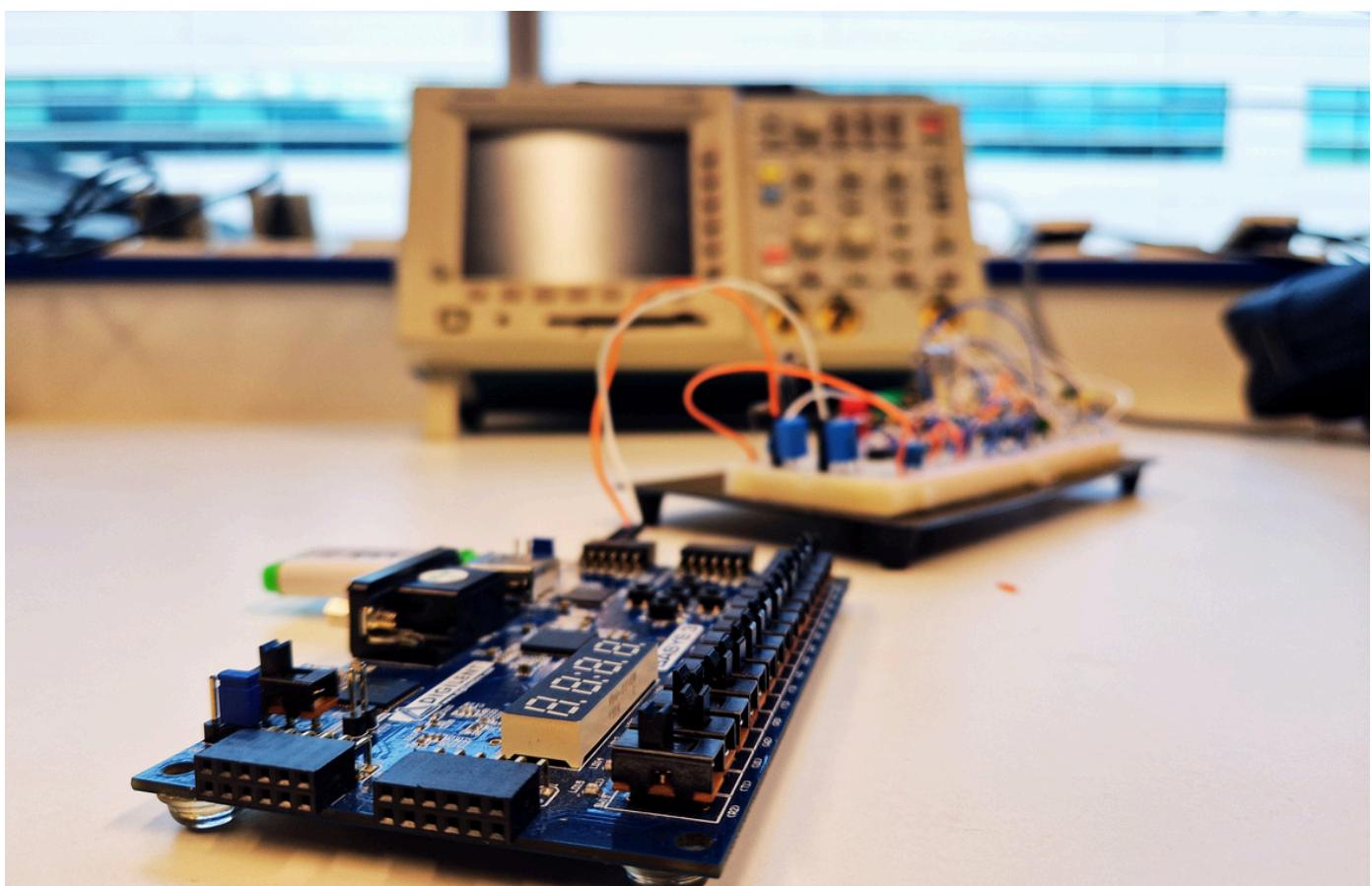


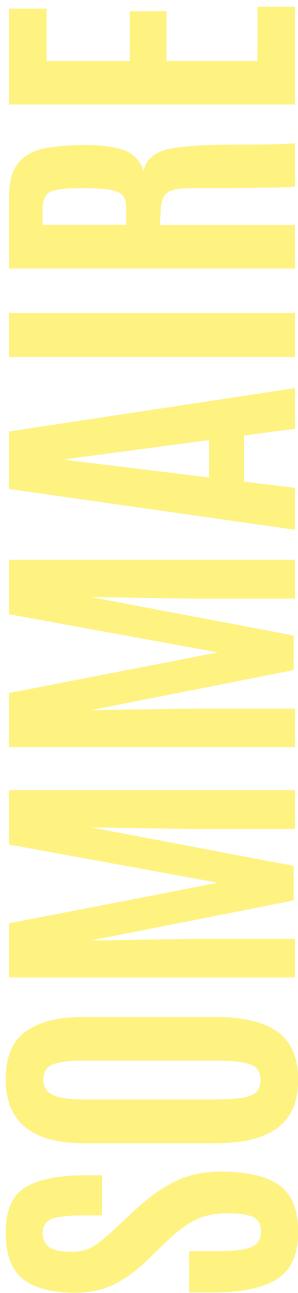


CHAUVELIERE MARTIN
NEHARI TAHA

EMETTEUR QPSK HETERODINE

RAPPORT TPE





01. Objectif	p.3
02. VHDL- Trame IQ	p.4
04. Astable	p.9
05. Filtre de Butterworth	p.13
06. DéphasEUR	p.20
07. Multiplieurs	p.23
09. Sommateurs	p.25
10. Caractérisation des éléments	p.27
11. Conclusion	p.32

OBJECTIF

Le présent projet vise à comprendre, concevoir et mettre en œuvre un émetteur QPSK (Quadrature Phase Shift Keying) hétérodyne. Ce type d'émetteur joue un rôle crucial dans divers standards de télécommunication tels que le Wi-Fi, le Zigbee, les communications par satellite, ainsi que les applications RFID.

L'objectif principal est de développer une compréhension approfondie des principes fondamentaux de la modulation QPSK et des techniques de conversion hétérodyne. La modulation QPSK, avec ses avantages en matière de spectre efficace et de robustesse contre les interférences, est largement utilisée dans les systèmes de communication modernes. La conception d'un émetteur QPSK nécessite une certaine compréhension des aspects théoriques et pratiques de la modulation de phase. La sélection des composants et la gestion des interférences est un autre aspect crucial de la conception, permettant d'obtenir une performance optimale en termes de qualité de signal et d'efficacité spectrale.

Ce projet vise également à développer des compétences pratiques dans l'utilisation d'outils de simulation et de conception de circuits électroniques.

Pour réaliser ce projet, nous allons d'abord créer une trame binaire pseudo-aléatoire que nous diviserons en deux trames, I et Q. Afin de transmettre l'information nous utiliserons un signal sinusoïdal de 400kHz. La modulation QPSK respecte la formule suivante :

$$s(t) = I(t)\cos(\omega t) + Q(t)\sin(\omega t)$$

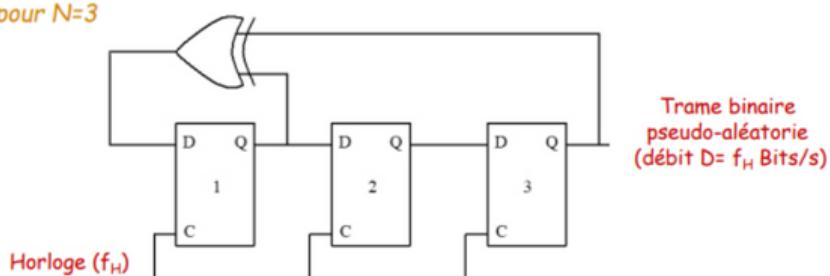
Après avoir modulé le signal, il reste à le transmettre à une fréquence élevée pour assurer la transmission de notre message.

VHDL

Objectif

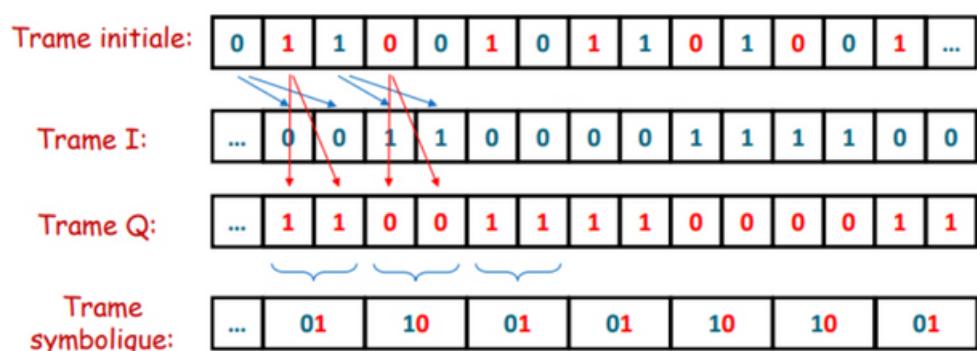
Tout d'abord il va nous falloir générer une trame binaire pseudo-aléatoire. Le principe est basé sur la théorie des séquences binaires de longueur maximum. Ces séquences sont générées par un registre à décalage comportant N bascules montées en série, complétées par un circuit de réaction réinjectant à l'entrée la somme modulo 2 des états de certaines bascules. Cette réaction permet au registre de passer par ses $L=2^{n-1}$ états possibles. L'ensemble est activé par une horloge à la fréquence $f_h = 1/T_h$

Exemple pour N=3



Nous générerons une trame avec $N = 11$ bascules avec un débit binaire $f_h = 20\text{Kbits/s}$ à l'aide d'une carte BASYS 3.

Après avoir généré notre trame binaire pseudo-aléatoire, il nous faudra réaliser une modulation numérique de type QPSK. Il faudra séparer le flux initial de débit D en 2 flux distincts, la trame I et la trame Q, afin d'obtenir un flux symbolique de débit D/2.



Réalisation

Dans un premier temps, il va nous falloir générer la clock $f_h = 1/T_h$, c'est-à-dire une horloge $f_h = 20\text{Khz}$.

```

signal compteur : integer range 0 to 2500;
signal clk_20k : std_logic := '0';

clock : process(clk)
begin
  if clk'event and clk='1' then
    if compteur < 2499 then
      compteur <= compteur +1;
    else
      clk_20k <= not(clk_20k);
      compteur <= 0;
    end if;
  end if;
end process;
```

Pour cela nous utilisons l'horloge interne de la Basys 3 de fréquence 100MHz, et à l'aide d'un compteur, nous créons notre clock de 20kHz.

Comment cela fonctionne ?
 $10\text{M} / 5000 = 20\text{k}$, en prenant en compte l'état haut et l'état bas de l'horloge, tous les 2500 coups d'horloge, on applique un not à notre horloge de 20kHz.

On commence ensuite par définir l'entité, ses entrées et ses sorties.

```

entity Trame_bin is
  port
  (
    clk : in std_logic;
    sortieIQ : out std_logic_vector (1 downto 0);
    sortieTrameinit_visu : out std_logic;
    interrupteur : in std_logic;
    clk_20k_visu : out std_logic
  );
end Trame_bin;
```

Nous mettons un interrupteur en entrée, lorsque celui-ci sera à 1 toutes les balances sauf la première seront définies à 0 et lorsque l'interrupteur passera à 0, la trame binaire commencera à se générer.

Ensuite nous créerons les balances à l'aide de signaux.

```
signal B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11 : std_logic ;
```

```

process (clk_20k)
begin
  if clk_20k'event and clk_20k='1' then
    if interrupteur = '1' then
      B1 <= '1';
      B2 <= '0';
      B3 <= '0';
      B4 <= '0';
      B5 <= '0';
      B6 <= '0';
      B7 <= '0';
      B8 <= '0';
      B9 <= '0';
      B10 <= '0';
      B11 <= '0';
    else
      B2 <= B1;
      B3 <= B2;
      B4 <= B3;
      B5 <= B4;
      B6 <= B5;
      B7 <= B6;
      B8 <= B7;
      B9 <= B8;
      B10 <= B9;
      B11 <= B10;
      B1 <= B11 xor B9;
    end if;
  end if;
end process;

```

Notre trame initiale maintenant initialisée, passons à la génération de nos trames I et Q. Voici comment nous allons procéder :

Tout d'abord à l'initialisation, notre premier coup de clock sera sauvegardé dans un signal mémoire. Ensuite, comme 1 coup d'horloge de trame IQ équivaut à 2 coups d'horloge de trame binaire, nous implémentons un nouveau compteur qui, lorsqu'il vaut 0, génère notre trame IQ et lorsqu'il vaut 1, enregistre simplement la valeur de sortie de la trame binaire dans mémoire.

```

process (clk_20k)
begin
  if clk_20k'event and clk_20k='1' then
    if (init =0) then
      memoire <= B11;
      init <= init +1;
    end if;
    if (compteurdm = 0) then
      TrameI <= memoire;
      TrameQ <= B11;
      compteurdm <= compteurdm + 1;
    elsif (compteurdm = 1) then
      memoire <= B11;
      compteurdm <= 0;
    end if;
  end if;
end process;

```

```

sortieTrameinit_visu <= B11;
sortieIQ{0} <= TrameI;
sortieIQ{1} <= TrameQ;
clk_20k_visu <= clk_20k;

```

Pour finir, on définit nos sorties. Celles-ci seront implémentées dans notre fichier de contrainte de la Basys3.

Fichier de contrainte :

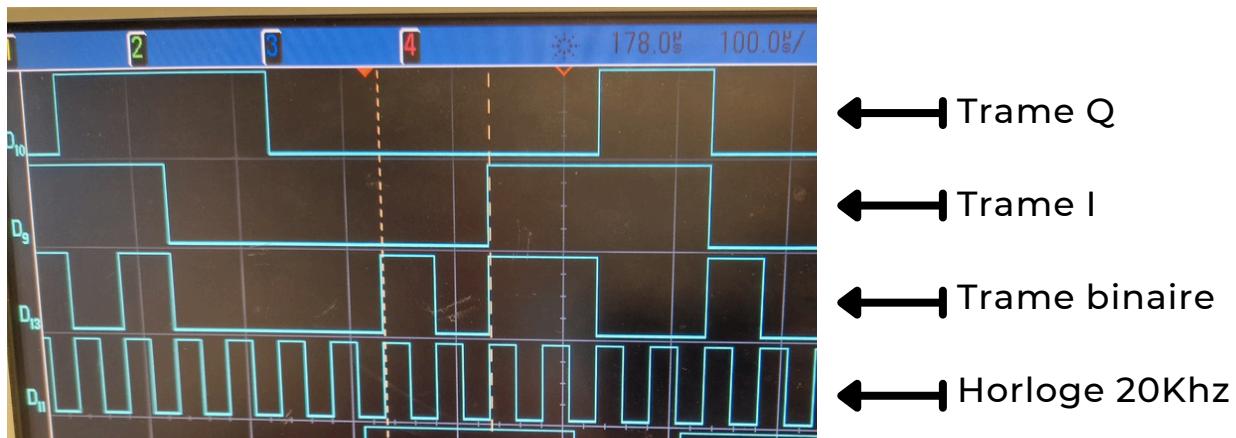
```

## Clock signal
set_property -dict { PACKAGE_PIN W5    IOSTANDARD LVCMS33 } [get_ports clk]
## Switches
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMS33 } [get_ports {interrupteur}]
##Pmod Header JB
set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMS33 } [get_ports {sortieTrameinit_visu}];
set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMS33 } [get_ports {sortieIQ[0]}];
set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMS33 } [get_ports {sortieIQ[1]}];
set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMS33 } [get_ports {clk_20k_visu}]

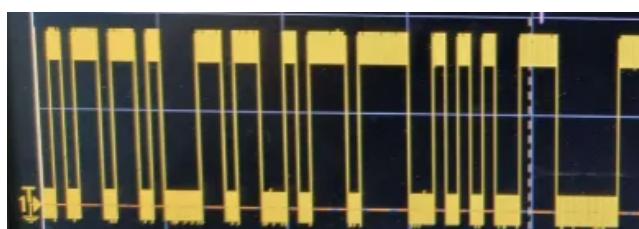
```

VHDL

Analyse



Notre trame IQ semble être fonctionnelle. Il y a seulement un petit changement par rapport à la consigne initiale, le décalage entre la trame binaire et la trame IQ n'est pas d'un coup d'horloge mais de deux. Cela n'aura pas d'incidence sur la suite du projet.



Les valeurs de tension associées à la trame binaire pseudo-aléatoire sont de 0V pour 0 et 3.72V pour 1.

Désormais, nous voulons que la trame binaire pseudo-aléatoire ait des niveaux de tensions symétriques aux états logiques 0 et 1 pour notre modulateur (-A pour 0 et +A pour 1).

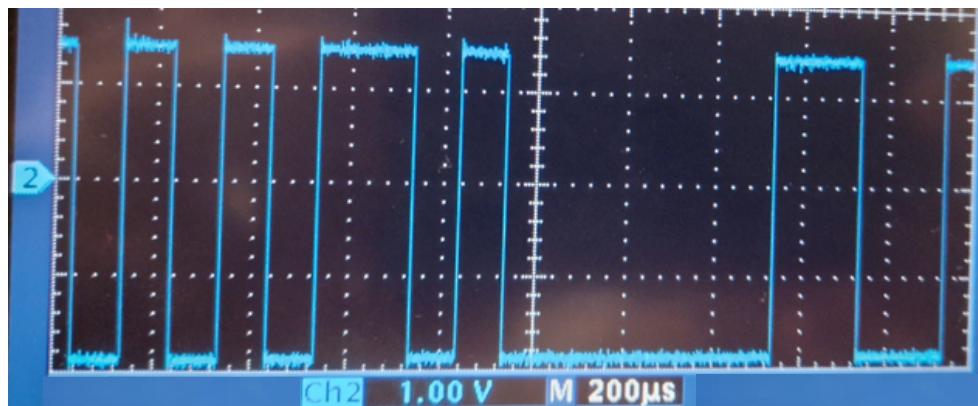
Pour cela, nous ajoutons un montage filtre Passe-Haut sur notre BreadBoard afin de supprimer cet offset.



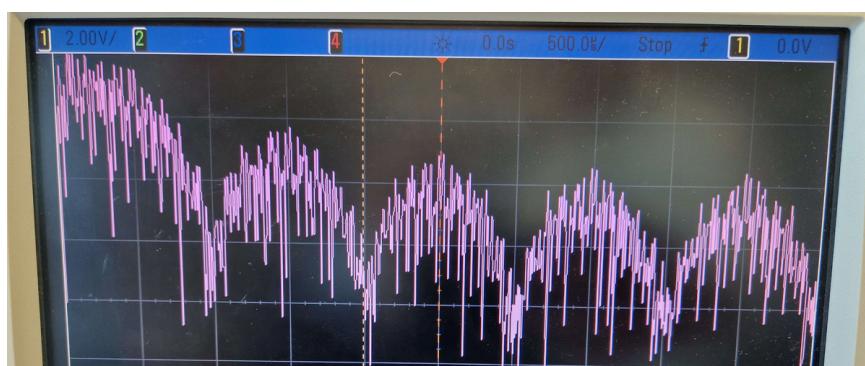
Ce montage est assez simple à mettre en place, il utilise une capacité de 470 nF et une résistance de 560 kΩ.

À noter que ce montage sera présent en double, une fois pour la trame I et une autre fois pour la trame Q.

Voici le résultat obtenu, la trame est beaucoup plus centrée en 0. Nous sommes à -1.8V pour l'état 0 et 1.6V pour l'état 1.



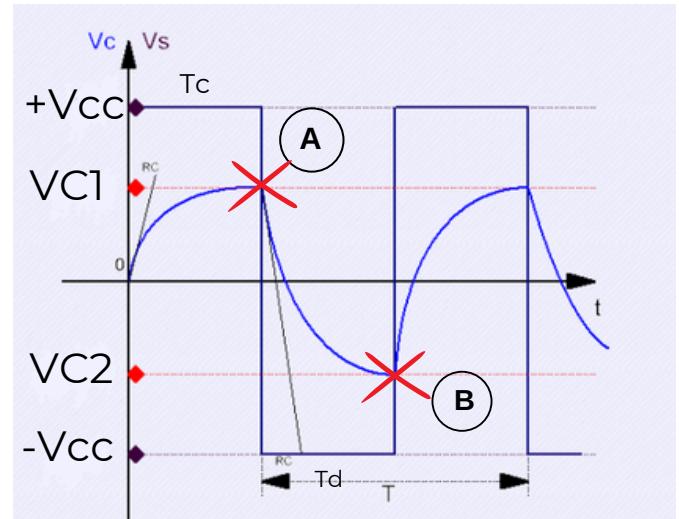
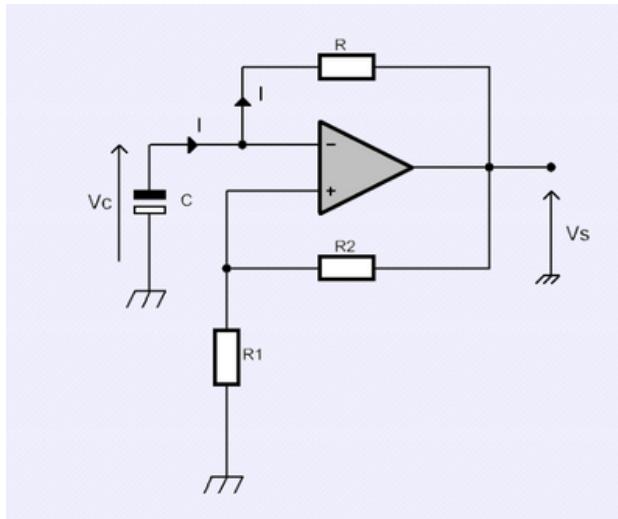
Enfin concernant la FFT, on peut observer la présence d'un lobe ayant une bande passante de 20kHz sur toutes les trames.



ASTABLE

Théorie

L'objectif de l'astable est de générer un signal rectangulaire de fréquence 400KHz.



Le condensateur C se charge à travers la résistance R vers $-V_{cc}$.
La tension sur la borne + devient alors :

$$V+ = \frac{R1}{R1 + R2} V_s \quad V^- = V_c$$

$$I = -C \frac{dV_c}{dt} = -C \frac{dV^-}{dt} \quad RI = V^- - V_s$$

$$Vs = RC \frac{dV_-}{dt}$$

A t=0

$V_- = 0$ (Capacité déchargée)
 $C = -Vs$, $Vs = Vcc$, on obtient alors :

$$V^-(t) = Vcc[1 - \exp(\frac{-t}{RC})]$$

Calcul de Tc:
 En A

$$Vcc = (1 - \exp(\frac{-Tc}{RC})) = \frac{R1}{R1 + R2} Vcc$$

$$\exp(\frac{-Tc}{RC}) = 1 - \frac{R1}{R1 + R2} = \frac{R2}{R1 + R2}$$

$$Tc = RCl \ln(1 + \frac{R1}{R2})$$

Calcul de Td:

On a $Td = Tc$ car la constante de temps τ est la même pour la charge et la décharge.

Maintenant, A t = Tc
 $V_- = Vc1$ $Vs = -Vcc$

Ce qui donne :

$$Vc1 = -\frac{R1}{R1 + R2} Vcc$$

$$V_- = C1 \exp(\frac{-Tc}{RC}) - Vcc = Vc1$$

$$C1 = (Vc1 + Vcc) \exp(\frac{Tc}{RC}) \exp(\frac{-t}{RC}) - Vcc$$

$$V_- = -\frac{R1}{R1 + R2} Vcc = (Vcc + \frac{R1}{R1 + R2} Vcc) \exp(\frac{Tc - Td}{RC}) - Vcc$$

$$Tc + Td = T = 2RCl \ln(1 + \frac{2R1}{R2})$$

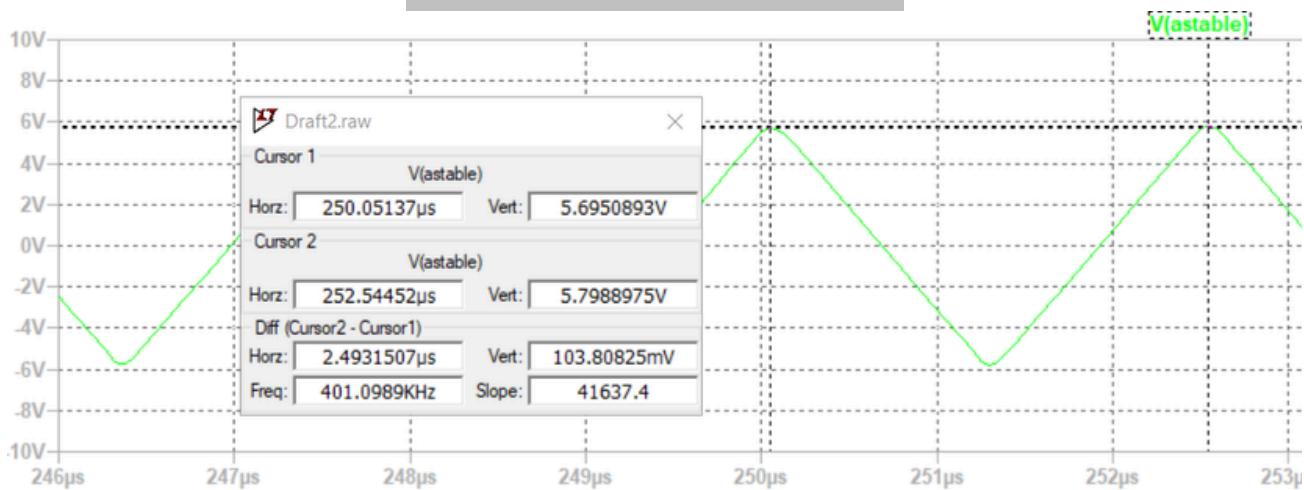
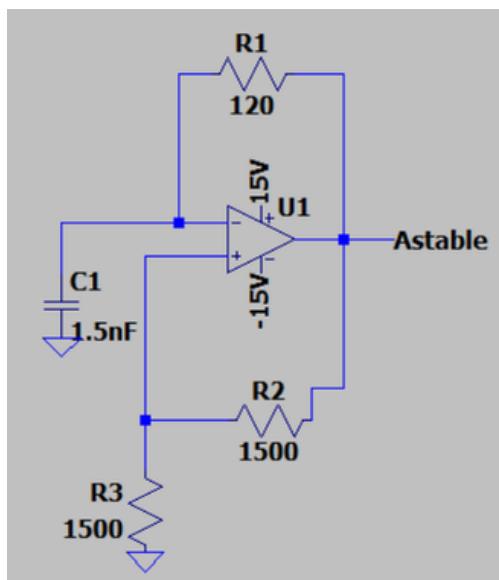
Finalement : $f = \frac{1}{T} = \frac{1}{2RCl \ln(1 + \frac{2R1}{R2})}$

En fixant les résistances R1 et R2 à 1.5 kΩ et la capacité C à 1 nF, on obtient alors $R = 1.2$ kΩ

Simulation

Toutes les simulations s'effectueront sur LT Spice :

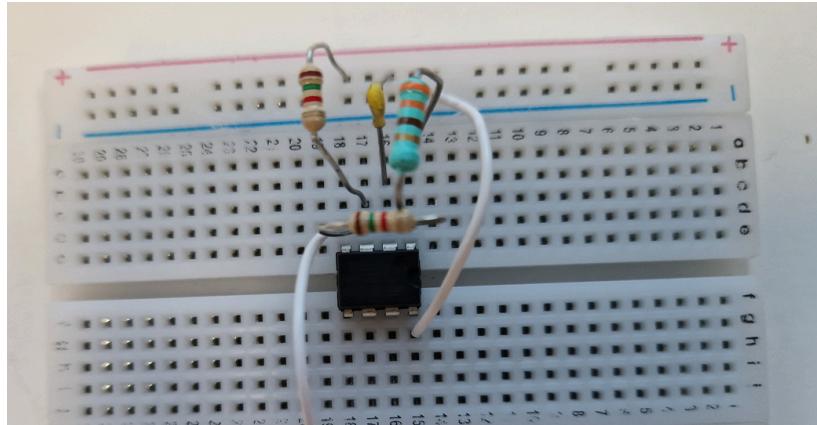
Les valeurs choisies théoriquement ne nous donnant pas de résultat concluant, nous décidons alors de modifier la valeur de certains composants. On choisit alors : $R = 120\Omega$ et $C = 1.5nF$



On observe que l'astable n'est pas de forme rectangulaire. En effet, nous n'arrivions pas à donner cette forme au signal tout en conservant une fréquence aux alentours de 400kHz. Nous avons donc préféré obtenir une bonne fréquence plutôt qu'une allure parfaitement rectangulaire. Cela est notamment dû à la charge et décharge de la capacité.

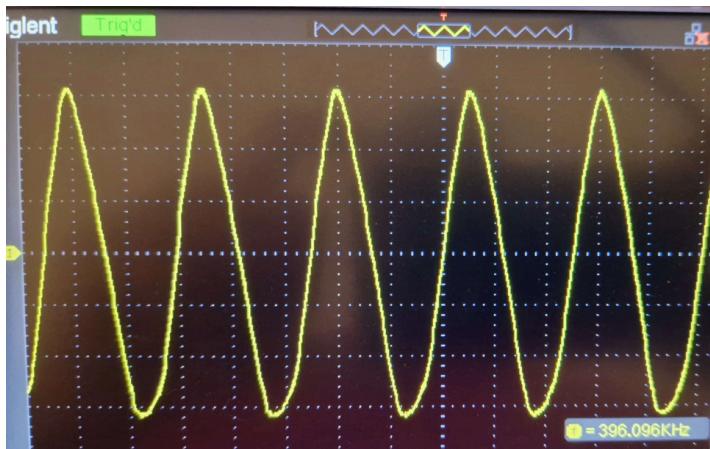
Réalisation

MONTAGE

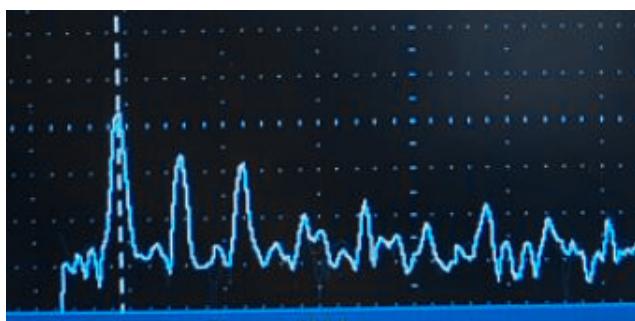


Expérimentalement, on observe une différence de fréquence avec ce qui a été simulé, une adaptation était donc nécessaire et les composants choisis sont alors :

$$R1 = 330\Omega \quad R2=R3=1.5k\Omega \quad C1 = 780pF$$



De cette façon, on retrouve bien un signal de 400kHz. Il n'est pas très rectangulaire, mais sa forme actuelle n'aura que très peu d'impact sur la suite du projet.



Petit mot sur la FFT :
Nous retrouvons bien le fondamental à 400kHz mais aussi les harmoniques tous les 400 kHz.

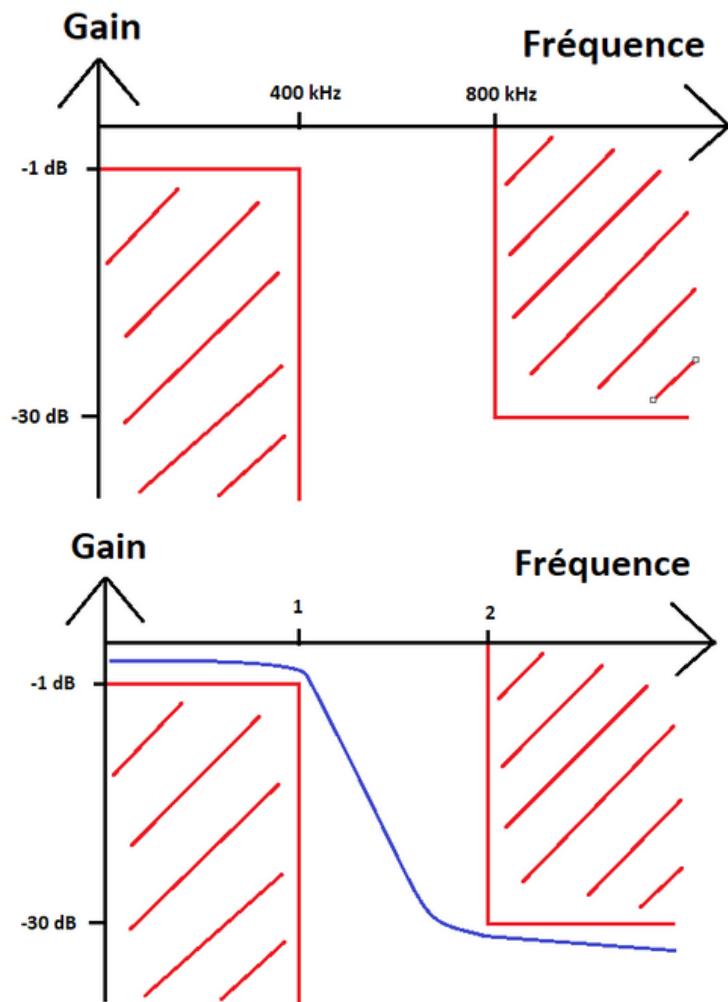
Filtrons maintenant ce signal afin de seulement garder le fondamentale.

FILTRE

Objectif

La forme d'onde en sortie d'astable étant quasi rectangulaire, on ne peut pas directement se servir de cette sortie comme oscillateur local, car sinon, le canal sera saturé avec toutes les harmoniques. Il faut donc transformer ce signal en une sinusoïde de fréquence 400kHz. Pour cela nous allons dimensionner un filtre de Butterworth.

Voici le gabarit du filtre que nous souhaitons réaliser :



Théorie

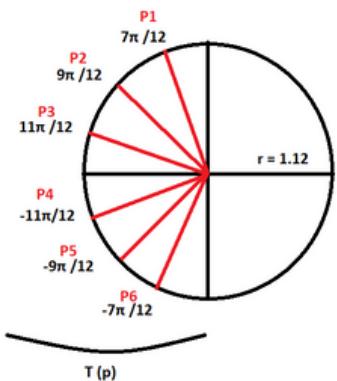
Commençons, d'abord par trouver la dimension du filtre n ainsi que ε .

$$n \geq \frac{1}{2} \frac{\ln(10^{\frac{-G_{max}}{10}} - 1) - 2\ln(\varepsilon)}{\ln(\frac{w_2}{w_1})} \quad \varepsilon \geq \sqrt{10^{\frac{-G_{min}}{10}} - 1}$$

Avec $G_{max} = -30$, $G_{min} = -1$, $w_1 = 400\text{kHz}$ et $w_2 = 800\text{kHz}$.

On trouve : $\varepsilon \geq 0.5$ $n \geq 5.95 \simeq 6$

Nous avons donc à placer sur notre cercle de rayon $\frac{1}{\varepsilon^{\frac{1}{n}}} = 1.12$
 des points espacés d'un angle $\frac{\pi}{n} = \frac{\pi}{6}$



Voici, ainsi le cercle obtenu avec les points P1 à P6 placés pour T(p)

On trouve :

$$P1 = R * (\cos(7\pi/12) + j \sin(7\pi/12)) = -0.29 + j 1.08$$

$$P2 = R * (\cos(9\pi /12) + j \sin(9\pi /12)) = -0.79 + j 0.79$$

$$P3 = R * (\cos(11\pi/12) + j \sin(11\pi/12)) = -1.08 + j 0.29$$

$$P_4 = P_3^* \quad P_5 = P_2^* \quad P_6 = P_1^*$$

On obtient alors :

$$T(p) = \frac{1}{\varepsilon} * \frac{1}{p^2 + p0.58 + 1.25} * \frac{1}{p^2 + p1.58 + 1.25} * \frac{1}{p^2 + p2.16 + 1.25}$$

A **B** **C**

On a donc un filtre de la forme :

$$T(j\omega) = \frac{1}{\varepsilon} * \frac{T_{oA}}{1 + 2m_A j \frac{\omega}{w_{1A}} + (j \frac{\omega}{w_{1A}})^2} * \frac{T_{oB}}{1 + 2m_B j \frac{\omega}{w_{1B}} + (j \frac{\omega}{w_{1B}})^2} * \frac{T_{oC}}{1 + 2m_C j \frac{\omega}{w_{1C}} + (j \frac{\omega}{w_{1C}})^2}$$

A: $T_A(p) = \frac{1.25^{-1}}{1 + \frac{0.58}{1.25}p + \frac{p^2}{1.25}}$ $T_A(j\omega) = \frac{0.8}{1 + 0.46 * 1.12 j \frac{\omega}{1.12w_1} p + (j \frac{\omega}{1.12w_1})^2}$

$$T_{oA} = 0.8 \quad w_{oA} = 1.12w_1 \quad m_A = 0.26$$

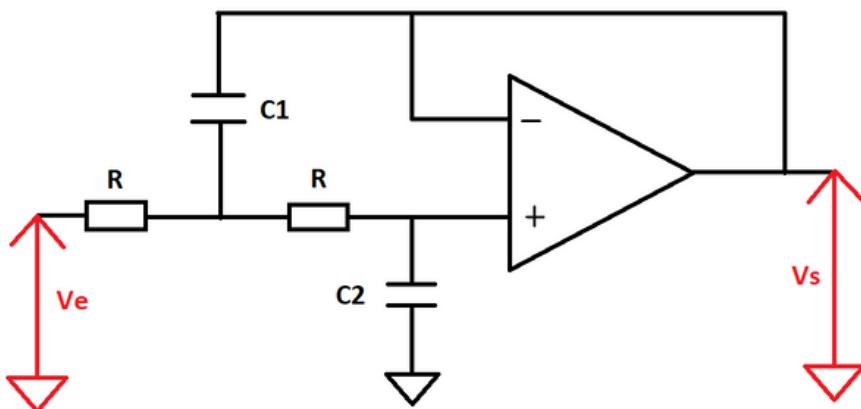
B: $T_B(p) = \frac{1.25^{-1}}{1 + \frac{1.58}{1.25}p + \frac{p^2}{1.25}}$ $T_B(j\omega) = \frac{0.8}{1 + 1.26 * 1.12 j \frac{\omega}{1.12w_1} p + (j \frac{\omega}{1.12w_1})^2}$

$$T_{oB} = 0.8 \quad w_{oB} = 1.12w_1 \quad m_B = 0.71$$

C: $T_C(p) = \frac{1.25^{-1}}{1 + \frac{2.16}{1.25}p + \frac{p^2}{1.25}}$ $T_C(j\omega) = \frac{0.8}{1 + 1.93 * 1.12 j \frac{\omega}{1.12w_1} p + (j \frac{\omega}{1.12w_1})^2}$

$$T_{oC} = 0.8 \quad w_{oC} = 1.12w_1 \quad m_C = 0.96$$

Pour réaliser notre filtre, nous décidons d'utiliser ce montage à amplificateur opérationnel idéal qui correspond à un filtre passe-bas d'ordre 2.



Plaçons la tension V' aux noeuds entre les 2 résistances R et la capacité $C1$. On trouve alors :

$$V_s = \frac{Zc2}{Zr * Zc2} V' = \frac{1}{1 + jRC2\omega} V'$$

$$(Ve - V')Y_r = (V' - V_s)Y_r + (V' - V_s)Y_{c1}$$

$$VeY_r + Vs(Y_r + Y_{c1}) = V'(2Y_r + Y_{c1})$$

$$VeYr + Vs(Yr + Yc1) = Vs(2Yr + Yc1)(1 + ZrYc2)$$

$$VeYr + Vs(Yr + Yc1 - 2Yr - 2Yc2 - Yc1 - ZrYc1Yc2) = 0$$

$$VeYr - Vs(2Yc2 + ZrYc1Yc2) = 0$$

$$\frac{Vs}{Ve} = \frac{Yr}{Yr + ZrYc1Yc2 + 2Yc2} = \frac{1}{1 + 2Yc2Zr + Yc1Yc2Zr^2}$$

Finalement:

$$\frac{Vs}{Ve} = \frac{1}{1 + 2jRc2w + j^2w^2R^2C1C2}$$

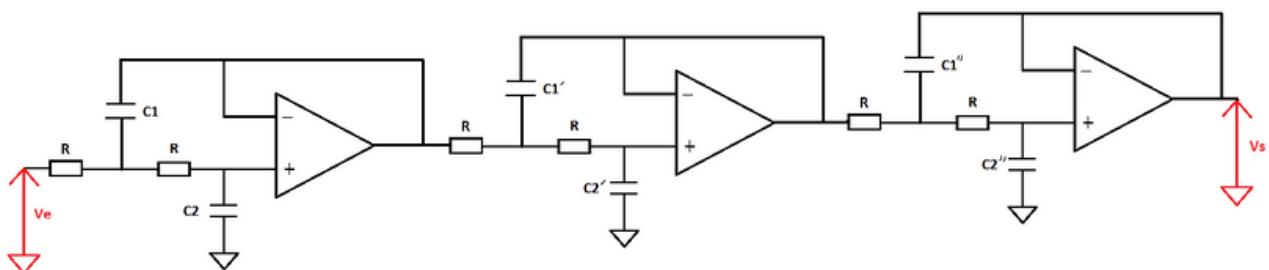
Cette fonction de transfert est de la forme :

$$\frac{Vs}{Ve} = \frac{T_o}{1 + 2mj\frac{w}{w_1} + (j\frac{w}{w_1})^2}$$

On en déduit donc :

$$T_o = 1 \quad w_o = \frac{1}{R\sqrt{C1C2}} \quad m = \sqrt{\frac{C2}{C1}}$$

Mettons maintenant en série notre montage afin de l'appliquer à notre filtre d'ordre 6 de fréquence de coupure 400 kHz.



Dimensionnons maintenant les éléments actifs de notre circuit.
Fixons toutes les résistances de notre montage en série à $1k\Omega$.

$$\text{On a } f_c, \text{ la fréquence de coupure de notre filtre: } f_c = \frac{1}{2\pi R\sqrt{C1C2}}$$

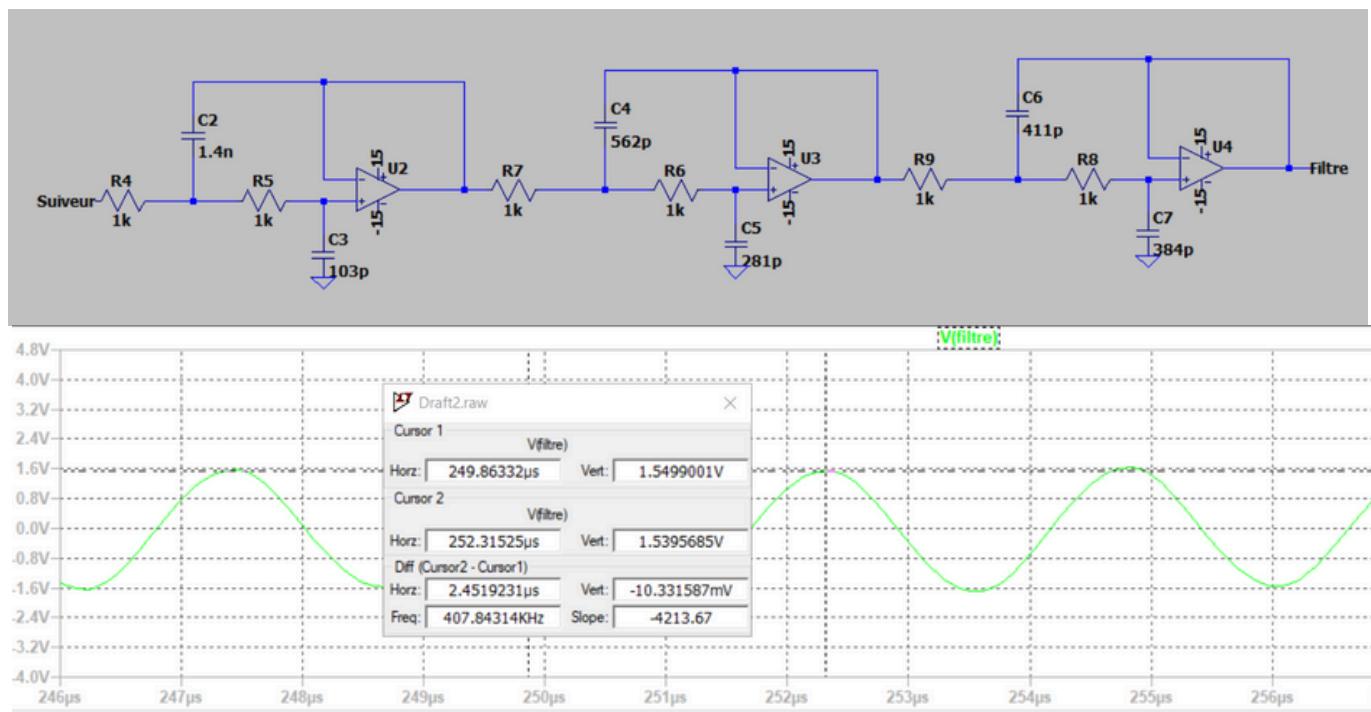
En utilisant les résultats trouvés précédemment, on obtient :

$$C1 = 1.4nF \quad C2 = 103pF$$

$$C1' = 562pF \quad C2' = 281pF$$

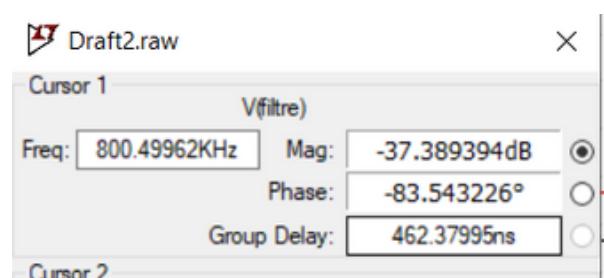
$$C1'' = 411pF \quad C2'' = 384pF$$

Simulation



La fréquence obtenue est bien de 400kHz est l'allure du filtre est bien sinusoïdale.

Regardons maintenant le diagramme de Bode de notre filtre, afin d'observer la fréquence de coupure obtenue.



Le diagramme de Bode semble plutôt correct, nous voulions obtenir -30dB à 800 kHz. Nous sommes à -37dB ce qui est légèrement au dessus mais cela ne devrait pas poser de problème pour la suite.

Réalisation

Pour réaliser notre filtre nous choisissons des composants normés.

On a alors :

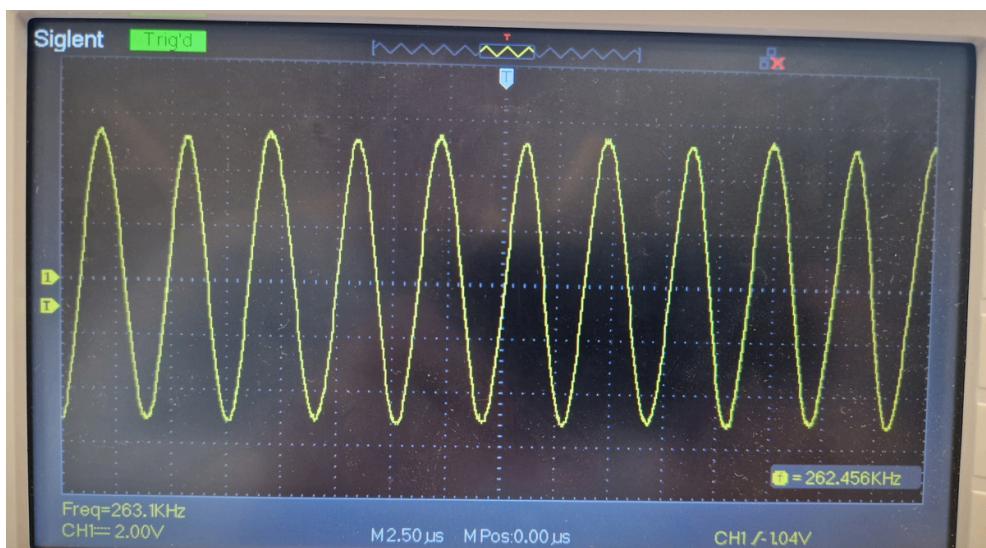
$$R = 1\text{k}\Omega$$

$$C1 = 1.5\text{nF} \quad C2 = 100\text{pF}$$

$$C1' = 560\text{pF} \quad C2' = 270\text{pF}$$

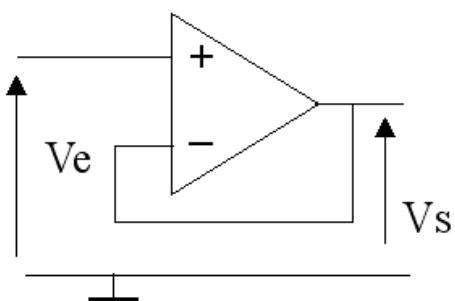
$$C1'' = 390\text{pF} \quad C2'' = 390\text{pF}$$

Nous obtenons quelque chose de peu satisfaisant, et en analysant de nouveau notre astable, voici ce que l'on obtient :

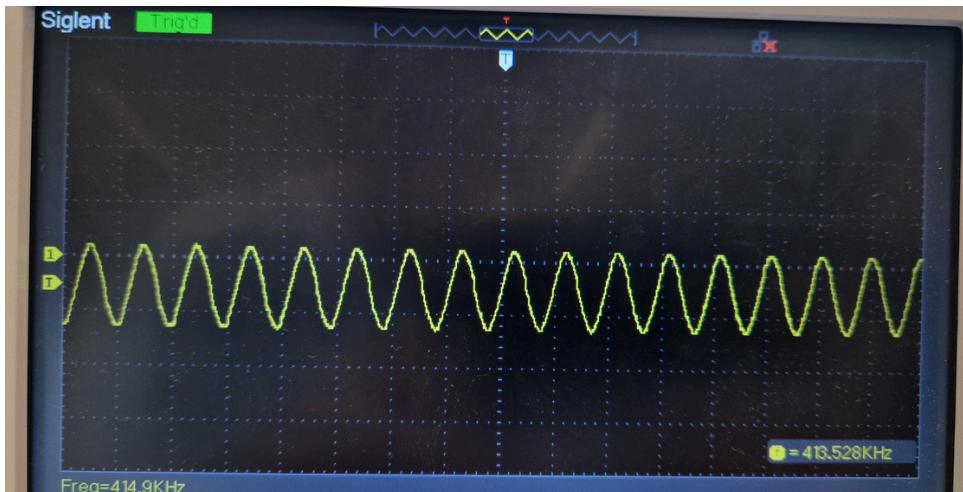


La fréquence de notre astable est descendue de 396 kHz à 262 kHz après l'ajout de notre filtre.

Nous décidons donc d'ajouter un montage Suiveur à l'aide d'un AOP.



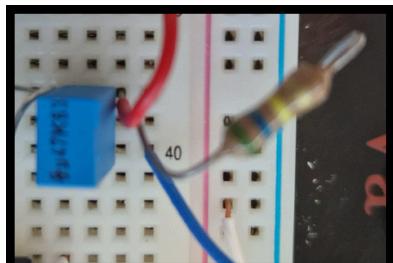
Ce montage a pour objectif d'isoler l'astable du reste du circuit, ici notre filtre, afin d'améliorer la qualité du signal et limiter les rétroactions de tension entre l'astable et le filtre.



La sortie de notre filtre est maintenant beaucoup plus acceptable, on retrouve bien la forme sinusoïdale cherchée avec une fréquence d'environ 400 kHz. Le signal est légèrement atténué, nous passons d'une amplitude crête à crête de 10V pour l'astable à 3V pour le filtre. Or, cette légère modification ne devrait pas avoir de répercussions sur la suite du projet.

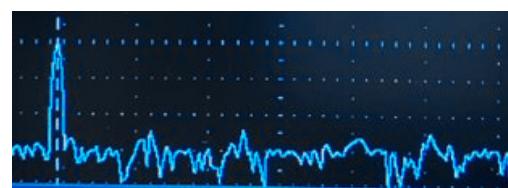
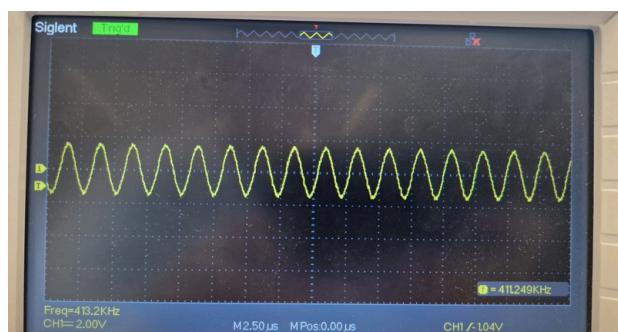
Ce qui est quelque peu gênant en revanche est la présence d'un offset sur notre filtre. En effet l'état haut est à peine au-dessus de la barre des 0V tandis que l'état bas lui est à -2.3V.

Afin de supprimer cet offset, nous décidons de placer à la suite du filtre passe-bas d'ordre 6, un filtre passe-haut RC.



Ce filtre passe-haut est composé, d'une résistance $R = 560\text{k}\Omega$ et d'une capacité $C = 470\text{nF}$. A lui seul, il suffira à bloquer la composante continue qui crée cet offset.

Le résultat obtenu est le suivant :



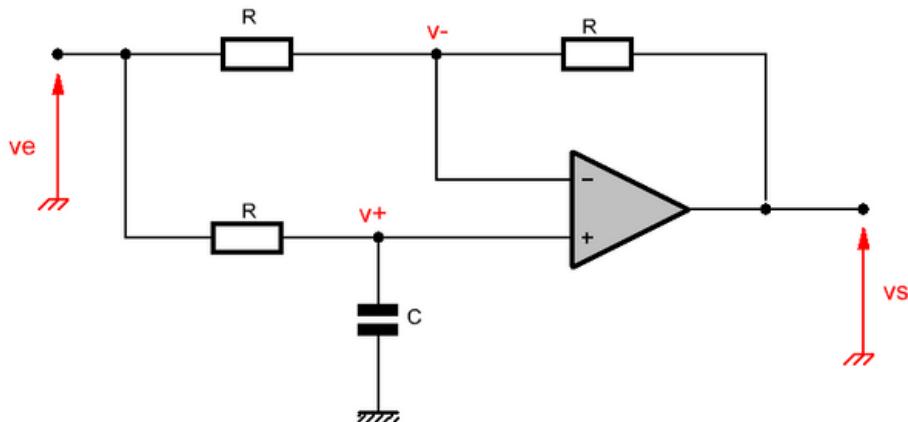
À noter que les harmoniques ont bien été atténuées de 30dB.

L'offset étant supprimé, nous pouvons donc passer à la suite du projet.

DÉPHASEUR

Théorie

Afin de réaliser notre modulateur IQ, il nous faut maintenant obtenir le même signal sinusoïdal précédemment conçu, mais déphasé de 90°.



ALI idéal, on a donc : $V_- = V_+$ et $i_- = i_+ = 0$

$$\text{Millmann en } - : V_- = \frac{V_e + V_s}{2}$$

$$\text{Millmann en } + : V_+ = \frac{V_e}{1 + jRC\omega}$$

$$\text{On obtient : } \frac{V_s}{V_e} = \frac{1 - jRC\omega}{1 + jRC\omega}$$

Le gain de la fonction de transfert est de 1.

La phase elle dépend de w. $\varphi(w) = -2\arctan(RCw)$

Nous devons donc à 400kHz avoir une phase de +- 90° ($\pi/2$).

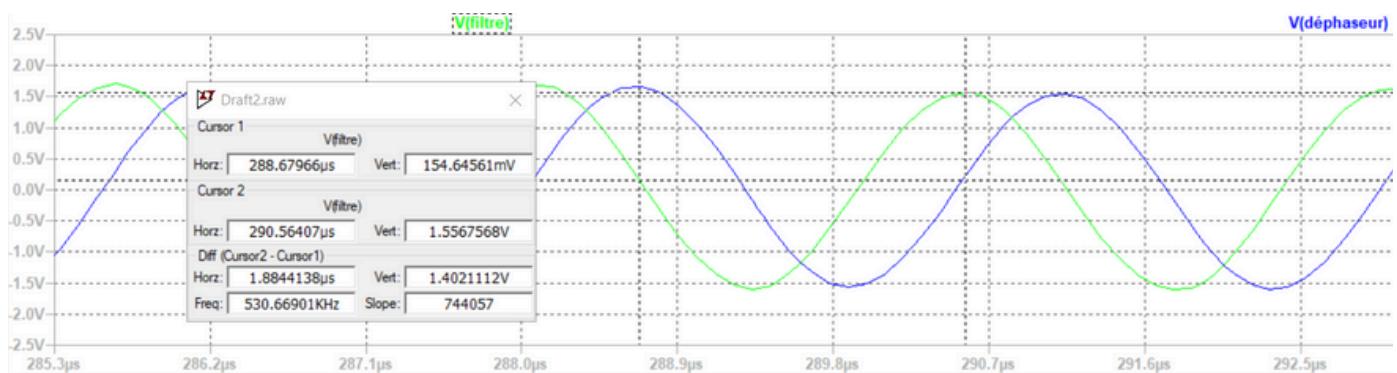
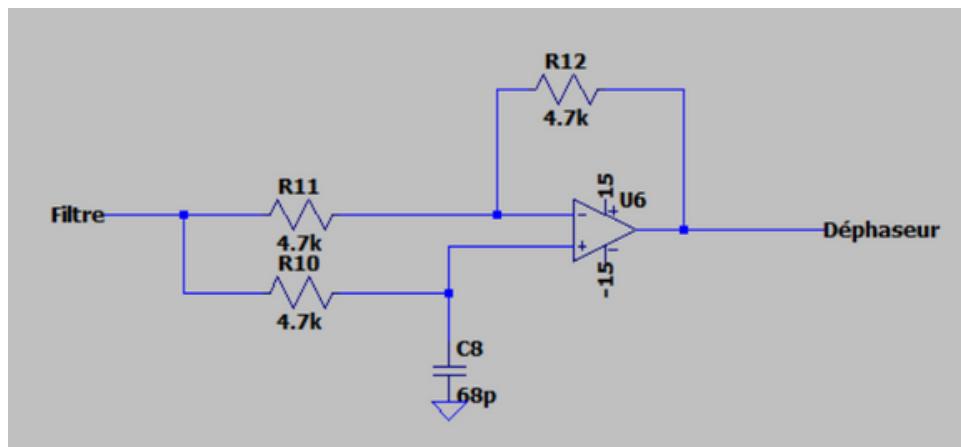
On en déduit que $RCw = 1$ et donc $f = \frac{1}{2\pi RC}$

On fixe alors $R = 4.7\text{k}\Omega$ et on détermine ensuite une capacité de 82pF .

Simulation

On note un angle légèrement trop supérieur à 90°.

Nous décidons alors de prendre une capacité de 68 pF.



Pour calculer l'angle sur LT Spice nous décidons de regarder la fréquence entre un maximum N du filtre et le maximum N+1 du déphaseur. Nous obtenons 530 kHz.

Après un petit calcul : $360^\circ - (400\text{kHz} / 530\text{kHz} * 360^\circ) = 90^\circ$

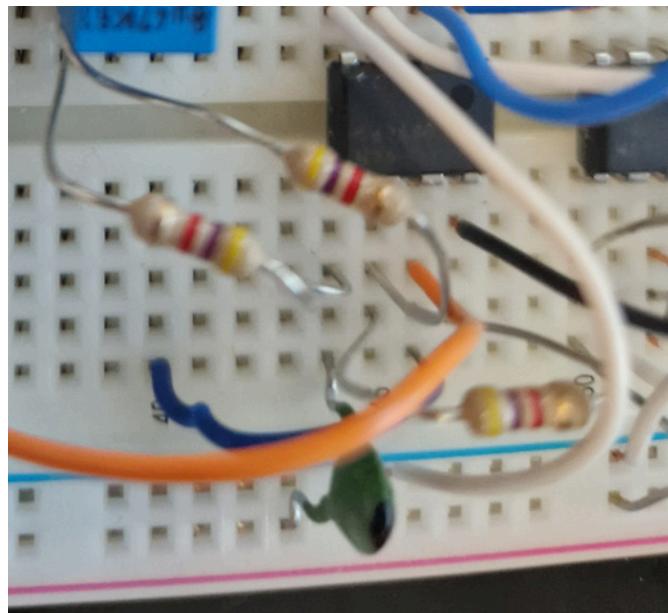
Le déphasage est maintenant donc bien de 90°.

On notera que l'amplitude n'a pas été modifiée.

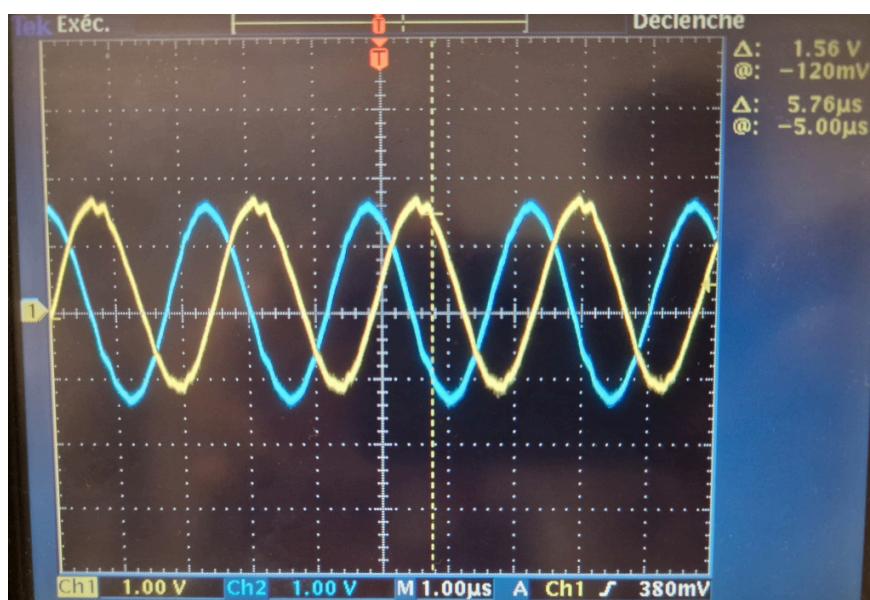
Regardons de suite ce qu'il en est pour la réalisation de ce montage.

Réalisation

MONTAGE



En prenant les valeurs précédemment choisies pour la simulation : $R_1=R_2=R_3=4.7\text{k}\Omega$ $C_1 = 68\text{pF}$, nous retrouvons bien notre déphasage de 90° sans perte d'amplitude.



Grâce à ce déphasage nous avons donc maintenant notre sinus et cosinus nécessaires à la modulation QPSK.

$$s(t) = I(t)\cos(\omega t) + Q(t)\sin(\omega t)$$

MULTIPLICATEURS

Mise en place

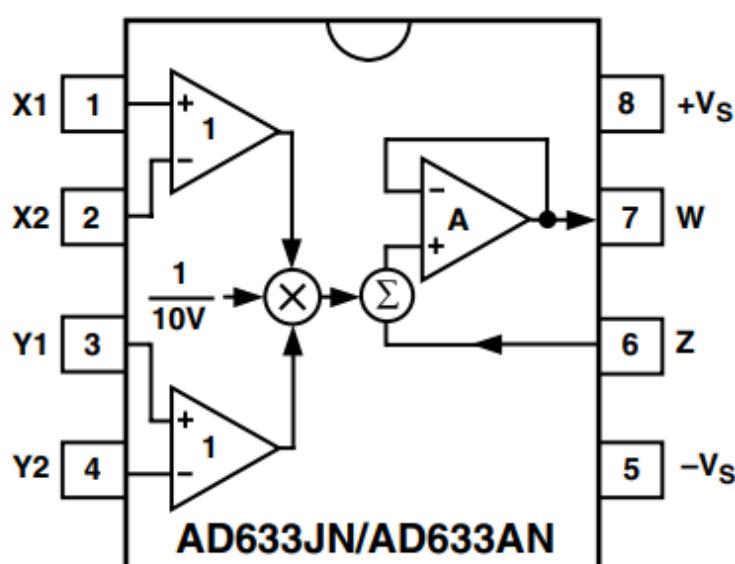
L'expression d'un signal QPSK est le suivant :

$$s(t) = I(t)\cos(\omega t) + Q(t)\sin(\omega t)$$

Dès lors, l'objectif est de multiplier les trames I et Q aux signaux créés précédemment.

Pour ce faire, on place le fichier .bit, obtenu après avoir généré le bitstream du code VHDL, dans une clé USB. Cette clé est ensuite placée dans la Basys-3 qui fera office de générateur de trame.

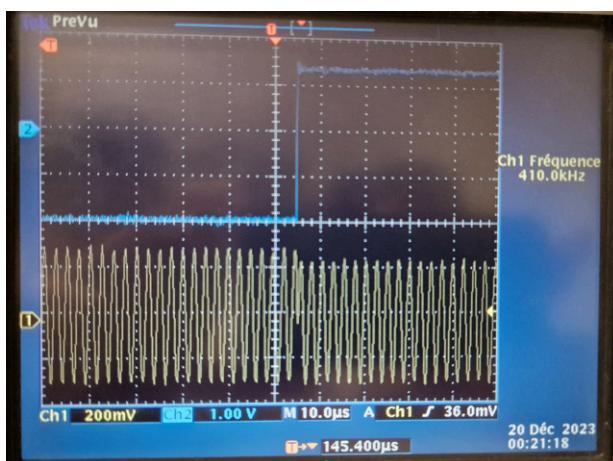
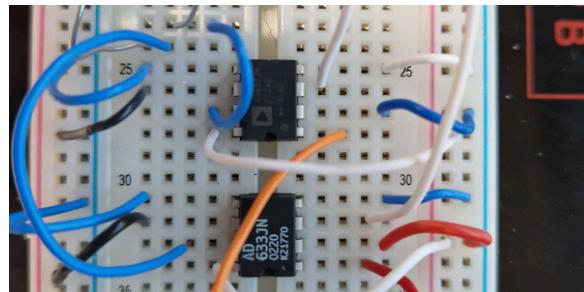
Voici un schéma fonctionnel du multiplicateur AD633 que nous utiliserons :



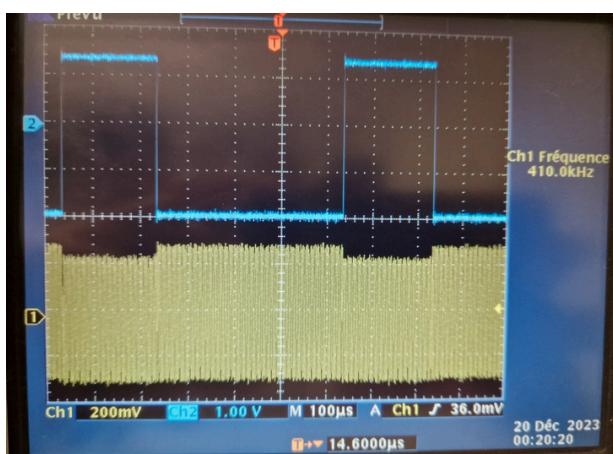
Les entrées - (X2, Y2) et Z sont sur la masse, les entrées + (X1, Y1) sont les signaux à multiplier, soit notre cosinus avec la trame I et notre sinus avec la trame Q.

Réalisation

MONTAGE



On remarque bien un saut de phase à chaque changement d'état d'une trame.



Et que les changements d'états ont bien la répercussion attendue.

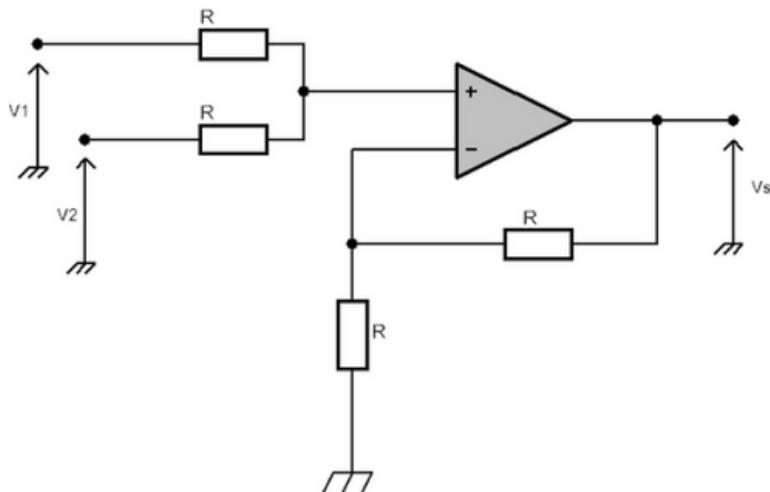
Il est important de préciser que l'ajout des multiplicateurs modifiait la masse générale de notre montage, nous avons donc rajouté des condensateurs de découplage entre notre masse et nos alimentations.

SOMMATEURS

Théorie

Il nous faut maintenant sommer nos deux sorties de multiplicateurs.

$$s(t) = I(t)\cos(\omega t) + Q(t)\sin(\omega t)$$



ALI idéal, on a donc : $V_- = V_+$ et $i_- = i_+ = 0$

$$\text{Millmann en } -: V_- = \frac{V_s}{2}$$

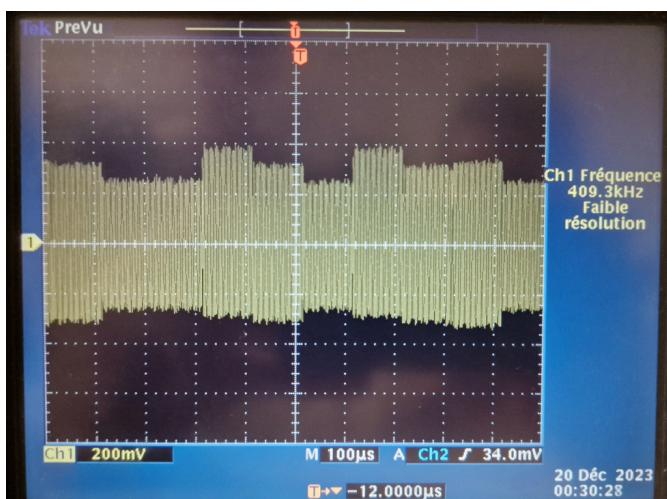
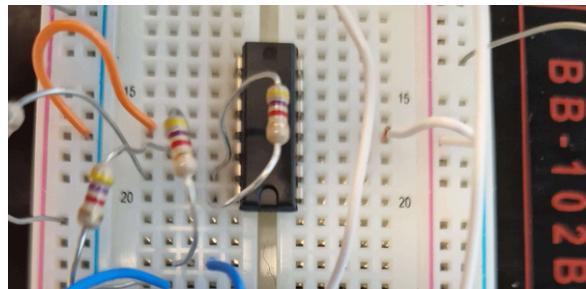
$$\text{Millmann en } +: V_+ = \frac{V_1 + V_2}{2}$$

$$\text{On a alors: } V_s = V_1 + V_2$$

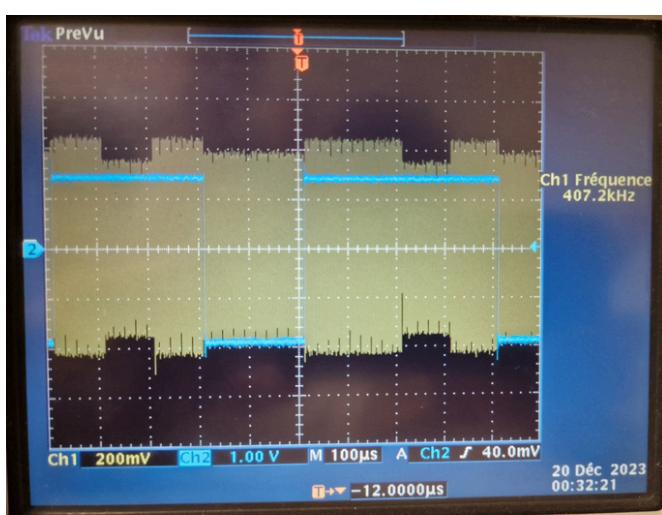
Nous obtenons donc bien notre modulateur QPSK.

Réalisation

MONTAGE



En sortie du sommateur, on observe bien les différents états du signal QPSK. Nous pouvons en distinguer 4, ce qui est, ce que l'on souhaitait obtenir.



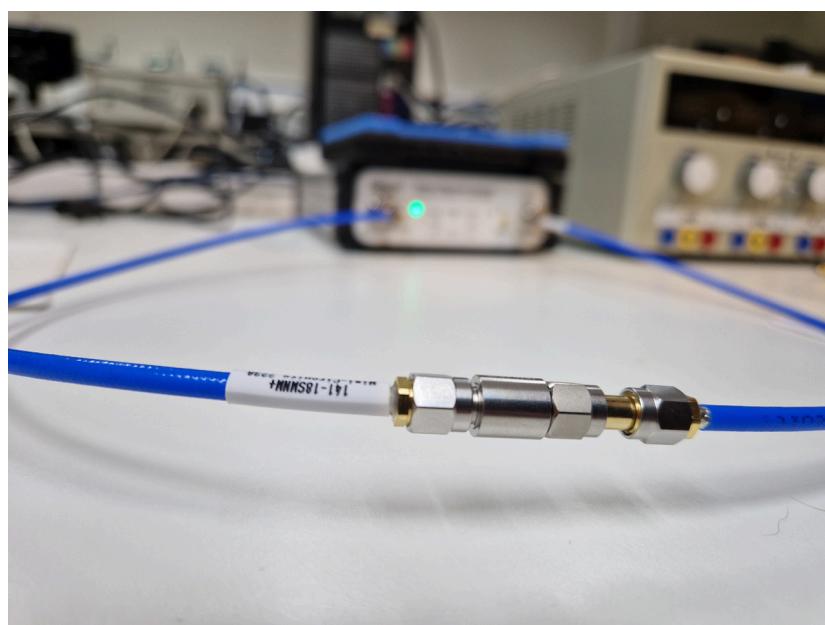
Superposé à la trame I, on observe bien les changements d'états de la sortie du sommateur liés aux changements d'états de la trame I.

Petit mot sur la FFT, on remarque que la bande passante du signal vaut 20kHz et non les 40kHz que l'on aurait dû obtenir si nous avions utilisé la trame binaire pseudo-aléatoire à la place de la trame IQ.

Notre modulateur QPSK est donc bel est bien terminé.

CARACTÉRISATION

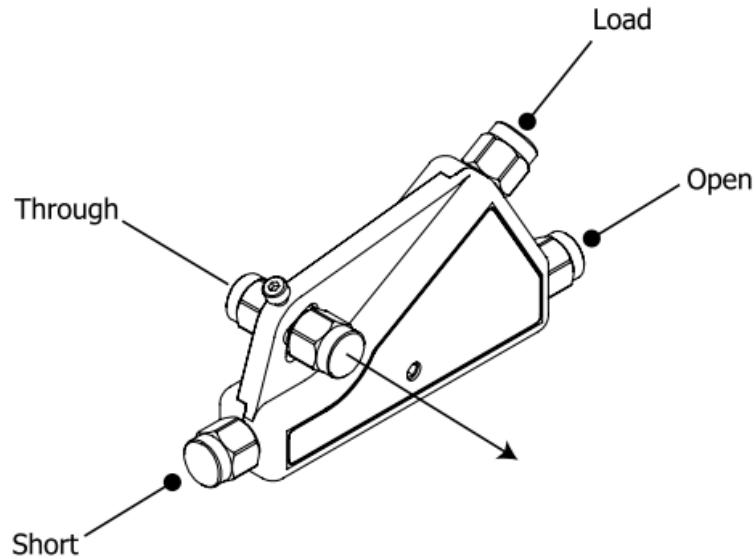
Pour finir ce projet, nous allons aborder la caractérisation de composants servant à la transmission de signaux en haute fréquence. Afin de mesurer les paramètres S d'un quadripôle, on utilise un analyseur de réseau vectoriel (VNA).



La première étape est donc le calibrage de cet analyseur de réseau vectoriel, effectué à l'aide de l'outil ci-dessous, qui sert de référence.



Pour ce faire, il suffit de suivre les étapes que le logiciel nous fournit, c'est-à-dire, connecter successivement les entrées Short-Open, Open-Short, Short-Load, Load-Short, et Through.



Une fois cela effectué, on vérifie à l'aide d'une simple connexion through si son paramètre S11 est bien au centre de l'abaque de Smith.



Ici c'est bien le cas, nous pouvons donc passer à la suite.

Mesurons maintenant le point de compression de l'amplificateur à l'aide d'une série de mesures faisant varier les dBm d'entrée.

Avant cela nous avons caractérisé un coupleur et un atténuateur pour les fréquences 400kHz et 800MHz.

Coupleur:

$$S_{400k} = \begin{bmatrix} -31.5 & -1.1 & -10.7 \\ -1.1 & -20.6 & -41.3 \\ -10.7 & -41.2 & -11.8 \end{bmatrix}$$

$$S_{800M} = \begin{bmatrix} -25.5 & -1.2 & -10.8 \\ -1.2 & -30.9 & -31.2 \\ -10.8 & -31.3 & -17 \end{bmatrix}$$

Atténuateur:

$$S_{400k} = \begin{bmatrix} -40.1 & -9.8 \\ -9.8 & -39.1 \end{bmatrix}$$

$$S_{800M} = \begin{bmatrix} -38.5 & -9.9 \\ -9.9 & -35.6 \end{bmatrix}$$

On en déduit par exemple pour l'atténuateur, une valeur d'atténuation de :

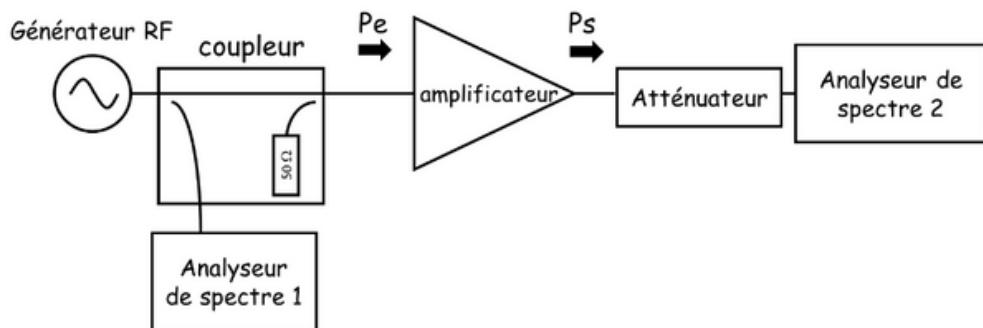
$$A = -20\log(9.8) = -19.8 \text{ dB à } 400\text{kHz}$$

$$A = -19.9 \text{ dB à } 800\text{MHz}$$

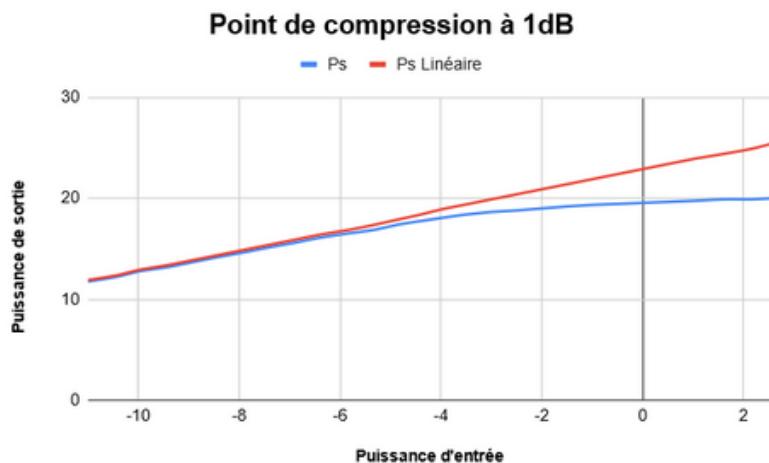
Cet atténuateur semble avoir la même atténuation peu importe la fréquence.

Point de compression à 1db de l'amplificateur:

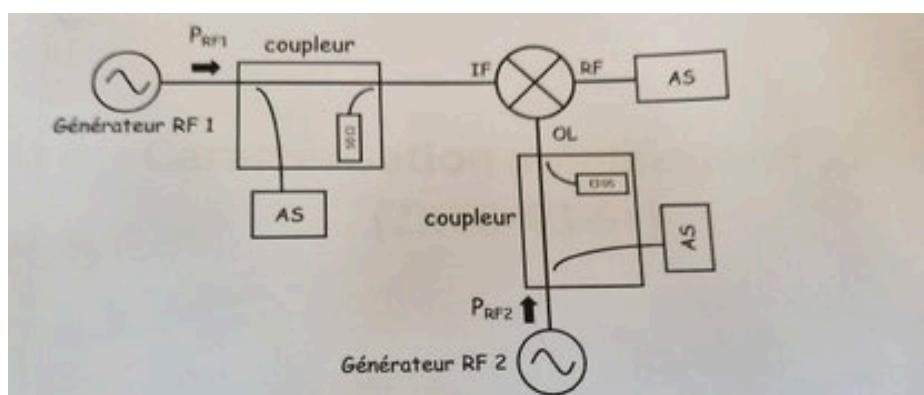
Nous réalisons maintenant le montage suivant, afin d'effectuer une série de mesures.



On trouve alors que le point de compression de cet amplificateur est à environ 3.5dBm sur l'entrée.



Pour finir nous avons caractérisé de la même manière un frequency mixer via l'intermédiaire de 3 manipulations.



1ere manipulation :

Tout d'abord nous voulons déterminer les isolations (OL-RF) et les pertes de conversion (IF-RF).

On fixe :

$P_{if} = -20 \text{ dBm}$, $P_{ol} = 10 \text{ dBm}$, $F_{if} = 400 \text{ kHz}$, $F_{ol} = 800 \text{ MHz}$

On obtient alors :

$$\text{Isolation (Ol-rf)} = P_{ol} (\text{voie OL}) - P_{ol} (\text{voie RF}) = 10 - (-23) = 33 \text{ dB}$$

$$\text{Isolation (If-rf)} = P_{if} (\text{voie IF}) - P_{if} (\text{voie RF}) = -20 - (-36) = 16 \text{ dB}$$

$$\text{Pertes de conversion (dB)} = P_{if} (\text{voie IF}) - P_{rf} (\text{voie RF}) = -20 - (-27) = 7 \text{ dB}$$

2eme manipulation :

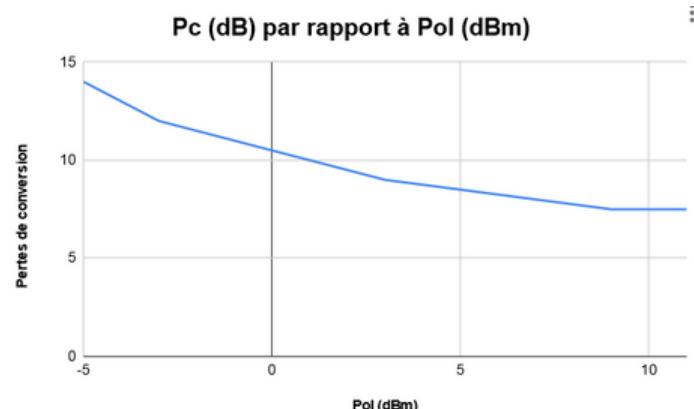
Ensuite traçons les pertes de conversions en fonction de Pol

On fixe :

$$P_{if} = -20 \text{ dBm}, F_{if} = 400 \text{ kHz}, F_{ol} = 800 \text{ MHz}$$

Nous obtenons comme résultats :

Pol (dBm)	Pif (dBm)	Prf (dBm)	Pc (dB)
-5	-20	-34	14
-3	-20	-32	12
-1	-20	-31	11
1	-20	-30	10
3	-20	-29	9
5	-20	-28,5	8,5
7	-20	-28	8
9	-20	-27,5	7,5
11	-20	-27,5	7,5



3eme manipulation :

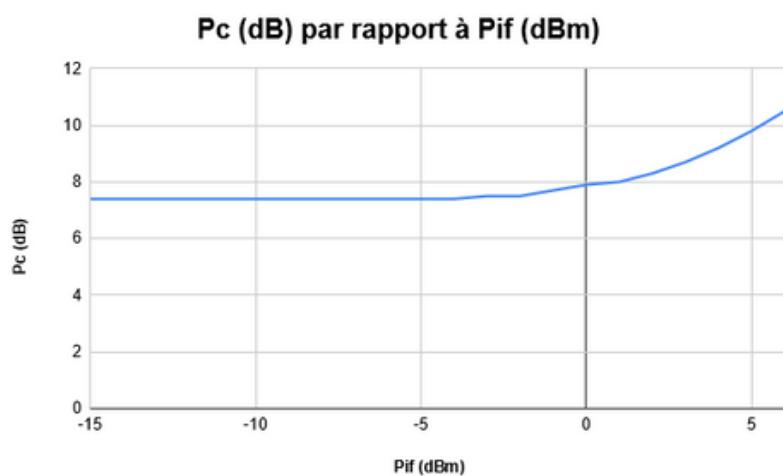
Pour finir, il nous faut tracer les pertes de conversion pour en déduire le point de compression.

On fixe :

$$Pol = 10 \text{ dBm}, F_{if} = 400 \text{ kHz}, F_{ol} = 800 \text{ MHz}$$

Pour Pif (voie IF) variant de -15dBm à 6dBm on obtient :

Pif (dBm)	Prf (dBm)	Pc (dB)
-15	-22,4	7,4
-14	-21,4	7,4
-13	-20,4	7,4
-12	-19,4	7,4
-11	-18,4	7,4
-10	-17,4	7,4
-9	-16,4	7,4
-8	-15,4	7,4
-7	-14,4	7,4
-6	-13,4	7,4
-5	-12,4	7,4
-4	-11,4	7,4
-3	-10,5	7,5
-2	-9,5	7,5
-1	-8,7	7,7
0	-7,9	7,9
1	-7	8
2	-6,3	8,3
3	-5,7	8,7
4	-5,2	9,2
5	-4,8	9,8
6	-4,5	10,5



Le point de compression est à 8.3dB.

CONCLUSION

En conclusion, ce projet de Travaux Pratiques a permis d'explorer et de mettre en œuvre la conception d'un émetteur QPSK hétérodyne. Les différentes étapes du projet, allant de la génération de la trame binaire pseudo-aléatoire à la caractérisation des composants, ont été abordées avec succès.

L'objectif principal était de développer une compréhension approfondie des principes fondamentaux de la modulation QPSK et des techniques de conversion hétérodyne. La modélisation VHDL a été utilisée pour générer une trame binaire pseudo-aléatoire, un filtre de Butterworth a été mis en place pour obtenir des signaux sinusoïdaux, et un déphaseur pour déphasier de 90 degrés.

La réalisation pratique a impliqué l'utilisation d'outils de simulation tels que LT Spice, ainsi que la mise en place de montages électroniques avec des composants réels. Des ajustements ont été nécessaires pour obtenir des résultats conformes aux attentes, démontrant ainsi l'importance de la phase expérimentale dans le processus de conception.

La caractérisation des composants, tels que l'amplificateur et le coupleur, a permis de mieux comprendre leur comportement dans des conditions spécifiques.

En fin de compte, ce projet a permis d'acquérir des compétences pratiques dans l'utilisation d'outils de simulation, la manipulation de composants électroniques, et la caractérisation des systèmes à haute fréquence. La réussite de ce projet contribuera à consolider les connaissances théoriques et pratiques en télécommunications, ouvrant ainsi la voie à une meilleure compréhension des systèmes de communication sans fil modernes.

Attention particulière aux professeurs et responsables de ce projet:

M.Defrance

Mme.Pichonat

M.Wichmann

