DELFOSSE Alexandre BARON Paul M'FOUKH Adel

Supélec

Rapport de mi-projet

Projet long

Supsat

Station sol

Sommaire

Introduction

Tout d'abord, Supsat est une association fondée en 2012 par trois étudiants de Supélec Rennes dont le but est, à terme, d'envoyer un nanosatellite (10*10*10cm, environ 1W, environ 1kg) dans l'espace, celui-ci étant entièrement conçu par des étudiants de Supélec.

L'objectif de Supsat est triple :

- *Pédagogique* : la conception d'un système aussi complexe qu'un satellite ne peut être que motivant et excitant pour un élève ingénieur qui a envie de s'investir dans un projet concret. Soutenu par des personnes issues de la recherche et de l'industrie, les étudiants conçoivent le satellite de A à Z. Ainsi, de la conception à l'assemblage des différentes parties, de l'assemblage à la vérification du design et enfin de la phase de tests au lancement réel ainsi qu'à son contrôle, les étudiants ont l'occasion de mettre en pratique et de confronter ce qu'ils ont appris sur un projet concret à la pointe de la technologie.
- *Scientifique* : bien que Supsat soit un nanosatellite, il embarque un micro-laboratoire qui est destiné à effectuer des tests relatifs à la reconfiguration partielle de FPGAs :
  - des tests de durcissement logiciel contre les SEEs (Single Event Effects)
  - des tests de reconfiguration sans interruption de service
- *Technologique* : des applications quant à l'utilisation de cette technologie peuvent apparaître si les expériences portent leurs fruits.

Dans le cadre de la réalisation du projet de Supsat, le présent rapport vise à dimensionner et réaliser la station sol qui permettra la transmission de données vers le satellite ainsi que la réception et le traitement des données envoyées par celui-ci, afin de déterminer les éléments qui la constituent accompagnés de leurs caractéristiques. Il a également pour but de présenter le bilan de liaison reliant les deux parties (station sol et satellite) et d'expliquer le principe de la communication entre ces deux parties.

La station sol Supsat peut être divisée, comme toute station sol, en deux unités :

- Le segment sol (Ground Station), emplacement où se trouve tout le matériel RF, avec les antennes ainsi que le système de suivi lié à ces antennes.
- La base de Contrôle (Mission Control) sur ordinateur en utilisant logiciels et programmes pour surveiller et contrôler le satellite.

Notre projet porte particulièrement sur le segment sol de la station, mais la partie émission et lecture des données transmises par le satellite sera également largement abordée.

Notre travail se concentre donc sur la partie « suivi du satellite » (Tracking), ainsi que sur la chaîne de réception, que nous avons décidée d'implémenter en radio logiciel (SDR) pour des raisons de flexibilité, de coûts, mais aussi pédagogiques.

I. *Aspects mécaniques de la station*

Le but de cette partie est de construire la base de la station sol qui aura pour finalité de soutenir l'antenne et de la diriger dans la direction souhaitée suivant deux axes de rotation. Cela nous permettra de suivre au mieux le satellite pour communiquer avec.

A. Problèmes rencontrés

Nous avons réutilisé la structure de la station sol construite par le groupe sur le même projet l'an dernier. Pour la construction, ils se sont basés sur l'open source SatNOGS pour les modèles. Elle est donc constituée d'une structure aluminium, d'un système de transmission ainsi que de deux moteurs pas-à-pas.

Ce système de transmission est composé d'une poulie-courroie de rapport $\frac{20}{36}$ et d'une roue de 30 dents avec une vis à un filet ce qui nous donne un rapport de transmission total de $\frac{1}{54}$.

Pour ce qui est de la précision du système, le moteur pas à pas réalise 200 pas pour un tour ce qui nous donne en sortie du moteur une précision de 1.8° et donc en sortie de la chaîne de transmission, une précision de $\frac{1.8}{54} = 0.03°$ ce qui est plus que suffisant.

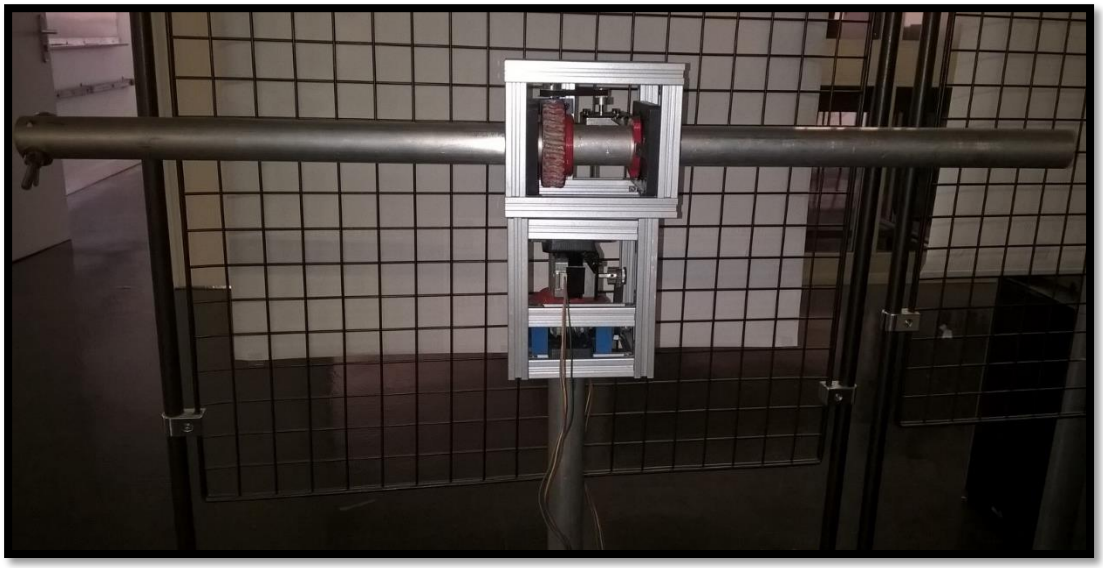


Figure 1. *Photo de la structure de la station sol*

Cependant, la structure présentait plusieurs défauts. En effet, les courroies étaient trop tendues et la roue-vis était bloquée ce qui empêchait les axes de tourner. Nous avons remarqué aussi la présence de jeux dans la structure et une certaine instabilité ; ce qui pose problème vu que l'on veut diriger l'antenne de façon assez précise. Nous nous sommes aussi posés la question des matériaux utilisés surtout concernant les pièces provenant de l'imprimante 3D. Leur usure plus ou moins rapide pourrait poser des problèmes à l'avenir.

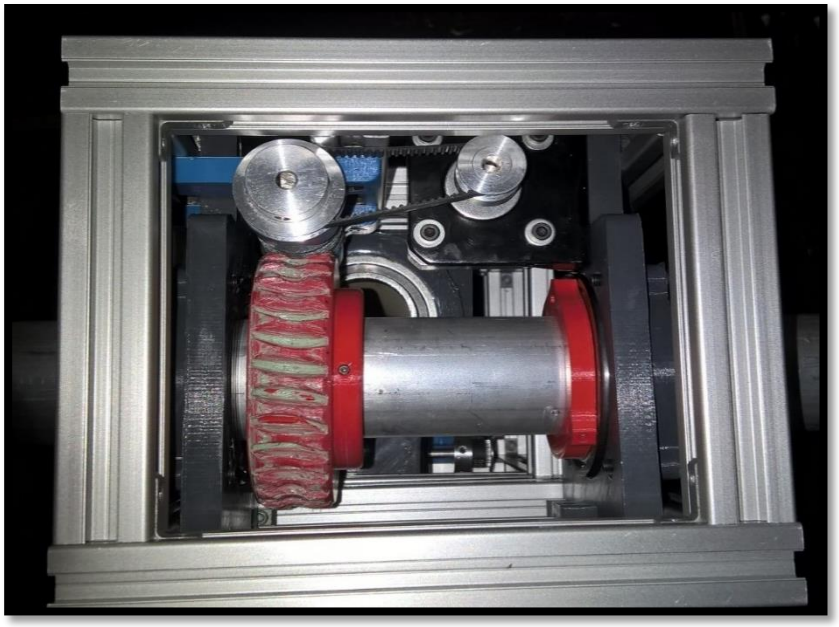


Figure 2. *Photo du haut de la structure avec les courroies et le moteur pas à pas*

B. Solutions apportées

Pour bien déterminer l'origine des problèmes, nous avons dû démonter la structure. Cela nous a permis de bien cibler les pièces qui posaient des difficultés à la mise en marche de la station.

La solution à ses différents problèmes fut simplement de réimprimer les pièces grâce à l'imprimante 3D mise à notre disposition par l'école et aux outils comme la lime, pour ajuster les pièces, prêtées par l'équipe de maintenance de l'école. Cela nous a demandé un travail de redimensionnement des pièces faites grâce au logiciel SolidWorks.

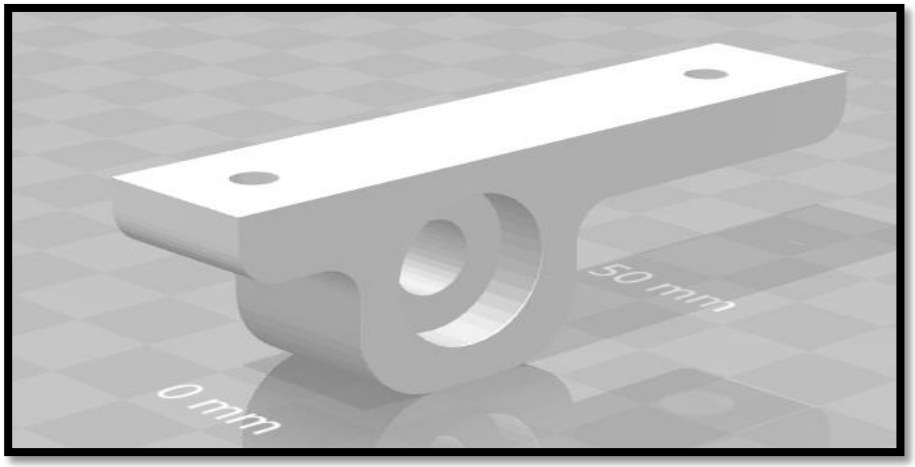


Figure 3. *Pièce modifée sous SolidWorks*

Cela nous a permis de résoudre le problème des courroies tendues et de la roue-vis bloquée. Le système peut tourner correctement maintenant.

Figure 4. *Photo de la structure avec les pièces modifiées (ici en bleu)*

C. Travail restant

Le travail restant concerne donc le problème de jeu, d'instabilité de la structure, et la question des matériaux utilisés.

Pour ce qui est des jeux présents, nous allons certainement utiliser des ressorts pour les supprimer. Cela devrait donc être assez simple et peu coûteux.

Le problème de l'instabilité sera réglé en améliorant le pied en modifiant sa mise en place actuelle et en fonction de l'antenne que l'on utilisera.

Pour la question des matériaux, à court terme, c'est-à-dire pour des tests, nous ne devrions pas avoir de problèmes vu que la structure fut déjà testée l'an passé par le groupe précédent.

II. *Contrôle du suivi de la station sol*

Cette partie se consacre à la description du fonctionnement de la station pour le suivi du cubesat.

Pour effectuer ce suivi, nous avons besoin des TLE ou Paramètres Orbitaux (voir Annexe). Ce sont des informations sur l'orbite des objets spatiaux régulièrement mis-à-jour par les agences spatiales. En connaissant la date et l'heure, on peut en déduire, pour un lieu donné, les angles d'azimut et d'élévation permettant de viser le satellite.

On contrôle ainsi l'orientation de l'antenne à l'aide de deux moteurs pas-à-pas.

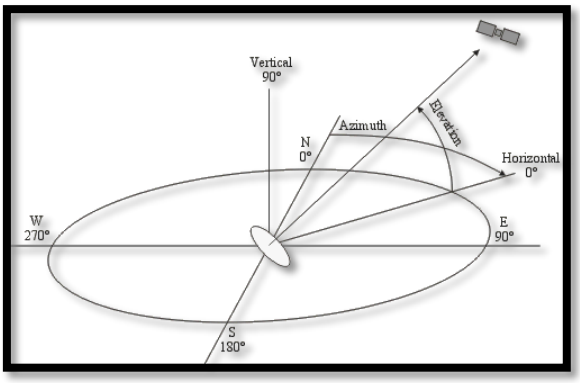


Figure 5. *Angles d'azimuth et d'élévation*

Nous allons décrire les différents éléments composant la chaîne d'information et d'énergie.

Nous avons décidé de remplacer certains éléments utilisés lors du projet de l'an dernier. Nous allons ainsi expliquer brièvement l'ancien fonctionnement avant de présenter les changements apportés et leurs raisons.

A. Conception du projet précédent

Initialement, un PC externe exécutait un script Python récupérant les TLE du satellite à suivre par Internet. Une suite de consignes en azimut et élévation étaient générés à différents instants à partir d'un instant donné. Ces consignes étaient enregistrées sur une carte SD.

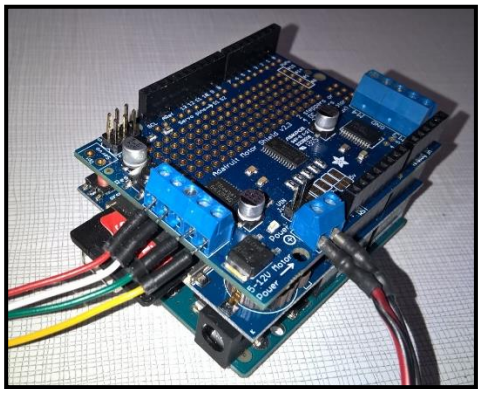


Figure 6. *Arduino Uno avec ses shields*

Puis, un Arduino Uno équipé d'un shield lecteur de carte SD, d'un RTC (Real Time Clock) et d'un shield driver de moteur pas-à-pas, lisait ces consignes et commandait alors les moteurs en conséquence à l'aide du RTC donnant la date et l'heure.

L'alimentation est assurée par une batterie de 1800mAh de 11,8V. Un convertisseur DC/DC ramène cette tension à 5V alimentant l'Arduino et les moteurs.

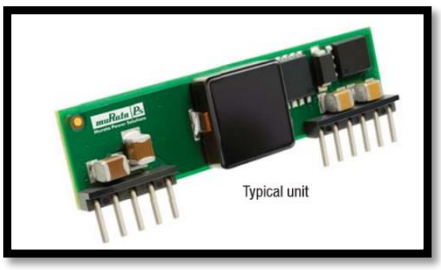


Figure 7. *Convertisseur DC/DC*

B. Nouvelle conception

Le passage par un PC externe nous semblait laborieux. Voulant rendre le système davantage autonome, nous avons décidé qu'un seul microcontrôleur serait chargé de trouver les TLE et de les convertir en temps réel en consignes d'angles.

De plus, la conception précédente obligeait à initialiser le RTC de l'Arduino lors du téléversement du code. Nous préférons alors, pour plus d'autonomie et de précisions, récupérer la date et l'heure sur Internet par NTP (Network Time Protocol) et actualiser ensuite le RTC.

Afin de pouvoir se connecter à Internet, nous avons choisi de passer par WiFi. L'ESP8266 nous a alors semblé un bon choix au vu de ses performance, son faible coût et l'expérience que nous avions sur l'IDE Arduino.

Nous utilisions un RTC et le shield motor driver de l'Adafruit.

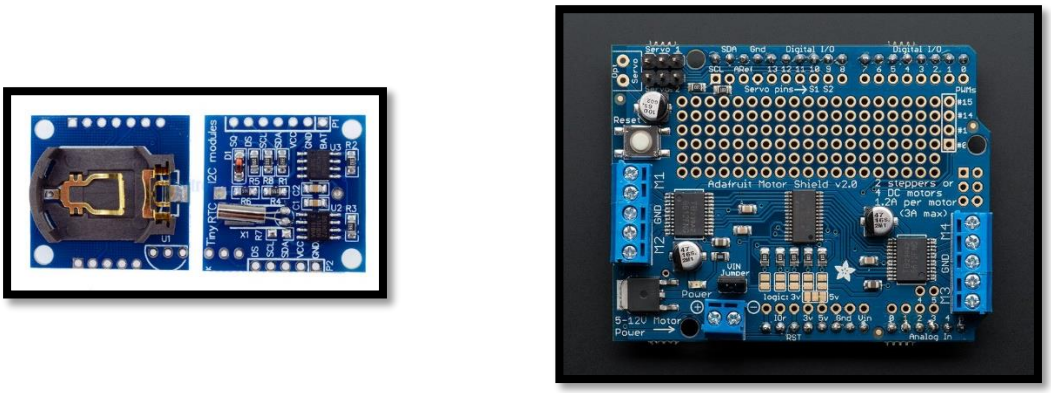


Figure 8. *RTC DS1307 et Adafruit Motor Shield Drive*

Cependant, nous avons rencontré de grandes difficultés à convertir les TLE en angles. En effet, les seules bibliothèques trouvées étaient disponibles en Python (PyOrbital) ou en C++ (OrbitalTools). La version C++ faisait appel en très grande partie à des fonctions standard de ce langage (STL, Standard Template Library). Après des recherches et des essais infructueux d'adaptation ou d'utilisation de bibliothèques intermédiaires (StandardCPlusPlus entre autres), la bibliothèque OrbitTools a semblé impossible à utiliser dans l'environnement Arduino.

Nous nous sommes alors penchés sur l'ESP32, version plus évoluée de l'ESP8266 pouvant fonctionner sous FreeRTOS. Nous utilisons à présent Eclipse pour adapter en C les fonctionnalités voulues. Cette plateforme permet un développement plus professionnel qu'Arduino. Sa puissance de calcul dépasse ce dont nous nécessitions *a priori*, mais cela permettra de développer potentiellement de nouvelles fonctionnalités par la suite sans changer de matériel.

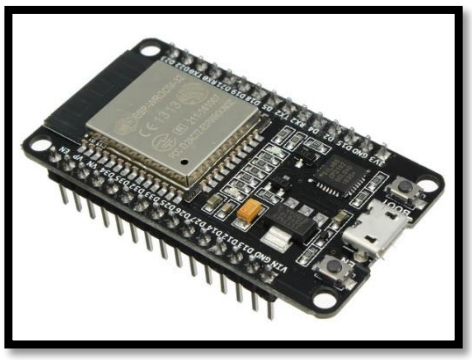


Figure 9. *ESP32, kit de développement*

De plus, l'ESP32 dispose d'un RTC interne nous permettant de nous dispenser *a priori* du DS1307. Il supporte également davantage de protocoles d'identification Wifi ; ce qui permettra de l'intégrer plus facilement à l'architecture réseau du Supélec.

L'utilisation de moteurs pas-à-pas permet de contrôler facilement l'angle simplement en donnant le nombre de pas. Pour information, nos moteurs font une rotation en 200 pas. Cependant, il est nécessaire d'avoir une position de référence pour commander en angles absolus.

Figure 10. *Moteur pas-à-pas*

Nous avons choisi comme référence l'horizontal pour l'élévation et le nord magnétique pour l'azimut.

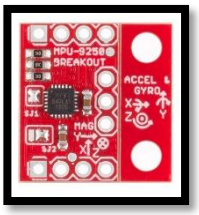


Figure 11. *IMU*

Dans l'idéal, nous pensons utiliser un capteur d'inclinaison et un magnétomètre pour repérer ces deux positions initiales. Mais un IMU (Inertial Measurement Unit) 9 axes faisant accéléromètre, gyroscope et magnétomètre, doit permettre d'obtenir l'orientation de l'antenne dans toutes les directions et donc de n'avoir qu'un seul capteur. L'utilisation d'un MPU9250 s'est avérée laborieuse, tant d'un point de vue de la calibration que de l'utilisation de quaternions pour en déduire l'orientation dans toutes les directions de l'espace.

Nous abandonnons donc l'utilisation d'un IMU pour favoriser des capteurs à effet Hall, beaucoup plus simples. Mais cela ne permettra pas à la station de s'adapter en autonomie si on la déplace. Si le temps nous le permet, nous reviendrons plus tard sur cette solution.

Nous utilisons toujours la batterie et le convertisseur DC/DC de l'an dernier, mais nous pensons peut-être passer par une alimentation secteur en fonction de la localisation de l'antenne. Ce point reste encore à clarifier. Nous avons cependant refait les soudures de la plaquette du convertisseur qui nous semblaient hasardeuses, et avons ajouté un fusible et un interrupteur pour imiter le schéma d'utilisation fourni par le constructeur.

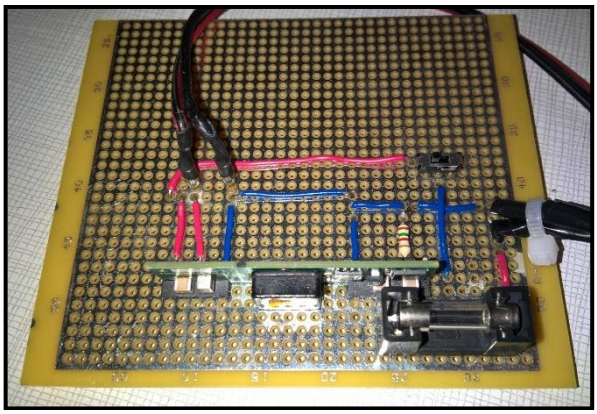


Figure 12. *Plaquette avec convertisseur DC/DC*

Afin de faciliter par la suite la vérification du fonctionnement de la station, nous intégrerons un écran sur la station affichant les informations essentielles de son état. Par exemple, les consignes calculées, l'heure récupérée par NTP, le succès de la récupération des TLE, etc…

Des caractéristiques plus précises de nos composants sont disponibles en annexe.

Sur les différentes plateformes utilisées, nous avons testé certains aspects du système final :

PLATEFORME	SHIELD MOTEURS	RTC	NTP	TLE INTERNET	CONVERSION TLE	IMU
ARDUINO UNO	Fonctionnel	Fonctionnel	Pas de connexion	Pas de connexion	x	Bibliothèque légère, pas de résultats concluants
ARDUINO MEGA	x	x	x	x	x	Bibliothèque lourde, peu de résultats
ESP8266	x	Fonctionnel	Fonctionnel	Fonctionnel	Incompatibilité	x
ESP32	A faire	Interne, fonctionnel	Fonctionnel	En cours	En cours	Proposé en projet de première année

Figure 13. *Fonctionnalités développées au cours des changements de plateforme*

Le regroupement de toutes ces fonctionnalités sur l'ESP32 est parfois laborieux car les bibliothèques utilisées précédemment ne sont pas toujours les plus adaptées et pertinentes.

Ainsi, l'architecture finale de notre système est :

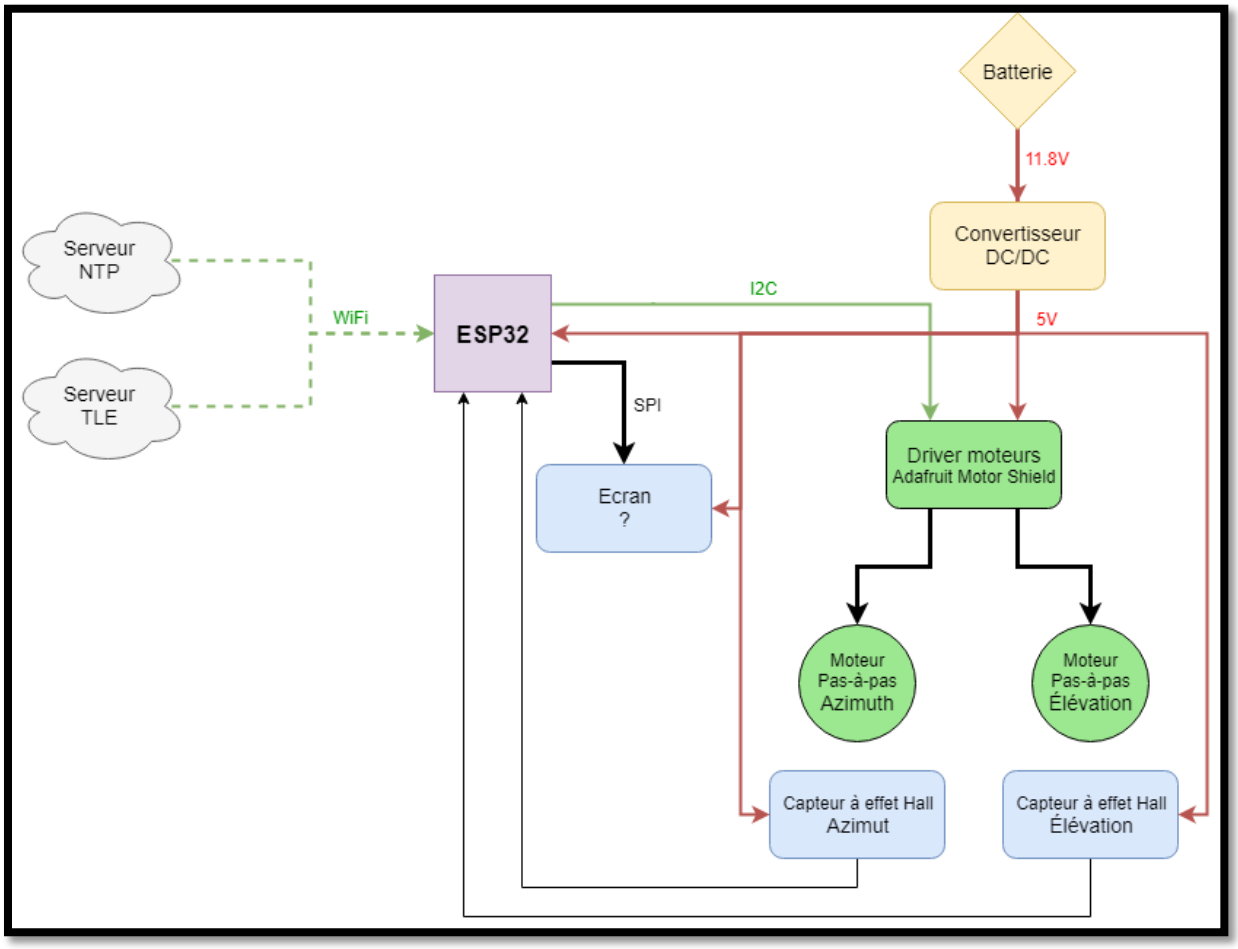


Figure 14. *Architecture actuellement en cours d'implémentation*

Le début du développement avec l'ESP32 a été difficile à cause de nombreuses erreurs de compilation dans l'IDE Eclipse. En effet, cette plateforme jeune ne dispose pas d'outils de développement aussi simples que l'Arduino. Il a donc fallu comprendre plus en profondeur le fonctionnement de ce composant et de FreeRTOS pour pouvoir faire fonctionner l'ESP32. A présent, nous développons et regroupons toutes les fonctionnalités de la station sol finale.

Nous vous présentons ainsi l'organigramme de l'aspect logiciel :

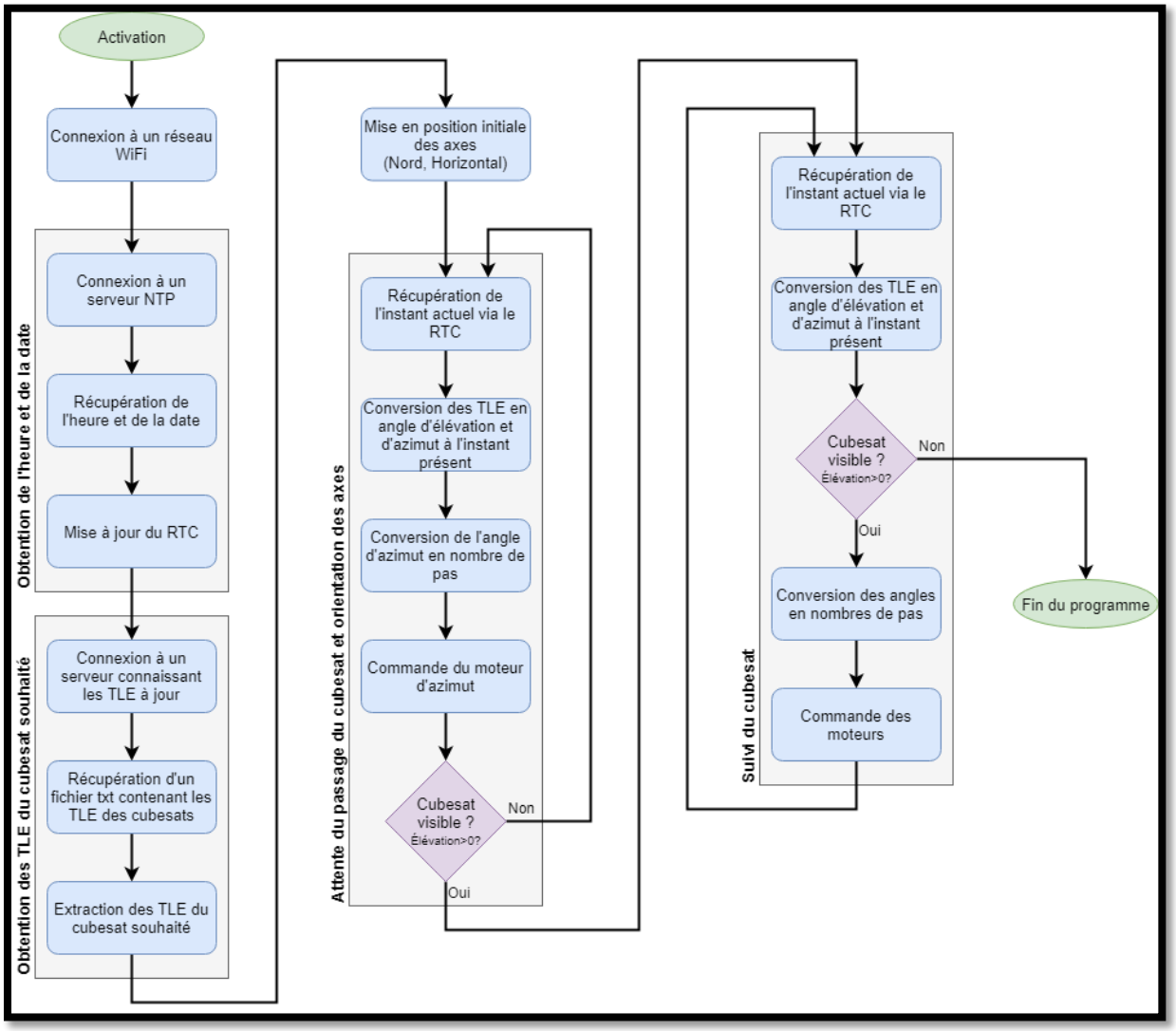


Figure 15. *Organigramme illustrant les actions de l'ESP32*

III. *Aspects théoriques de la communication*

A. Bilan de liaison descendant

Afin d'établir le bilan de liaison, il est important de rappeler les caractéristiques principales de la mission.

Le CubeSat est conçu pour être en orbite basse (LEO : Low Earth Orbit) circulaire autour de la terre, à une altitude de 400km. On peut alors déterminer sa vitesse orbitale moyenne (en négligeant la masse du satellite devant celle de la terre, ce qui n'est pas ridicule…):

$$v = \sqrt{\frac{M.G}{r+h}}$$

Avec $G = 6{,}67\ 59.10^{-11}\ m^3.kg^{-1}.s^{-2}$ la constante de la gravitation ; $M = 5{,}972\ .10^{24}\ kg$ la masse de la terre ; $r = 6\ 371.10^3 m$ le rayon de la terre et $h = 400.10^3 m$ la hauteur de l'orbite.

On trouve $v = 7{,}7.10^3 m.s^{-1} = 7{,}7\ km.s^{-1}$

Cette vitesse est assez élevée pour provoquer un décalage de fréquence par effet Doppler d'environ 10KHz lors du passage du satellite au-dessus de la station sol, décalage qu'il faudra prendre en compte et corriger avec soin.

Le satellite étant en orbite circulaire, il est vu, par la station sol, à une distance qui dépend de l'angle d'élévation (en approximant que sa trajectoire passe quasi parfaitement « au-dessus de notre tête », ce qui n'est pas forcément le cas en réalité et rajoute des pertes. Mais afin de simplifier les calculs, on peut faire cette approximation).

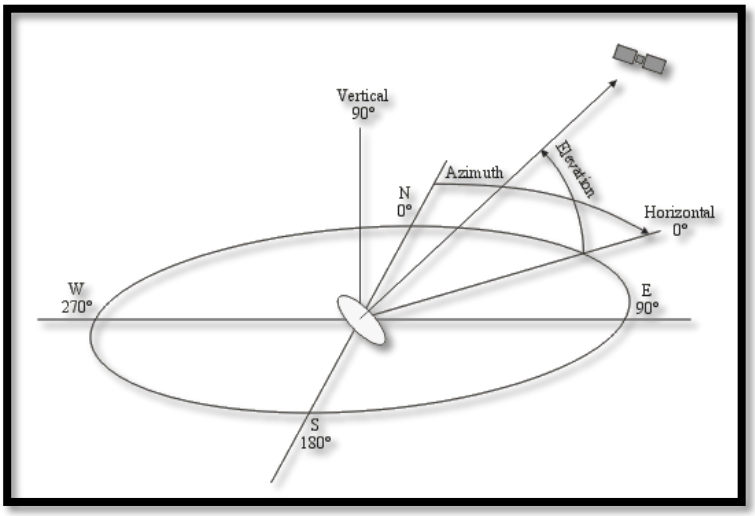


Figure 16. *Anlges d'azimuth et d'élévation*

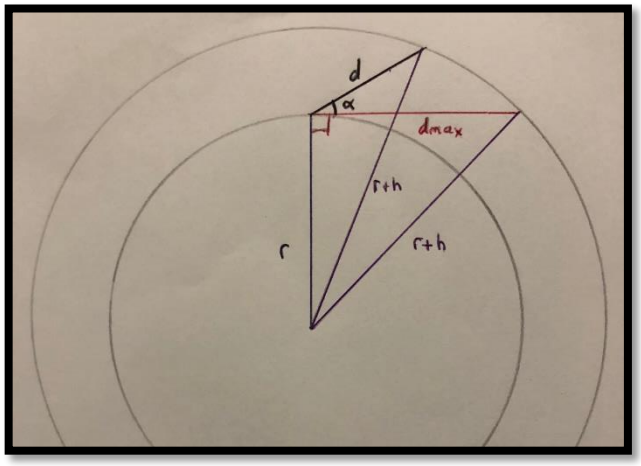


Figure 17. *Détermination de la distance entre la station sol et le cubesat*

On peut exprimer l'un en fonction de l'autre. Cela permettra par la suite d'exprimer les pertes en espace libre (dues à la distance parcourue par le signal) en fonction de l'angle d'élévation de l'antenne, et déterminer une plage utile d'élévation.

Soit $r = 6\,371.10^3 m$ le rayon de la terre et $h = 400.10^3 m$ la hauteur de l'orbite, α l'angle d'élévation et d la distance séparant le satellite de la station sol.

On peut déterminer la distance maximale d_{max} entre le satellite et la station, qui correspond à un angle d'élévation $\alpha = 0°$, à l'aide du théorème de Pythagore :

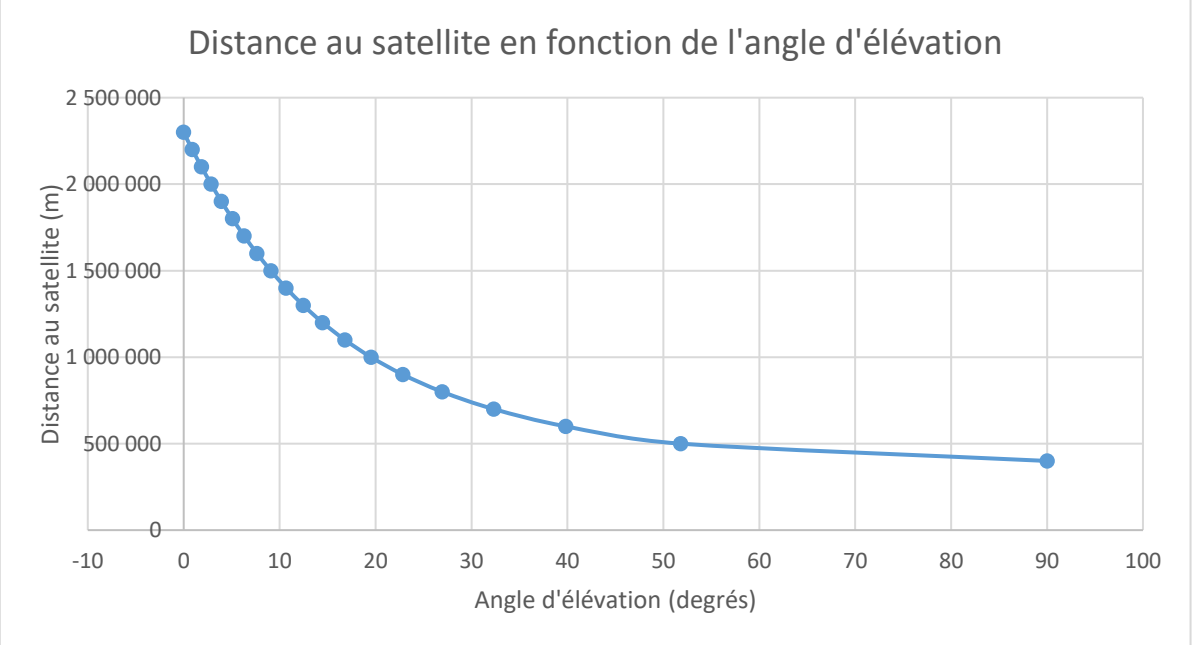
$$d_{max}^2 + r^2 = (r + h)^2$$

$$d_{max} = \sqrt{(r + h)^2 - r^2} = 2298km$$

Par construction $d_{min} = h = 400km$ pour une élévation $\alpha = 90°$. Pour toutes les autres positions intermédiaires, on doit appliquer le théorème d'Al-Kashi :

$$\boxed{\alpha = \arccos\left(\frac{r^2 + d^2 - (r + h)^2}{2.r.d}\right) - \frac{\pi}{2}}$$

Ainsi, on est capable de déterminer les valeurs d'élévation correspondantes pour des distances satellite – station sol variant de 400 à 2300km :

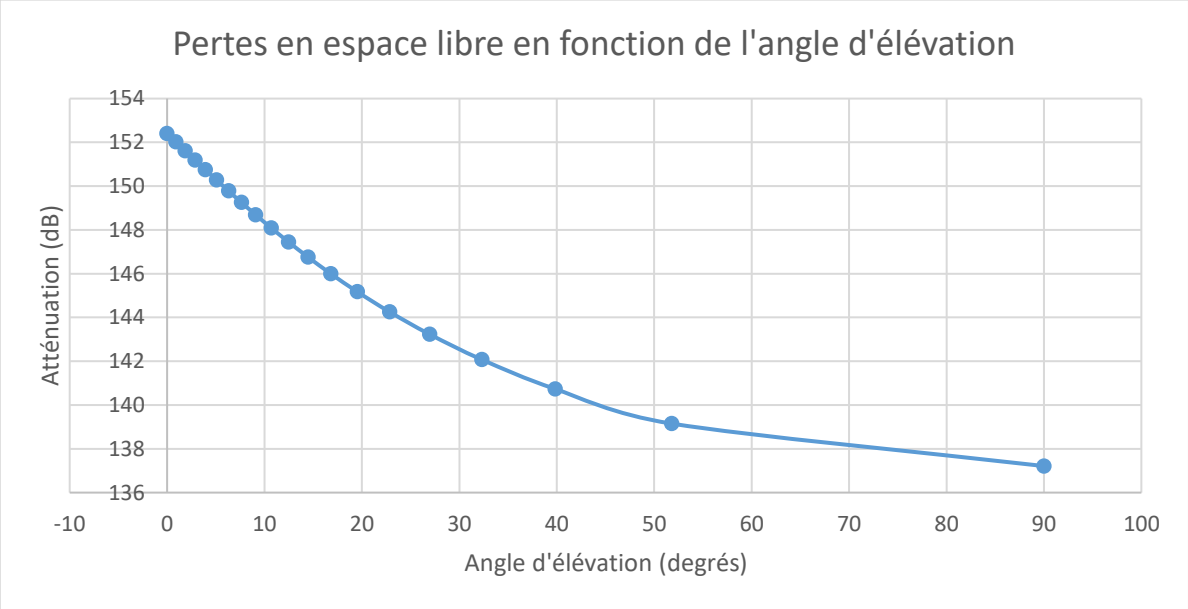


La fréquence utilisée pour la communication TM/TC (Télémétrie / Télécommande) est 433,5MHz pour les deux liens, montant et descendant.

On calcule alors les pertes en espace libre (Free Space Loss) en fonction de la distance à laquelle on voit le satellite :

$$\boxed{FSL = \left(\frac{4.\pi.d.f}{c}\right)^2 \Rightarrow FSL|_{dB} = 20.\log\left(\frac{4.\pi.d.f}{c}\right)}$$

On peut ensuite, connaissant le lien entre la distance au satellite et l'angle d'élévation, tracer l'atténuation due aux pertes en espace libre en fonction de l'angle d'élévation :



La puissance reçue vaut en dB :

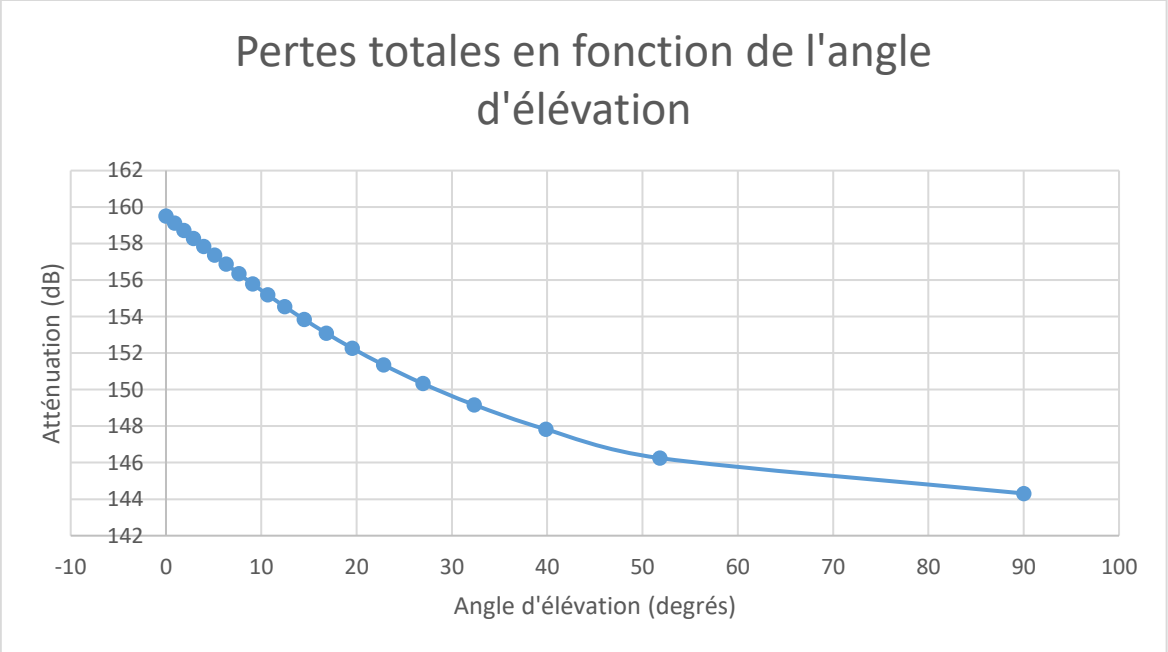
$$\boxed{P_{reçue} = P_{émise} + G_{sat} + G_{sol} - Pertes}$$

Le terme de pertes comprenant l'ensemble des pertes : polarisation, atmosphère, ionosphère, câbles, etc.

En ajoutant le gain du LNA et les pertes engendrées par le câble entre le LNA et l'USRP et l'USRP lui-même, on obtient en sortie de l'USRP :

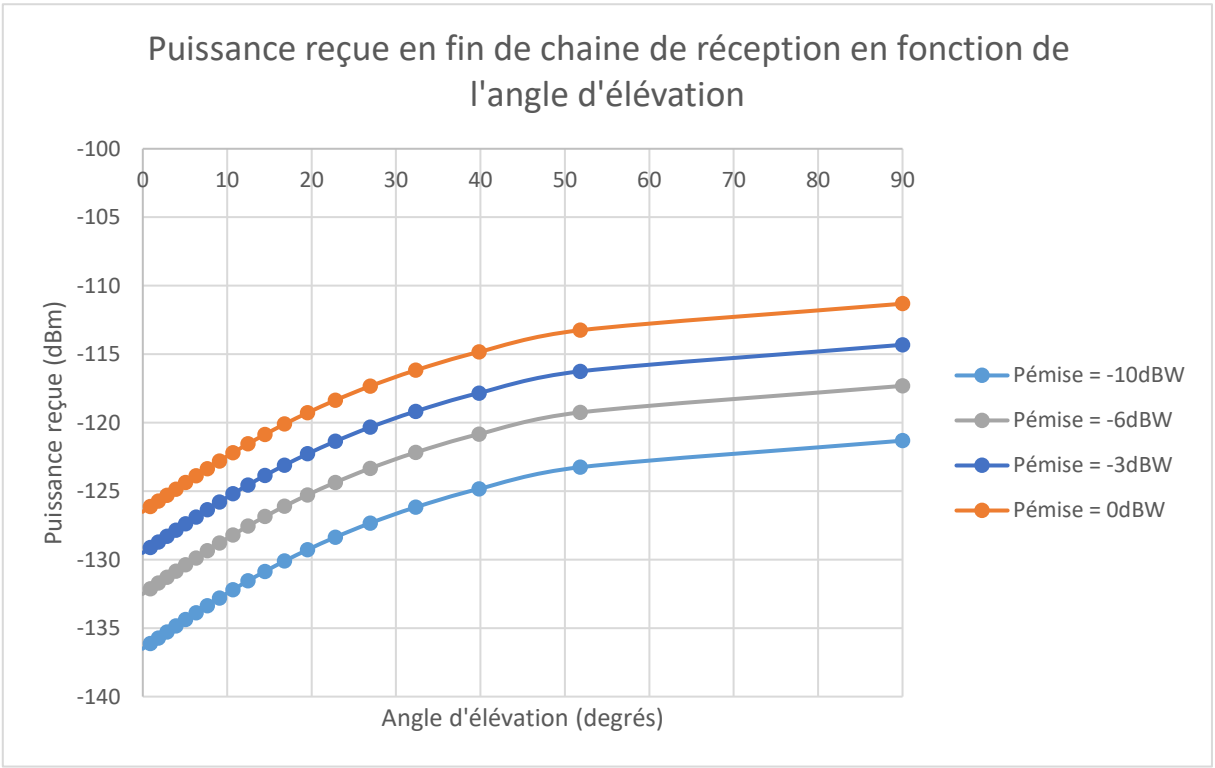
$$P'_{reçue} = P_{émise} + G_{sat} + G_{sol} - Pertes' + G_{LNA}$$

On peut alors tracer l'ensemble des pertes jusqu'à l'USRP en fonction de l'angle d'élévation :



On fixe typiquement $G_{sat} = 0dBi, G_{sol} = 15dBi \; et \; G_{LNA} = 18dB$

On obtient alors, en fonction de l'angle d'élévation et de la puissance émise par le satellite :



Pour calculer la marge de liaison ainsi que le taux d'erreur binaire, il faut définir le « Carrier to Noise Ratio » - CNR :

$$\boxed{\frac{C}{N} = \frac{E_b}{N_0} . \frac{D}{B_w}}$$

Ce ratio correspond au « Signal to Noise Ratio » - SNR (rapport signal sur bruit) mais pour le signal modulé.

E_b est l'énergie par bit, $N_0 = k.T$ la densité spectrale de bruit ou $k =$ la constante de Boltzmann et T la température de bruit de l'ensemble de la chaîne de réception, D est le débit binaire et B_w la largeur de bande considérée.

On peut exprimer :

$$\frac{E_b}{N_0} = \frac{C}{N} . \frac{D}{B_w}$$

Il existe des courbes représentant le taux d'erreur binaire (BER) en fonction de ce rapport :

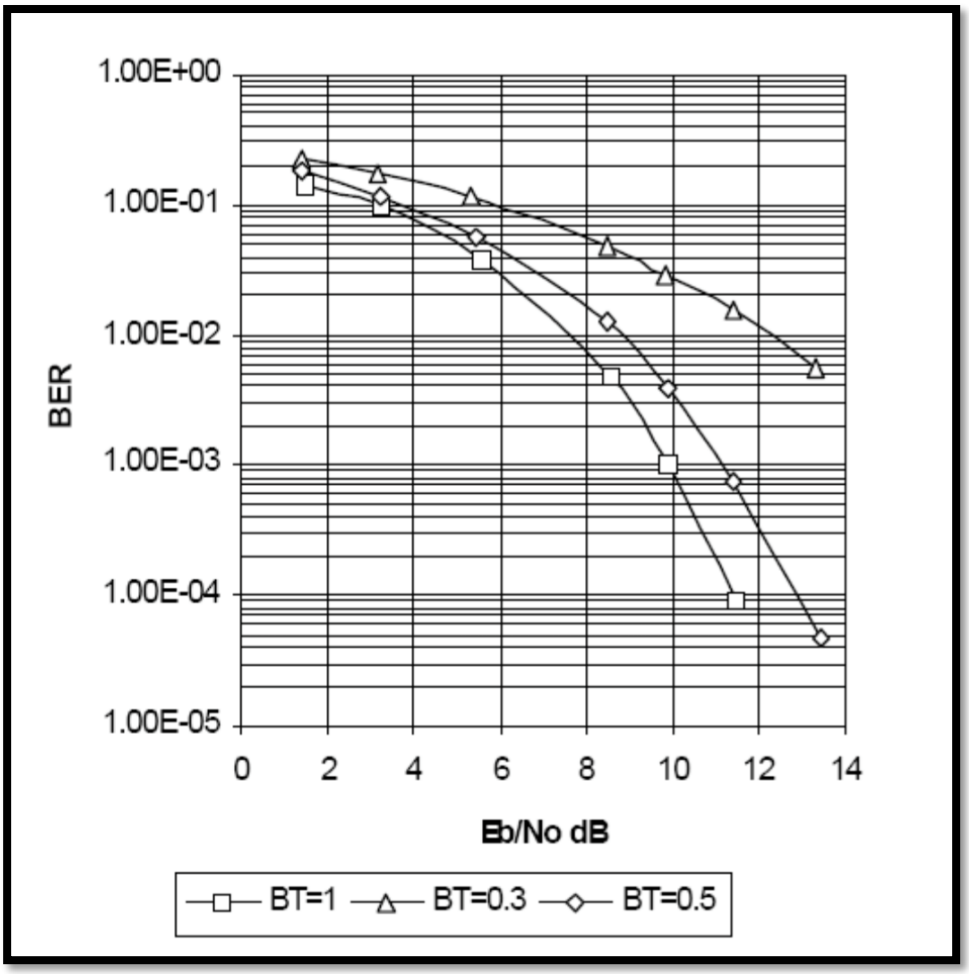


Figure 18. $BER = f\left(\frac{E_b}{N_0}\right)$ *pour un signal GMSK*

On peut donc estimer celui-ci. Dans notre cas, on a :

$$\boxed{\left.\frac{E_b}{N_0}\right|_{dB} = C_{dB} - N_{dB} + 10 * \log\left(\frac{D}{B_w}\right)}$$

avec $N_{dB} = 10 * \log(k.B_w.T)$ et $T \approx 1000K$, $B_w \approx 500kHz$, et $D = 9{,}6kbps$

On trouve dans ce cas un rapport $\left.\frac{E_b}{N_0}\right|_{dB}$ toujours supérieur à 22dB ce qui donne une marge de liaison pour un BER de 10^{-5} en GMSK de :

$$\boxed{LM = \left.\frac{E_b}{N_0}\right|_{dB} - \left.\frac{E_b}{N_0}\right|_{dB\ /\ BER=10^{-5}} > 7dB}$$

Pour la liaison montante, on sait que le bilan est au moins aussi favorable, puisque l'on peut techniquement émettre beaucoup plus de puissance que le satellite ne le peut, n'étant pas soumis, au sol, aux mêmes contraintes d'énergie.

B. Modulation

Les deux types de modulation qui peuvent être utilisées sur la carte communication du CubeSat (Texas Instruments **CC1101**) sont les modulations de fréquence et de phase 2-FSK et GMSK.

1. La modulation 2-FSK

La modulation FSK (Frequency Shift Keying) est une modulation de fréquence discrète. Les différents symboles sont représentés par différentes fréquences. Dans le cas de la 2-FSK, on utilise deux fréquences pour représenter des bits : la fréquence porteuse plus ou moins une déviation en fréquence.

On appel f_p est la fréquence porteuse et Δf la déviation de fréquence, on transmet :

$$\begin{cases} Un\ «1»\ lorsque\ f(t) = f_p\ + \Delta f \\ Un\ «0»\ lorsque\ f(t) = f_p - \Delta f \end{cases}$$

La FSK peut être cohérente ou non-cohérente.

Dans le cas de la FSK non-cohérente, il y a discontinuité dans la phase aux changements de fréquences. La FSK non-cohérente peut se réaliser par exemple, par simple commutation entre deux oscillateurs non synchrones.

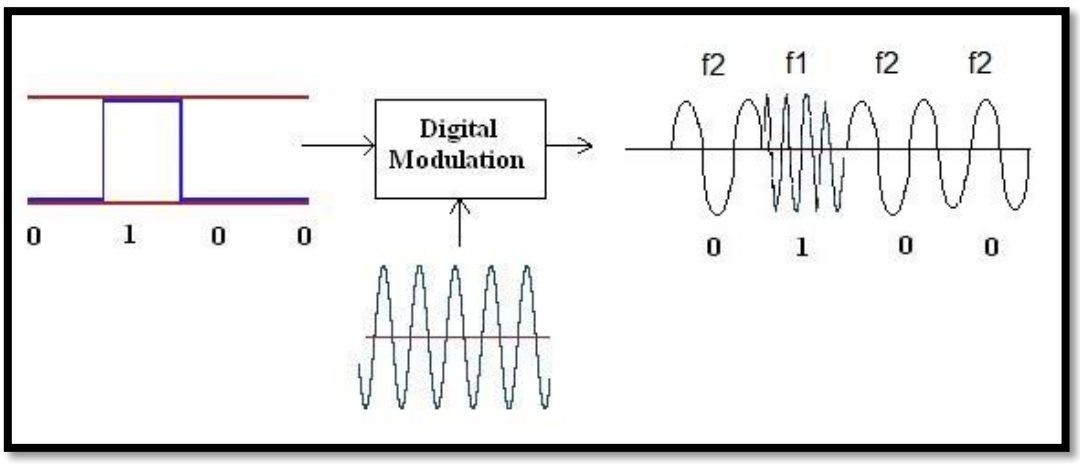


Figure 19. *2-FSK non cohérente*

Dans le cas de la FSK cohérente, il n'y a pas de discontinuité de phase lors des changements de fréquences. On réalise ce type de modulation en faisant par exemple, varier la fréquence d'une PLL (boucle à verrouillage de phase). C'est ce type de FSK qui est utilisé par la carte communication.

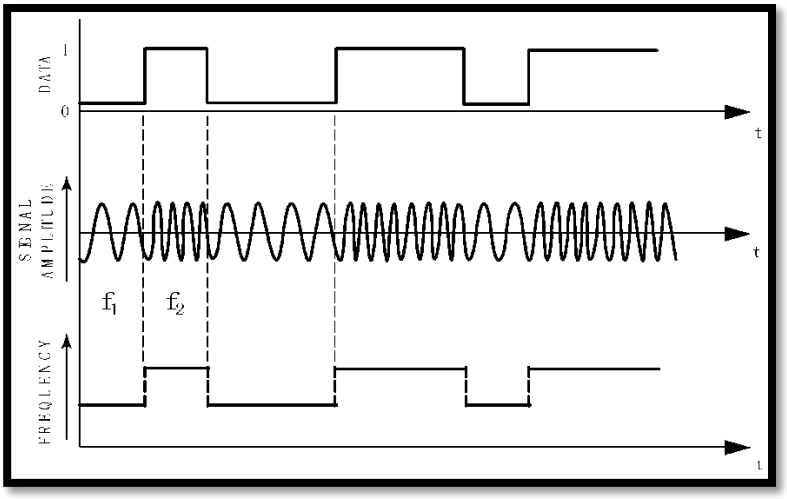


Figure 20. *FSK cohérente*

On définit l'indice de modulation m tel que :

$$\boxed{m = \frac{\Delta f}{f_m} = \frac{2 * \Delta f}{D}}$$

avec D le bitrate et $f_m = D/2$ la fréquence moyenne du signal modulant.

Remarque : Pour la fréquence envisagée ici (433MHz), cela donne une déviation de l'ordre de 10KHz.

L'indice de modulation va influencer l'encombrement spectral.

On peut déterminer la bande passante utile par la règle de Carson :

$$\boxed{B = 2 * (2.f_m + \Delta f)}$$

Au niveau fréquentiel, les discontinuités de phase de la FSK non-cohérente vont entraîner un élargissement du spectre.

Dans le cas $m = 0.5$, la largeur de bande du signal est minimale. On appelle cette modulation MSK (Minimum Shift Keying).

2. La modulation GMSK

Un des problèmes de la modulation FSK est sa faible efficacité spectrale, le spectre n'est pas suffisamment étroit, le niveau des lobes secondaires est trop élevé, et des débits binaires élevés induisent systématiquement des largeurs de canaux importantes. Le niveau des lobes secondaires est trop élevé.

La modulation GMSK permet d'améliorer cette efficacité spectrale en réduisant le niveau des lobes secondaires. La GMSK est utilisée dans des systèmes qui nécessitent un débit de données important et plusieurs canaux proches. Le GSM en est l'exemple le plus répandu.

Un filtre gaussien appliqué au train binaire à moduler permet de diminuer l'énergie des lobes secondaires. Ce filtre est caractérisé par la valeur du produit $B.T_b$ où B est la fréquence de coupure du filtre et T_b la durée d'un bit.

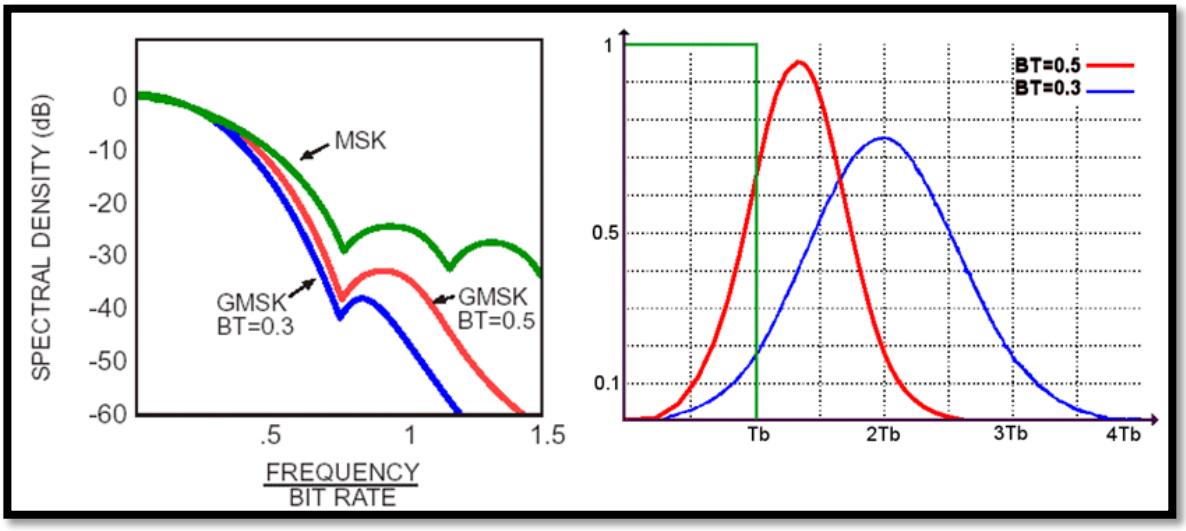


Figure 21. *Comparaison des efficacités spectrales et réponse impulsionnelle du filtre Gaussien*

On remarque que la valeur d'un bit influence la valeur du signal filtré sur plus d'une période Tb. Pour $B.T_b = 0{,}5$, l'influence s'étend sur environs 2 bits. Cette période d'influence donne naissance à un phénomène d'interférence inter-symbole (ISI) qui rend plus difficile la démodulation du signal.

Ainsi, une valeur faible de $B.T_b$ rend le spectre plus compact mais requiert un meilleur rapport signal/bruit (SNR) pour démoduler les bits. C'est pourquoi généralement on choisit $B.T_b = 0{,}5$.

A la sortie du filtre gaussien, on retrouve un signal NRZ dont les flancs ont été atténués plus ou moins fortement suivant la valeur de BT.

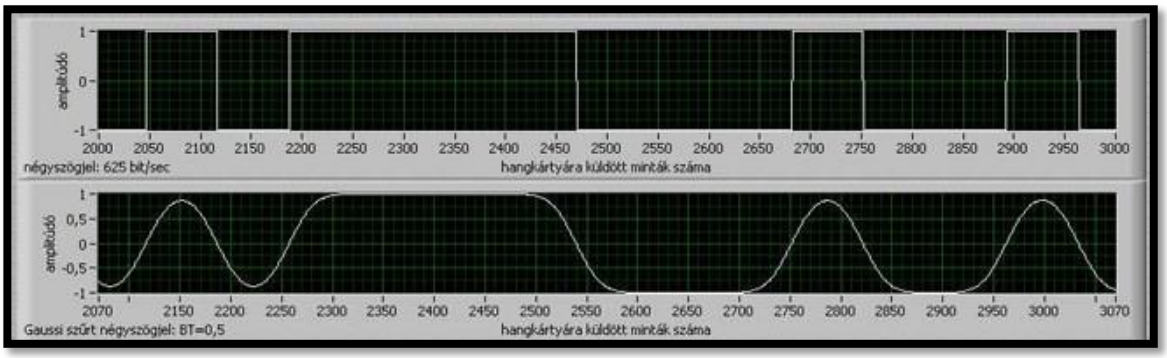


Figure 22. *Signal à la sortie du filtre gaussien*

3. Performance

On évalue la qualité d'une transmission numérique à l'aide du taux d'erreur binaire (BER pour Bit Error Rate). Ce BER est le rapport entre le nombre de bits erronés et le nombre de bits transmis. C'est le protocole utilisé et l'application qui vont imposer le BER minimum à atteindre pour que les données transmises soient exploitables.

Le BER dépend de la modulation et du rapport $\frac{Eb}{N_0}$. E_b est l'énergie par bit à l'entrée du démodulateur et N_0 est la densité de puissance du bruit.

$E_b = P_s . T_b$ et $P_N = N_0 . B$ avec $\frac{1}{T_b} = D$ le data rate et B la largeur de bande du signal.

De plus, la largeur de bande du signal est proportionnelle au data rate : $B = K . \frac{1}{T_b}$

$$\frac{E_b}{N_0} = \frac{P_s . T_b}{N_0} = \frac{P_s . K}{N_0 . B} = K . \frac{P_s}{P_N}$$

On réutilise le même graphe que précédemment :

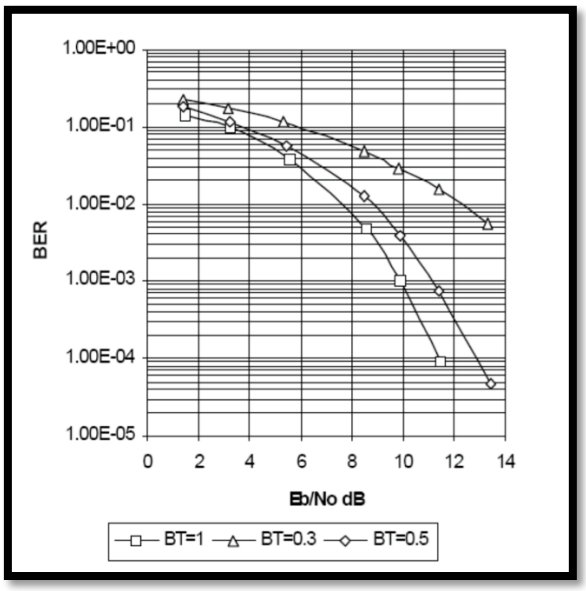


Figure 23. $BER = f\left(\frac{E_b}{N_0}\right)$ *pour un signal GMSK*

Il faut trouver un compromis entre efficacité spectrale et rapport signal/bruit…

On peut également exprimer le taux d'erreur par paquets (PER pour Packet Error Rate) :

$$\boxed{PER = f(L, BER) = 1 - (1 - BER)^m}$$

avec L la longueur du message en bytes.

C. Protocole de communication

Le protocole qui sera utilisé pour la communication avec le CubeSat est le AX.25, protocole radioamateur dérivé du X.25 (utilisé par le minitel), qui est largement utilisé pour les télécommunications avec les Cubesat.

Flag	AX.25 Transfer Frame Header (128 bits)				Information Field	Frame-Check Sequence	Flag
	Destination Address	Source Address	Control Bits	Protocol Identifier			
8	56	56	8	8	32-2048	16	8

Figure 24. *Trame du protocole AX.25*

Il est également envisageable d'utiliser le protocole NGHam, protocole lui-même basé sur le AX.25 et modifié afin d'être plus robuste aux bruits et avoir une meilleure efficacité spectrale ainsi qu'un haut débit utile.

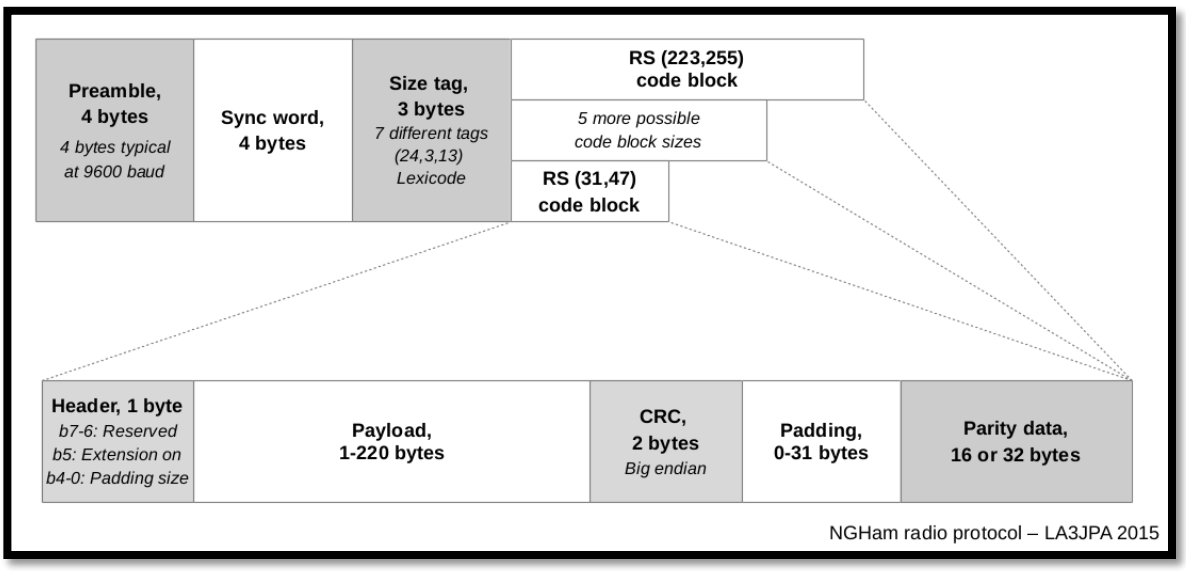


Figure 25. *Trame du protocole NGHam*

Nous souhaitons également implémenter un **codage source** ainsi qu'un **codage canal** (code cyclique ou convolutif) afin de rendre la communication plus efficace. Pour cela, nous sommes actuellement en train de nous concerter avec le groupe qui travaille au développement de la carte communication du cubesat.

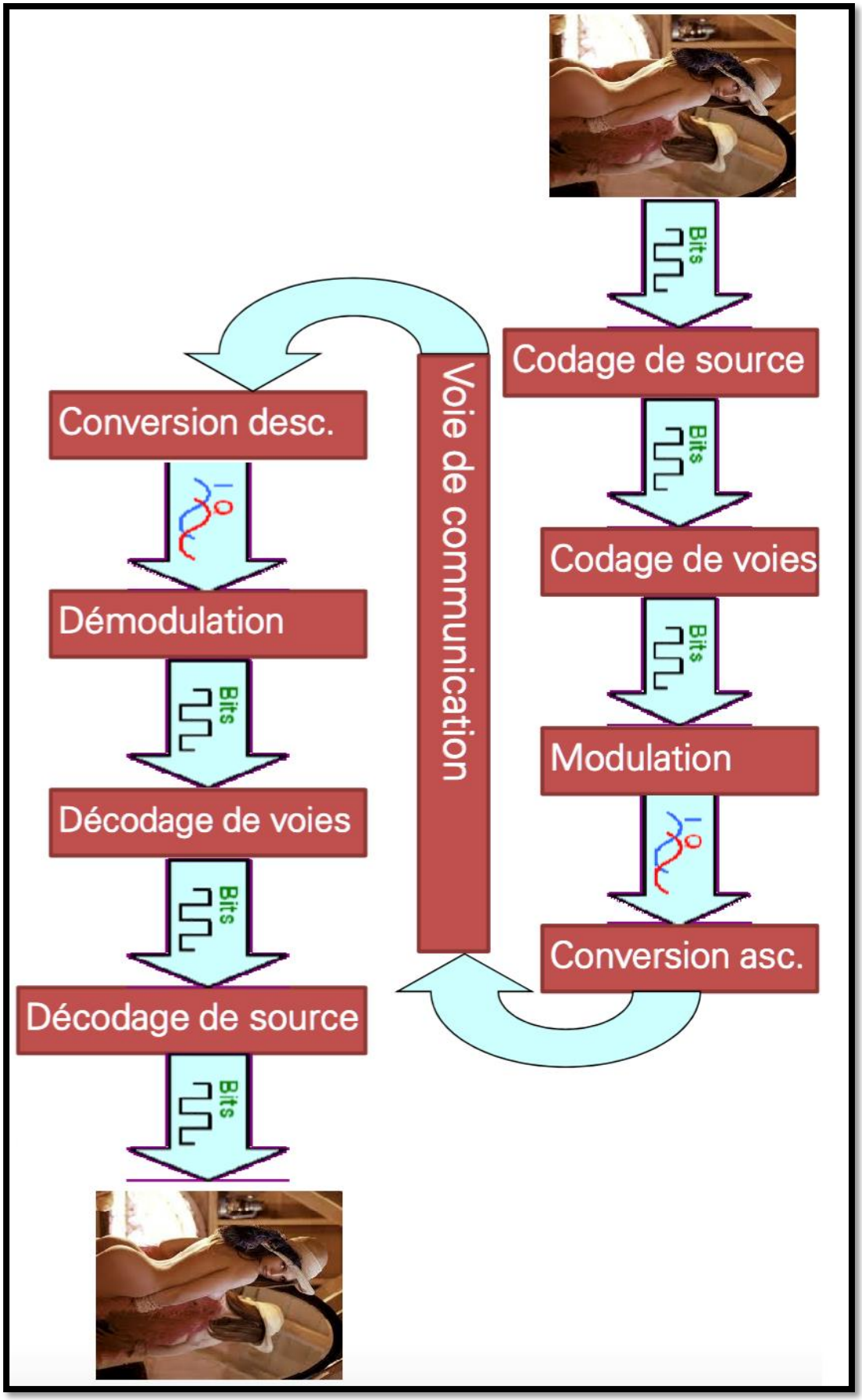


Figure 26. *Système de communication numérique*

IV. *Conception de la chaîne de réception*

A. Choix de l'antenne

1. Théorie

On utilise beaucoup d'antennes à polarisation circulaire pour les communications spatiales car même si l'onde émise par le satellite est polarisée linéairement, on peut réceptionner celle-ci avec 3dB de pertes dans toutes les positions du satellite. Si on utilisait des antennes à polarisation linéaire, il faudrait veiller à l'alignement des polarisations, ce qui n'est pas évident, afin d'éviter des pertes de signal trop importantes qui peuvent atteindre 30dB.

L'utilisation de la polarisation circulaire à une extrémité seulement donne une perte d'environ -3 dB. Afin de bénéficier de tous les avantages de la polarisation circulaire, les liens montant et descendant doivent l'utiliser. Le tableau ci-dessous montre la relation entre polarisations Horizontal, Vertical, RHCP (polarisation circulaire droite) et LHCP (polarisation circulaire gauche) et les pertes en dB (la polarisation RHCP est également connu sous le nom de polarisation CW (ClockWise) et la LHCP en tant que polarisation CCW (Counter ClockWise)).

	Horizontal	Vertical	RHCP	LHCP
Horizontal	0	30	3	3
Vertical	30	0	3	3
RHCP	3	3	0	30
LHCP	3	3	30	0

Figure 27. *Pertes selon les différentes polarisations*

2. Cahier des charges

Pour construire l'antenne de la station sol, adaptée au Cubesat de l'école, celle-ci doit respecter certains critères, à savoir :

- Fréquence centrale de 433,5MHz
- Polarisation circulaire
- Gain élevé (>=15dB)
- Directive

Pour cela, il existe principalement deux types d'antennes utilisables : l'antenne hélice axiale et l'antenne crossed-Yagi.

3. Antenne hélice axiale

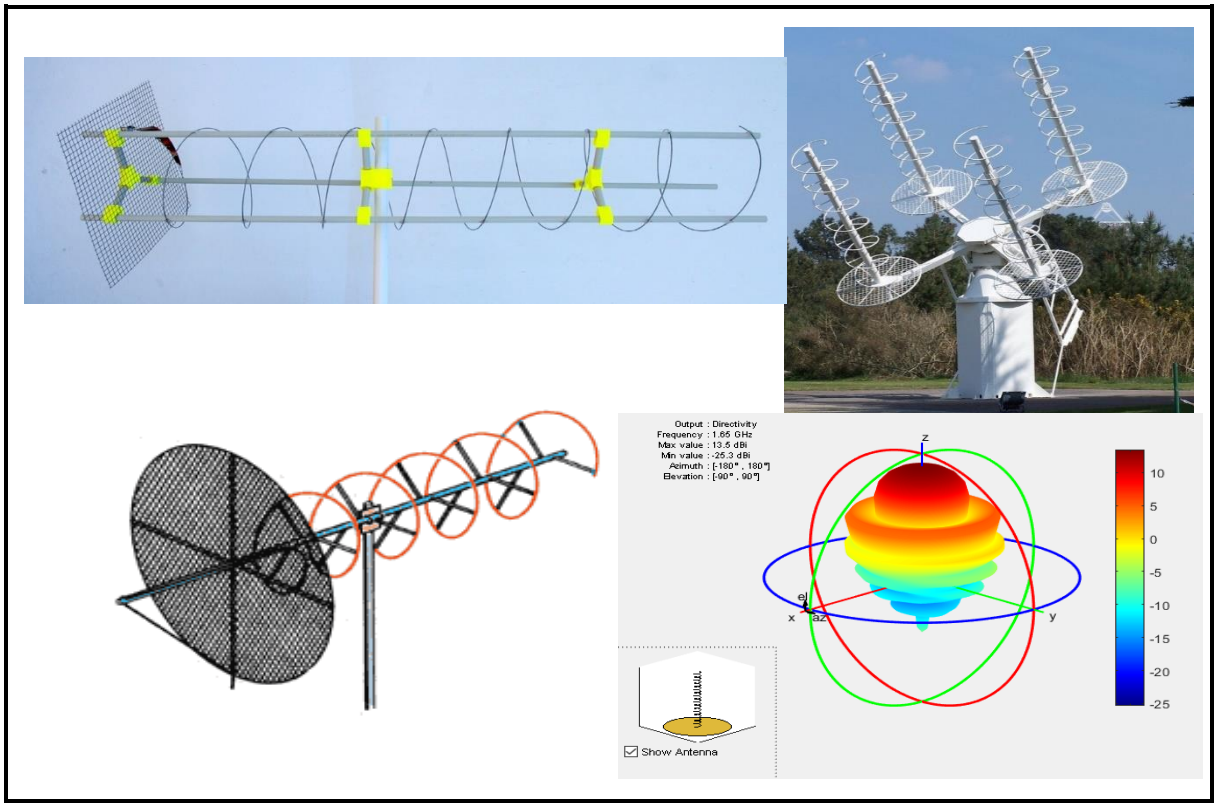


Figure 28. *Exemples d'antennes hélices axiales et diagramme de directivité typique*

Avantages : facilité de construction ; coût relativement faible ; possibilité de gain assez élevé si assez grande ; bonne directivité

Inconvénients : Polarisation donnée non modifiable ; pas de modèle déjà disponible à Supélec

4. Antenne Crossed-Yagi

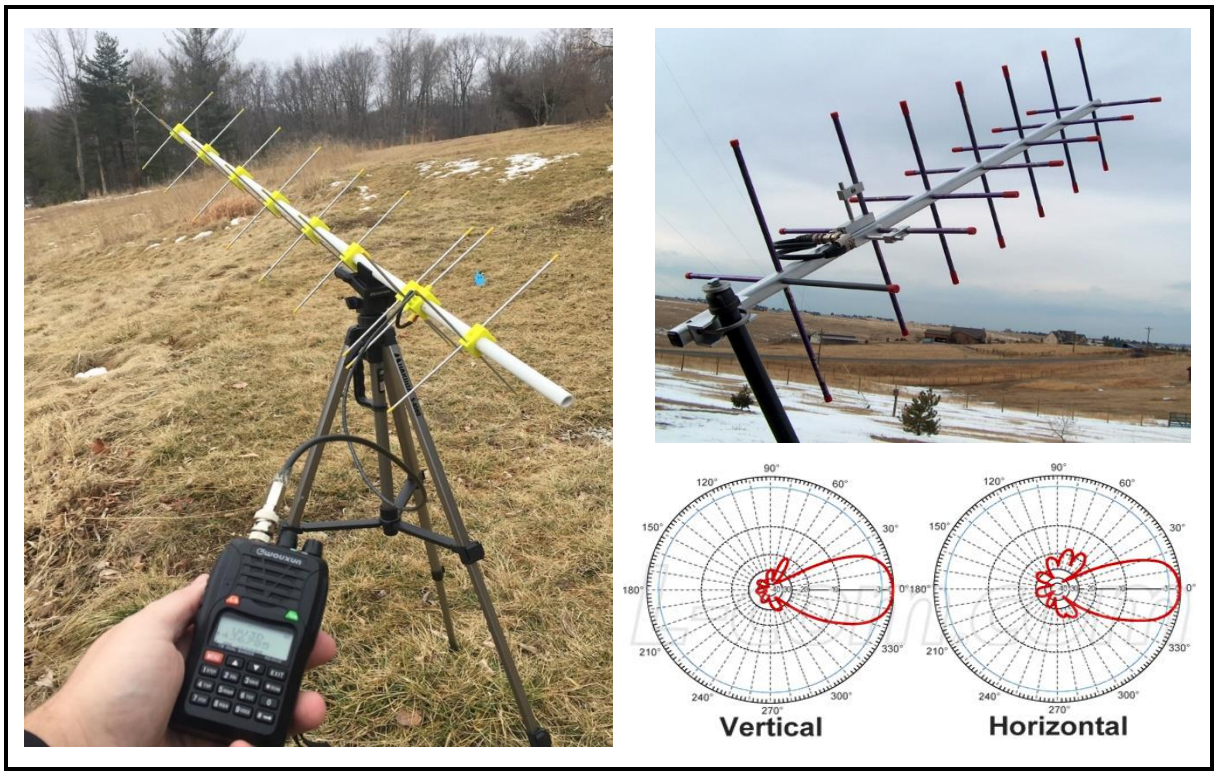


Figure 29. *Exemples d'antennes Yagi croisées et diagramme de directivité typique*

Avantages : Antenne Yagi linéaire à la bonne fréquence déjà disponible à Supélec ; gain élevé ; possibilité de switcher entre les deux polarisations (RHCP & LHCP) ; bonne directivité

Inconvénients : Fabrication plus technique ; coût légèrement plus élevé

L'antenne hélice axiale est naturellement polarisée circulairement. En ce qui concerne la Crossed-Yagi, il s'agit de deux antennes Yagi différentes disposées à 90°, il faut donc les « coupler » afin d'obtenir la polarisation circulaire souhaitée. Pour cela, il y a deux étapes à effectuer :

- **L'adaptation des antennes pour en faire une unique antenne adaptée également**

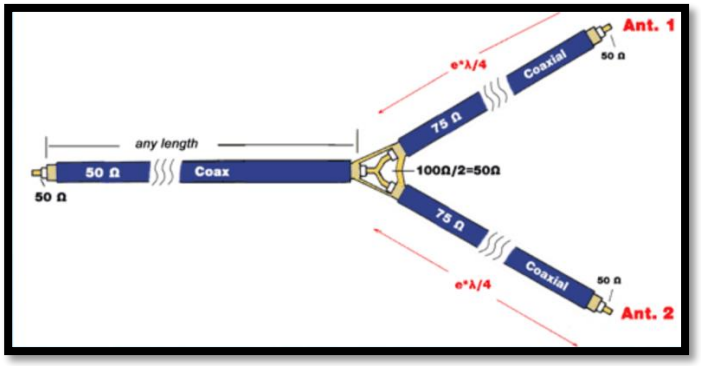


Figure 30. *Matching stubs*

- **Il faut avoir un retard de phase (phase delay) afin de générer une polarisation circulaire**
  Pour cela deux méthodes :

  - *Méthode électrique* (à l'aide d'un stub) :

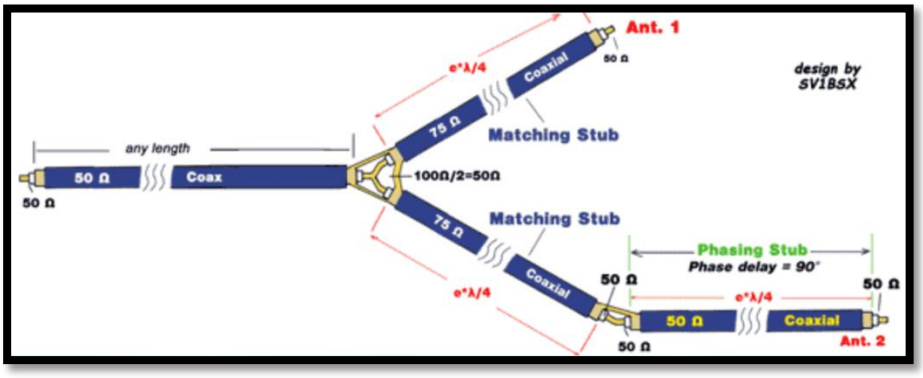


Figure 31. *Méthode électrique*

  - *Méthode physique* (en espaçant judicieusement les éléments):

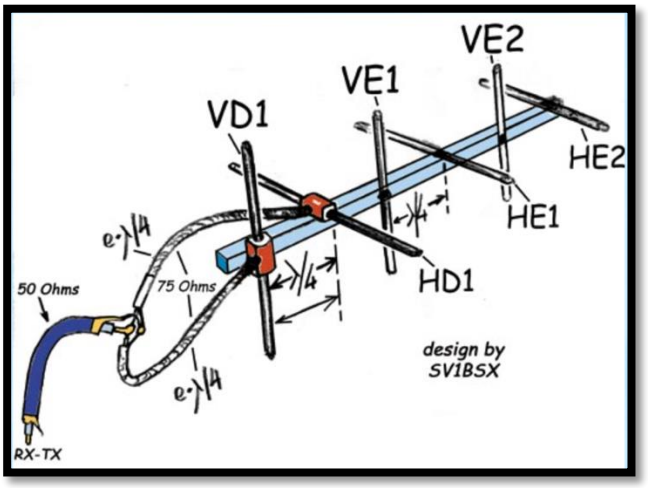


Figure 32. *Méthode physique*

B. LNA

Afin de garantir la qualité du signal qui sera traité numériquement, il faut impérativement ne pas l'abîmer après sa réception par l'antenne. Mais ce signal au niveau de l'antenne est tellement faible (-145dBm) qu'il faut tout de même l'amplifier. D'où l'utilité du LNA (Low Noise Amplifier), qui est un amplificateur à faible bruit, c'est-à-dire qu'il amplifie très bien le signal dans la bande de fréquence pour laquelle il est conçu pour fonctionner tout en ajoutant très peu de bruits (facteur de bruit < 1).

La bande de fréquence dans laquelle nous travaillons étant la bande ISM (433MHz), nous avons cherché des LNA fonctionnant dans cette bande et possédant de bonnes performances.

	Down East Microwave L432LNA (GaAs pHEMPT)	Advaced Receiver P432VDG (GaAsFET)	DG0VE LNA70-1	RF Bay LNA-750(GaAsFET)
Gain	17 dB	18 dB	17-19 dB	40 dB
Facteur de bruit	< 0,5	0,5	0,4-0,45	0,5
P1dB	+19 dBm	+12 dBm	?	+19 dBm
Prix	85 $	100 $	80 €	260 $

Figure 33. *Comparaison de LNA*

Il est indispensable d'acquérir un de ces LNA ou un autre disposant de caractéristiques similaires.

C. Interface de radio logicielle : l'USRP

L'interface que nous allons utiliser est celle disponible à l'école, à savoir une carte USRP (pour Universal Software Radio Peripheral). Le modèle à notre disposition est l'**USRP N210**. Il s'agit d'une interface possédant une architecture modulaire permettant de travailler sur une très large bande de fréquence (DC-6GHz) qui inclut bien évidement la nôtre.

Le modèle utilisé bénéficie d'une connexion par câble Internet Gigabit, qui lui permet de pouvoir effectuer des communications en full duplex jusqu'à 25MS/s (MS = Mega Sample) de données complexes I/Q codées sur 16 bits.

Cette interface peut être utilisée sous les logiciels **GNU Radio**, LabVIEW et Simulink.

Figure 34. *USRP N210*

En suivant une architecture courante de radio définie par logiciel, le matériel NI USRP offre un frontal analogique de conversion directe avec des convertisseurs analogiques/numériques (C A/N) haute vitesse et des convertisseurs numériques/analogiques (C N/A) dotés d'un FPGA à fonctionnalité fixe pour les étapes de conversion par abaissement de fréquences numériques (DDC) et conversion par élévation de fréquences numériques (DUC). La chaîne du récepteur commence par un frontal analogique très sensible capable de recevoir de très petits signaux et de les numériser en utilisant la conversion par abaissement de fréquences directe vers des signaux I et Q en bande de base. La conversion par abaissement de fréquences est suivie d'une conversion analogique/numérique haute vitesse et d'une conversion par abaissement de fréquences numériques qui réduit la vitesse

d'échantillonnage et rassemble I et Q pour la transmission vers un PC hôte, en utilisant une liaison Gigabit Ethernet pour un traitement ultérieur. La chaîne de transmission commence par le PC hôte là où les I et Q sont générés et transférés sur le câble Ethernet vers le matériel NI USRP. La conversion par élévation de fréquences numériques (DUC) prépare les signaux pour le C N/A après lequel le mélange I-Q a lieu pour directement convertir par élévation les signaux afin de produire un signal de fréquence RF, qui est ensuite amplifié et transmis.

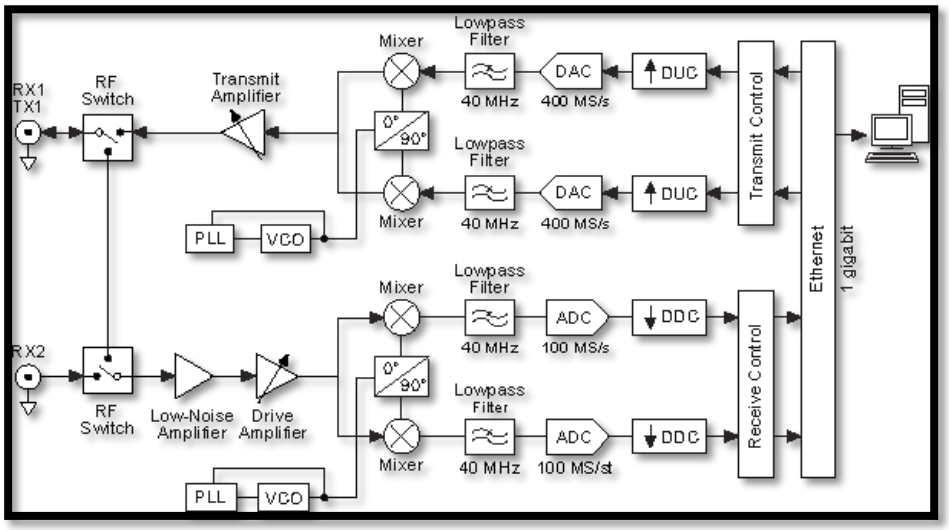


Figure 35. *Diagramme au niveau système*

Supélec dispose de plusieurs cartes USRP connectées en réseau, ainsi que d'un Octo-clock permettant de les synchroniser. On peut accéder à ces cartes en utilisant un logiciel de contrôle à distance afin de se connecter à l'ordinateur sur lequel elles sont branchées.

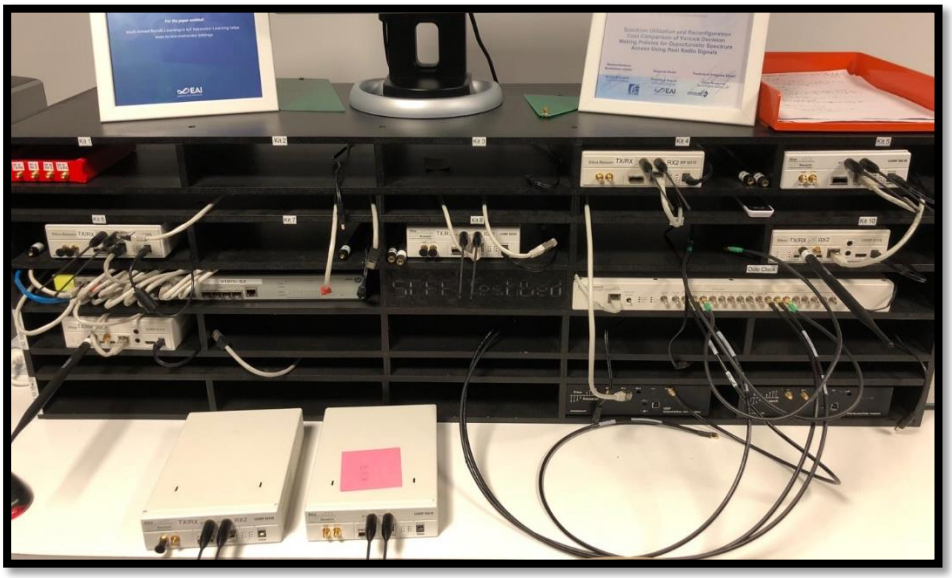


Figure 36. *USRP de Supélec*

D. Radio logicielle

La radio logicielle à partir de laquelle nous allons effectuer les traitements de signaux est GNU Radio, qui est un logiciel gratuit et multiplateforme.

On peut créer des chaînes de traitement du signal graphiquement en utilisant l'interface du logiciel GNU Radio - Companion, qui crée des graphiques « flowcharts » ; ou en codant sous python (le flowchart générant lui-même un code python qui est ensuite exécuté).

1. Prise en main

Afin de prendre en main ce logiciel, nous avons effectué quelques essais simples.

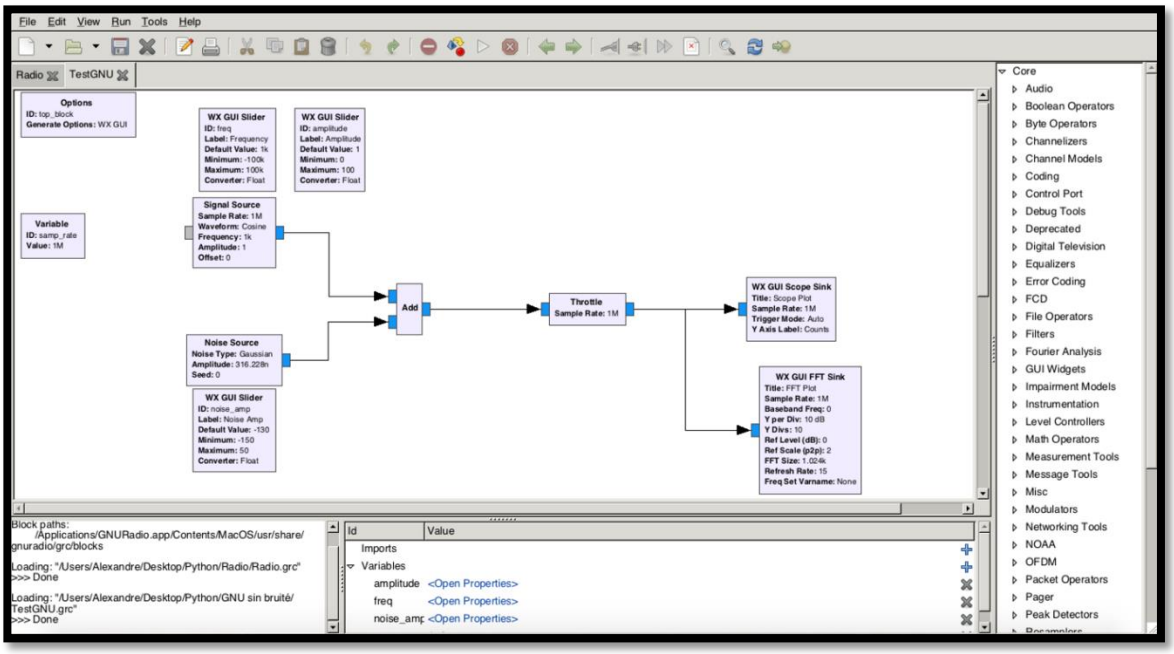


Figure 37. *Génération d'un cosinus et de sa transformée de Fourier*

Ce premier schéma permet de créer un cosinus dont on peut faire varier fréquence et amplitude, ainsi que d'ajouter à ce cosinus un bruit blanc gaussien ; puis de visualiser le signal somme et sa FFT.

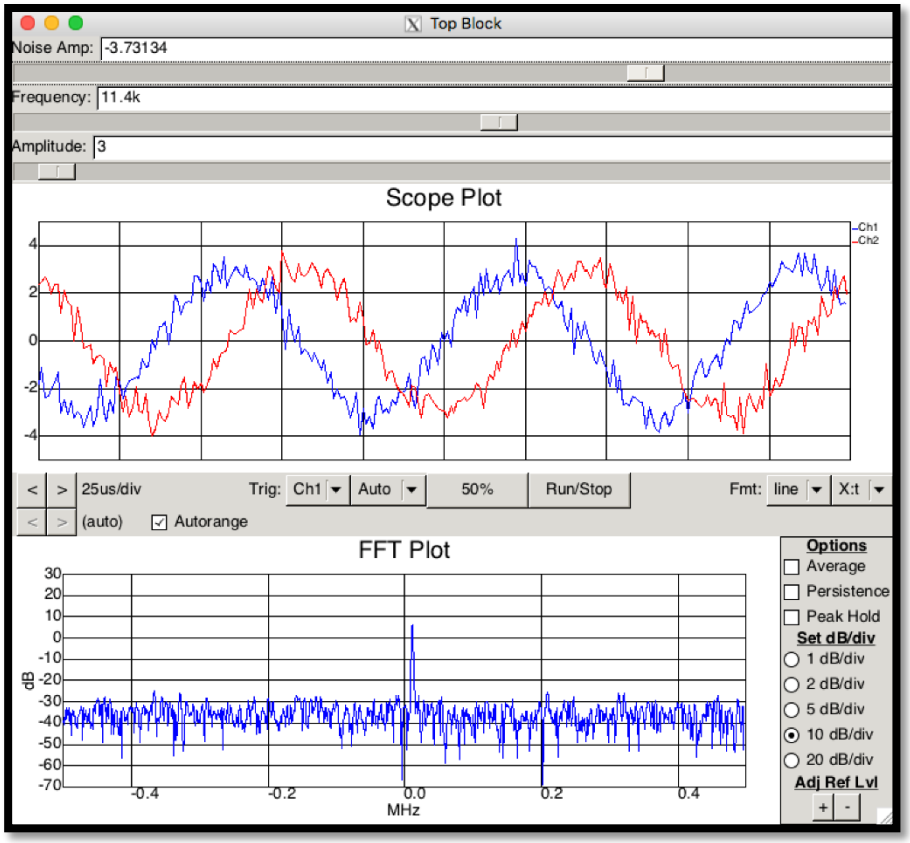


Figure 38. *Visualisation du cosinus et de sa transformée de Fourier*

Après avoir testé le fonctionnement des variables et compris la méthode d'implémentation, nous avons ensuite développé une radio FM utilisant un dongle SDR (RTL2832U) destiné à la réception de la radio et de la télé numérique.

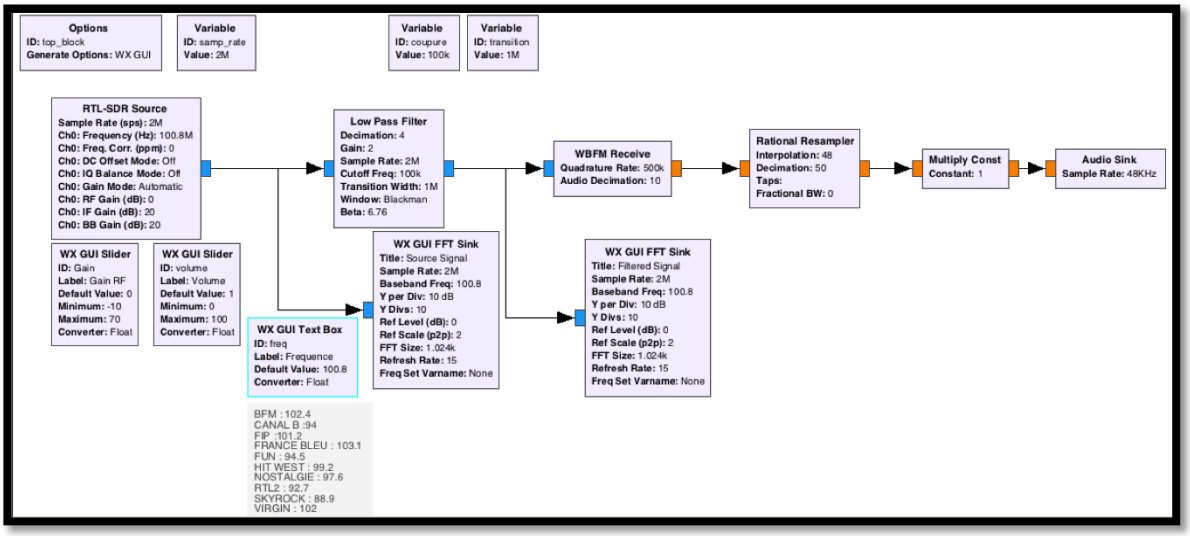


Figure 39. *Chaine de réception FM*

Nous avons ainsi pu tester la réception de la radio suivant le gain que l'on impose. Cette radio fonctionne très bien.

Nous avons alors commencé à tester la modulation en GMSK afin de tenter de calculer un BER selon le modèle de chaîne de transmission. Cependant, le bloc Error Rate étant obsolète, les résultats obtenus n'ont aucun sens…

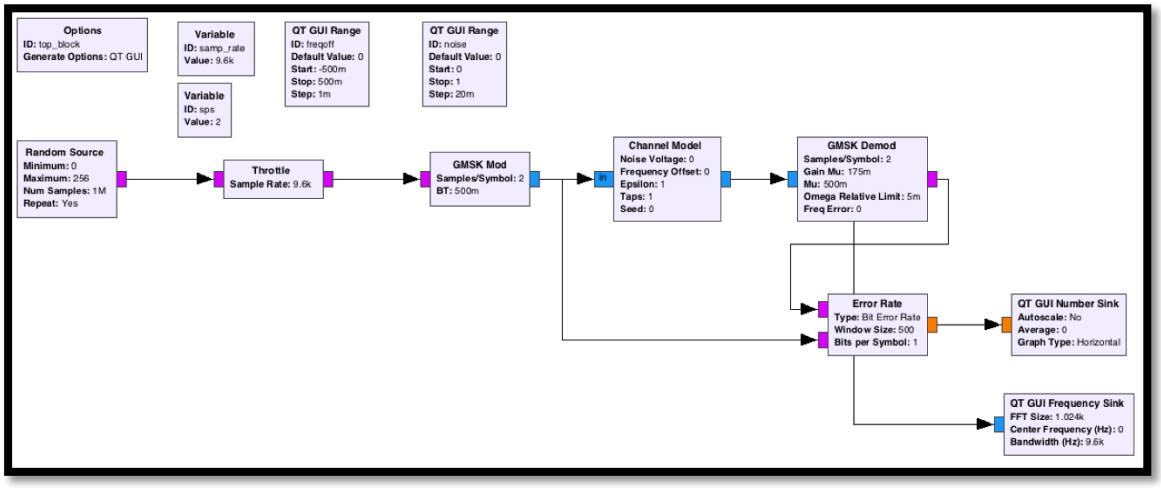


Figure 40. *Essais avec le bloc obsolète Error Rate*

Après cette rapide prise en main de GNU radio, nous avons obtenu les codes d'accès à distance à l'ordinateur de Supélec où sont branchés les USRP et qui dispose de GNU Radio, pour commencer les tests sur ce matériel, dans l'objectif d'implémenter la chaîne de réception finale.

2. Utilisation de l'USRP

Nous avons donc aussi fait des tests d'émission et de réception grâce aux USRP. Pour cela, nous avons créé sur GNU-Radio deux chaînes : une d'émission où un USRP est utilisé en tant qu'émetteur (USRP Sink) et une de réception, où un autre USRP est utilisé en tant que récepteur (USRP Source). La modulation utilisée est le GMSK, ce qui sera probablement la modulation utilisée par Supsat, et les données ont été empaquetées grâce au « Packet Encoder » de GNU radio, qui offre des caractéristiques basiques d'empaquetage (header, code d'accès et préambule).

Nous avons réalisé ce premier essai en envoyant une image au format jpg (la fameuse Lena…).

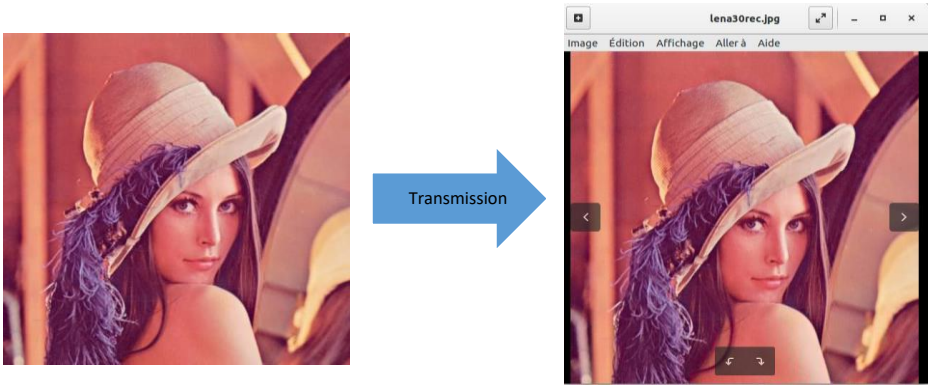


Figure 41. *Transmission de lena.jpg (photo originale et après transmission)*

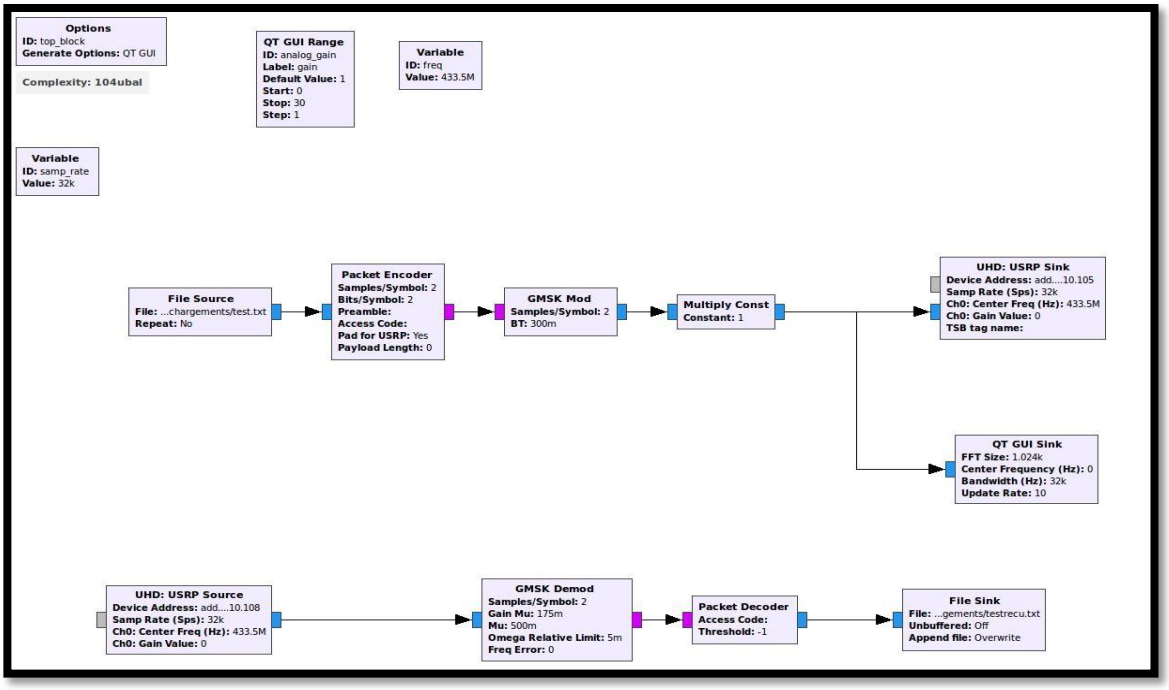


Figure 42. *Schéma GNU-Radio utilisé*

Dans cette configuration (les deux USRP sont très proches, quelques dizaines de centimètres), les erreurs semblent tellement minimes qu'elles sont imperceptibles à l'œil. Nous avons donc souhaité calculer les erreurs de bits malgré l'obsolescence du bloc « Error Rate ». Pour réaliser cette comparaison des deux images, émise et reçue, nous les avons importées dans Matlab afin de calculer l'erreur sur les pixels.

```
BERT.m | readjpg.m | +
1 -   clear all;
2
3 -   Lena=imread('lena30.jpg');
4
5 -   Lena_rec=imread('lena30rec.jpg');
6
7 -   bert = sum( sum( Lena ~= Lena_rec ) ) / numel(Lena);
```

Figure 43. *Script Matlab pour calculer l'erreur sur les pixels*

```
bert(:,:,1) =

    0.2011


bert(:,:,2) =

    0.1840


bert(:,:,3) =

     0
```

Figure 44. *Résultat*

On constate qu'il y a bien eu des erreurs lors de la transmission, mais que celles-ci restent minimes pour le résultat visible à l'œil nu.

Il nous faut alors développer une chaîne d'émission-réception robuste qui sera capable de décoder un signal très faible provenant du Cubesat, incluant la modulation GMSK et les codages source et canal selon l'arrangement avec le groupe qui s'occupe de la carte communication du Cubesat.

Nous effectuerons alors des tests de réception en utilisant soit directement la carte communication (Texas Instruments CC1101), soit un émetteur du type RFM69 qui peut émettre sur la bande de fréquence souhaitée (433MHz) ; en faisant rayonner une puissance très faible afin de simuler les niveaux de puissance qui seront obtenus lorsque le Cubesat sera en orbite (voir bilan de liaison).

Conclusion

Pour conclure, de nombreux problèmes ont été découverts bien que le projet fut avancé l'an passé. Nous avons cependant pu apporter des solutions aux anomalies et lacunes du projet précédent.

Ainsi, le mécanisme de mouvement de la station sol fonctionne correctement, son contrôle se simplifie et gagne en autonomie, et une chaîne de réception viable est imaginée.

Cependant, ce projet n'est pas terminé. En effet, la station doit encore être stabilisée mécaniquement pour gagner en stabilité, le développement de l'aspect logiciel du contrôle de la station est en cours et la chaîne de réception complète doit encore être réalisée avec le choix très important de l'antenne et du LNA. Le développement de la partie radio logicielle concernant l'émission et la réception des données reste également à faire.

Malgré les difficultés rencontrées, ce projet reste fort intéressant par son aspect pratique comprenant une véritable finalité plus que motivante. La multidisciplinarité qu'impose la conception d'une station sol, à savoir concernant l'électronique, la mécanique et la télécom principalement, nous permet d'approfondir nos compétences et méthodes afin de découvrir des solutions viables à des problématiques réelles. L'acquisition de ces connaissances sera un atout certain dans notre future carrière professionnelle.

Annexes

A. TLE (Two Lines Element) ou Paramètres Orbitaux

Ces paramètres décrivent la trajectoire de nombreux objets spatiaux (satellite, nano satellites, débris, etc..). Ils sont régulièrement mis à jour par les agences spatiales, notamment la NORAD (North American Aerospace Defense Command).

On peut retrouver ces paramètres ici : *http://celestrak.com/NORAD/elements/*

Par exemple, pour le SwissCube de l'EPFL, on retrouve :

```
SWISSCUBE
1 35932U 09051B   18043.92525997  .00000102  00000-0  33982-4 0  9993
2 35932  98.5251 192.2353 0006949 224.9949 135.0686 14.56119658445318
```

L'interprétation de ces lignes est décrite sur Wikipédia :
https://fr.wikipedia.org/wiki/Param%C3%A8tres_orbitaux_%C3%A0_deux_lignes

B. Caractéristiques des composants

1. Moteurs pas-à-pas

NEMA 17 Stepper Motor 17HS3001-20B

200 steps per revolution (1.8 deg/step)
2 Phase bipolar 4 wires
Rated Voltage 2V DC (not working voltage, no problem to use higher voltage)
Rated Current 1.2A
Phase Resistance: 1.7 Ohm ± 10% (20°c)
Phase inductance: 4.5 mH ± 20% (1kHz 1 V rms)
Holding torque: 0.4 N.m Min

2. Shield Arduino

Adafruit Motor Shield v2.3
4.5-13.5 VDC Motor Power
Communication I2C (0x60)
1.2A par moteur pas-à-pas

3. Convertisseur DC/DC

OKX T/10 - D12 Series
Adjustable DOSA 10-Amp SIP DC/DC Converters
8.3-14 VDC input voltage range
Programmable output voltage from 0.7525-5.5Vdc

4. Autres

RTC : DS1307
IMU : MPU9250

C. Diagramme global de la liaison descendante

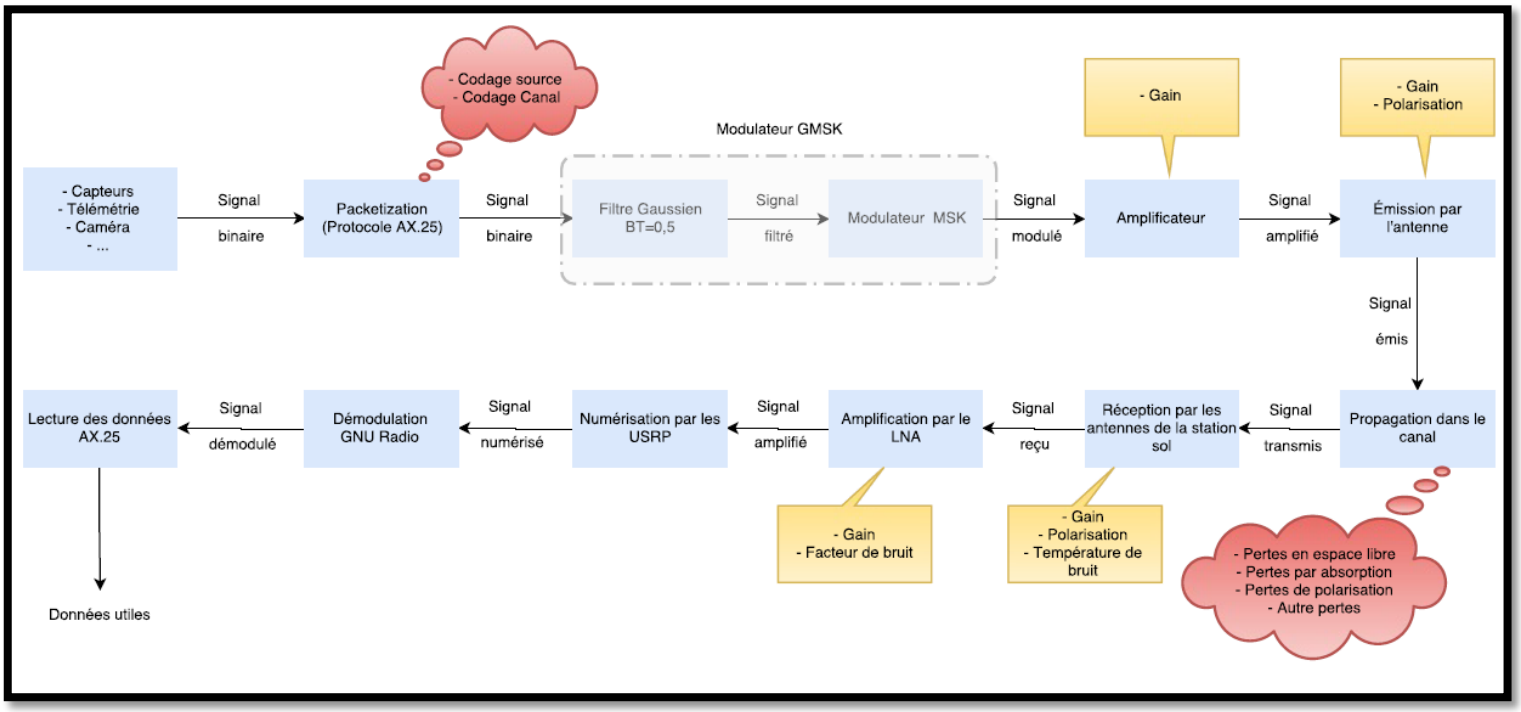


Figure 45. *Diagramme global de la liaison descendante*

Sources

http://www.qsl.net/sv1bsx/antenna-pol/polarization.html

http://nuts.cubesat.no/om

https://satnogs.org

https://www.rtl-sdr.com/

http://www.ti.com/product/CC1101

https://www.ettus.com/

https://wiki.gnuradio.org/i

https://github.com/zacmanchester/kicksat/wiki/Setting-Up-A-Ground-Station

http://www.space.aau.dk/cubesat/

https://fr.mathworks.com/

http://swisscube.epfl.ch/

http://www.n2yo.com/satellites/?c=18

http://cubestar.no

http://igosat.in2p3.fr/fr/page-d-exemple/

https://services.renater.fr/ntp/serveurs_francais

http://celestrak.com/NORAD/

http://exploreembedded.com/wiki/

https://esp-idf.readthedocs.io/

https://github.com/nkolban/esp32-snippets/

https://github.com/AnalogLamb/ESP32-ApplicationNote/

https://github.com/espressif/esp-idf/

http://www.zeptomoby.com/satellites/

https://www.n2yo.com/