



LORASPACE

Réalisé par :

- AMAZOUZ Sofiane
- BENOMAR Seifedinne
- CHEBIHI Hichem
- CHEMALI Redha

M2PGI

2017-2018

Sommaire :

- Conception et architecture générale du projet
- Réalisation du support
- Configuration de la chaine
- Technologies utilisées
- Fonctionnement de l'application
- Visualisation temps réel des données avec JYSE
- Simulation de la communication bidirectionnelle
- Difficultés rencontrées
- Conclusion

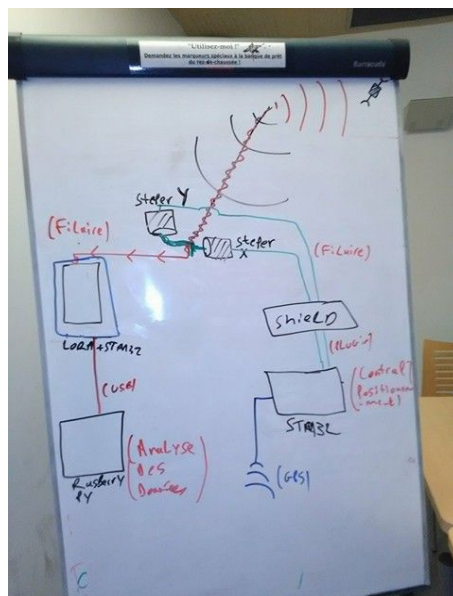
Context :

Le but de ce projet est de réaliser un prototype qui permet de préparer le terrain pour imaginer une communication bidirectionnelle basse consommation d'énergie pour les nano-satellites en utilisant des moyens peu coûteux comme le protocole Lora .

Afin de réaliser ceci nous avons commencé un long cycle de réflexion pour imaginer l'ensemble des éléments de la chaîne afin que la circulation des données se fait de la façon la plus optimal , vous trouverez dans ce document l'ensemble des étapes pour la réalisation de notre projet

Conception et architecture générale du projet

Le choix de l'architecture et des outils est très important pour la suite c'est pourquoi on a consacré quelques jours à la documentation sur le sujet et les diverses techniques et bonnes pratiques utilisé, la première architecture n'a pas tardé à voir le jour



Après avoir constaté quelque défaillance et après consultation et quelque conseil de la part du prof nous avons conclu sur une **architecture finale** qui sera comme suit :

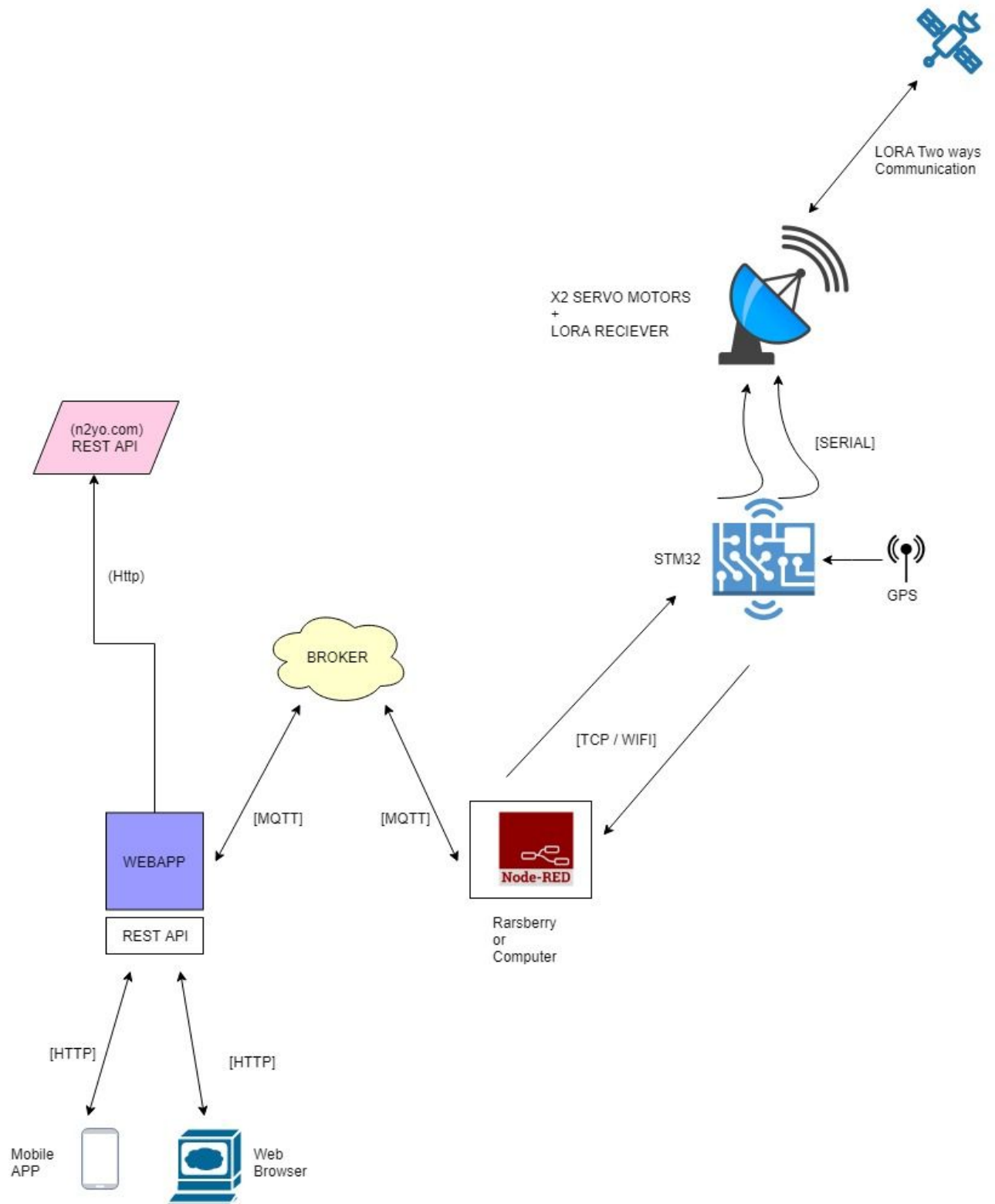


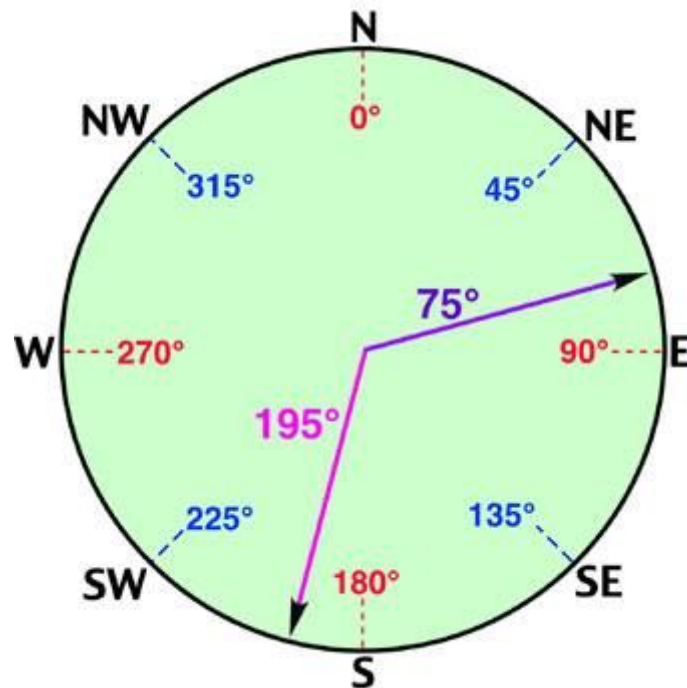
Figure 1 : "Architecture générale du projet"

Réalisation du support :

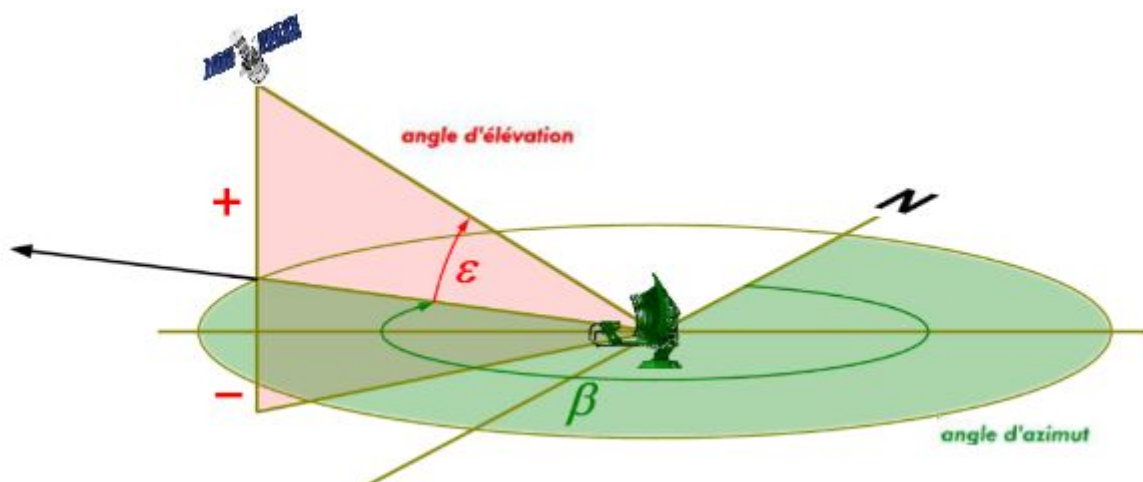
Dans cette partie nous allons détailler l'ensemble du processus pour réaliser la structure .

1 Conception et plans :

Tout d'abord il faut savoir comment orienter l'antenne vers un satellite et pour se faire il nous faut deux angles le 1er l'Azimut qui est l'angle formé par la direction de l'antenne par rapport au nord de telle sorte que si l'antenne est orientée vers le nord l'angle égal à 0 et si elle est orientée vers l'est l'angle égal à 90 etc...,comme le montre la figure ci-dessous.



Le 2eme angle est l'angle formé par l'antenne et la surface de la terre et qu'on appelle Élévation, les deux angles sont bien illustrés dans la figure ci-dessous où l'angle β est l'Azimut et l'angle ϵ est l'Élévation:



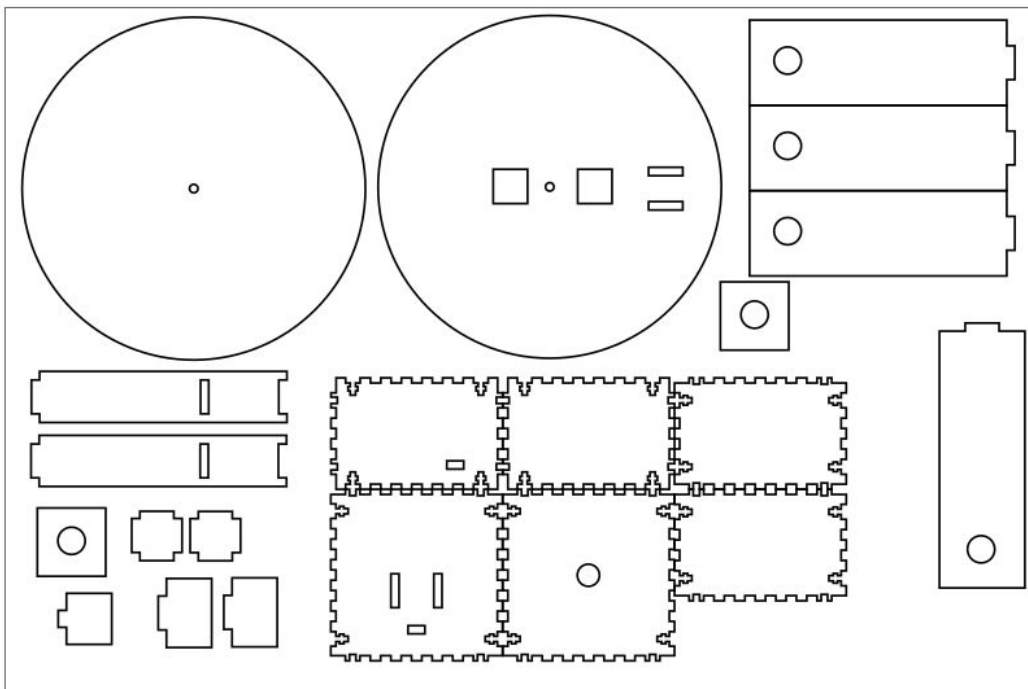
Dans un premier lieu nous avons imaginé plusieurs formes de structures pour permettre à l'antenne d'être orientée vers le satellite souhaité et finalement nous avons choisi une

structure à deux servomoteurs un pour former le 1er angle (Azimut) et le second pour former le 2ème angle (Élévation)



2 Réalisation:

Pour réaliser cette structure nous avons utilisé 2 logiciels **Inkscape** pour les pièces au-dessus de la base et **Makercase** pour la base qui est une boîte, ces deux logiciels nous ont permis de faire les plans ensuite passer à la machine qui découpe au laser.



Cette figure représente les pièces en bois de la structure .

Nous avons utilisé 2 servomoteurs de type tower pro mg995r avec une possibilité d'angle de 180 degrés.



Nous avons aussi utilisé:

- 2 tiges filetées de 5 mm
- 9 roulement 5*16*5 mm
- un roulement axial (butée à billes) de 50*70*14 mm.



Tige filetée 5mm



Roulement 5*16*5 mm



Roulement axial 50*70*14 mm

3 Fonctionnement :

Le premier servomoteur est à l'intérieur de la base et il est relié au disque horizontal, qui porte le reste de la structure ,avec une tige filetée pour le faire tourner en fonction de l'Azimut.

Le deuxième servomoteur ,qui est sur un support , est relié au disque vertical qui va porter l'antenne et ce servomoteur va faire tourner le disque en fonction de l'Élévation.

Les deux servomoteurs ont une possibilité de rotation de 180 degré mais aussi le sens de cette rotation est de droite à gauche alors que l'angle de l'Azimut est représenté de gauche à droite en prenant comme base le Nord donc nous avons implémenté une fonction (côté code source) qui nous permet d'avoir le bon Azimut et la bonne Élévation nous allons l'expliquer en deux exemples.

- **Azimut <= 180 :**

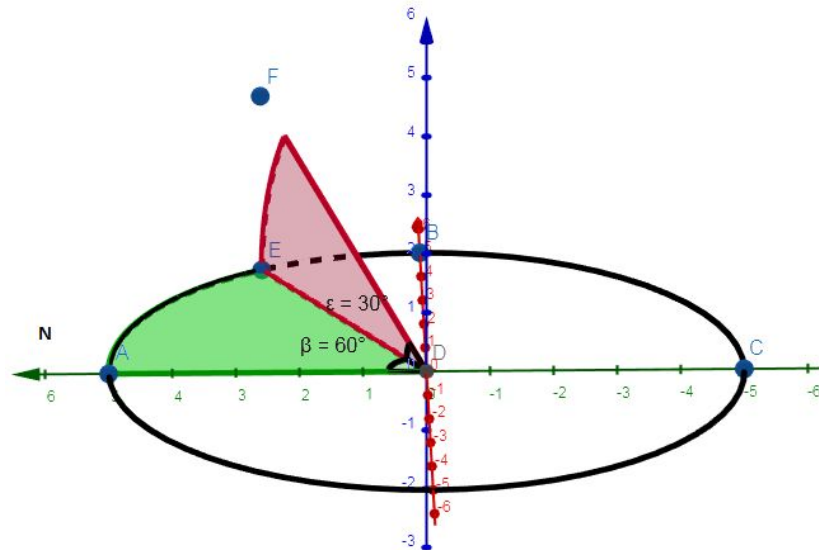
Si l'azimut est inférieur ou égal à 180 degrés alors on applique ces deux instructions :

Azimut=180 - Azimut;

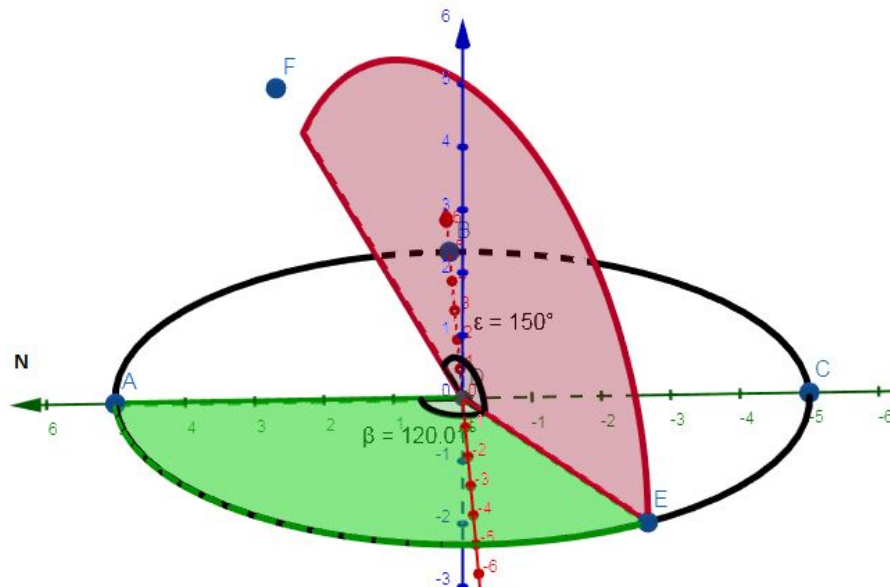
Élévation=180 - Élévation;

Exemple :

Un satellite avec l'Azimut = 60° et l'Élévation = 30° , pour bien orienter l'antenne au satellite il faut d'abord la pointer au nord ensuite la faire tourner de gauche à droite avec un angle de 60° ensuite de bas en haut avec un angle de 30° comme le montre la figure ci dessous.



Mais pour notre cas notre servomoteur tourne de droite à gauche donc on pointe l'antenne au nord et on la fait tourner de droite à gauche avec un angle de $180-60=120$ degrés et de bas en haut avec un angle de $180-30=150$ degrés comme le montre la figure ci-dessous .



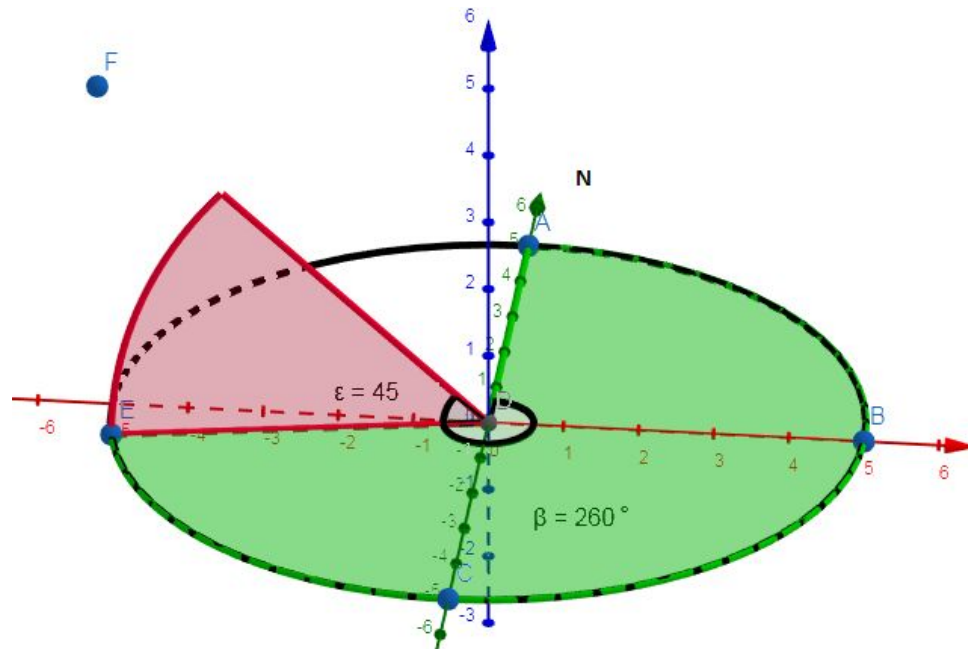
- **Azimut > 180 :**

Si l'azimut est supérieur à 180 degrés alors on applique une seule instruction :

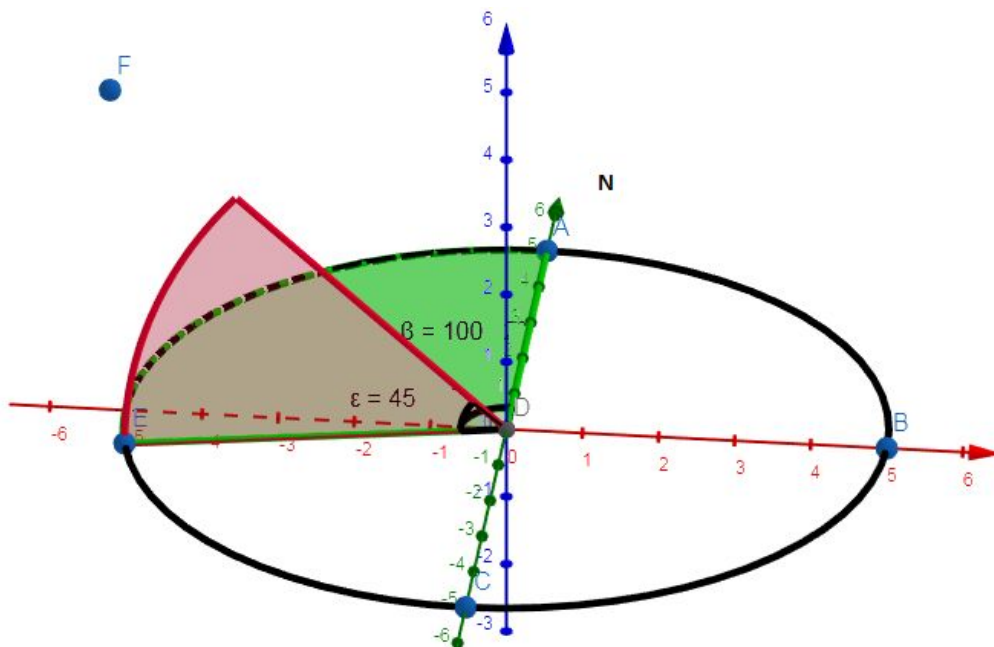
- $\text{Azimut} = 360 - \text{Azimut}$;

Exemple :

Un satellite avec l'Azimut = 260 et l'Élévation = 45°, pour bien orienter l'antenne au satellite il faut d'abord la pointer au nord ensuite la faire tourner de gauche à droite avec un angle de 260 degrés ensuite de bas en haut avec un angle de 45 degrés comme le montre la figure ci-dessous.



Mais pour notre cas notre servomoteur tourne de droite à gauche donc on pointe l'antenne au nord et on la fait tourner de droite à gauche avec un angle de $360 - 260 = 100$ degrés et on ne change pas l'élévation dans ce cas alors on tourne de bas en haut avec un angle de 45 degrés comme le montre la figure ci-dessous.



Configuration de la chaîne

La solution proposée permet de donner l'accès aux utilisateurs de choisir un satellite ensuite orienter l'antenne vers le satellite sélectionné et commencer la récupération des données envoyées par une communication LORA tout en gardant la solution simple et extensible (réplication facile).

La figure 1 sur la page 4, montre les différentes composantes qui constitue notre chaîne. La chaîne commence par l'application web (webapp) qui communique avec un service externe (n2yo.com) pour avoir la liste des satellites qui est sur le ciel de notre structure (antenne) en utilisant les coordonnées GPS (position de l'antenne remontée par le STM) (on expliquera ça plus tard). Notre application web peut être directement utilisée depuis un navigateur ou bien depuis une application mobile (qu'on a déjà développé) en utilisant la REST API exposé par notre l'application web.

Après d'avoir choisi un satellite depuis la liste, notre webapp envoie les paramètres d'orientation de la structure (Azimut, Élévation) vers un broker (hivemq) en utilisant un protocole Mqtt sur un topic spécial de configuration.

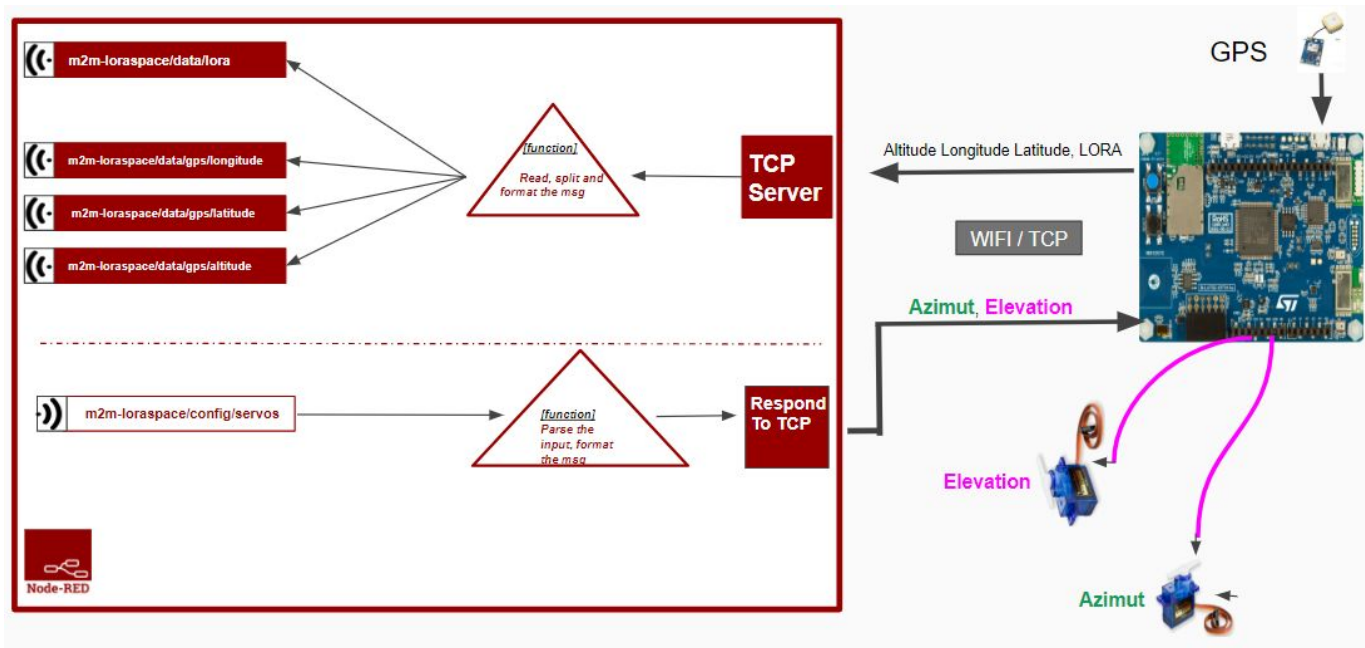


Figure “Communication STM32 Node-Red-MQTT”

De l'autre côté nous avons un Node Red installé sur une machine locale (ou sur une raspberry-pi) récupère ce msg de configuration MQTT ensuite le transformant en

des données qui peuvent être lues par la STM32 et l'envoyer par TCP vers cette dernière.

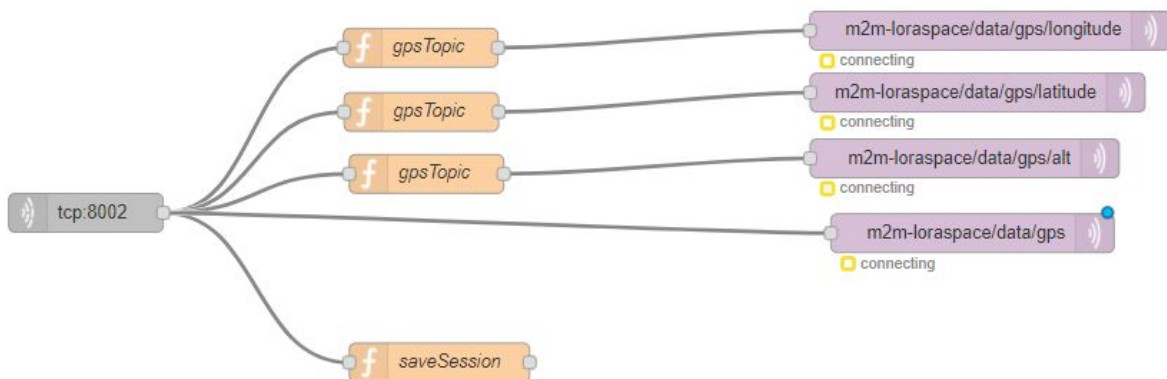


Figure “pipeline node-red responsable de l’envoi des messages de configuration des servos”

La STM 32 récupère les valeurs de rotation des servomoteurs envoyé par node-red, elle les convertit et ensuite elle fait bouger les servomoteurs qui vont tourner la structure.

Après d’avoir orienté notre structure (antenne) vers le bon satellite on commence la communication bidirectionnelle des données en LORA, la STM demande des informations ensuite le satellite envoie ces informations) ces informations sont remontés vers le broker ensuite récupérer depuis jayse pour faire des beaux dashboards. (Comme le module LORA consomme tous les pins de la STM, nous avons utilisé une deuxième carte NUCLEO pour faire la récupération des données LORA)

La STM permet ainsi de récupérer les coordonnées GPS (Altitude, Longitude, Latitude) de la structure (position de l’antenne) via un module externe ensuite d’envoyer ces coordonnées par TCP/WIFI vers le node red qui va les convertir ensuite les envoyer par MQTT vers le broker sur des topics de données GPS spéciaux.



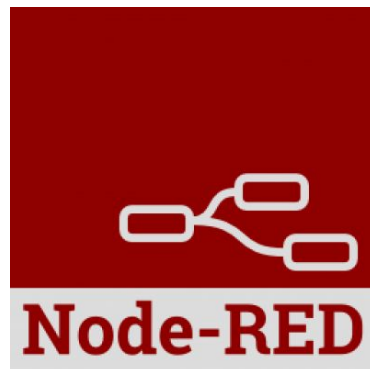
Dans ce projet nous avons utilisé la plateforme mbed os sur cloud pour programmer nos cartes stm pour la flexibilité et la facilité qu’apporte cette plateforme.

Nous avons utilisé node red comme une passerelle entre la stm32 et le Broker vue qu’on a pas pu faire une communication directe depuis la stm vers un broker.

Le node red communique avec la stm via un réseau TCP et avec le broker via MQTT et avec le board nucleo (pour les données Lora) avec serial.

Technologies utilisées :

NodeRed :



Outil visuel pour connecter et relier des dispositifs matériels des api et des services en ligne

Mbed :



mbed™ est une plateforme en ligne, regroupant différents outils pour un développement rapide sur architecture ARM® cortex-M.

.

MQTT :



MQTT (MQ Telemetry Transport) est un protocole de messagerie publish-subscribe basé sur le protocole TCP/IP

Web Service :



Après beaucoup de recherche nous avons décidé d'utiliser une api fourni par le site n2yo.com , la principale difficulté durant cette étape et que le webservice été encore en cours de développement , après avoir contacté l'administrateur celui-ci nous a fourni une ancienne version en soap , et plus tard la version rest (quand l'administrateur s'est rendu compte que des gens s'interesse il la sorti plus tôt)

NodeJs & Express JS :



On a utilisé node js comme serveur web pour sa **bonne performance** et le **support** disponible ,il forme un couple parfait avec son **framework express.js** pour gérer le routage des pages ainsi que les différentes requêtes **get/post** pour l'échange de données , il peut communiquer directement avec le webservice, la communication avec **mqtt** est facile à comprendre , on l'utilisera pour la suite pour créer une application WEB qui va gérer la communication avec les satellites

Application Android

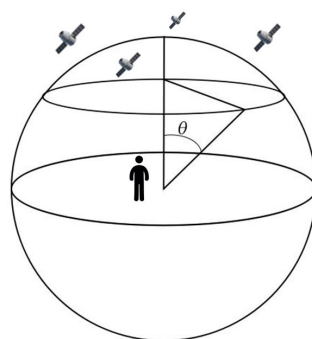


L'application Android va servir de télécommande pour lancer la configuration de l'antenne c'est l'application principale pour gérer la structure

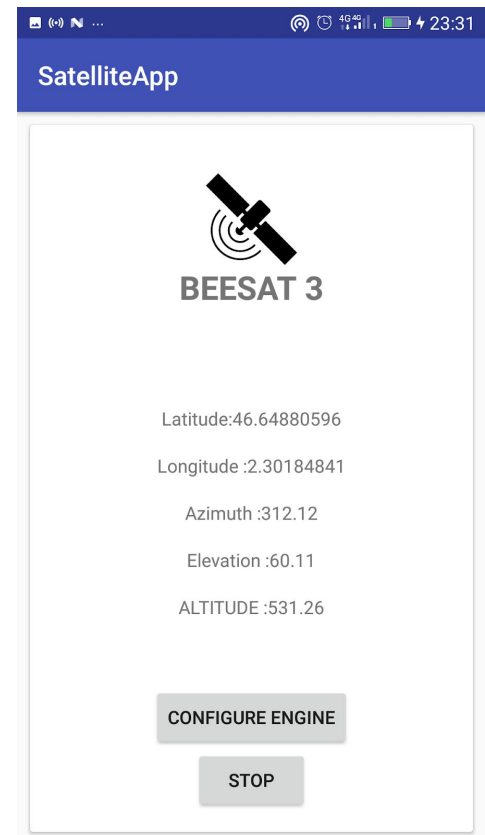
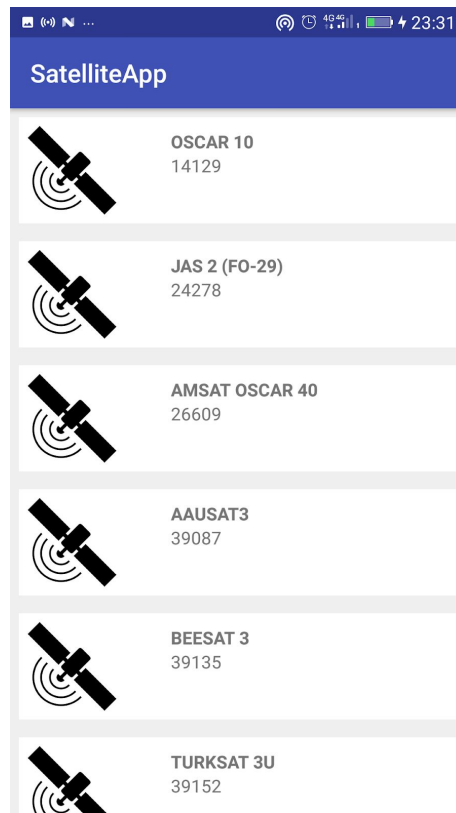
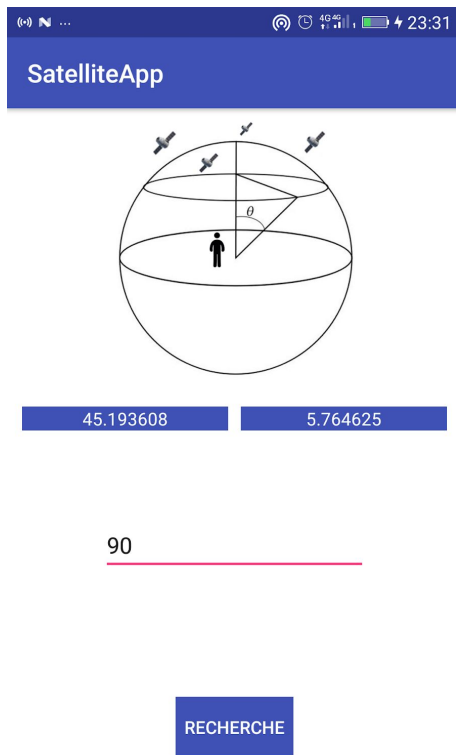
Fonctionnement de l'application:

Lorsque l'utilisateur sélectionne un angle l'application effectue une requête http au serveur web pour récupérer les données du satellites , pour recevoir ces données depuis le webservice nous avons besoin de spécifier les coordonnées gps de la structure (**pour recevoir les données selon la localisation de la structure**), l'application va chercher l'information depuis un canal mqtt qui reçoit en continue les données de localisation du gps de la structure, , une fois les données obtenues par celle ci , l'application peut interroger correctement le webservice qui va nous fournir la liste des satellites

L'angle représente un rayon de recherche donné au-dessus de l'emplacement de l'observateur. La portée du rayon de recherche est de 0 à 90 degrés, 0 signifi que seuls les satellites passent juste au-dessus de la position de l'observateur, tandis que 90 degrés ramènent tous les satellites au-dessus de l'horizon(azimut). Cette image peut offrir une meilleure explication



Une fois la liste affichée , l'utilisateur peut visualiser les informations plus détaillées du satellite puis en appuyant sur le bouton **“configure engine”** , va envoyer en continu l'azimut et l'élévation du satellite à l'engin via un canal mqtt , ce dernier va ensuite interpréter les résultats et s'orienter vers le satellite



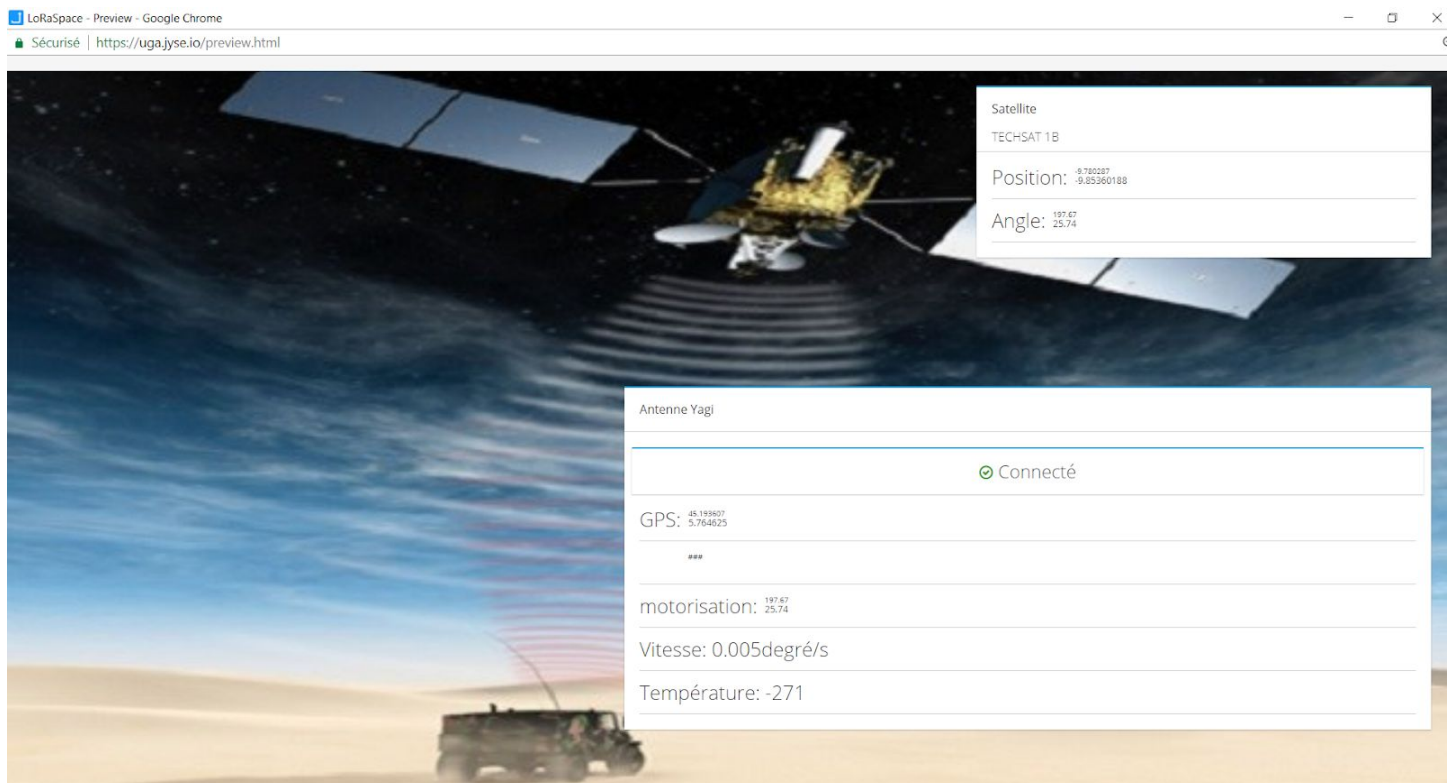
Visualisation des données avec JYSE:

Cette solution permet de récupérer des données via plusieurs protocoles et plusieurs format de données.

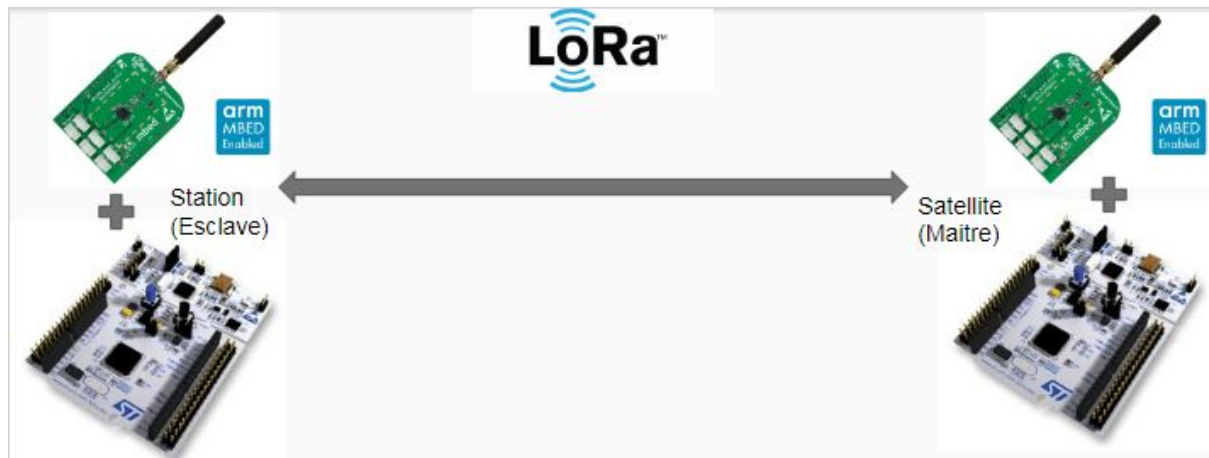
Dans notre cas, nous avons utilisé le protocole MQTT en utilisant plusieurs topics pour les données GPS, les données du moteurs et les données échangées en LoRa. Ces derniers envois des données variées (entiers, chaînes de caractères, fichiers JSON...).

Ainsi, nous avons pu afficher les données relatives à la station terrestre (antenne), notamment, si elle est connectée à un satellite ou non, les données GPS (latitude, longitude et altitude), les angles de rotations du moteur et les données reçues en LoRa.

De plus, nous avons affiché le données du satellite connecté, position et angle (élévation et azimuth)



Simulation de la communication bidirectionnelle :



Nous avons simulé la communication entre la station et le satellite avec 2 antennes LoRa , dans cette communication il y a une phase de connexion ensuite une phase d'envoi de données et une phase de déconnexion, cette communication est synchrone.

Console de la station:

messages reçu par la station.

```
SX1276 Demo Application

> Board Type: SX1276MB1MAS <

> LORA Mode <

Satellite LoRaSpace 38400
> OnRxTimeout
Conexion d'un appareil...
Collecte de donnees...
Envoi de donnees...
Appareil deconnecte
```

Console du satellite:

messages reçu par le satellite.

```
SX1272 Demo Application

> Board Type: SX1272MB2xAS <

> LORA Mode <

Station IM2AG
Detection du satellite...demande de connection!!!
connecte au satellite LoRaSpace... demande de donnees
Donnees:
Name: LoRaSpace
ID: 38
Temperature: -271
Vitesse: 0.005degre/s
Deconnexion...
```

Difficultés rencontrées :

- Configuration matérielle
- Captation signal GPS (lors des testes à l'intérieur)
- Matériel pour le montage de la structure
- Le webservice à changé en cours de développement

Conclusion :

Ce projet nous a permis de nous familiariser avec les nombreuses technologies utilisées dans le monde des objets connectés, ainsi que de concevoir une architecture de bout en bout pour circuler une information d'un matériel à un autre , nous avons pu mettre en pratique de la créativité pour imaginer une structure qui fait partie intégrante de notre chaîne .

Ce projet nous a permis surtout de préparer le terrain à une éventuelle expérimentation d'un nouveau protocole d'échange de données basses consommation avec les satellites qui peut nous permettre de réduire le coût considérablement