

Remerciements

Pour nous avoir soutenu tout au long de ce projet, l'équipe du projet souhaite remercier M. Lagrèze, M. Pastorino et M. Genon-Catalot pour leur disponibilité et leur aide. Nous remercions aussi M. Bernardi pour ses conseils et pour nous avoir guidé quand nous étions dans le doute.

Le projet n'aurait aussi pu être possible sans Planètes-Sciences qui ont initié le projet « Ballon » dans les écoles et les clubs.

Sommaire

Introduction.....	3
I. Présentation.....	4
1. Cahier des charges et spécifications.....	4
2. Frequency Shift Keying.....	9
3. Code correcteurs: Hamming.....	12
II. Développement.....	17
1. Modulateur FSK.....	18
2. Démodulateur FSK.....	25
3. Logiciel.....	33
III. Résultats des tests.....	39
Conclusion.....	56
Bibliographie/Webographie.....	58
Lexique.....	60

Annexes

- I. Schématics des modulateurs/démodulateurs
- II. Documentation technique du XR2206
- III. Documentation technique du XR2211
- IV. Documentation technique du KIWI
- V. Manuel d'utilisation du Puxing
- VI. Bibliothèque C embarquée: hammingUtils.c/h
- VII. Diagramme des classes des applications de test DataSender / DataReceiver

Introduction

Notre projet, « Réalisation d'une chaîne de transmission radio pour ballon stratosphérique », s'inclue directement dans le projet global « Ballon » mené par l'IUT de Valence, en collaboration avec l'association Planète-Sciences (dépendante elle-même du CNES). Des étudiants en licence professionnel CASIR et en DUT Informatique travaillent en parallèle sur le ballon afin d'embarquer les composants nécessaires à son instrumentation: module GPS, capteurs de pression, température, ...

La chaîne de transmission que nous devons élaborer entre le ballon et la station au sol, doit permettre la réception et le traitement au sol des paramètres physiques mesurés par le ballon, en respectant les contraintes imposées par le professeur tuteur et le CNES.

Dans un premier temps, un cahier des charges a été établi afin de définir les besoins en fonctions des contraintes fixées en terme de qualités de transmission (rapidité, fiabilité, ...). A partir de ces spécifications, nous avons pu commencer à rechercher différentes solutions et développer par la suite nos prototypes matériels et logiciel. Afin de vérifier le bon fonctionnement de notre transmission, nous avons réalisé plusieurs tests, en fonctionnant par étapes. Différentes erreurs ont pu être corrigées, et des optimisations effectuées, grâce aux résultats de ces tests.

I. Présentation

CAHIER DES CHARGES ET SPÉCIFICATIONS

Avant de débiter notre projet, il a été nécessaire d'établir un cahier des charges complet indiquant les objectifs à atteindre et les contraintes auxquelles nous avons été soumis. Nous avons aussi du penser à la manière d'implémenter la solution proposée, que nous avons détaillée dans les spécifications techniques.

Ce projet prévoit la réalisation de la transmission physique utilisée entre un ballon stratosphérique et une station au sol. Afin de prendre connaissance de l'existant de manière détaillée, et du projet Ballon en général mené par l'IUT de Valence, nous utilisons le site internet réalisé par un étudiant en licence professionnelle SIRE.

<http://rom.buisson.free.fr/ballon/>

Un système avec un modulateur externe et un émetteur KIWI existe déjà. Nous partons de cette base pour construire les autres éléments constituant la nouvelle chaîne (modulateur, émetteur, récepteur VHF, démodulateur).

La transmission doit respecter certains critères imposés par l'environnement mais aussi par le responsable du projet.

- rapidité

Un débit binaire de 4800bits/s en modulation FSK doit-être assuré entre le ballon et la station de réception au sol.

- *fiabilité*

Les données transmises ne doivent pas être corrompues. Les trames envoyées doivent donc permettre de détecter et de corriger les erreurs. La transmission doit-être sûre avec un faible taux d'erreur bit.

- *performance*

Le dispositif de transmission, embarqué dans la nacelle du ballon, ne doit pas perturber les autres éléments électroniques. Il faudra donc surveiller sa consommation électrique et sa dissipation thermique. De plus, il faudra garantir une autonomie supérieure à 3H (durée moyenne d'un vol).

- *réutilisabilité*

Le modulateur doit-être simple d'utilisation et doit-être documenté en indiquant ses différents modes de fonctionnement. De même pour le démodulateur. Par ailleurs, le projet prévoit la programmation d'une plateforme de test pour vérifier le bon fonctionnement du codage Hamming et du modulateur/démodulateur.

- *Adaptabilité*

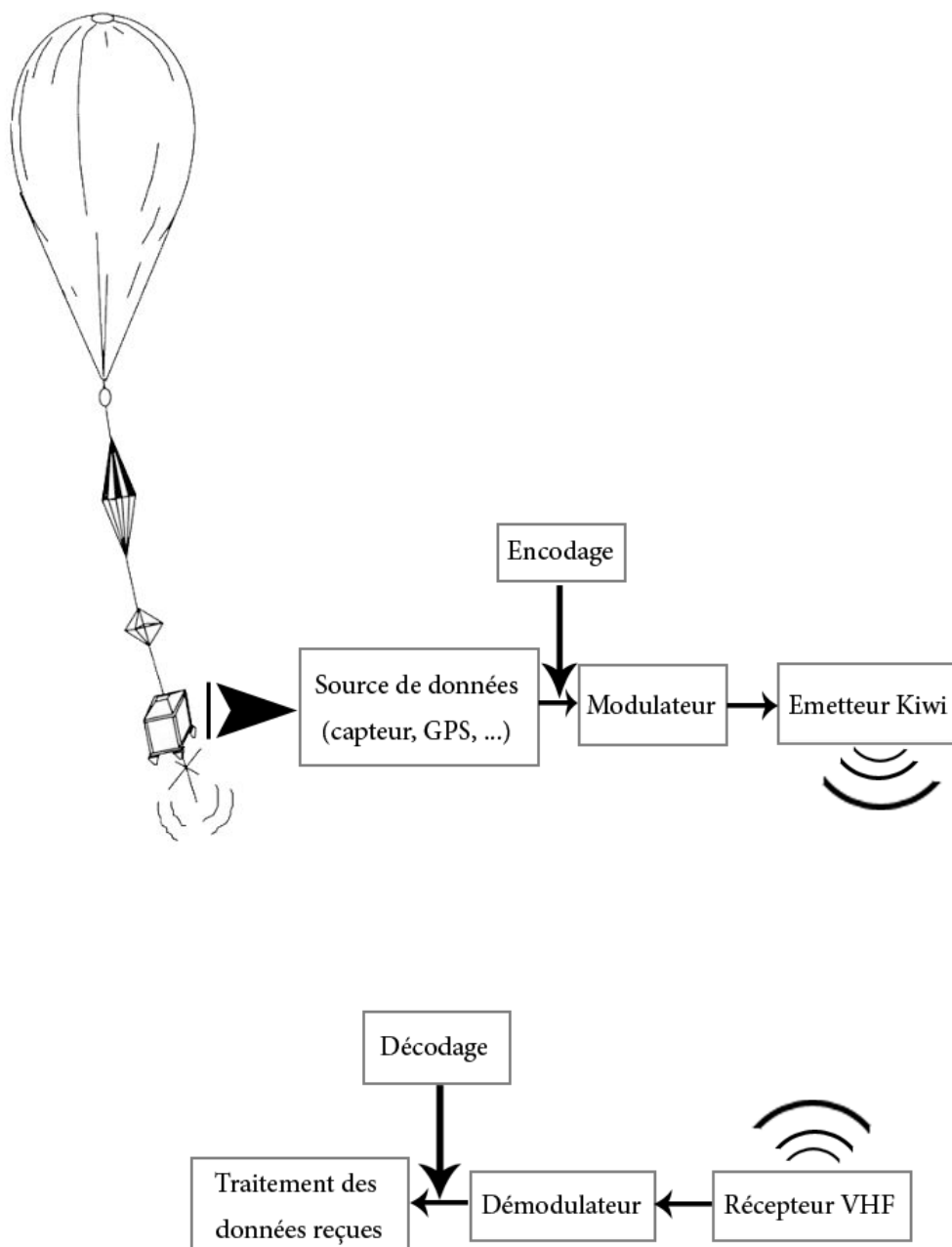
Il est bien sûr convenu que le dispositif doit pouvoir être testé sur une autre plateforme en ayant le maximum de compatibilité (notamment entre les différents systèmes d'exploitation).

Voici la présentation détaillée de la solution technique envisagée pour la réalisation de la chaîne de transmission:

- Mise en œuvre de l'émetteur à 138 MHz (dénommé KIWI) fourni par le CNES.
- Mise au point d'un modulateur FSK permettant la modulation du KIWI (en réutilisant le composant XR2206 déjà utilisé pour le modulateur existant).
- Réalisation d'un démodulateur FSK permettant la démodulation des données numériques reçues (composant XR2211 ~4€). Par la suite, ce démodulateur pourra être logiciel en utilisant les interfaces d'une carte son.
- Réalisation d'une bibliothèque logicielle écrite en C permettant l'encapsulation des données utiles en rajoutant des détections et éventuellement des corrections d'erreurs (code de Hamming).
- Réalisation d'une plateforme, matérielle et logicielle, permettant le test de la qualité de la liaison et, en particulier, permettant la mesure du Taux d'Erreur Bit (TER).

Dans un premier temps, cette plateforme sera réalisée en utilisant deux ordinateurs (interface écrite en JAVA) reliés par un lien série (introduction volontaire d'erreur). Ensuite viendra le test de la transmission sans fil en utilisant un émetteur à 144Mhz (bande libre utilisée par les radioamateurs).

Schéma récapitulatif de l'ensemble:



Le ballon embarque un émetteur Kiwi qui émet le signal que nous avons préalablement modulé. Le signal radio est reçu au sol par un récepteur VHF qui se charge alors de renvoyer le signal analogique au démodulateur.

Cette transmission fait intervenir plusieurs composants tels qu'un modulateur/démodulateur, une bibliothèque C embarquée dans le ballon pour encoder les données, un programme permettant la désencapsulation et le décodage des données ainsi qu'un couple d'émetteur/récepteur VHF.

FREQUENCY SHIFT KEYING

Pourquoi avoir choisi la modulation FSK?

Tout d'abord, la modulation FSK est une modulation de signaux numérique, par opposition à une modulation de signaux analogiques. Or, il est plus aisé pour un démodulateur de reconstituer un signal comportant un nombre fini d'états plutôt qu'un nombre infini.

D'autres avantages non négligeables ont été décisifs dans le choix de cette modulation:

- Étant donné qu'il s'agit d'une modulation de fréquence, elle est peu sensible aux variations d'amplitude induites par les perturbations sur la transmission.
- Il est possible d'éviter la saturation du récepteur, par un système de contrôle de gain, qui va amplifier différemment les signaux de faible amplitude et de forte amplitude.
- La transmission asynchrone est rendue possible car un « 0 » est codé par une fréquence particulière, le différenciant ainsi de l'absence de signal sur la ligne.
- Pour finir, nos débits de transmission sont faibles.

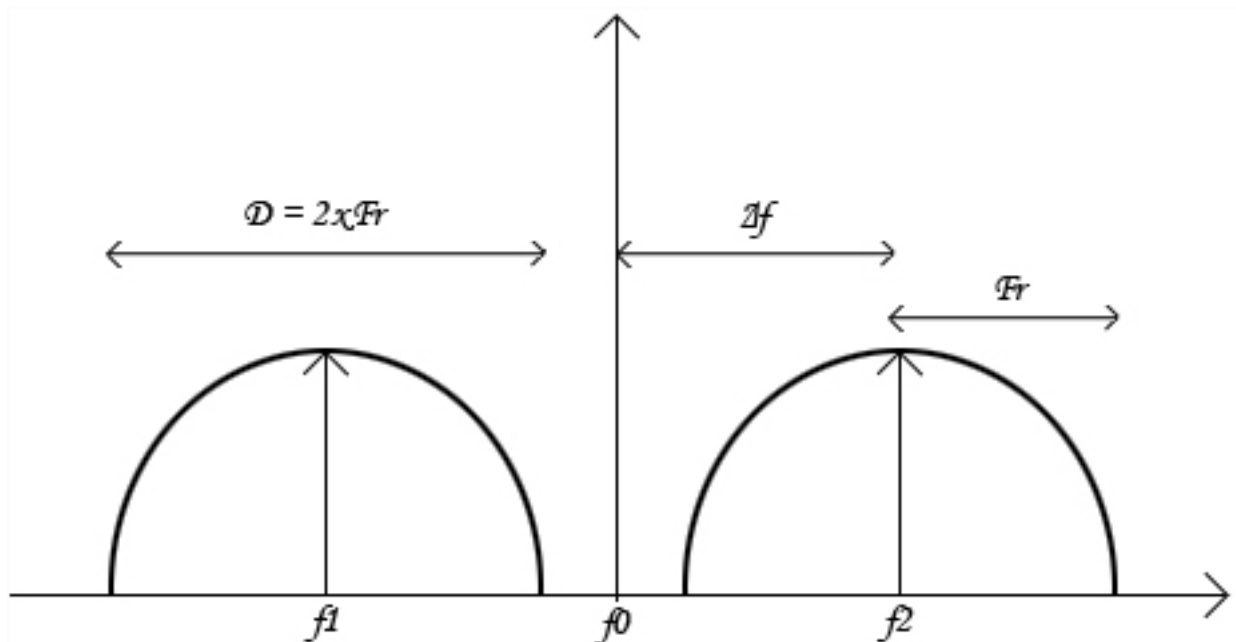
La modulation FSK est donc toute indiquée. Mais avant de la mettre en place de manière concrète, il est nécessaire de bien la comprendre.

Principe de fonctionnement global:

La modulation FSK est une modulation de fréquence dont le spectre dépend principalement de son indice de modulation m . Il est défini par la formule suivante :

$$m = (f_2 - f_1) / 2F_r = \Delta f / F_r \quad \text{avec} \quad \Delta f = (f_2 - f_1) / 2$$

f_1 et f_2 sont les fréquences porteuses de chacun des deux lobes composant le spectre global, centrées autour de f_0 . La largeur d'un lobe est égal au double de la fréquence rythme (F_r).

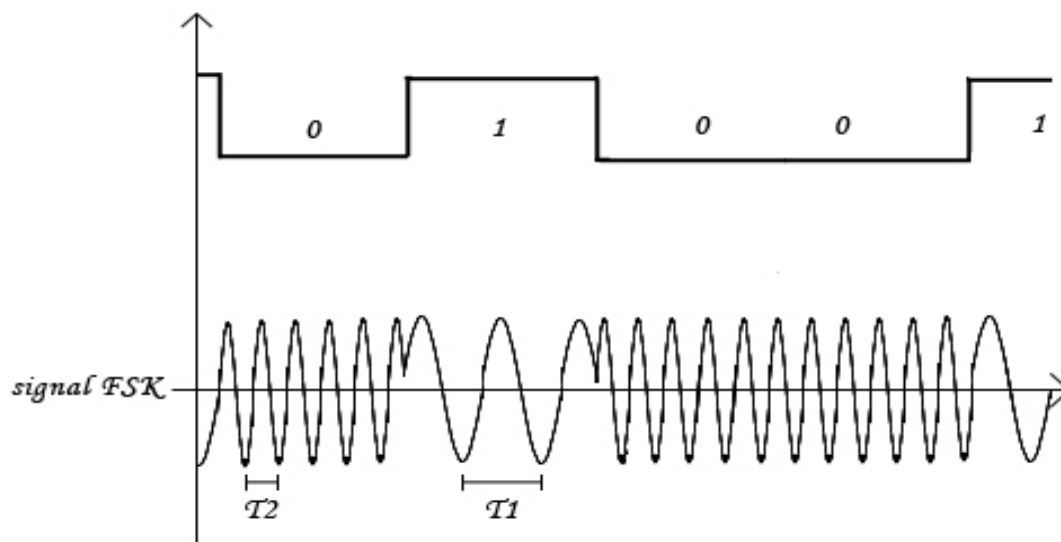


L'indice de modulation détermine donc l'occupation spectrale de la modulation FSK. Il est nécessaire que m soit supérieur à 1 afin d'éviter tout recouvrement de spectre, phénomène dans lequel les lobes viennent se superposer. Cette superposition serait source de bruit, et il deviendrait difficile pour le récepteur de différencier un niveau haut d'un niveau bas.

Nous choisissons volontairement un indice de modulation supérieur à 2 car nous utilisons comme canal de transmission l'air, source naturelle de bruit, pouvant entraîner une distorsion de fréquence, et donc un recouvrement de spectre.

L'occupation spectrale de la modulation FSK (bande passante) est donc, d'après la définition de la bande Carson:

$$BT = 2(\Delta f + Fr)$$



Voici le signal temporel observé lors d'une modulation FSK. Les 0 et les 1 sont codés respectivement par les fréquences $f_1=1/T_1$ et $f_2=1/T_2$.

CODES CORRECTEURS: HAMMING

Afin de pouvoir rendre notre liaison la plus fiable possible, nous avons mis en place, conformément au cahier des charges, un code correcteur d'erreur. Avant de présenter la solution mise en place, il peut-être bon de présenter rapidement ce que sont les codes correcteurs.

Les codes correcteurs d'erreurs, comme leur nom l'indique, sont des mécanismes de codage qui permettent de corriger un message si celui-ci arrive endommagé (par les perturbations électromagnétiques lors d'une transmission dans l'air par exemple). Afin de réaliser cette correction, il est bien sur évident que le message envoyé doit-être modifié en augmentant sa taille (on y inclue des données supplémentaires). Il est alors inclue dans un ensemble plus grand formant les mots du code.

Soit x les données initiales de longueur k bits, C la fonction de codage:

$$\begin{array}{c} C \\ x \rightarrow m \end{array}$$

m est le message final qui sera envoyé et de taille n bits: c'est un mot du code C .

Le code C est tel qu'à chaque message x , on associe un mot de code m : l'ensemble des mots du code C contient alors exactement 2^k éléments. On dit que C est un code de longueur n à 2^k éléments.

En notant T la transmission, on obtient le schéma suivant:

$$T_{m \rightarrow m'}, \text{ et, en notant } e = m' - m = m' + m,$$

$$T_{m \rightarrow m'} = m + e$$

où e représente l'erreur de transmission.

Si le message reçu m' est un mot du code C , alors on considère qu'il n'y a pas eu d'erreur de transmission et on récupère les données x d'origine.

Si le message reçu m' n'est pas un mot du code de C , alors on est sûr qu'il y a eu au moins une erreur et le décodage du message reçu doit nous permettre de retrouver le message d'origine ($m = m' - e$). Pour cela, on utilise la distance de Hamming, qui est la différence de bit entre deux ou mots, noté δ .

Par exemple, la distance de Hamming des deux mots m_1 et m_2 , si $m_1 = 011$ et $m_2 = 100$, est de 3.

Ainsi, suivant la distance de Hamming qui sépare le message reçu m' d'un mot du code m , et si cette distance est plus faible qu'avec un autre mot du code, alors on peut émettre l'hypothèse que le message qui a été envoyé est m (m' est plus proche de m qu'un autre mot du code).

Exemple:

Soit un code C de longueur 3 :

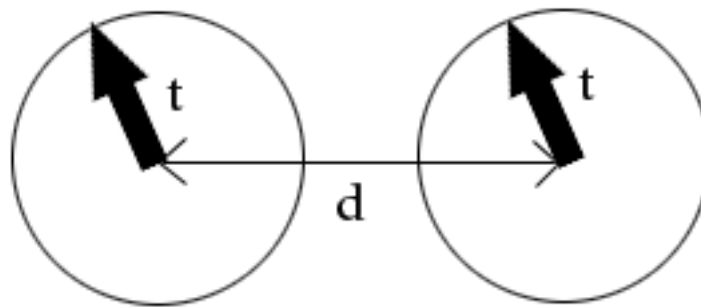
x	m = C(x)
0	101
1	010

Le code C contient deux mots: $m_1 = 101$ et $m_2 = 010$. Si on reçoit $m' = 110$, qui est à une distance de 2 de m_1 et de 1 de m_2 , alors il est plus probable que le message qui a été envoyé soit m_2 (donc 1) !

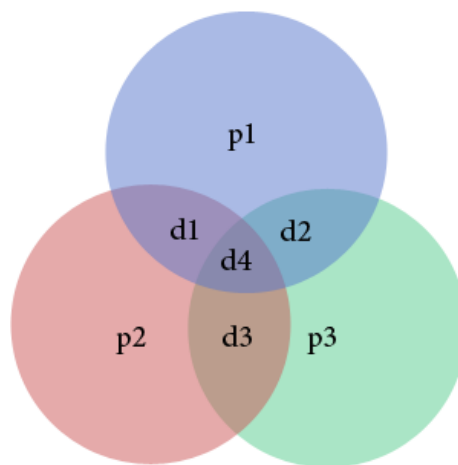
Cette manière de décoder le message reçu (décodage à distance minimum) dépend beaucoup de la distance minimale du code, qui est la distance entre deux mots distincts du code C, noté d . Si la distance qui sépare le message reçu m' d'un mot probable du code est inférieur à d , alors nous avons détecté une erreur ($0 < \delta(m, m') < d$) car m' n'est pas un mot du code. Au maximum, on pourra détecter $d-1$ erreurs. Pour corriger l'erreur, il faut donc choisir m pour que la distance qui le sépare du message reçu, $\delta(m, m')$, soit la plus petite possible. L'algorithme de décodage du code C choisi doit justement se charger de cette tâche.

Le codage de Hamming que nous avons utilisé est un codage binaire de type $[7, 4, 3]$: il transforme chaque message de 4 bits en un mot du code de 7 bits et la distance minimale entre chaque mot du code est de 3. Il permet donc en théorie de détecter 2 erreurs et d'en corriger 1 (c'est un code 1-correcteur): la distance qui sépare deux mots du code est de 3, et le seul élément qui existe pour que $\delta(m, m') \leq t$, est pour $t=1$.

De plus, le codage de Hamming est un code linéaire parfait: les cercles de centre le mot du code et de rayons t ne se touchent pas.



Le codage de Hamming repose sur le calcul de bits de parité suivant le schéma ci-dessous:



$d1$, $d2$, $d3$ et $d4$ représente les bits de données et $p1$, $p2$, $p3$ les bits de parités paire sur ces bits de données:

- $p1 = \text{paritéPaire}(d1, d4, d2)$
- $p2 = \text{paritéPaire}(d1, d4, d3)$
- $p3 = \text{paritéPaire}(d3, d4, d2)$

On construit le mot de code suivant la séquence $p1p2d1p3d2d3d4$.

La propriété de ce codage est telle que lors du décodage, il suffit de vérifier la parité. Si celle-ci est mauvaise, elle nous indique l'emplacement du bit où une erreur est apparue. Il faut donc vérifier, dans le message reçu $d_1d_2d_3d_4d_5d_6d_7$ que les parités sont toujours respectées:

- $r_1 = 0 = \text{paritéPaire}(d_1, d_3, d_5, d_7)$
- $r_2 = 0 = \text{paritéPaire}(d_2, d_3, d_6, d_7)$
- $r_3 = 0 = \text{paritéPaire}(d_4, d_5, d_6, d_7)$

Si il y a une erreur sur un bit, les parités ne sont plus nulles et on sait qu'il y a eu une erreur sur le bit $r_3r_2r_1$!

II. Développement

Pour faire une modulation/démodulation de type FSK, nous avons utilisé deux composants principaux: un XR2206 et un XR2211. Vous trouverez les documentations de ces deux composants dans les annexes.

Le XR2206 implémente un VCO, dont le but est de transformer une tension continue en un signal sinusoïdal dont la fréquence dépend de la tension en entrée.

Le XR2211 permet de réaliser une PLL: à partir d'une fréquence donnée, elle est capable de produire une tension continue.

Nous avons fait deux modulateurs et deux démodulateurs. Le premier modulateur/démodulateur à 1200 bauds permet de tester la modulation avec les Puxing. Son spectre de fréquence est identique à celui de la voix humaine.

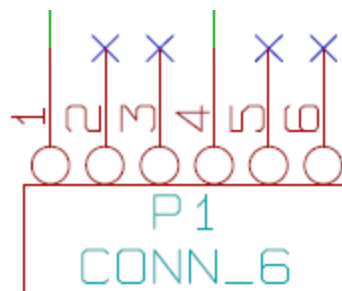
Le second modulateur/démodulateur à 4800 bauds. Il n'est pas adapté pour être testé avec les Puxing mais il est opérationnel avec l'émetteur radio KIWI.

Vous trouverez en annexe I les schématics des modulateurs 1200 et 4800 bauds,

MODULATEUR FSK

Modulateur FSK à 1200 Bauds pour les tests avec les Puxing.

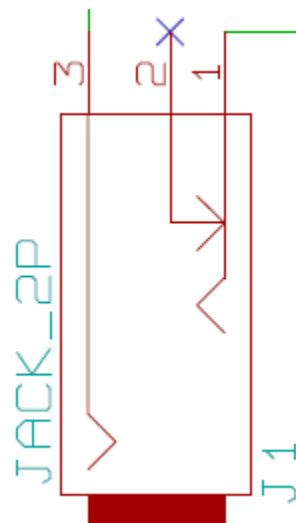
Dans ce schéma nous distinguons plusieurs parties de traitement ainsi que différents connecteurs. Ci-dessous nous allons vous expliquer le fonctionnement de chaque partie de traitement ainsi que les fonctions des connecteurs.



Connecteur de l'embout du câble FTDI (uniquement pour les tests)

Broche 1: Masse

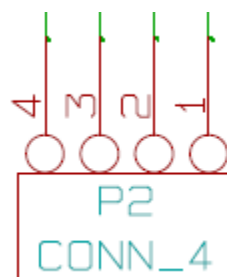
Broche 4: Sortie qui transmet les données en asynchrone du PC vers le modulateur



Prise Jack 3.5mm en sortie du modulateur, vers le Puxing (uniquement pour les tests)

Broche 1: Entrée des données modulées (uniquement pour les tests avec le 1200 bauds)

Broche 3: Masse



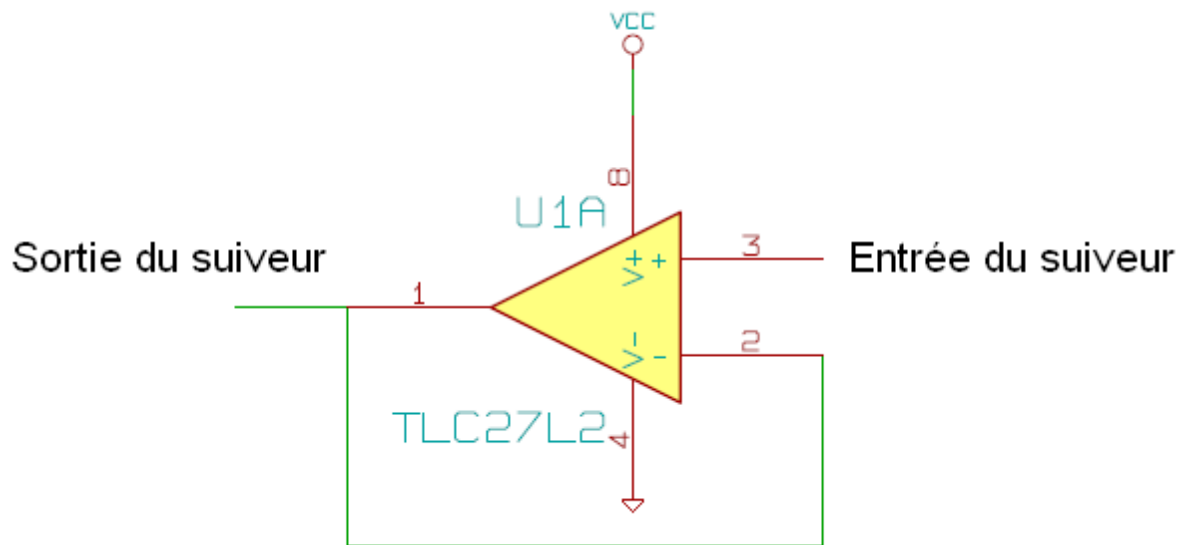
Connecteur pour relier le modulateur à la carte de développement PIC embarquée.

Broche 1(jaune) : Sortie des données

Broche 2(marron) : Masse

Broche 3 (orange): Entrée des données

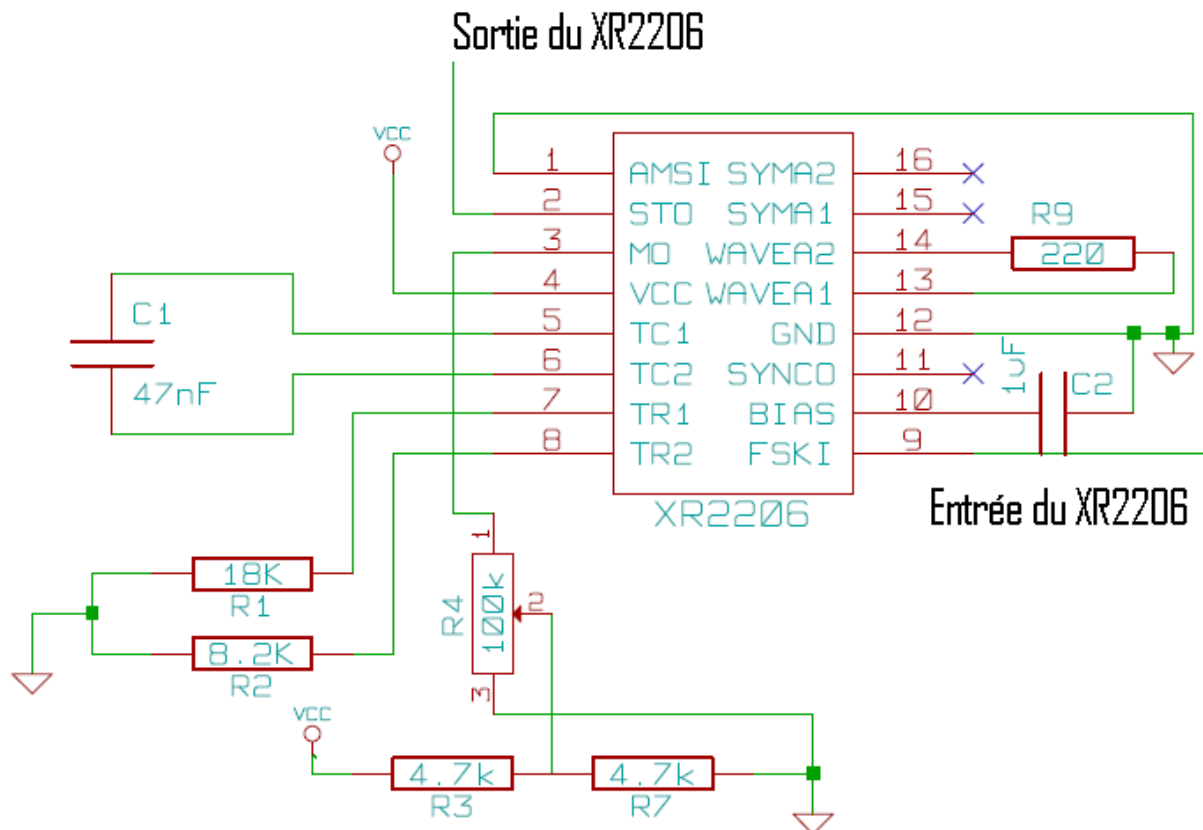
Broche 4 (rouge): Alimentation venant de la batterie embarquée du ballon (9V)



Montage suiveur.

On utilise un montage suiveur pour régler le problème d'impédance entre la connexion USB et le XR2206. Les tensions en sortie du connecteur USB varient entre 0 et 3.3V. Ce sont des tensions continues. Un « 1 » logique a comme tension 3.3V, et un « 0 » logique 0V. La tension en entrée est égale à la tension en sortie.

Schéma de montage du XR2206



Le XR2206 est un VCO. Il convertit des tensions en fréquences, qui varient en fonction de la tension d'entrée.

Tension d'alimentation $V_{cc}=9V$.

Le schéma ci-dessus est fourni par le constructeur du XR2206. Les seuls composants qui peuvent varier sont R1, R2 et C1. Ces composants servent à déterminer les fréquences de modulation F1 et F2. Ces fréquences correspondent respectivement aux niveaux logique « 1 » et « 0 ».

$$F1 = 1 / R1 * C1$$

$$F2 = 1 / R2 * C1$$

Le Puxing possède un filtre sur l'entrée micro ne laissant passer que la voix (300Hz à 3400Hz). Notre bande passante est donc de 3100Hz. La fréquence centrale théorique du spectre est de 1850Hz. Les fréquences F1 et F2 sont respectivement à 1225Hz et 2665Hz. On prend **C1 = 47nF**.

$$1 / 1225 = R1 * C1$$

$$R1 = 17 \text{ k}\Omega = \mathbf{18\text{k}\Omega}$$

$$1 / 2665 = R2 * C1$$

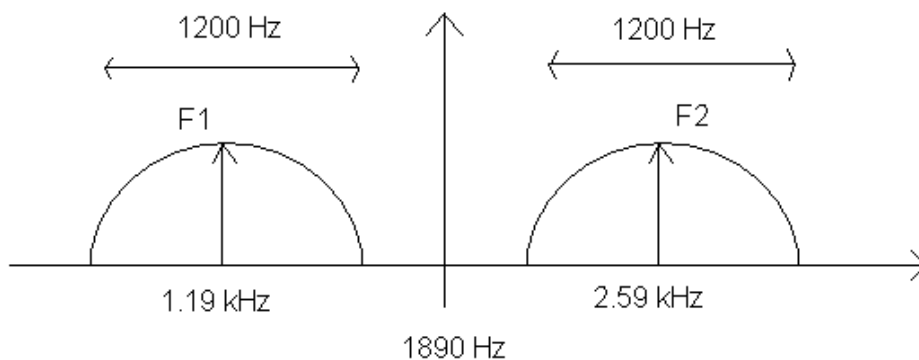
$$R2 = 7.9 \text{ k}\Omega = \mathbf{8.2\text{k}\Omega}$$

Les fréquences F1 et F2 en fonction des composants R1, R2 et C1 obtenus:

$$\mathbf{F1 = 1.19 \text{ kHz}}$$

$$\mathbf{F2 = 2.59 \text{ kHz}}$$

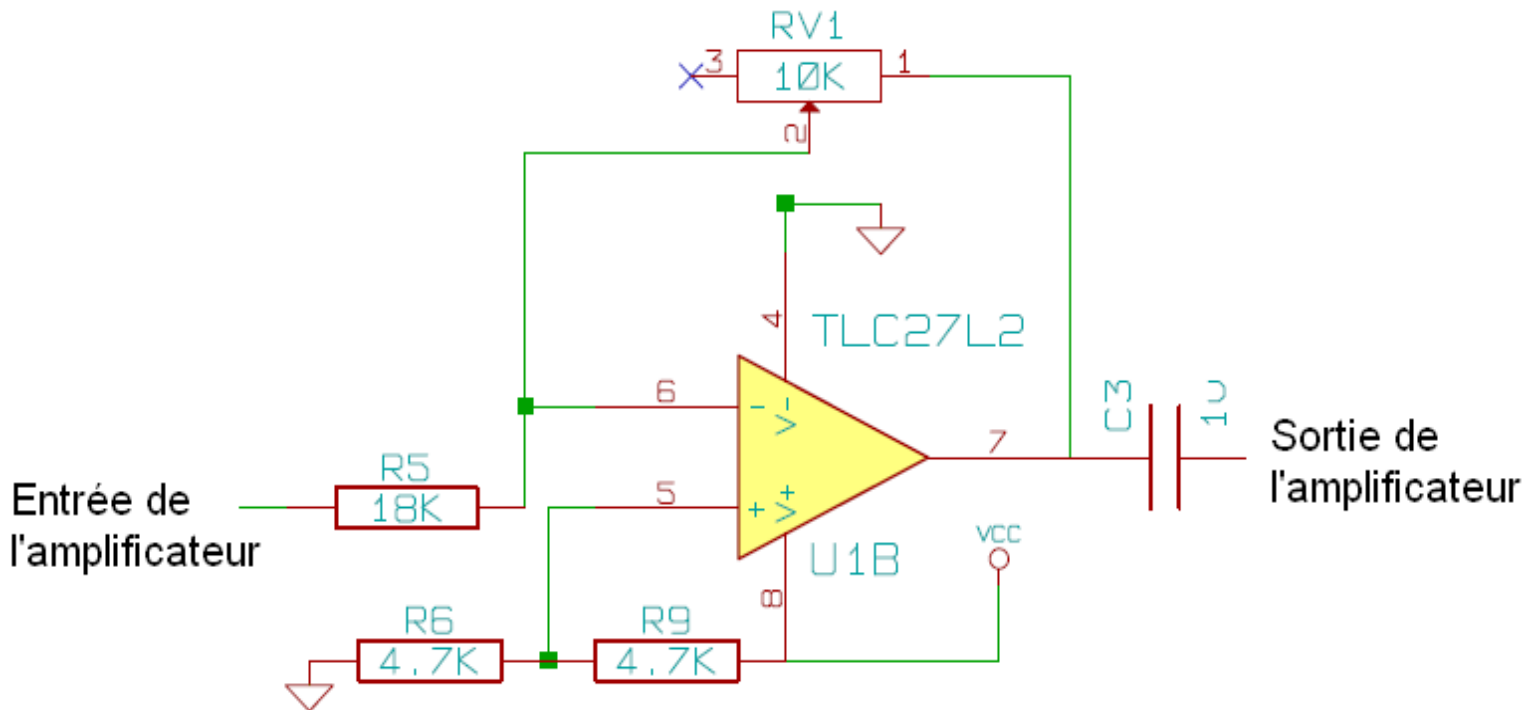
La fréquence centrale du spectre est donc 1.89 kHz.



En sortie du XR2206, les tensions continues sont transformées en tension sinusoïdale variant de 1.5V à 4.5V:

- Quand la fréquence de ce signal est de 1.19kHz, c'est un « 1 » logique
- Quand la fréquence est de 2.59kHz, c'est un « 0 » logique.

Amplificateur Opérationnel Inverseur



La tension d'alimentation V_{cc} est de 9V.

L'entrée micro du Puxing ne peut supporter qu'un signal de faible amplitude variant de +300mV à -300 mV. Il faut donc diminuer la tension du signal en sortie du XR2206 à l'aide d'un amplificateur opérationnel inverseur.

$$V_s = V_e * (-RV1 / R5)$$

Plus la résistance du potentiomètre **RV1** est petite, plus V_s deviendra faible.

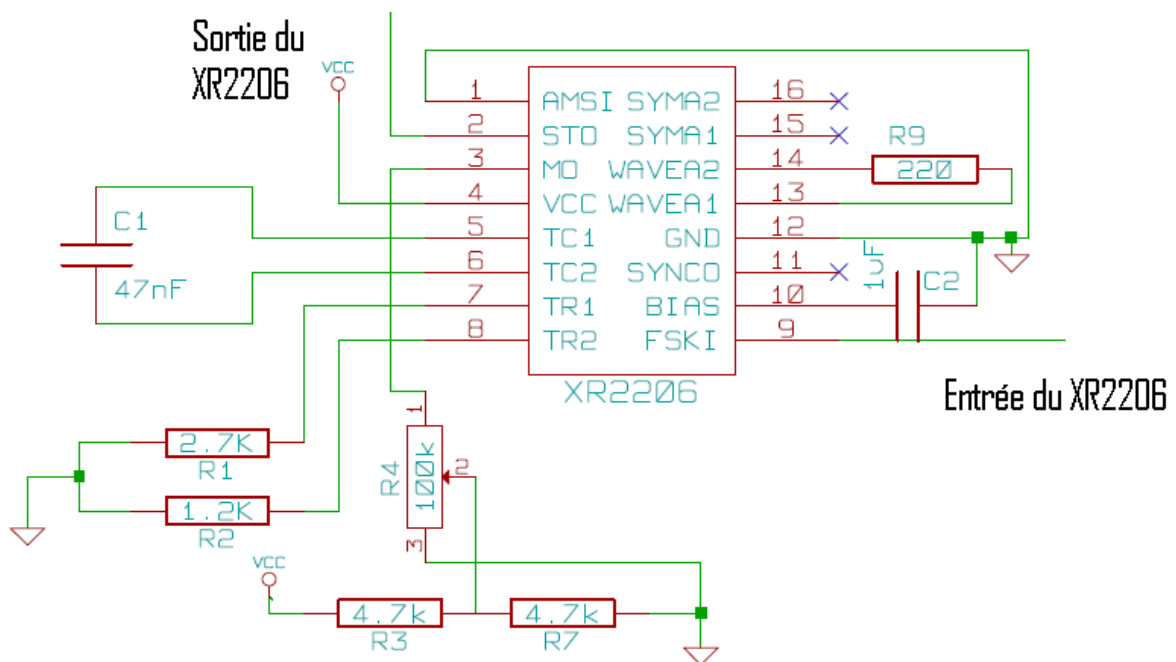
L'amplificateur recentre le signal en entrée autour d'une composante continue. Cette composante continue entre dans la broche 5 de l'amplificateur. Elle est déterminée par les résistances **R6** et **R9**: $V = V_{cc} * R8 / (R8+R6)$. Cependant cette composante est supprimée grâce au condensateur C3.

En sortie de ce montage il ne nous reste qu'une tension sinusoïdale de très faible amplitude qui peut alors rentrer dans l'entrée micro du Puxing.

Modulateur FSK à 4800 bauds

La connectique est similaire à celle du modulateur à 1200 Bauds.

Pour la modulation en 4800 bauds, nous avons pris comme fréquences $F1 = 7.8 \text{ kHz}$ et $F2 = 17.4 \text{ kHz}$.



Les calculs sont exactement identiques au modulateur à 1200 bauds, sauf que les fréquences $F1$ et $F2$ ont changé.

$$F1 = 1 / R1 * C1$$

$$R1 = 2.7 \text{ k}\Omega$$

$$F2 = 1 / R2 * C1$$

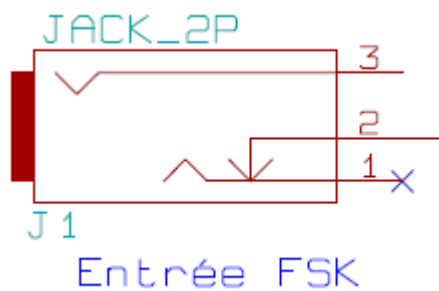
$$R2 = 1.2 \text{ k}\Omega.$$

On garde $C1 = 47 \text{ nF}$.

Le modulateur peut fonctionner à 4800 bauds, et aux vitesses de transmissions inférieur.

DÉMODULATEUR FSK

Démodulateur FSK à 1200 Bauds pour les tests avec les Puxing.

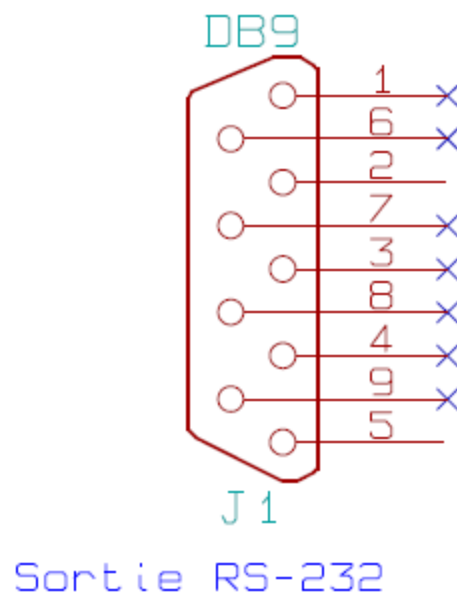


Ceci est l'entrée Jack 2,5mm. Le signal vient de la sortie haut-parleur du Puxing.

Broche 2: Entrée du signal à démoduler (uniquement pour les tests à 1200 bauds)

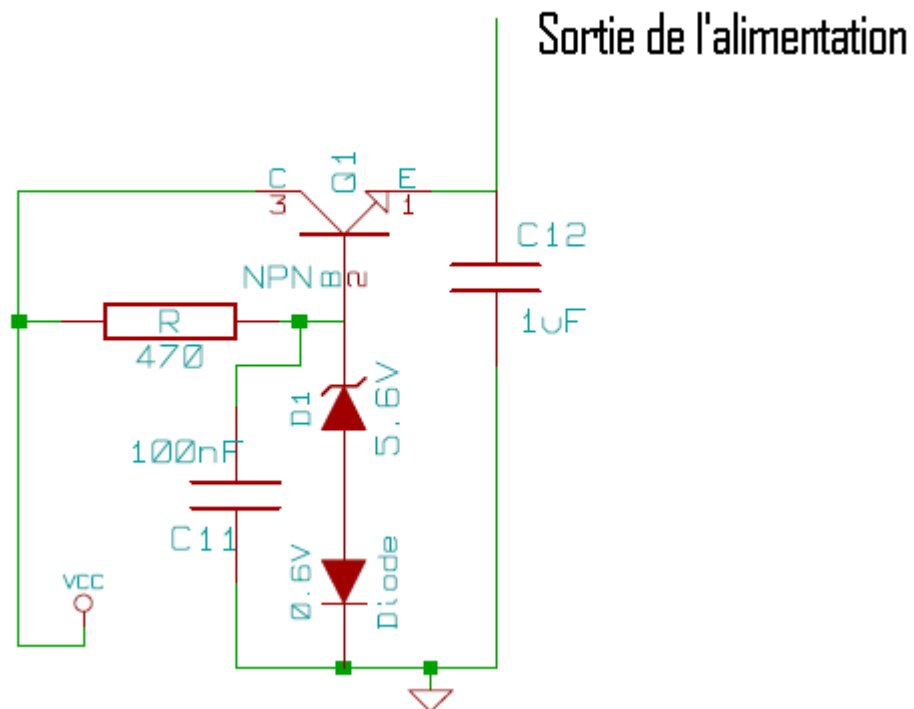
Broche 3: Masse

Connecteur femelle DB9.

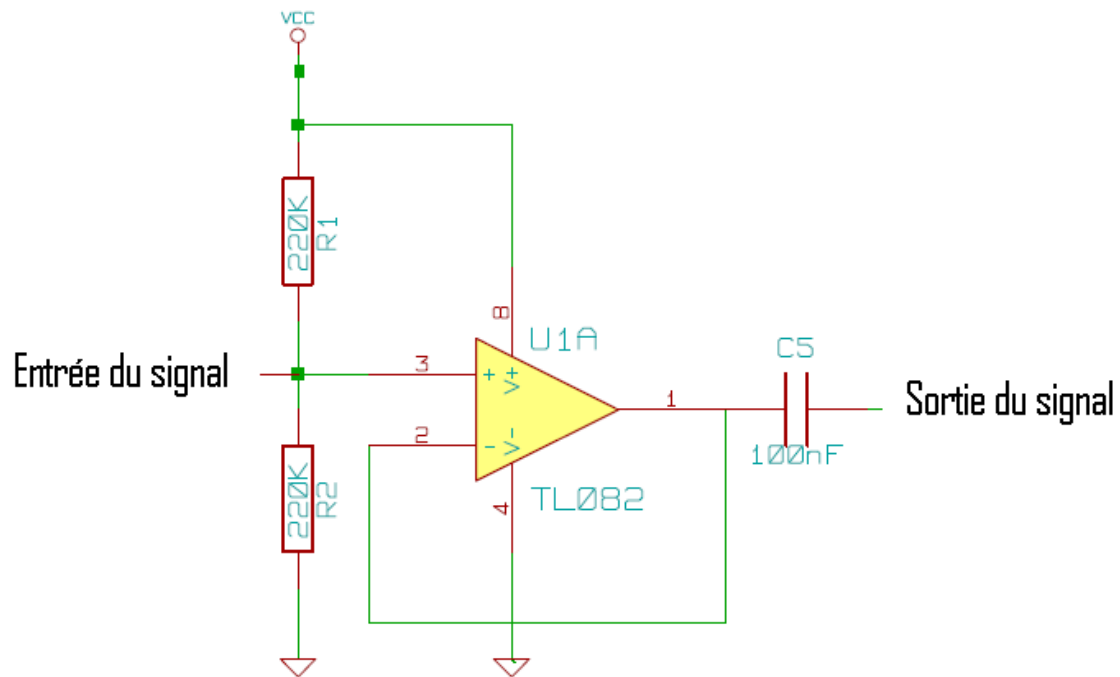


Broche 2: Entrée du signal démodulé vers le PC

Broche 5: Masse



On a une tension d'alimentation $V_{cc}=9V$. Pour alimenter le MAX232 il faut une tension de 5V. Cette partie du montage sert à transformer une tension continue de 9V en une tension continue de 5V.

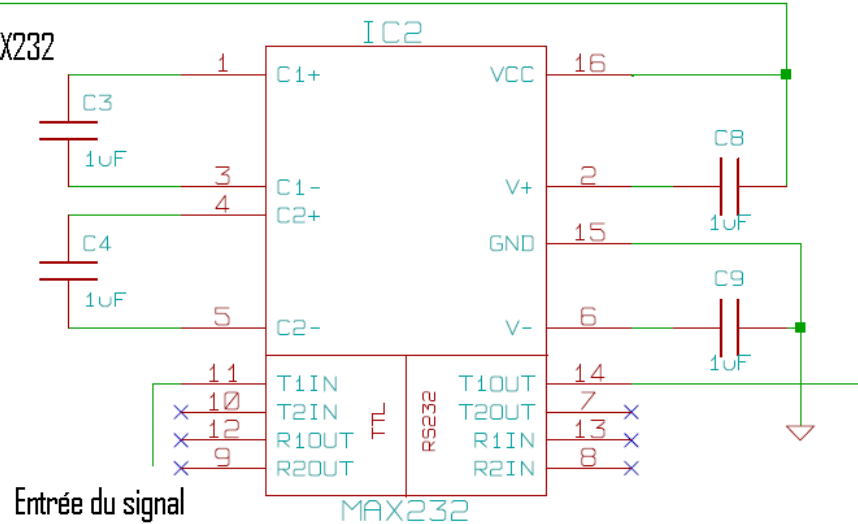


L'alimentation Vcc de l'amplificateur est de 9V.

Cet amplificateur suiveur recentre le signal avant de le faire entrer dans le XR2211. Ce schéma était fourni avec le modulateur 1200 bauds.

Sortie du signal

Alimentation du MAX232

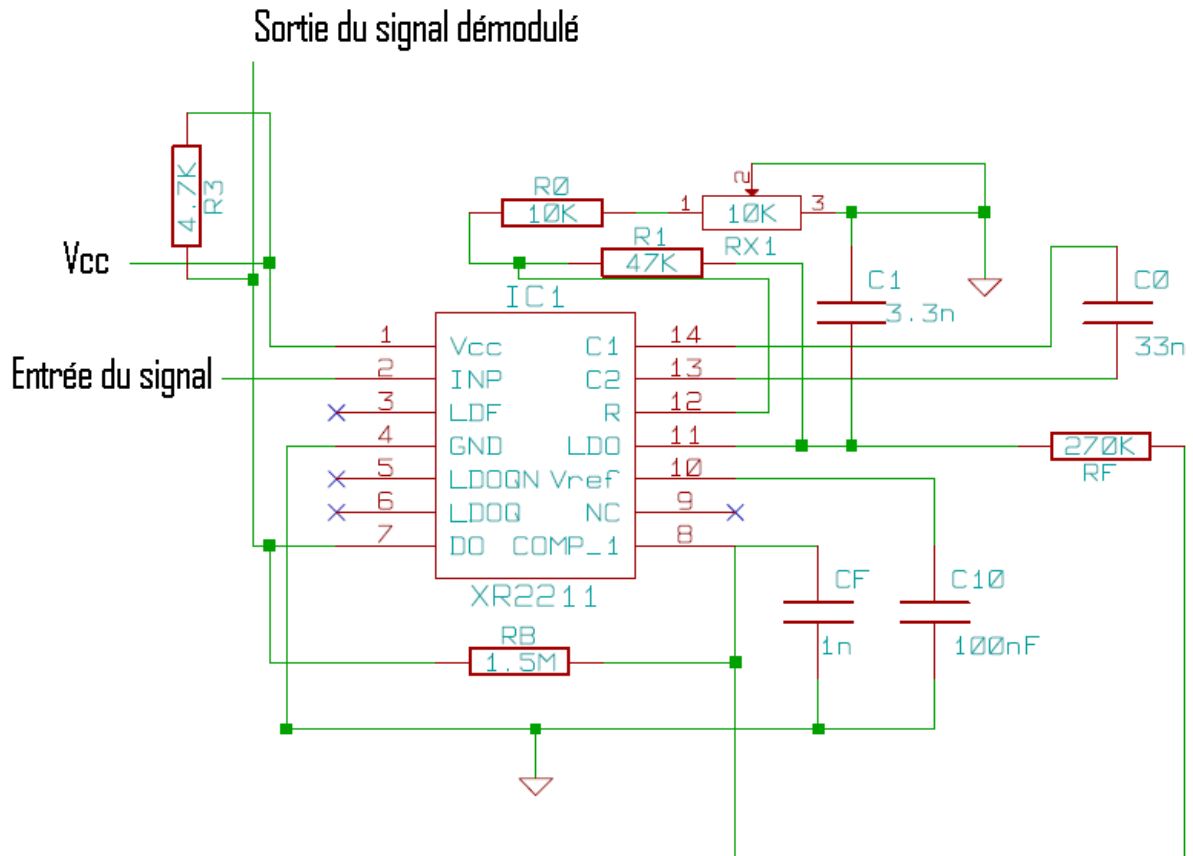


L'alimentation du MAX232 est de 5V.

Le MAX232 sert à refaire les niveaux (en +12V -12V) respecter la norme RS232 des liens séries. Le schéma du montage est fourni par le constructeur du MAX232.

L'alimentation Vcc est de 9V.

Le schéma du montage est fourni par le constructeur. Ci-dessous nous avons fait les calculs des composants pour une démodulation à 1200 bauds.



$$f_0 = \sqrt{(f_1 * f_2)} = \sqrt{(2590 * 1190)} = 1755\text{Hz}$$

$$R_0 = 10000 + 10000/2 = 15\text{k}\Omega$$

Le potentiomètre **RX1** sert à ajuster la résistance globale **R0**.

$$C_0 = 1 / (R_0 * f_0) = 1 / (15000 * 1755) = 39\text{nF}$$

$$\text{On prend } C_0 = 33\text{nF} \text{ donc } R_0 = 1 / (1755 * 33 * 10^{-9}) = 17\text{ k}\Omega$$

$$R_1 = (R_0 * F_0) * 2 / (F_1 - F_2) = (20000 * 1755) * 2 / (2590 - 1190) = 50\text{ k}\Omega$$

$$\text{On utilisera } R_1 = 47\text{k}\Omega$$

$$C1 = (1250 * C0) / (R1 * \zeta^2) = (1250 * 33 * 10^{-9}) / (50000 * 0,5^2) = 32 \text{ nF}$$

On prend **C1 = 33 nF**

$$R_F = R1 * 5 = 51000 * 5 = 255 \text{ k}\Omega$$

On prend **R_F = 270 kΩ**

$$R_B = R_F * 5 = 270000 * 5 = 1,35 \text{ M}\Omega$$

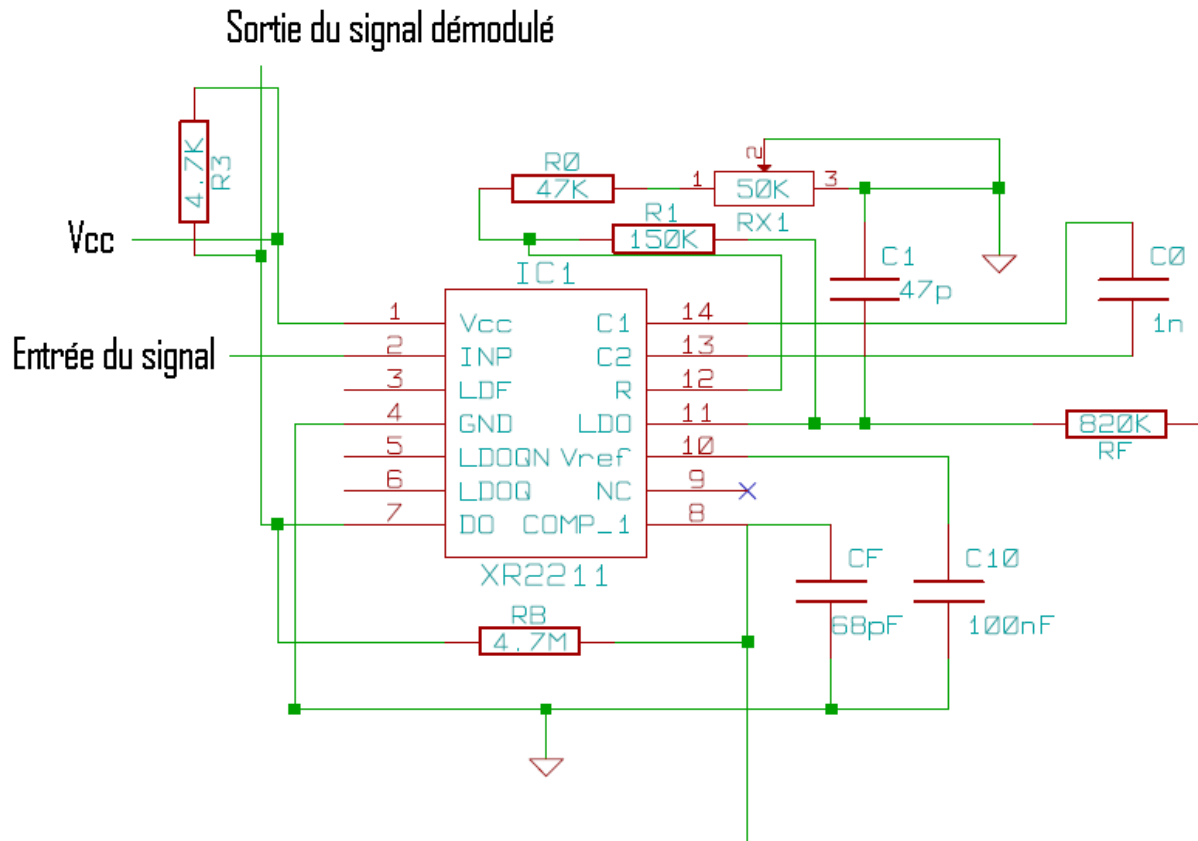
On prend **R_B = 1,5 MΩ**

$$R_{\text{sum}} = (R_F + R1) * R_B / (R_F + R_B + R1) = (270 + 51) * 1500 / (270 + 15 + 1500) = 240 \text{ k}\Omega$$

$$C_F = 0.25 / (R_{\text{sum}} * \text{Baud Rate}) = 0.25 / (240\,000 * 1200) = 0.9 \text{ nF}$$

On prend **C_F = 1 nF.**

Démodulateur FSK à 4800 Bauds



L'alimentation Vcc est de 9V.

C'est le même schéma que pour le démodulateur 1200 bauds. Ci-dessous nous avons fait les calculs des composants pour une démodulation en 4800 bauds. Les résultats sont théoriques, et après manipulations, il s'avère que les valeurs du schéma conviennent mieux à notre démodulateur.

Cependant, le démodulateur n'était plus en état de marche à la fin du projet. il faut donc reprendre le schéma en s'appuyant sur les calculs ci-dessous et la documentation du XR2211.

$$f_0 = \sqrt{(f_1 * f_2)} = \sqrt{(7800 * 17400)} = 11649 \text{ Hz}$$

$$R0 = 84000 + 10000/2 = 89 \text{ k}\Omega$$

Le potentiomètre **RX1** sert à ajuster la résistance globale R0.

$$C0 = 1 / (R0 * f0) = 1 / (89000 * 11649) = 964\text{pF}$$

On prend **C0 = 1nF** donc **R0 = 1 / (11649 * 1*10⁻⁹) = 86 kΩ**

$$R1 = (R0 * F0) * 2 / (F1 - F2) = (94000 * 11649) * 2 / (17400 - 7800) = 228.27 \text{ k}\Omega$$

On prend **R1 = 220 kΩ**

$$C1 = (1250 * C0) / (R1 * \zeta^2) = (1250 * 1*10^{-9}) / (22000 * 0,5^2) = 22\text{pF}$$

On prend **C1 = 22pF**

$$RF = R1 * 5 = 220000 * 5 = 1.1 \text{ M}\Omega$$

On prend **RF= 1.5 MΩ**

$$RB = RF * 5 = 1500000 * 5 = 7.5 \text{ M}\Omega$$

On prend **RB = 6.8 MΩ**

$$Rsum = (RF + R1) * RB / (RF + RB + R1) = (1500 + 220) * 6800 / (1500 + 6800 + 220) = 1372 \text{ k}\Omega$$

$$CF = 0.25 / (Rsum * \text{Baud Rate}) = 0.25 / (1372000 * 4800) = 37\text{pF}.$$

On prend **CF = 33pF**

LOGICIEL

Afin de pouvoir utiliser le codage de Hamming [7, 4, 3], nous avons dû développer une bibliothèque C embarquée dans le ballon qui encapsule les données à émettre, et une application au sol pour décoder les données. Le code source de la bibliothèque (hammingUtils.c et hammingUtils.h) est fourni en annexe VI, avec un exemple d'utilisation simple. L'application permettant le décodage n'est pas présentée ici, uniquement l'algorithme le sera.

Aussi, notre transmission étant orientée caractère (transmission de paquet de 8bits), il nous restait un bit que nous n'utilisons pas (puisque le codage Hamming transforme 4 bit de données en un mot de 7 bits). Pour éviter une désynchronisation au niveau du récepteur, nous avons donc décidé d'utiliser ce bit comme bit de poids pour signaler un quartet de poids fort (1) ou de poids faible (0).

L'algorithme de codage et de décodage étant uniquement basé sur le calcul de bit de parité, il n'a pas fallu beaucoup de temps pour réaliser un premier exemple de codage en python. Par la suite, une fois le principe bien cerné, nous sommes passés au développement en langage C.

Voici comment l'algorithme expliquant le fonctionnement du codage tel qu'implémenté dans la bibliothèque C embarquée (sauf qu'au lieu d'afficher les résultats, ils sont contenus dans une nouvelle chaîne de caractères qui est retournée). La syntaxe `quartet[0]` représente le bit 0 de l'octet quartet !

{Fonction de calcul de la parité paire sur 3 bits, assimilé à des octets du type 0000 000x}

Fonction paritéPaire3(b1, b2, b3: octet): octet

DEBUT:

 {Sommes des bits modulo de 2}

SI ((b1+b2+b3)%2 = 0)

ALORS

 Retourne 0

SINON

 Retourne 1

FINSI

FIN

{Fonction de calcul de la parité paire sur 4 bits, assimilé à des octets du type 0000 000x}

Fonction paritéPaire4(b1, b2, b3, b4: octet): octet

DEBUT:

 {Sommes des bits modulo de 2}

SI ((b1+b2+b3+b4)%2 = 0)

ALORS

 Retourne 0

SINON

 Retourne 1

FINSI

FIN

{Algorithme de codage de données utilisant le codage Hamming
[7, 4, 3]}

VARIABLES:

donnéesSource: chaîne de caractères
bp, p1, p2, p3: octets {bits}
b1, b2, b3, b4: octets {bits}

DEBUT:

```
{Parcours de la chaîne de caractère}
POUR CHAQUE caractère DANS donnéesSource:
    bp <- 1 {quartet de poids fort en premier}

    {Parcours du caractère pour traiter les deux
quartets (octet du type 0000 xxxx)}
    POUR CHAQUE quartet DANS caractère:

        {On prend chaque bit du quartet}
        b1 <- quartet[0] {bit 1}
        b2 <- quartet[1] {bit 2}
        b2 <- quartet[2] {bit 3}
        b3 <- quartet[3] {bit 4}

        {Calcul de la parité paire}
        p1 <- paritéPaire3(b1, b2, b4)
        p2 <- paritéPaire3(b1, b3, b4)
        p3 <- paritéPaire3(b2, b3, b4)

        {Construction du nouvel octet et affichage}
        Ecrire(bp, p1, p2, b1, p3, b2, b3, b4)

        {On reboucle sur le quartet de poids faible}
        bp <- 0
    FINPOUR
FINPOUR
FIN
```

L'algorithme de décodage est basé exactement sur le même principe en vérifiant la parité.

{Algorithme de décodage d'octets utilisant le codage Hamming [7, 4, 3]}

Fonction decode(caractèresSource: tableau de 2 octet): octet

DEBUT:

bp <- 1

POUR CHAQUE octet DANS caractèresSources

{Recupération de la valeur des bits}

p1 <- octet[1]

p2 <- octet[2]

b1 <- octet[3]

p3 <- octet[4]

b2 <- octet[5]

b3 <- octet[6]

b4 <- octet[7]

{Vérification de la parité}

p1 <- paritéPaire4(p1, b1, b2, b4)

p2 <- paritéPaire4(p2, b1, b3, b4)

p3 <- paritéPaire4(p3, b2, b3, b4)

{Calcul du bit sur lequel il y a eu une erreur}

errorBit <- p3 * 4 + p2 * 2 + p1

SI errorBit != 0

ALORS

{Erreur détectée sur le bit errorBit}

{Correction de l'erreur sur le bit où il y a eu une erreur avec un OU-Exclusif...}

FINSI

{Construction du nouvel octet}

SI bp = 1

ALORS

oRecu <- b1*128 + b2*64 + b3*32 + b4*16

bp <- 0

SINON

oRecu <- b1*8 + b2*4 + b3*2 + b4

FINSI

FINPOUR

Retourne oRecu

FIN

Le premier bit de poids nous permet d'éviter un décalage qui pourrait survenir par la perte d'un paquet. Il n'est malheureusement pas possible de corriger une erreur sur ce bit, à moins de réfléchir par rapport à la suite de bit reçu.

{Algorithme vérifiant la synchronisation des octets reçus.
Lire() permet de récupérer un octet depuis le flux d'entrée.}

VARIABLES:

data: tableau de 2 octets
oRecu: octets

DEBUT:

{Récupération de deux octets depuis le flux d'entrée}
data[0] <- Lire()
data[1] <- Lire()

TANT QUE vrai

{Vérification des bits de poids}

SI data[0][0]=1 **ET** data[1][0]=0

ALORS {Aucune erreur}

Ecrire(decode(data))

data[0] <- Lire()

data[1] <- Lire()

SINON {Perte d'un paquet ?}

oRecu <- Lire()

SI data[0][0]=1 **ET** data[1][0]=1

ALORS

SI oRecu[0]=0

ALORS {Perte de paquet !}

data[0] <- data[1]

data[1] <- oRecu

SINON {Erreur sur le bit de poids

précédent}

Ecrire(decode(data))

data[0] <- oRecu

data[1] <- Lire()

FINSI

SINON **SI** data[0][0]=0 **ET** data[1][0]=1

ALORS {Perte de paquet !}

data[0] <- data[1]

data[1] <- oRecu

```

    SINON SI data[0][0]=0 ET data[1][0]=0
    ALORS
        SI oRecu[0]=0
        ALORS {Perte ou erreur ?}
            data[0] <- data[1]
            data[1] <- oRecu
            Ecrire(decode(data))
        SINON {Erreur sur le bit de poids}
            Ecrire(decode(data))
            data[0] <- oRecu
            data[1] <- Lire()
        FINSI
    FINSI
FINSI
FINTANT QUE
FIN
```

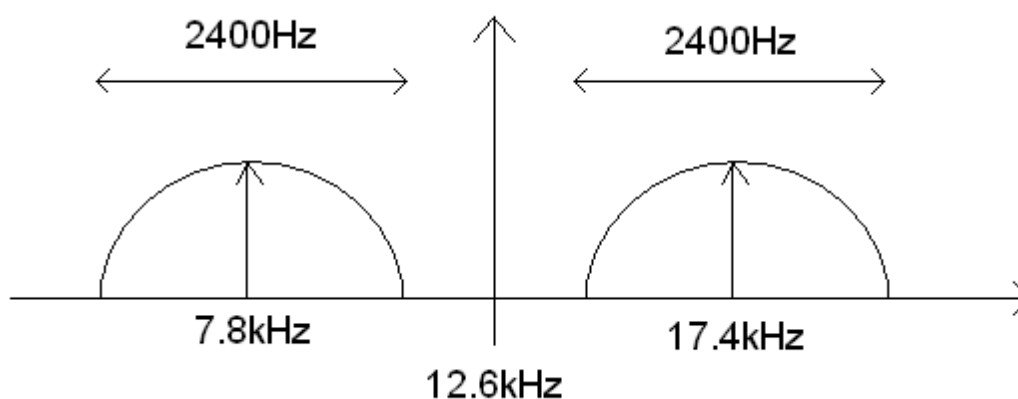
Cet algorithme plutôt complexe permet de gérer la totalité des suites d'octets reçus. Sa mise en œuvre n'est pas compliquée mais, par peur d'obtenir de mauvais résultats, il n'a jamais été testé car il nous est impossible de détecter une erreur sur le premier bit de poids !

III) Résultats des tests

Après avoir réalisé les montages électronique des modulateurs et démodulateurs, nous sommes passés aux tests sur des prototypes de modulateur/démodulateur à 4800bauds. Le modulateur/démodulateur à 1200bauds est venu par la suite pour s'adapter aux tests avec les Puxing.

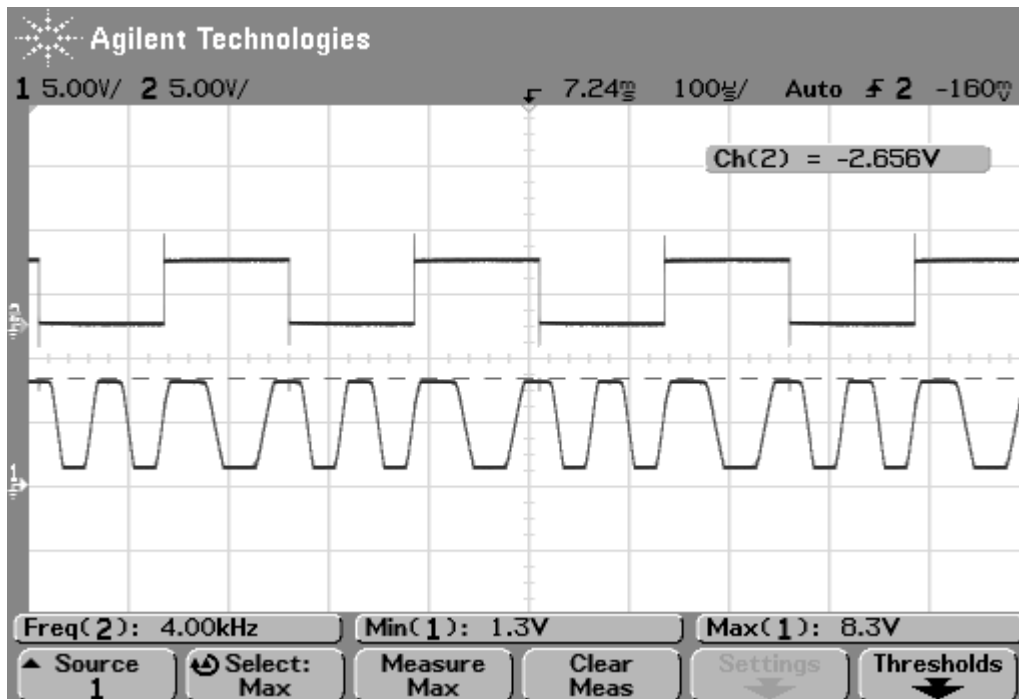
MODULATEUR 4800 BAUDS

Le premier modulateur réalisé. Les fréquences f_1 et f_2 qui modulent un « 1 » et un « 0 » sont 7.8kHz et 17.4kHz. Ces fréquences ont été choisies de sorte qu'à une vitesse de 9600 bauds, nous obtenions bien une période par niveaux; les résistances utilisées ont ensuite fixés ces valeurs. Ci-dessous le spectre de fréquences autour de $f_0 = 12.6\text{kHz}$.

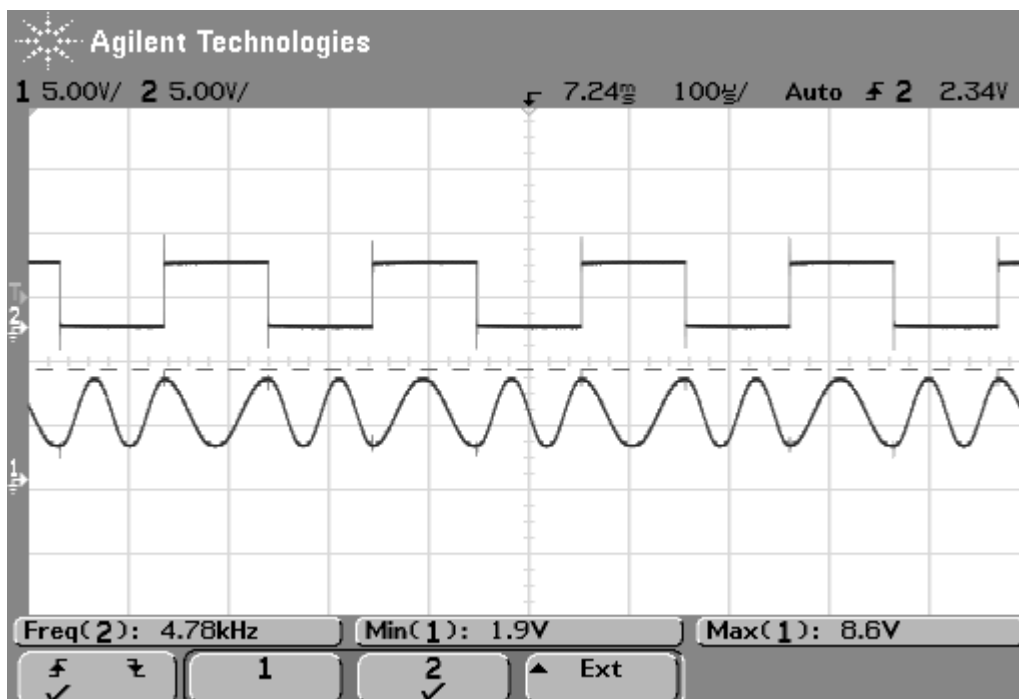


Le modulateur supporte des vitesses de modulation de 4800/2400/1200/600 bauds. Les tests réalisés sont très concluants sur le montage. Le seul souci est venu du potentiomètre R4 qui permet d'affiner le signal sinusoïdal de sortie.

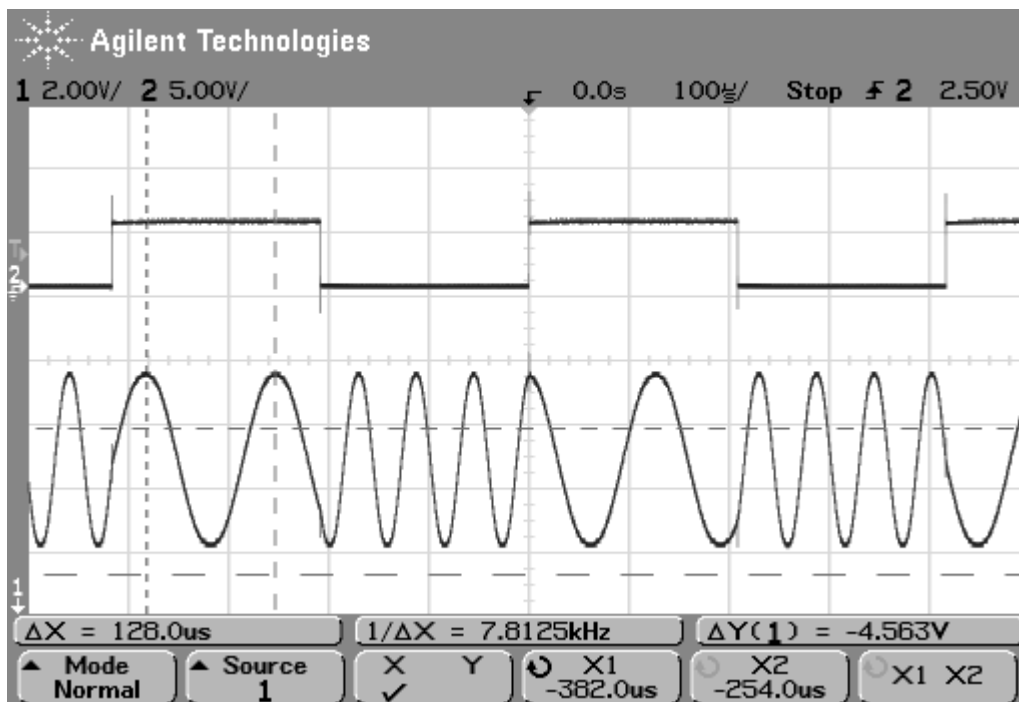
En réglant le potentiomètre sur 90K Ω :



En réglant le potentiomètre sur 36K Ω :

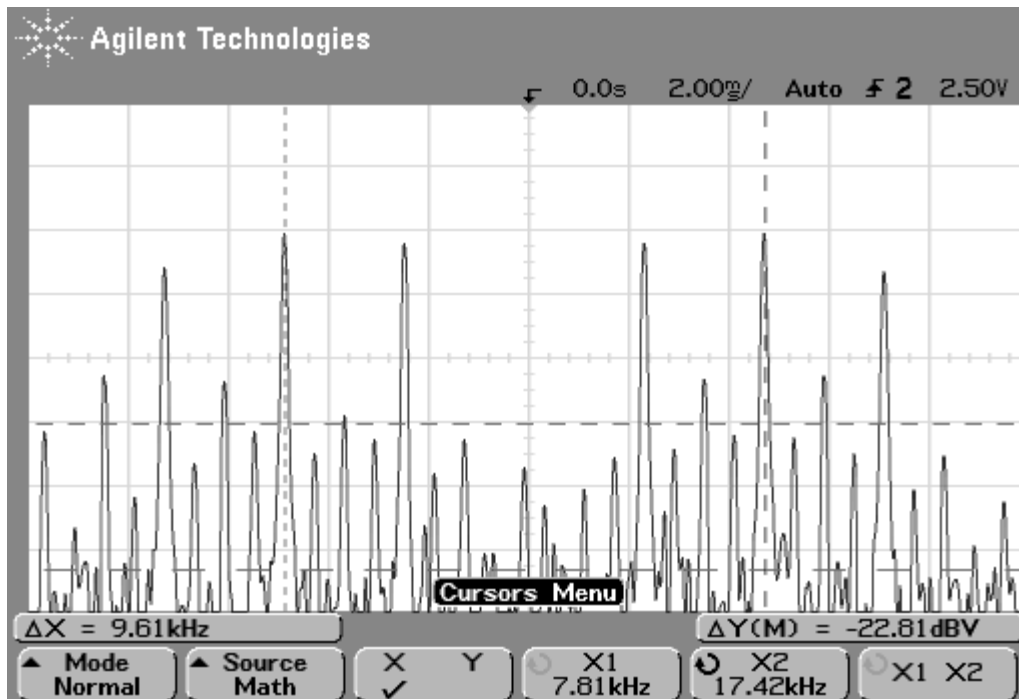


Ci-après, un schéma montrant le signal en sortie du modulateur (voie 1) en faisant entrée un signal carré à 2400Hz (soit 4800bauds).



Nous observons donc que notre modulation est parfaite. Les « 1 » et les « 0 » sont modulés correctement. La composante continue du signal est toujours présente car nous n'avons pas encore ajouté l'AOP inverseur et la capacité en sortie.

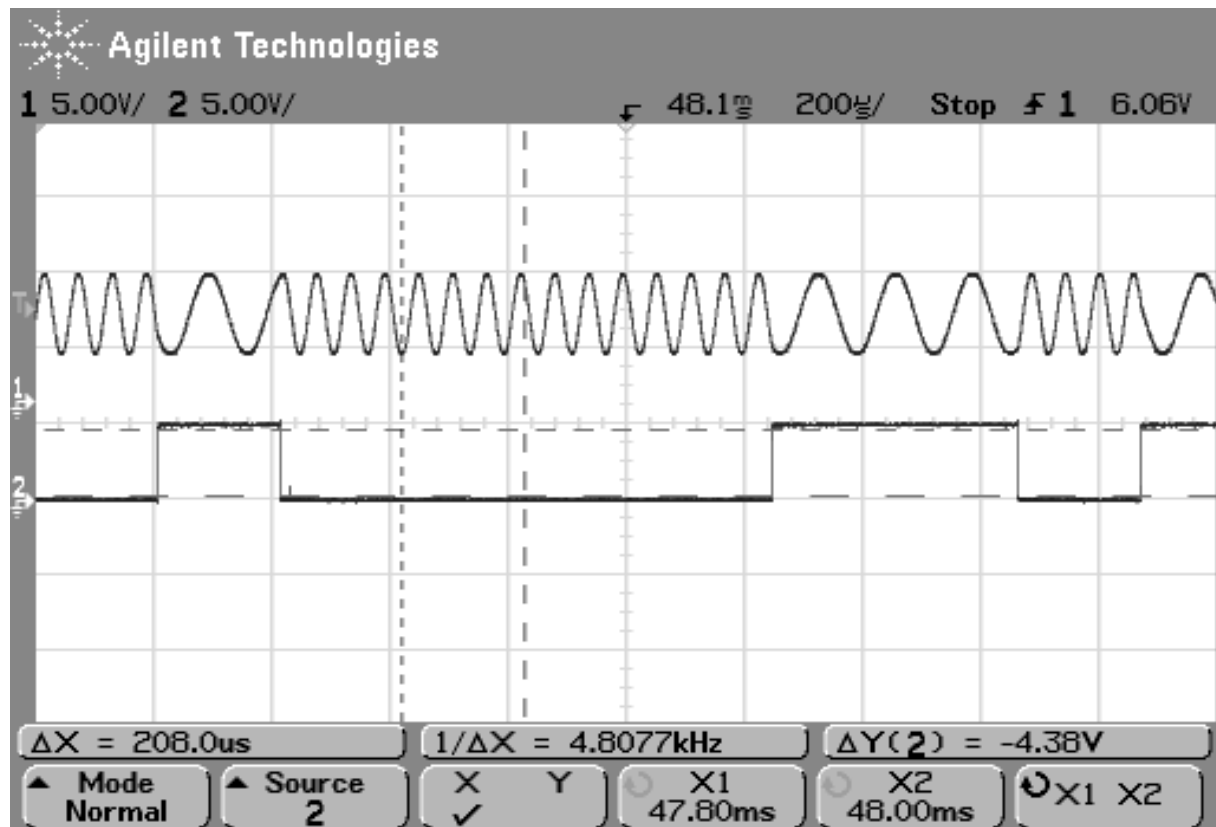
Voici la FFT du signal en sortie du modulateur:



Les deux lobes sont extrêmement bien discernables. On en attendait pas moins avec un indice de modulation de $2 = (17400 - 7800) / 4800$.

Nous avons ensuite voulu tester notre modulateur en émettant un caractère depuis le PC.

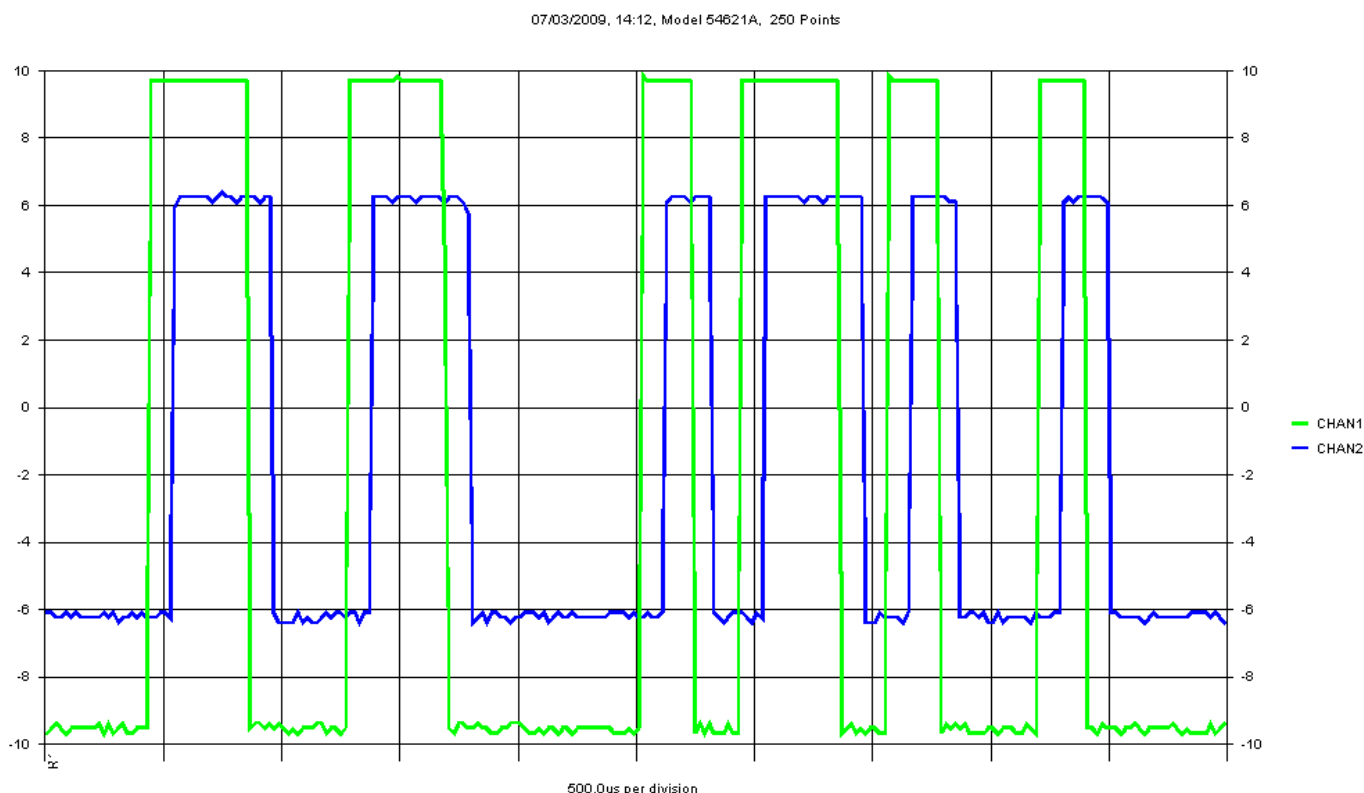
Test en transmettant le caractère « a »: 0110 0001



Notre modulateur n'a aucun mal et se charge très bien de sa fonction. Le problème vient maintenant du démodulateur qui n'était pas très fonctionnel...

DEMODULATEUR 4800 BAUDS

Notre démodulateur fonctionne sur des fréquences de 6.7KHz à 18.8KHz avec une fréquence centrale autour de 11.3KHz et 12.6KHz. Voici un exemple de démodulation que nous avons réussi à obtenir en transmettant un « a », mais en incluant cette fois-ci la correction d'erreur (1110 0110 0110 1001).



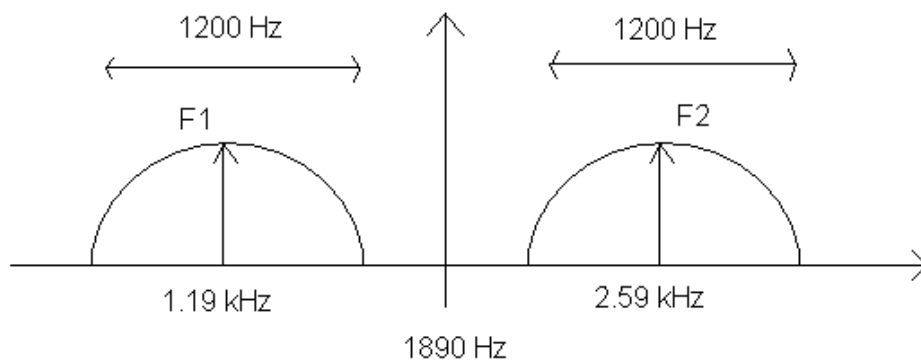
Le signal en sortie du démodulateur (bleu) est presque identique au signal en entrée du modulateur (vert). On note juste la présence d'un déphasage de 103µs, qui représente le temps de traitement dans le XR2211.

L'utilisation des différentes vitesses de transmission (2400/1200/600) ne modifie pas nos signaux: le déphasage va en augmentant mais le signal en sortie du démodulateur reste toujours fidèle au signal d'entrée du modulateur.

Le principal problème que nous avons rencontré avec ce modulateur/démodulateur 4800bauds et son incompatibilité totale avec les Puxing777 nous servant aux tests. En effet, ceux-ci possèdent un filtre passe-bas dont la fréquence de coupure est à 4KHz dans l'entrée micro. Nous avons donc développé une solution afin de nous adapter: un modulateur/démodulateur à 1200bauds pour exploiter au maximum notre bande de fréquence de 300 à 4KHz.

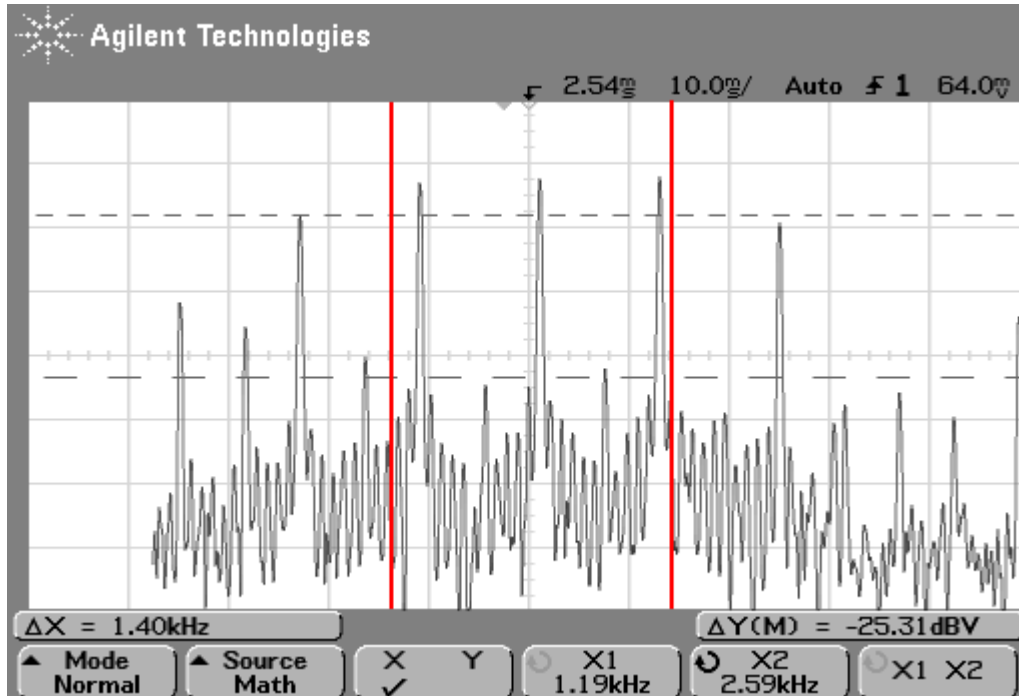
MODULATEUR 1200 BAUDS

Ce modulateur exploite donc au maximum la bande de fréquence de 300 à 4KHz comme le montre le spectre des fréquences:



La bande passante transmise est donc de $2(700 + 600) = 2600\text{Hz}$, de 590Hz à 3190Hz environ, avec un indice de modulation de 1,16. Ce cas est vraiment juste pour la bande fréquence que nous utilisons et l'indice de modulation ne nous permet pas d'émettre à de trop grande distance (la démodulation n'est plus garantie).

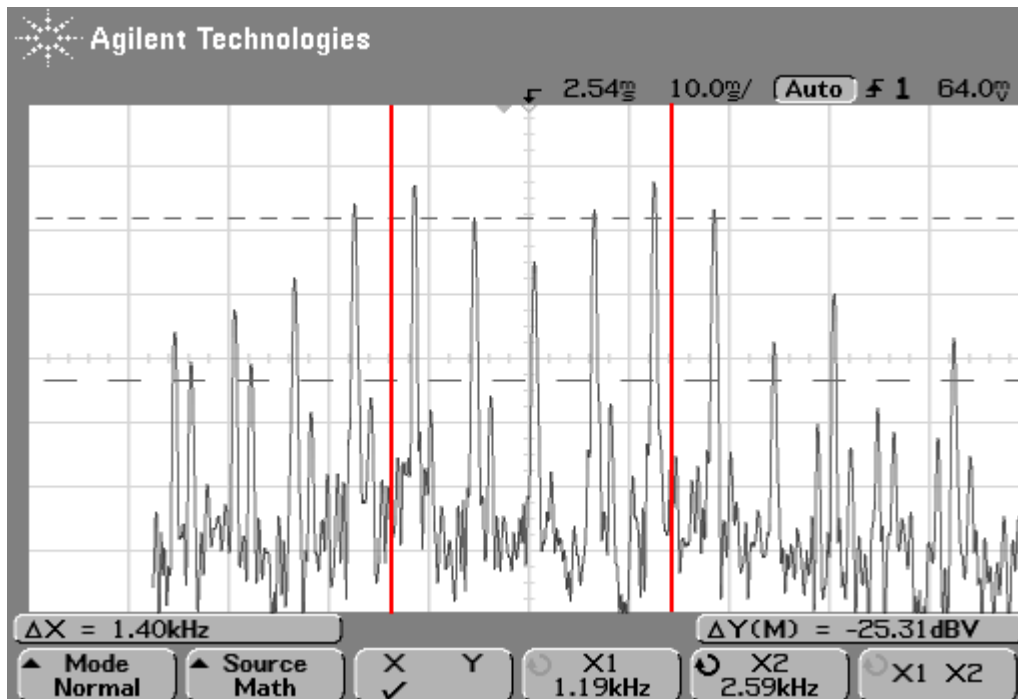
La FFT sur un signal carré modulé nous affiche très bien ce problème:



Les deux lobes sont indiscernables. La démodulation est bien entendu toujours possible, mais elle sera de moins bonne qualité à grande distance. L'augmentation de l'indice de modulation aura pour effet d'éloigner les deux lobes.

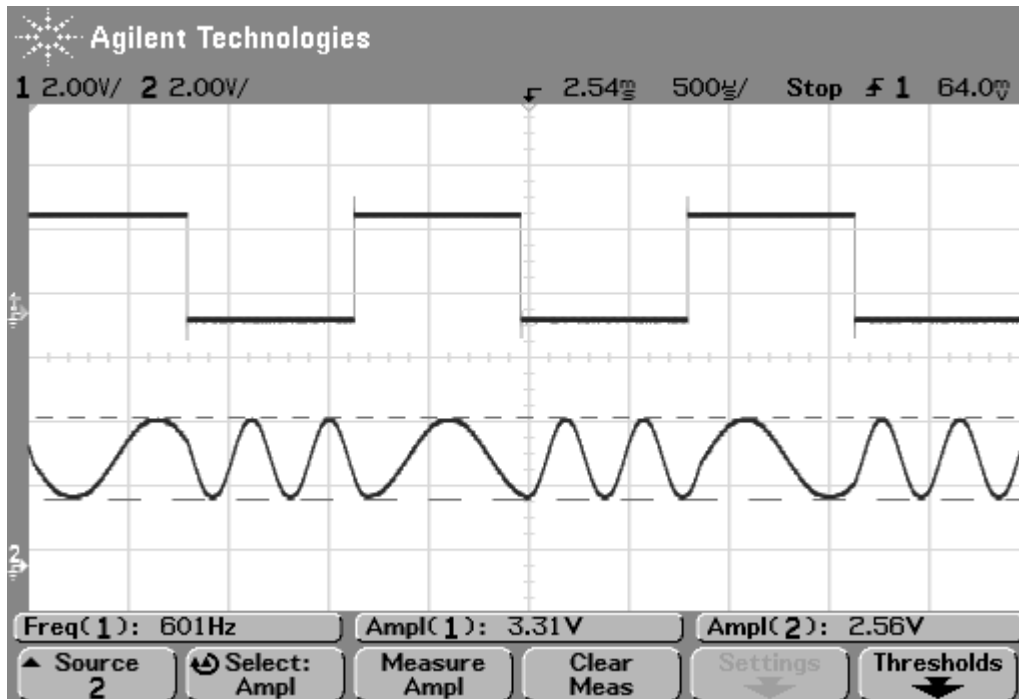
Nous discernons plus facilement ce phénomène en utilisant une vitesse de transmission de 600bauds. L'indice de modulation est deux fois plus élevé (étant donné que la vitesse de transmission est divisée par 2): $m = 700/300 = 2,3$.

Ci-après la FFT du signal de sortie en utilisant un signal d'entrée carré:



Dans ce cas, les deux lobes sont distincts. La démodulation sera alors de meilleure qualité en cas de déformation du signal.

Voici un exemple de modulation à 1200bauds qui nous a servi à réaliser les transformée de Fourier:

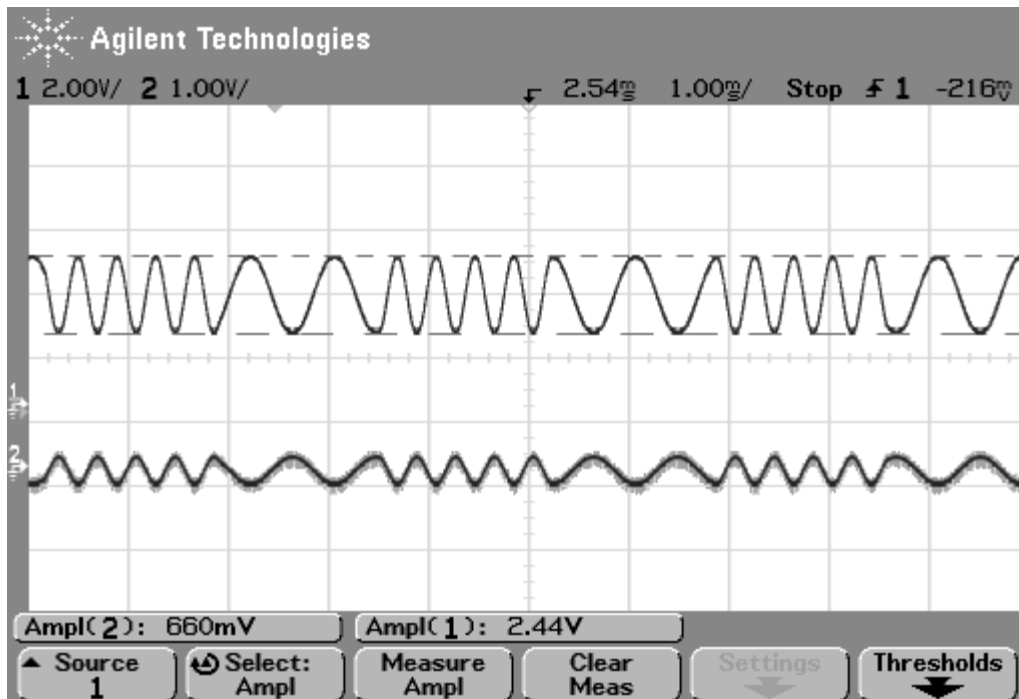


En entrée (voie 1), nous avons un signal carré d'amplitude 3.3Vcc (ce sont les niveaux actuels du FTDI).

Nous remarquons que nous n'avons pas tout à fait une période de sinus ($600 * 2 > 1190$) lorsque nous modulons un état haut. Cela ne dérange pas beaucoup la démodulation, mais comme pour l'indice de modulation, la démodulation sera de moins bonne qualité. A 600bauds ce problème ne se pose évidemment pas. L'état haut (« 1 » binaire) est modulé par la fréquence f1 1190Hz et l'état bas (« 0 » binaire) est modulé par la fréquence f0 2590Hz.

Le signal passe ensuite dans l'amplificateur opérationnel pour ajuster l'amplitude du signal et un condensateur pour supprimer la composante continue. Notre signal ne doit pas dépasser les 750mV pic à pic.

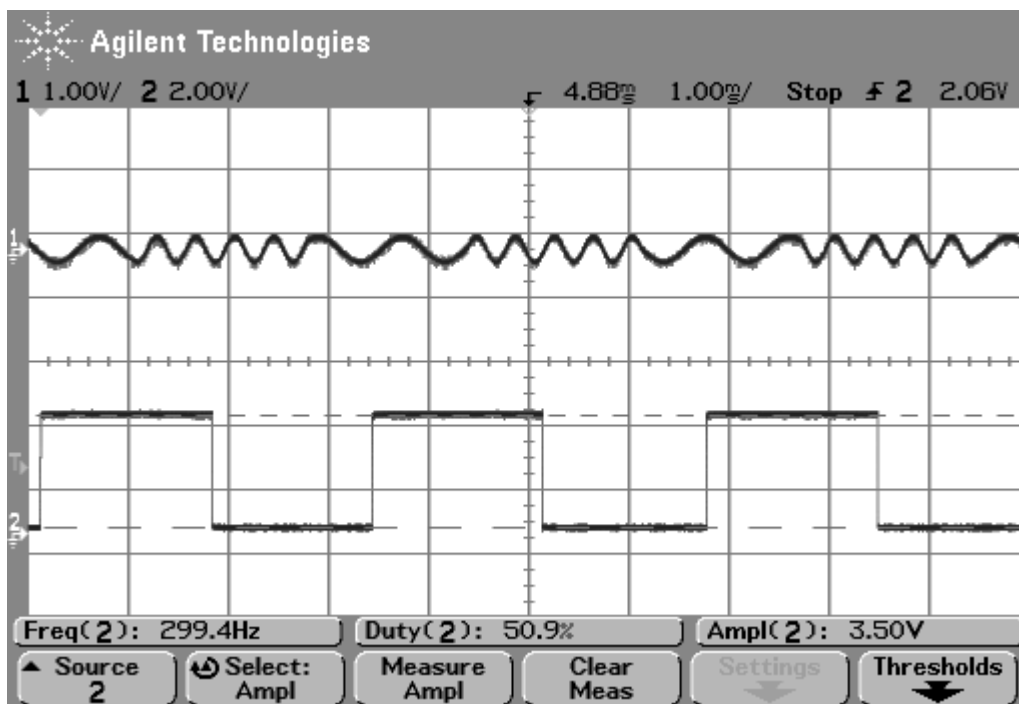
En sortie de l'AOP:



Le signal de sortie est obtenu sur la voie 2. Il est d'amplitude 660mV pour entrer dans le Puxing via l'entrée micro. Nous observons que le signal est inversé mais cela n'est pas un souci car nous travaillons en fréquence (la fréquence reste identique).

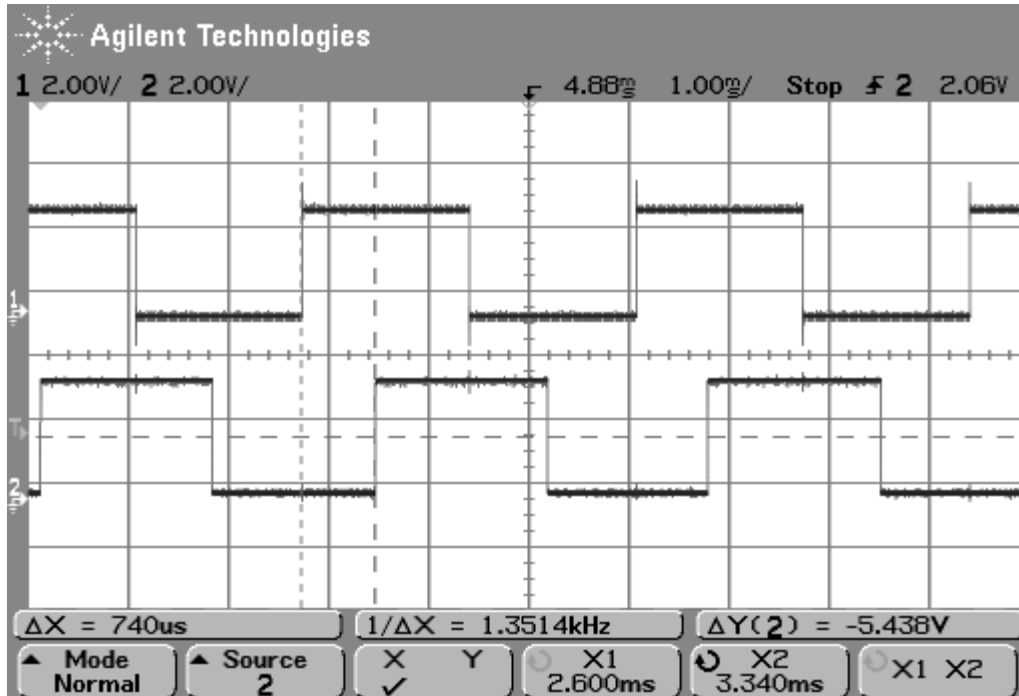
DÉMODULATEUR 1200 BAUDS

Le signal précédent a été mis en entrée de notre démodulateur. Voici le résultat:



Le signal obtenu est bien proche des 300Hz avec un rapport cyclique d'environ 50%. Nous retrouvons donc notre signal d'origine d'amplitude 3.5V. Il passe ensuite dans un max232 pour ajuster les niveaux à la norme RS232.

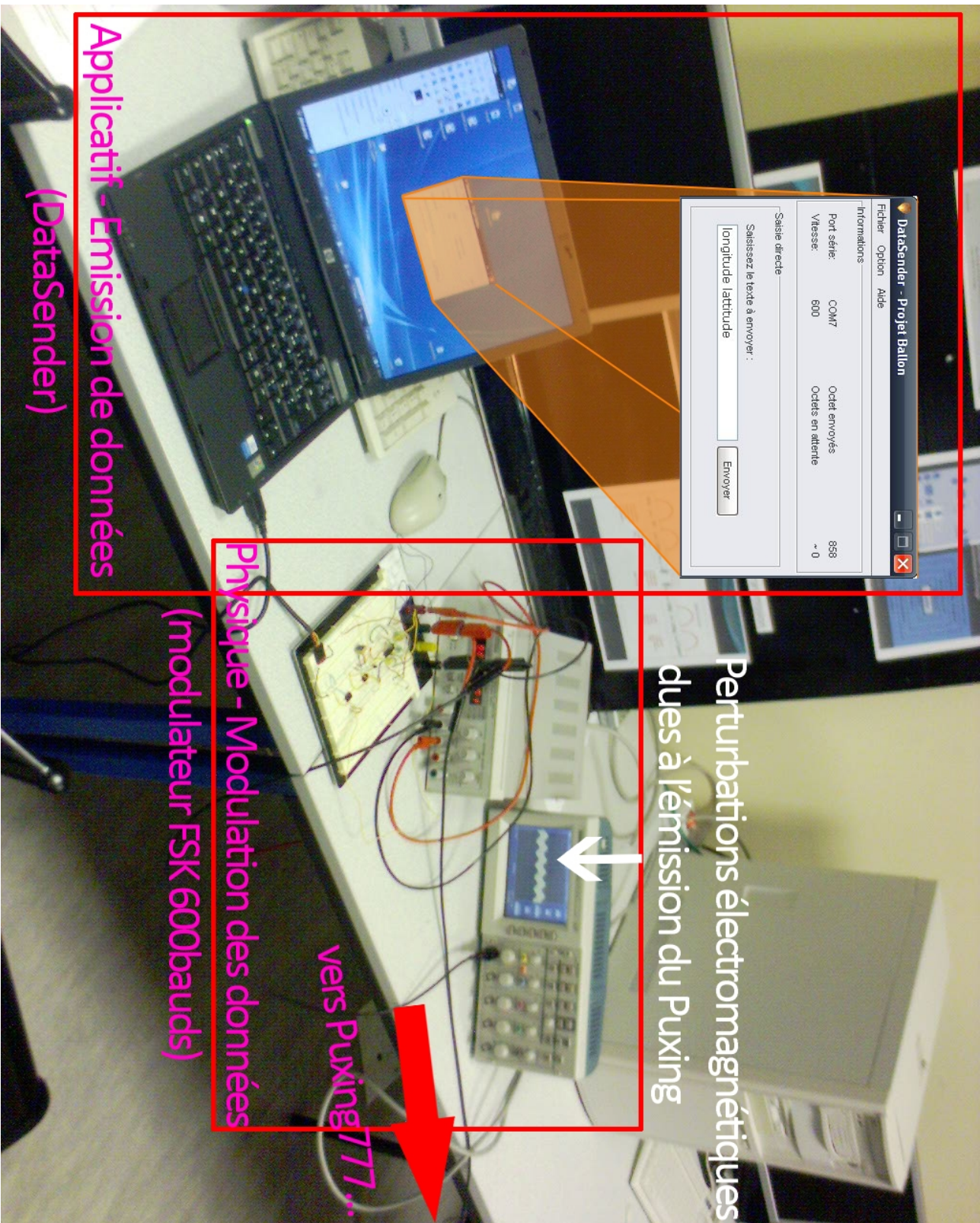
Il a été intéressant de comparer les deux signaux:



Comme on peut le constater, les deux signaux sont comparables avec la présence d'un déphasage de $740\mu s$. A 1200 bauds, ce déphasage est de $650\mu s$.

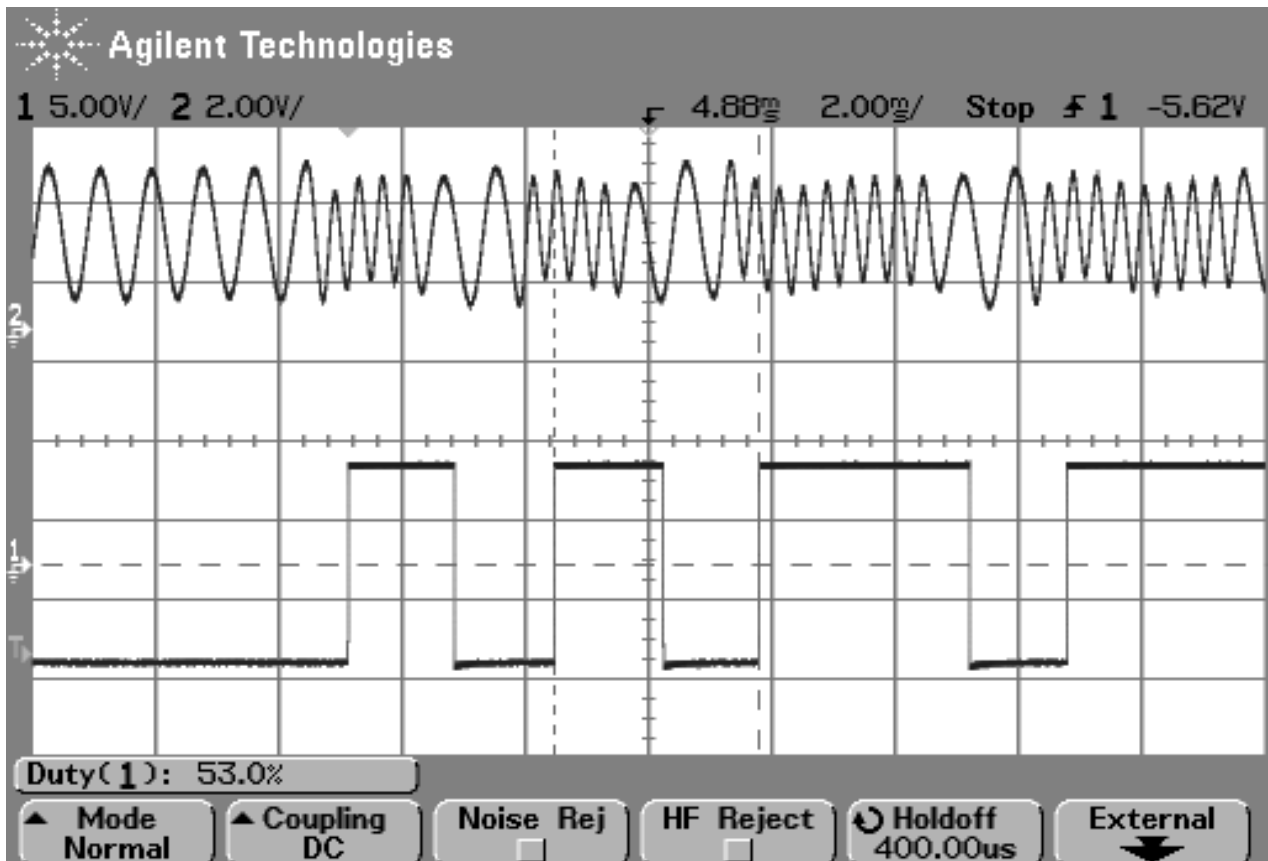
Afin de pouvoir conclure cette phase de test, nous avons réalisé un test de la transmission avec les émetteurs radio Puxing777. Le modulateur était placé à la plateforme Technologique et le démodulateur (avec le récepteur) dans le bâtiment C. Le modulateur était branché à l'entrée micro du Puxing qui émettait sur la bande de fréquence 144.3MHz (fréquence libre radioamateur). Le démodulateur, quant à lui, était connecté à la sortie haut-parleur du Puxing placé en réception.

Les mots de test sont « longitude » et « latitude » comme présentés sur les photos.



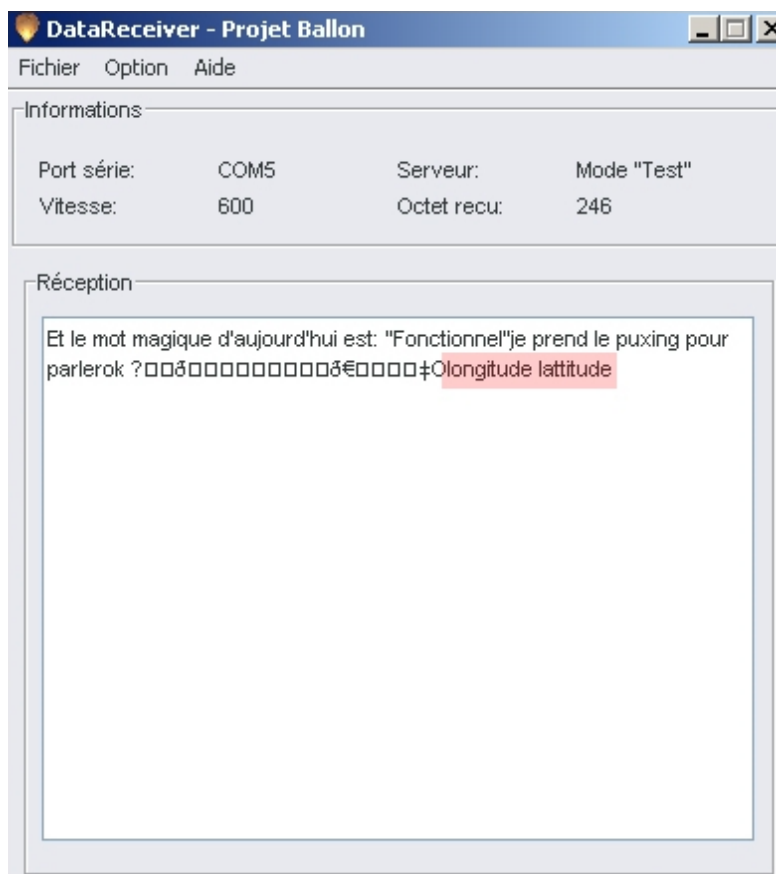


Voici le signal que nous obtenons en réception:



Nous arrivons bien à démoduler le signal reçu, qui a été amplifié par le Puxing à l'aide de la molette de réglage du volume.

Voici ce qu'affichait l'écran de l'ordinateur lors de cette transmission. On voit bien apparaître les mots « longitude » et « latitude ». Les caractères non affichables apparaissant entre « parler ok? » « Longitude » sont seulement dus au fait que nous avons déconnecté puis reconnecté le Puxing durant la transmission.



Notre transmission est donc bien fonctionnelle et performante à 600 bauds avec les Puxing. Nous n'avons aucune erreur à 600baud même en transmettant un gros fichier de plusieurs kilo-octets. En revanche, à 1200bauds, nous avons reçu la moitié du message. Le reste étant coupé en entrée du Puxing par le filtre passe-bas à 4KHz. En utilisant un émetteur Kiwi, qui possèdent une bande passante en entrée de 500Hz à 50KHz, nous n'aurions pas eu ce problème et nous aurions pu tester notre transmission à 4800 bauds sans avoir à nous adapter.

Conclusion

Notre chaîne de transmission nous permet donc d'émettre des données numériques, provenant du ballon stratosphérique, aux débits suivants: 600, 1200, 2400 et 4800 bauds. Ces valeurs sont fixées par les composants associés au modulateur, le démodulateur étant dépendant de ce dernier. Un programme codé en C implémenté sur un PIC, sert au codage des données avant leur émission. Un autre programme codé en JAVA, exécuté par un ordinateur au sol, permet le décodage et l'affichage des données. Les résultats obtenus sont tout à fait satisfaisants, puisque nous sommes capables de transmettre les données de manière fiable, à des débits correspondant aux besoins. En revanche, des problèmes de portabilité des équipements restent à résoudre, notamment au niveau du modulateur, puisque celui-ci sera intégré dans le ballon.

Cependant, au cours de la dernière semaine de projet, nous avons réalisé que certains des composants que nous utilisons n'étaient plus fabriqués par le constructeur. Ainsi, nous en avons recherché un autre, toujours disponible, réalisant à la fois les fonctions de conversion tension/fréquence et fréquence/tension (qui puisse réaliser la modulation et la démodulation). L'AD9834 est un composant réalisant une DDS (Direct Digital Synthesis): il peut générer un signal sinusoïdal à des fréquences pouvant atteindre les 75 Mhz, variant en fonction du signal d'entrée.

Ce projet nous a permis de mettre en pratique les connaissances acquises au cours de notre formation. En faisant intervenir des notions de transmission numérique de données, d'électronique et de programmation, nous avons abordé la plupart des composantes du diplôme universitaire technologique de Réseaux et Télécommunications.

Les compétences de chacun des membres du groupe nous ont permis d'atteindre nos objectifs, qui étaient de parvenir à faire fonctionner l'émission, le codage, la réception et le décodage des données.

Ce projet nous a aussi permis de développer notre autonomie, ainsi que de mettre à profit notre aptitude à nous organiser, en suivant un planning et un cahier des charges basés sur des contraintes temporelles et de disponibilité des équipements. Nous sommes ainsi parvenus à tenir les délais, malgré quelques retards induits par l'indisponibilité des salles durant la période de grèves.

Grâce à ce travail, nous avons pu appréhender concrètement les difficultés de mise en œuvre d'un projet technique. Nous nous sommes aussi et surtout enrichi d'une nouvelle expérience dans laquelle nous avons utilisé et structuré nos acquis, dans le but de concrétiser les compétences acquises au cours de notre formation.

Bibliographie

Cours de télécommunications de Denis Genon-Catalot intitulé Télécommunications numériques : Modulation de fréquence et modulation FSK

Mathématiques à l'usage des informaticiens, Thierry Brugère, Alain Mollard, Henri Habrias, *Ellipses*, 2003

Au coeur de Java 2, volume 1 et 2, Cay S. Horstmann , Gary Cornell , Campus press, 2005

Webographie

<http://uuu.enseirb.fr/~dondon/transnum/modnumFSK/ModnumFSK.html>: Site explicatif du principe de la modulation/démodulation FSK.

<http://rom.buisson.free.fr/ballon>: Projets Ballon 2007-2008 : Projets Ballon 2007-2008

<http://www.planete-sciences.org/espace/basedoc/index.php/Catégorie:Télémesures>: Wiki Planètes-science :

http://f4buc.chez-alice.fr/page_web_ballon/index.htm: Projet Ballon du Radio Club de Vauréal (95)

http://radioamateur.f1rfn.online.fr/jcm_reglementation2.htm#tableauITU: Listes des fréquences libres pour radio-amateurs (Région 1)

http://fr.wikipedia.org/wiki/Code_correcteur: Wikipédia Code Correcteur: Wikipédia Code Correcteur

<http://pagesperso-orange.fr/ts.icare/>: Le site des ballon sonde ICARE du BTS
Électronique du lycée Alfred Kastler de Talence (33)

[http://missiontice.ac-
besancon.fr/sciences_physiques/physique_appliquee/electronique/mod_numerique/in
dex.php](http://missiontice.ac-besancon.fr/sciences_physiques/physique_appliquee/electronique/mod_numerique/index.php): Transmissions numériques sur onde porteuse

http://pmr446.free.fr/index_px777.htm: Récepteur radio Puxing 777

Lexique

AOP: montage amplificateur permettant de réaliser différentes fonctions (suiveur inverseur etc...).

Bande passante: largeur du spectre occupé par le signal.

Bauds: nombre de symboles/seconde.

Bits de parité: dépend du nombre de bits à "1" dans un mot de bits. Si ce nombre est pair, on dit que la parité est paire, impaire sinon.

Bruit: perturbations sur le signal du à des interférences sur le canal de transmission.

C: langage de programmation impératif.

Canal de transmission: médium par lequel transitent les informations.

CNES: Centre national d'étude spatiales.

Distorsion de fréquence: Lors d'une transmission, la distorsion de fréquence peut ajouter ou supprimer les harmoniques d'un signal.

Filtre: permet de conserver uniquement le signal utile, en coupant à une fréquence donnée.

Fréquence Rythme Fr: Inverse de la durée consacrée à la transmission d'un symbole élémentaire.

FSK: Frequency Shift Keying: modulation numérique à saut de fréquence (cf. page 9).

Hamming (code): code mathématique permettant la détection et la correction d'erreurs.

Indice de modulation: Faire varier l'indice de modulation m permet d'augmenter ou de diminuer l'écart entre la fréquence la plus grande et la plus petite.

Java: langage de programmation orienté objet multi-plateforme.

PLL: Phase Locked loop: Boucle à verrouillage de phase. Permet de convertir une fréquence en une tension.

Python: langage de programmation.

Récepteur VHF: Very High Frequency: récepteur radio très haute fréquence (de l'ordre du Mhz).

Signaux analogiques: signaux comportant une infinité de valeurs.

signaux numériques: signaux composés d'un nombre fini de valeurs.

Stratosphère: seconde couche de l'atmosphère située entre la troposphère, au-dessus, et la mésosphère, au dessous

TER: taux d'erreurs bit mesuré lors de la réception du signal

Trames: ensemble de bits que l'on envoie.

Transmission asynchrone: mode de transmission où l'émetteur et le récepteur ne sont pas synchronisés. L'horloge n'est pas transmise.

VCO: Voltage Controlled Oscillator: Oscillateur commandé en tension. Permet de convertir une tension en une fréquence.