

Table des matières

1 Introduction.....	1
2 Principe.....	2
3 Diagramme de classes Dds Fsk Psk.....	4
4 Applications.....	5
4.A) Codage FSK.....	5
4.B) Test avec Audacity de la classe Dds et Fsk.....	6
4.C) Introduction au codage PSK :.....	9
4.D) Le codage différentiel (classe Psk):.....	9
4.E) Le codage convolutif (classe Psk).....	9
4.F) Test de l'inversion de phase avec Audacity.....	11

1 Introduction

Notre objectif ici est de fournir au lecteur une compréhension du fonctionnement et de quelques utilisations des classes Dds Fsk et Psk dans les applications de communication radiotélégraphique.

Diverses approches peuvent être utilisées, pour générer un signal périodique, classiquement, lors de la conception d'un nouveau synthétiseur de fréquence, deux approches de base sont communes : les boucles à verrouillage de phase (PLL) et la synthèse numérique directe.

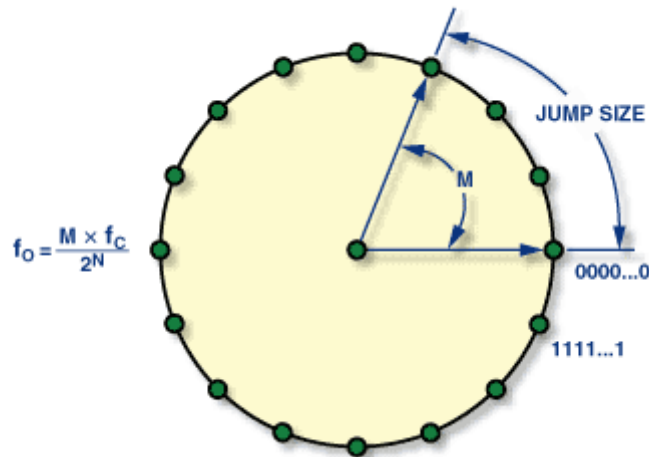
Toutefois la plus flexible est le synthétiseur numérique direct (DDS).

La méthode Dds, (direct digital synthesizer), produit une forme d'onde analogique, en générant un signal variant dans le temps sous forme numérique, puis en effectuant une conversion numérique-analogique. Elle permet une commutation rapide entre les fréquences de sortie et une résolution de fréquence fine. De plus, elle permet aussi de modifier la phase et l'amplitude.

La classe Dds, avec son mot de réglage de 32 bits (attribut **incrementPhase**), permet un réglage très précis de la fréquence de référence, ce qui se traduit par un réglage fin de la fréquence de sortie beaucoup plus pratique que par l'utilisation d'une PLL.

2 Principe

La synthèse DDS produit un signal périodique à une fréquence donnée. La fréquence dépend de deux variables, la fréquence d'échantillonnage (**samplingFrequency**) et la valeur programmée pour le saut de phase (**incrementPhase**).



La forme de l'onde dépend de la table de consultation. Ici nous avons une table de consultation sinusoïdale avec 512 points ($N = 9$). Elle permet de faire la relation phase/amplitude. Le signal sinusoïdal est généré autour d'une composante continue (valeur 127).

incrementPhase fournit la valeur à ajouter à l'accumulateur de phase pour passer à l'échantillon suivant.

accumulateur. L'accumulateur contient la phase (angle) pour l'échantillon courant. La table de consultation permet d'associer la valeur de l'amplitude correspondant au sinus de l'angle de phase contenu dans l'accumulateur. Cette valeur est transférée dans le registre d'entrée du DAC après éventuellement une atténuation de l'amplitude ou un déphasage.

Le DAC, à son tour, convertit ce nombre en une valeur correspondante de tension analogique.

L'ensemble de ce calcul est effectué dans la fonction nommée interruption.

```
accumulateur += incrementPhase; // accumulateur de phase
phase = ((accumulateur >> 23 + dephase) & 0x1ff); //ajoute le déphasage
sinus = pgm_read_byte(&(sinusTable[phase])); //lecture de l'amplitude dans la
table de consultation sinus
dac_output_voltage(dacChannel, sinus); //envoi de la valeur vers le dac
```

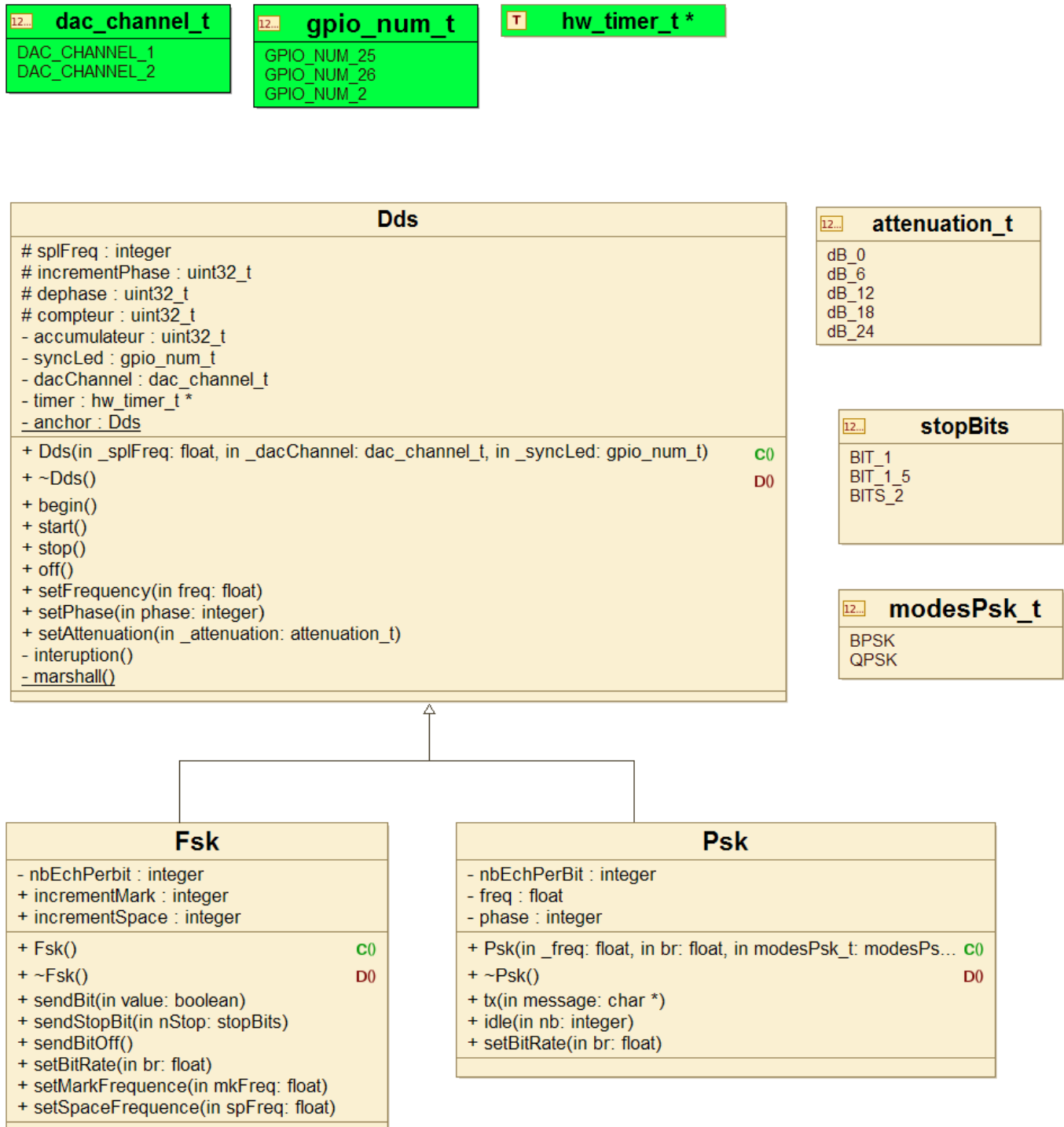
Pour générer une onde sinusoïdale à fréquence fixe, une valeur constante est ajoutée à l'accumulateur de phase à chaque cycle d'interruption.

L'incrément de phase est déterminé par la fonction

computeIncrementPhase(float freq).

L'accumulateur de phase est en fait un compteur modulo-M qui incrémente son nombre stocké à chaque appel de la routine d'interruption. Plus la taille du saut est grande, plus l'accumulateur de phase déborde rapidement et termine son équivalent d'un cycle sinusoïdal.

3 Diagramme de classes Dds Fsk Psk

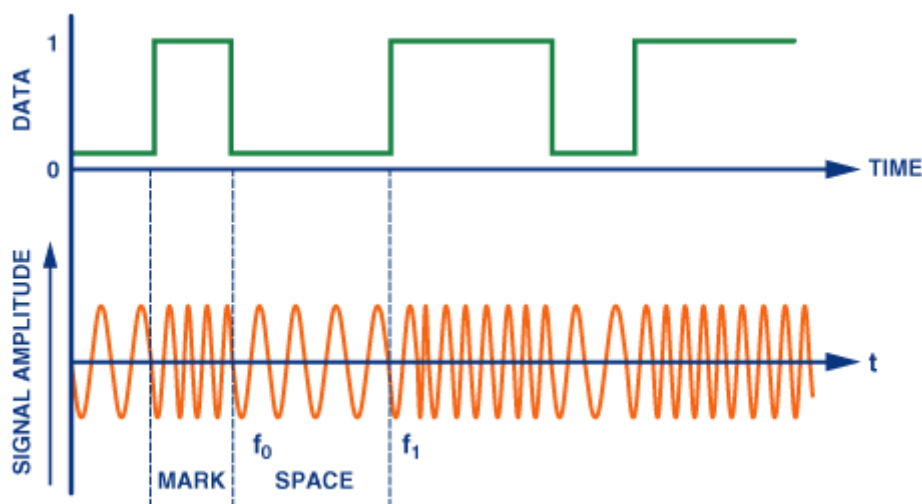


4 Applications

Étant donné que la classe Dds facilite le réglage de la fréquence et de la phase, elle est particulièrement adaptée pour coder des données par modulation de phase et de fréquence sur une porteuse. Voici deux applications qui remontent aux premiers jours de la radiotélégraphie.

4.A) Codage FSK

La modulation par saut de fréquence binaire (FSK) est l'une des formes les plus simples de codage de données. Les données sont transmises en décalant la fréquence d'une porteuse continue vers l'une ou l'autre de deux fréquences discrètes (opération binaire). Une fréquence (f_1) est désignée comme la fréquence «**mark**» (un binaire) et l'autre (f_0) comme la fréquence «**space**» (zéro binaire). La figure ci dessous montre la relation entre les données et le signal transmis.



Ce schéma de codage est facilement implémenté à l'aide de la classe Fsk. Le mot d'accord de fréquence (attribut **incrementPhase**) représentant les fréquences de sortie change afin de générer f_0 et f_1 en synchronisme avec le motif de 1 et de 0 à transmettre.

Les méthodes `Fsk::setMarkFrequency(float mkFreq)` et `Fs::setSpaceFrequency(float spFreq)` doivent être utilisées pour configurer la fréquence à générer.

Une utilisation possible de la classe Fsk est la modulation de type *audio FSK* (RTTY45) pour transmettre des caractères alphanumériques à 45 bits par seconde

La fréquence *mark* est réglée à 1000 Hz, et la fréquence *space* à 1170 Hz.
En dépit de son faible débit de transmission, cette modulation reste le standard pour les transmissions de données entre radioamateurs.
En VHF la fréquence est 144,800 Mhz.

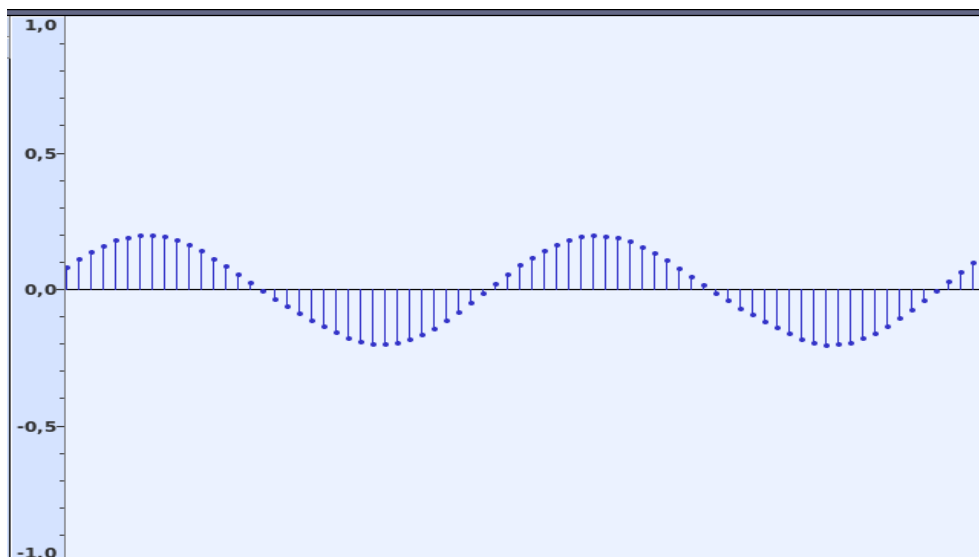
4.B) Test avec Audacity de la classe Dds et Fsk

Raccorder le connecteur J14 de la carte à un jack micro, suivant plan de raccordement fourni.

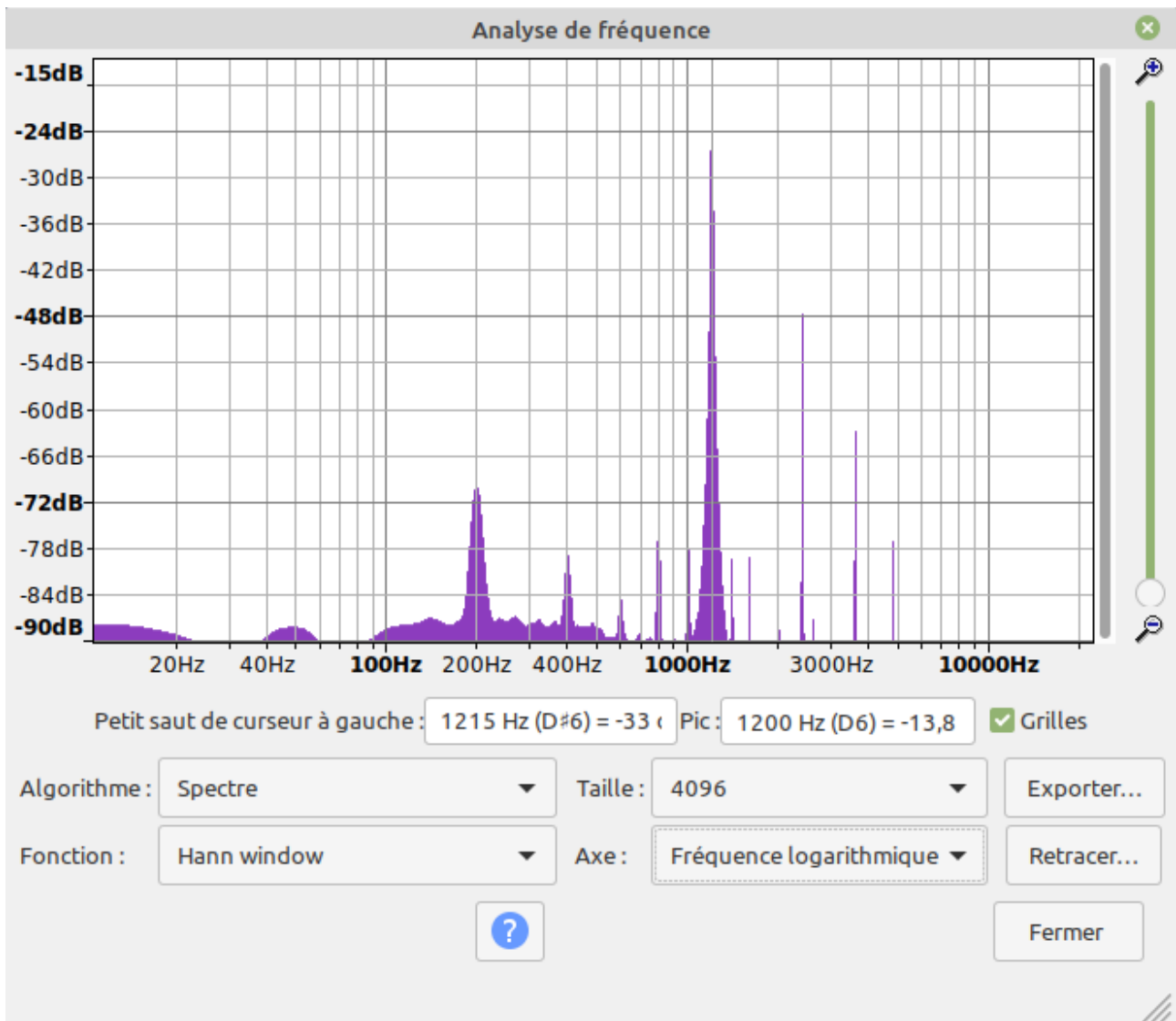


Compiler puis télécharger le programme testDds.

Comme le montre la copie d'écran suivante Nous obtenons un signal sinusoïdal.

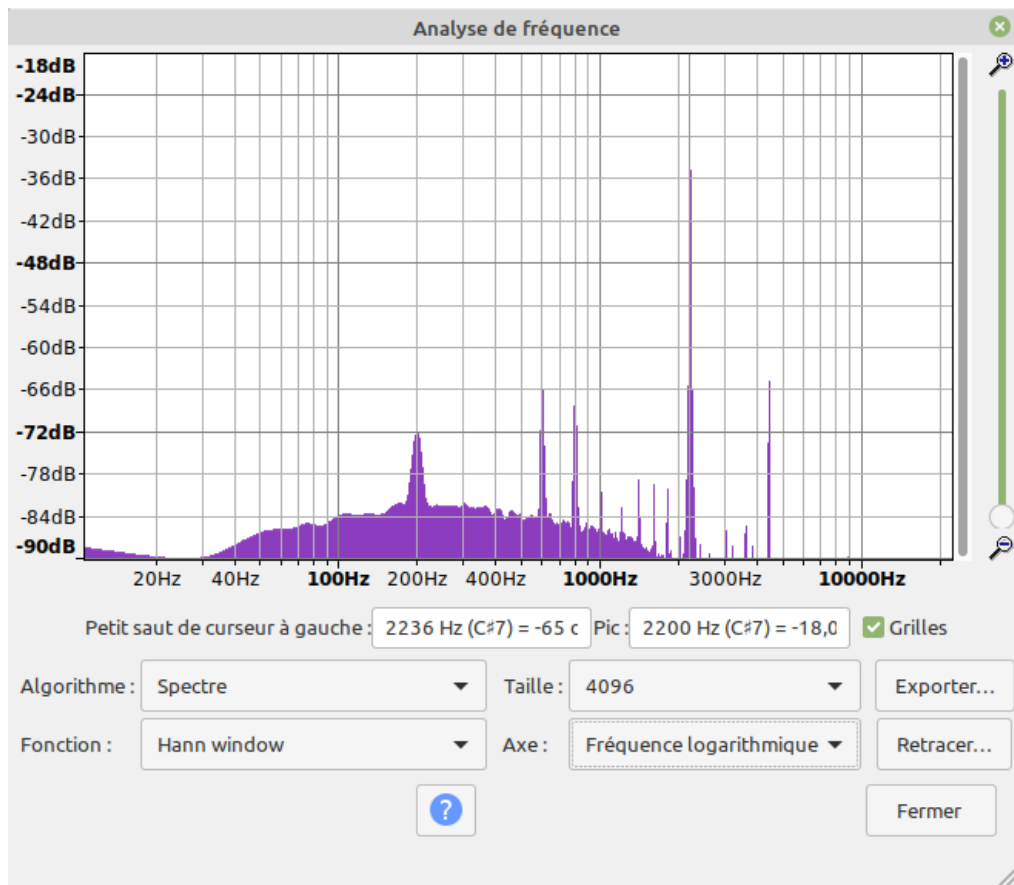


Nous obtenons 1200 Hz amplitude -13,8 dB (potentiomètre au max)

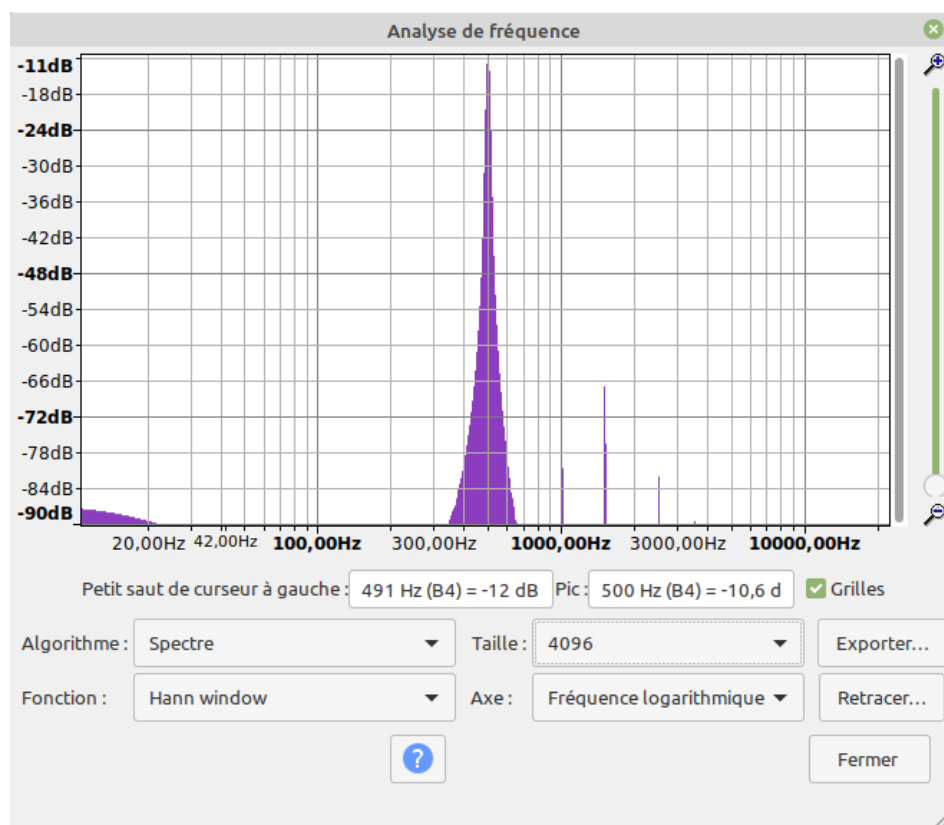


Pour la fréquence Space (2200 Hz , -18,0 dB)

L'atténuation de l'amplitude est dû à la présence d'un filtre.



Pour la fréquence 500 Hz
Nous obtenons 500Hz -10,6 dB



4.C) Introduction au codage PSK :

La modulation par déplacement de phase (PSK) est une autre forme simple de codage de données. En PSK, alors que la fréquence de la porteuse reste constante, la phase du signal transmis est modifiée pour véhiculer l'information. Il existe différents schémas qui peuvent être utilisés pour accomplir la PSK.

La méthode la plus simple, utilise seulement deux phases de signal 0° et 180° . Elle est communément appelée PSK binaire (**BPSK**).

L'encodage PSK est facilement implémenté avec la classe Dds. La classe possède un attribut qui peut être chargé avec une valeur de phase **Dds::setPhase(0)** ou **Dds::setPhase(180)**. Cette valeur est directement ajoutée à la phase de la porteuse sans changer sa fréquence. La modification du contenu de cet attribut module la phase de la porteuse (générant ainsi un signal de sortie PSK).

4.D) Le codage différentiel (classe Psk):

l'idée du codage différentiel est de faire en sorte que les bits d'information codent finalement **non pas l'état** des symboles mais leur **changement d'état**.

L'état de chaque symbole envoyé est déterminé en fonction de l'état du symbole précédent.

- le bit = 0 crée un saut de phase de π sur le symbole $a[m]$ par rapport au symbole précédent $a[m-1]$,
- le bit = 1 ne crée aucun saut de phase
-

En repérant à la réception, la présence ou non d'un saut de phase, on pourra décider du bit transmis. Si la phase de l'onde ne change pas, le bit d'information est un état logique 1. Si la phase de l'onde s'inverse, c'est-à-dire change de 180° , le bit d'information est un état logique 0.

L'envoi d'une séquence connue du récepteur en début de transmission (ou périodiquement répété en début d'une trame) permet la synchronisation des états logiques.

La méthode **Psk::idle(nb)** permet d'envoyer une suite de nb bits 0.

4.E) Le codage convolutif (classe Psk)

Afin de minimiser le risque d'erreur de transmission du aux bruits, on peut passer au QPSK avec le codage convolutif.

Dans ce mode, quatre angles de phase sont utilisés. QPSK (Quadrature Phase Shift Keying). Les angles de phase possibles sont 0° , $+90^\circ$, -90° et 180° .

les données brutes sont transformées du binaire (1 sur 2) au quaternaire (1 sur 4) de telle manière qu'il existe un motif connu avec précision dans la séquence de symboles quaternaires.

Dans le code utilisé dans PSK31, chaque symbole quaternaire transmis est dérivé d'une série de 5 bits de données consécutifs. Cela signifie que chaque bit binaire à transmettre génère une séquence de 5 symboles, chevauchant les séquences de bits adjacents, d'une manière prévisible que le récepteur peut utiliser pour estimer la séquence correcte même en présence de corruptions dans des parties de la séquence.

Le code convolutif

le tableau suivant contient les 32 combinaisons d'une série de cinq bits, transmis, avec les sauts de phase correspondants à appliquer à la porteuse, en degrés.

00000	180	01000	0	10000	+90	11000	-90
00001	+90	01001	-90	10001	180	11001	0
00010	-90	01010	+90	10010	0	11010	180
00011	0	01011	180	10011	-90	11011	+90
00100	-90	01100	+90	10100	0	11100	180
00101	0	01101	180	10101	-90	11101	+90
00110	180	01110	0	10110	+90	11110	-90
00111	+90	01111	-90	10111	180	11111	0

Prenons un exemple pour illustrer l'encodage convolutif

soit à transmettre un seul caractère l'espace (codé sur un seul bit 1 en varicode)
le caractère est encadré d'un préambule 5 fois 0 et d'un postambule 5 fois 0

0 0 0 0 0 1 0 0 0 0 0		→ Phase 0°
0 0 0 0 0	saut A 180	→ Phase 180°
0 0 0 0 1	saut D +90	→ Phase 270°
0 0 0 1 0	saut C -90	→ phase 180°
0 0 1 0 0	saut C -90	→ phase 90°
0 1 0 0 0	saut B 0	→ phase 90°
1 0 0 0 0	saut D +90	→ Phase 180°
0 0 0 0 0	saut A 180	→ Phase 0°

Notez qu'une séquence continue de zéros (la séquence inactive) donne des inversions de phase continues, les mêmes que BPSK.

4.F) Test de l'inversion de phase avec Audacity

La classe Psk hérite de Dds. Elle permet d'inverser la phase avec atténuation de l'amplitude.

Test avec audacity du signal obtenu lors de l'inversion de phase

