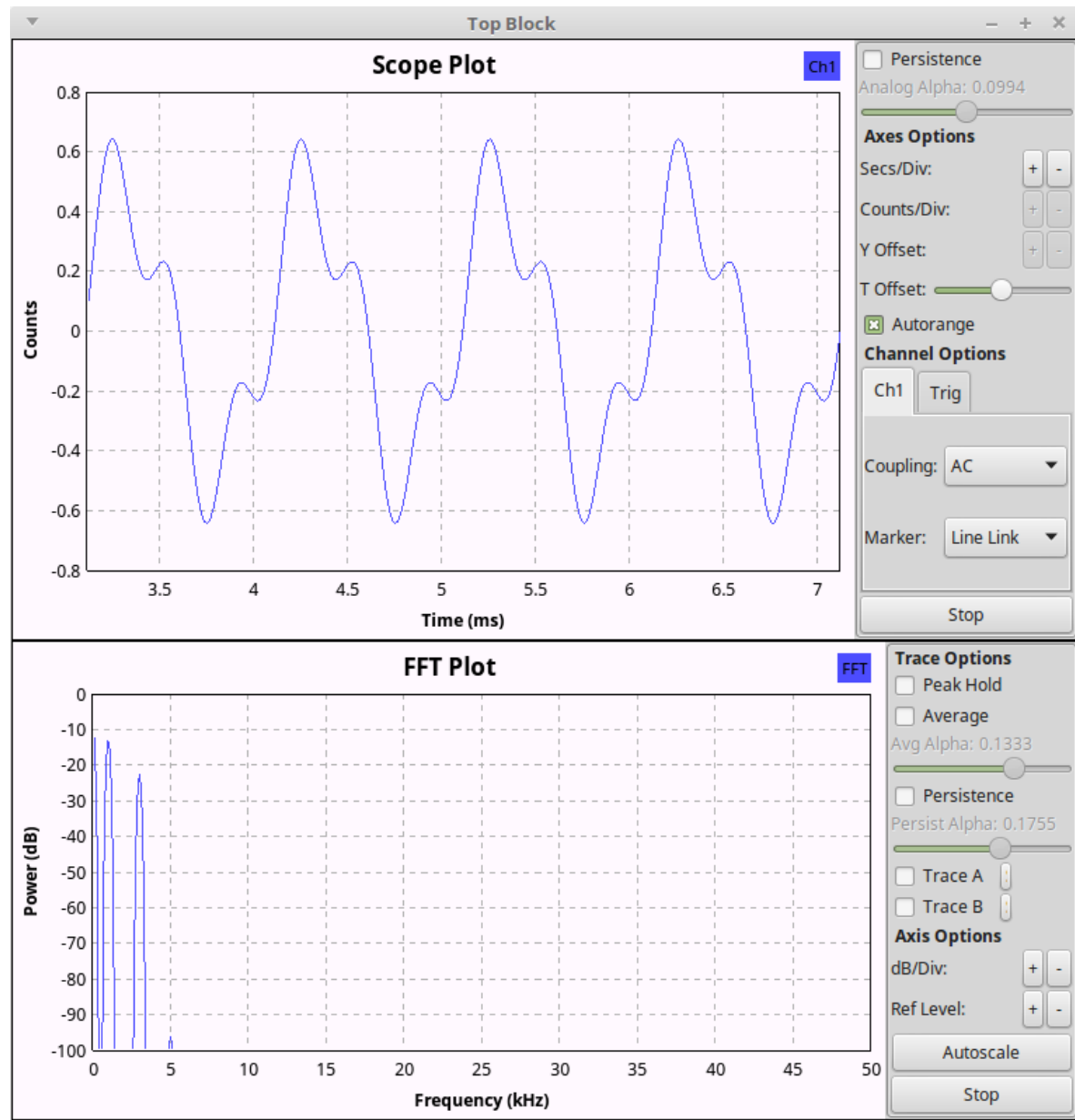
7) Je refais la manipulation précédente pour une fréquence de coupure de 8kHz :



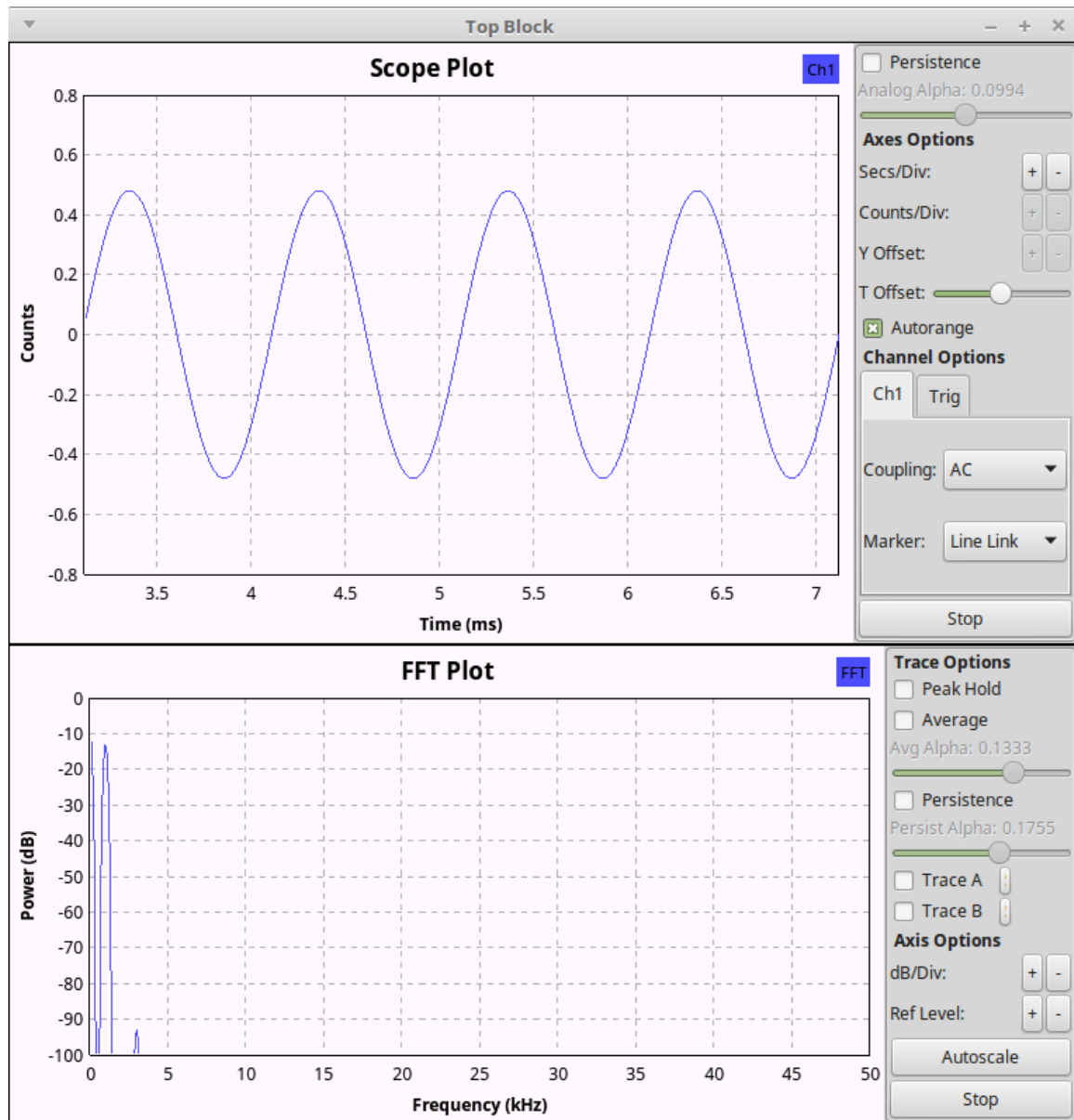
Pour une fréquence de coupure de 6kHz :



Pour une fréquence de coupure de 4kHz :



Pour une fréquence de coupure de 2kHz on retrouve un signal sinusoïdal :



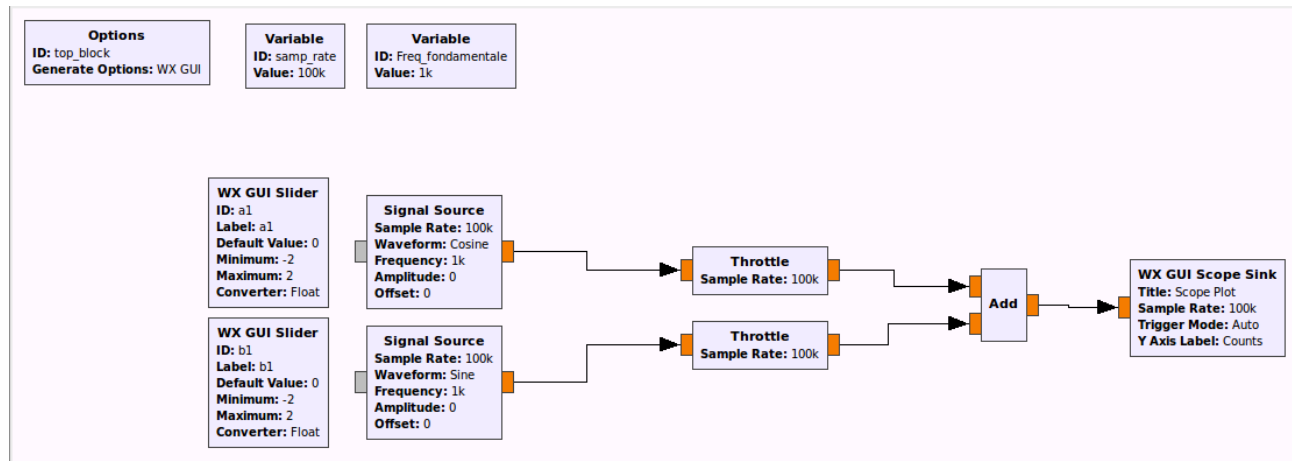
8) Plus la fréquence de coupure est basse plus le signal sera atténué et il s'approchera de la forme d'un signal sinusoïdal. Pour avoir un signal le plus ressemblant possible au signal transmis il faut laisser des fréquences assez hautes passer. Il faut donc laisser une fréquence de coupure assez haute.



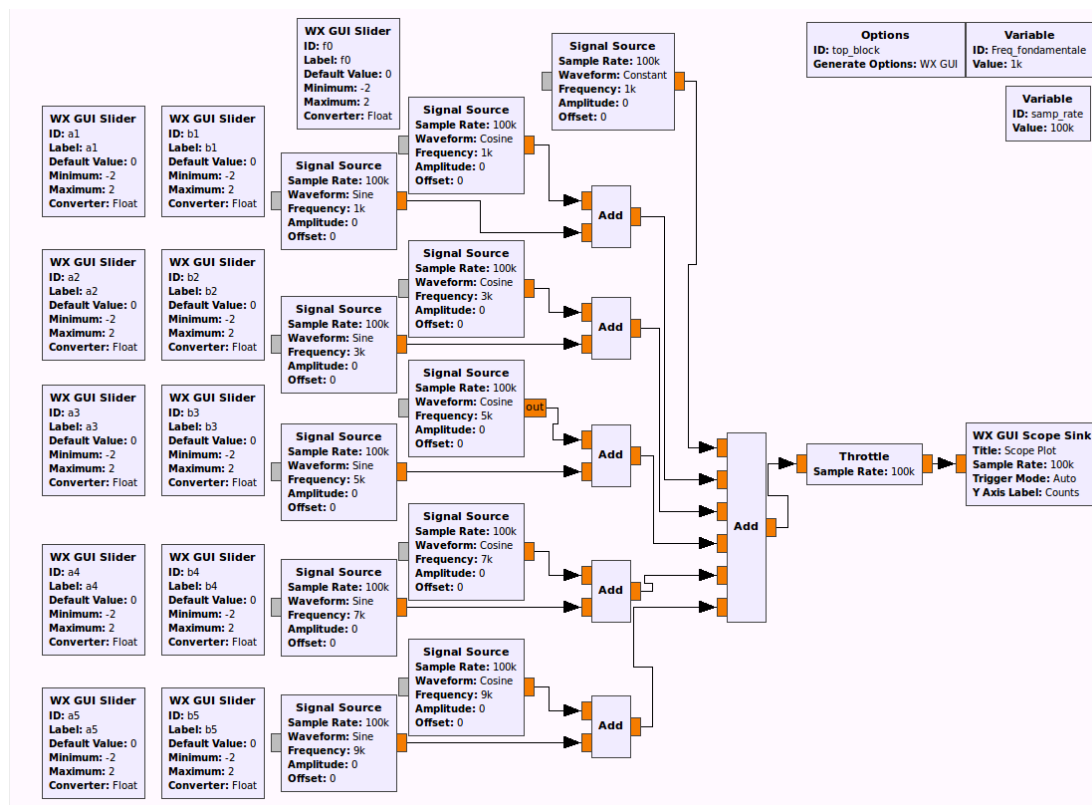
## 2) Réalisation d'un synthétiseur universel de signaux

1) J'ouvre le fichier Partie21.grc. Je vois que ce diagramme additionne deux signaux. L'un est un cosinus et l'autre un sinus.

2) Le bloc signal source permet de générer un signal. Le bloc throttle permet d'éviter au pc de buguer. Le bloc add permet d'additionner les deux signaux. Le bloc WX GUI Sink permet de visualiser l'addition des deux signaux. Le bloc WX GUI Slider permet d'identifier un bloc.



3) Je complète le fichier de façon à obtenir une composante continue et 5 harmoniques.



4) J'utilise le synthétiseur pour obtenir un signal rectangulaire. En définissant les fréquences j'ai sauté les harmoniques 2, 4 et 6 ce qui fait que je n'ai pas mis 0 à b2 et b4.

