PARENT Aldric
MASTER 1 - Sciences de l'Univers et Technologies Spatiales

Observatoire de Paris – Université PSL
61 avenue de l'Observatoire, 75014 Paris - France

Déploiement d'une station sol dédiée aux communications avec des nanosatellites

Du 13 Janvier au 20 Juin 2025

Organisme d'accueil : Observatoire de Paris – CENSUS – PSL
61 avenue de l'Observatoire, 75014 Paris - France

Tuteur de Stage : M. Boris SEGRET, Directeur Technique CENSUS - LIRA

Enseignant référent : M. Cédric LEYRAT, M1 co-directeur et astronome

I. REMERCIEMENTS

Je souhaite avant tout remercier Boris Segret pour son encadrement, sa disponibilité constante et la qualité des connaissances qu'il m'a transmises. Son exigence et son sens du détail m'ont poussé à toujours chercher à m'améliorer, à explorer de nouvelles pistes et à réfléchir à des idées innovantes. J'ai grandement apprécié les discussions enrichissantes que nous avons eues, ainsi que le partage de son expérience, tant sur les sujets spécifiques à ce stage que sur des thèmes plus généraux liés à l'espace. Un grand merci.

Je tiens également à remercier chaleureusement l'ensemble de l'équipe CENSUS : Elisa, Chiara, Yanis, Kai, Kouassi, Antonin. C'était un plaisir de travailler avec vous au quotidien. J'ai apprécié vos retours, vos conseils et vos idées vis-à-vis du travail que j'ai mené. Votre présence a aussi contribué aux aspects plus informels et conviviaux de ce stage. C'était tous les jours un plaisir de me rendre au travail.

Merci au personnel du LIRA (Laboratoire d'Instrumentation et de Recherche en Astrophysique) et plus particulièrement à l'équipe informatique que j'ai pu solliciter lors de divers problèmes, et à l'équipe de Mustapha dont l'aide a été précieuse pour identifier le problème de signal UHF. Je me suis senti chanceux de pouvoir m'appuyer sur l'aide d'experts quand je traitais des sujets que je ne maîtrisais pas.

Enfin, merci aux membres et contributeurs de la communauté de la Libre Space Foundation, en particulier à Jan et Cristian, qui ont été d'une remarquable gentillesse et disponibilité pour répondre à mes questions et m'aider à résoudre les problèmes rencontrés.

II. TABLE DES MATIÈRES

III.1 Context

In recent years, the miniaturization of space technologies has led to a growing use of nanosatellites for scientific missions. To ensure the efficient operation of such satellites, ground stations must provide reliable communication capabilities. Within this context, the Paris Observatory – PSL established a ground station in Meudon in 2017. Following the end of the PicSat mission, the infrastructure required a full redesign to meet new technical and operational objectives.

III.2 Aims

The goal of this internship was to redesign and reimplement the software architecture of the Meudon ground station (SSOL), enabling a reception mode operation based on the open-source SatNOGS network for communication with nanosatellites. The system had to fulfil four major requirements: cross-platform compatibility, modularity and maintainability, reproducibility of the installation, and long-term reliability.

III.3 Methods

The work relied on three main technical choices:
(1) The use of the SatNOGS client for core operations
(2) The automation of the installation process using an Ansible playbook
(3) Containerization through Docker to ensure independence from the host operating system.
A structured and progressive approach was adopted, beginning with a thorough analysis of the project's technical and organizational context, followed by hands-on learning of the tools and technologies involved, practical implementation of software solutions, and continuous testing and documentation to ensure long-term sustainability.

III.4 Results

The redesigned software architecture now allows full automated deployment of the ground station, verified through successful satellite signal receptions in both VHF and UHF bands. The new system meets all predefined requirements and has restored the station's operational capabilities within the SatNOGS network. A hardware issue affecting UHF signal quality was diagnosed and attributed to a degraded LNA, which has been replaced based on a long-term performance strategy aligned with the NANOSATGRID project.

IV.1 Contexte du stage

Depuis plusieurs années, le secteur spatial est profondément transformé par l'émergence du New Space, une dynamique favorisée par l'essor des technologies de miniaturisation, l'optimisation des architectures systèmes et la réduction des coûts, masses et volumes des plateformes spatiales, visant à rendre l'espace plus accessible, flexible et économique. Au cœur de cette révolution, les nanosatellites (de petits satellites aux dimensions et coûts réduits) jouent un rôle clé, notamment dans le domaine scientifique. Leur faible encombrement et coût de production permettent non seulement de multiplier les campagnes expérimentales, mais aussi d'intégrer et de valider rapidement des instruments innovants, offrant ainsi un banc d'essai idéal pour de nouvelles technologies spatiales.

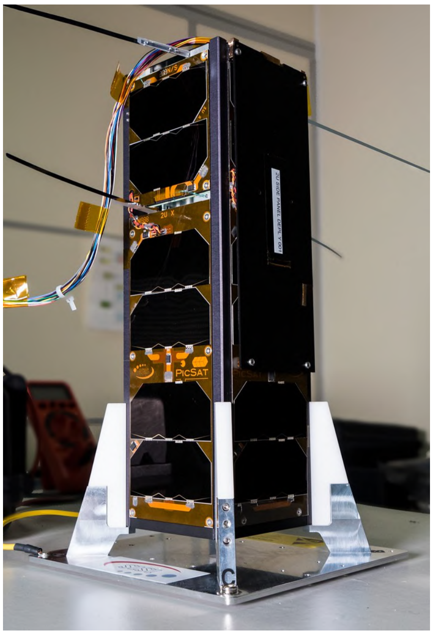


Figure 1 : Nanosatellite Picsat

Cependant, pour tirer pleinement parti du potentiel des nanosatellites, il est essentiel de disposer d'une infrastructure robuste et efficace pour assurer les communications entre la Terre et ces engins spatiaux. En effet, la transmission des commandes, la réception des données, ainsi que la gestion opérationnelle nécessitent une station sol dédiée, capable de suivre ces satellites en orbite et de garantir un lien continu et fiable. C'est justement le suivi d'une mission spatiale : la mission PicSat (1), qui a motivé la mise en place d'une station sol au sein de l'observatoire de Paris – PSL en 2017 (2). L'objectif principal de cette mission était l'observation du transit exo planétaire de Beta Pictoris b. Le projet de station sol a été financé par le CCERES, aujourd'hui rebaptisé CEntre pour les Nanosatellites en Sciences de l'Univers (CENSUS), qui fait partie intégrante du pôle spatial de l'Université de Recherche PSL et est engagé dans plusieurs programmes spatiaux.

Si la première version de la station sol a permis de communiquer efficacement avec le nanosatellite PicSat, la perte de ce dernier a conduit à une réévaluation nécessaire des objectifs et du rôle de la station sol de l'Observatoire de Paris – PSL. Ces objectifs redéfinis s'articulent désormais autour de quatre axes principaux (2) :

- **O1** : La station sol doit constituer une infrastructure visible et reconnue au sein de la communauté nanosatellite, et contribuer activement au développement du secteur scientifique régional.
- **O2** : La station doit assurer les communications avec les nanosatellites lancés par l'Observatoire de Paris – PSL ou ses partenaires, afin de faciliter leur exploitation opérationnelle.
- **O3** : La station doit également être capable de recevoir les signaux émis par d'autres nanosatellites, contribuant ainsi à une démarche collaborative d'écoute.
- **O4** : Le projet NANOSATGRID, représentant les futurs développements du segment sol, devra permettre d'élargir les capacités de CENSUS à opérer les nanosatellites de ses partenaires, en intégrant la station dans un réseau plus large de stations sol collaboratives.

Cependant, la mise en œuvre de ces nouveaux objectifs a rapidement mis en lumière plusieurs limitations techniques et opérationnelles de la station sol existante, soulignant l'urgence d'une refonte complète de l'infrastructure (4). Parmi les principaux défis identifiés :

- **Emplacement inadapté** : La station précédente était directement installée au sol, sur une pelouse, sans aménagements spécifiques ni dispositif de sécurisation. Cette exposition à l'environnement extérieur compromettait la sécurité des équipements électroniques. De plus, la faible hauteur d'installation limitait la qualité des observations satellitaires, en raison d'un champ de vision obstrué et de mauvaises conditions de propagation des signaux.
- **Faible stabilité structurelle** : La structure manquait de rigidité, rendant l'ensemble peu fiable et difficile à entretenir sur le long terme. Cette instabilité nuisait aux performances et à la durabilité des équipements.
- **Absence de protection environnementale** : Le manque de dispositifs de protection adaptés augmentait le risque de dégradation mécanique des composants et affectait la précision des mesures, notamment à cause des conditions météorologiques variables.
- **Problèmes de compatibilité logicielle** : Les deux modes de fonctionnement de la station, le mode interne CENSUS et le mode collaboratif SATNOGS (voir description qui suit), nécessitaient des environnements logiciels distincts (Debian et Ubuntu), ce qui empêchait leur cohabitation sur un même système d'exploitation Linux et compliquait leur gestion.
- **Défaillances matérielles fréquentes** : Des pannes régulières des composants électroniques et des systèmes embarqués multipliaient les interruptions de service et accentuaient les difficultés d'exploitation.

Ces constats ont servi de moteur à une réorganisation profonde de la station sol, afin d'en améliorer la fiabilité, les performances et d'assurer une exploitation pérenne et collaborative, pleinement alignée avec les ambitions scientifiques de CENSUS.

C'est dans ce contexte de refonte et de montée en capacité de la station sol de Meudon que s'inscrit mon stage, réalisé dans le cadre du Master 1, *Sciences de l'Univers et Technologies Spatiales* de l'Observatoire de Paris – PSL.

Figure 2 : Précédente implémentation de la station sol

IV.2 Objectifs du stage

Les quatre objectifs de fonctionnement de la station sol (SSOL) évoqués précédemment ont permis d'imaginer deux modes de fonctionnement de cette dernière :

- **M1 – « SatNOGS mode »** : dans ce mode, la SSOL est intégrée au réseau international SatNOGS (Satellite Networked Open Ground Station) (3), une infrastructure collaborative et open source de stations sol. Il s'agit d'un mode purement réceptif, destiné uniquement à la réception de données émises par des satellites, sans possibilité de leur transmettre des commandes.
- **M2 – « CENSUS mode »** : ce mode est réservé aux utilisateurs autorisés connectés au réseau interne de l'Observatoire. Il permet non seulement d'écouter les signaux de satellites déployés dans le cadre de missions scientifiques, mais également de leur envoyer des commandes. Ce mode offre ainsi une capacité de télécommande complète, spécifiquement dédiée aux nanosatellites opérés par l'Observatoire de Paris – PSL.

La mise en œuvre de ces deux modes de fonctionnement s'est amorcée avec le développement initial du « SatNOGS mode (M1) », dans le cadre d'un stage de Master 2. Ce travail préliminaire a constitué la première étape de la réimplantation du segment sol, en posant les bases techniques de la nouvelle infrastructure.

a. Travail réalisé en amont

Le projet de refonte du segment sol de l'Observatoire de Paris peut être divisé en deux volets principaux : un volet matériel (hardware) et un volet logiciel (software). Le travail réalisé lors du stage précédent s'est concentré essentiellement sur la réorganisation matérielle de la station.
Ainsi, au début de mon stage, j'ai hérité d'une infrastructure déjà partiellement opérationnelle, reposant sur la configuration suivante (2) :

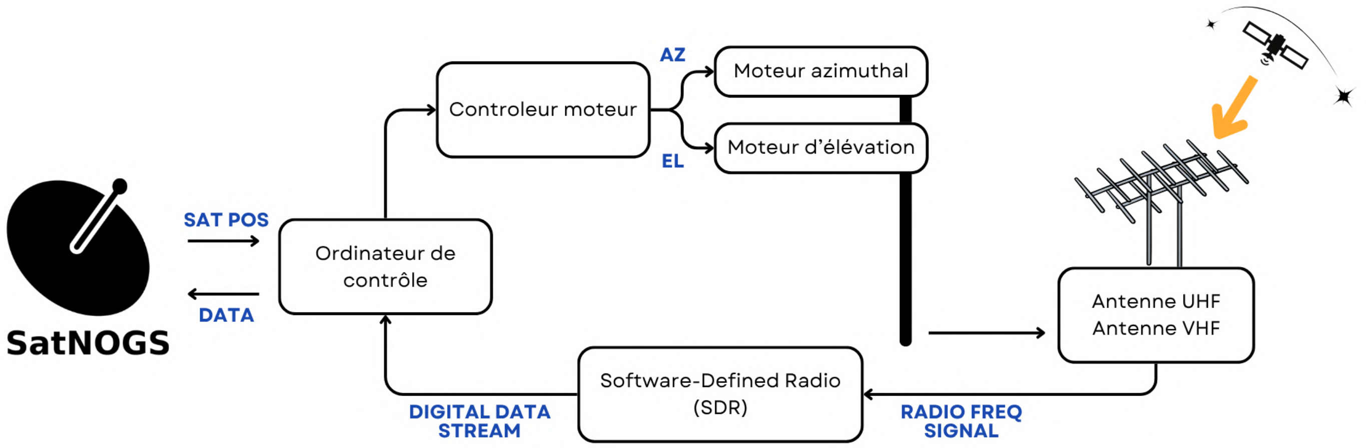


Figure 3 : Principaux composants du fonctionnement en mode « réception » (M1 – SatNOGS mode)

Dans le mode de fonctionnement M1, l'utilisateur sélectionne un satellite via l'interface en ligne du réseau SatNOGS, le système calcule alors ses passages visibles depuis la station grâce à ses données orbitales. L'observation est ensuite planifiée et les données du satellite sont transmises à l'ordinateur de contrôle. Avant chaque passage, les équipements se configurent automatiquement : le SDR est réglé sur la fréquence appropriée et l'antenne est orientée en conséquence. Pendant la traversée du satellite, le système assure le suivi de sa trajectoire tout en captant et en enregistrant son signal. Si le protocole de transmission est connu, les données sont décodées, puis toutes les informations collectées sont transmises à la base de données SatNOGS, contribuant ainsi à une collecte collaborative et ouverte des données satellitaires.

La partie hardware de la station se compose des éléments principaux suivants :

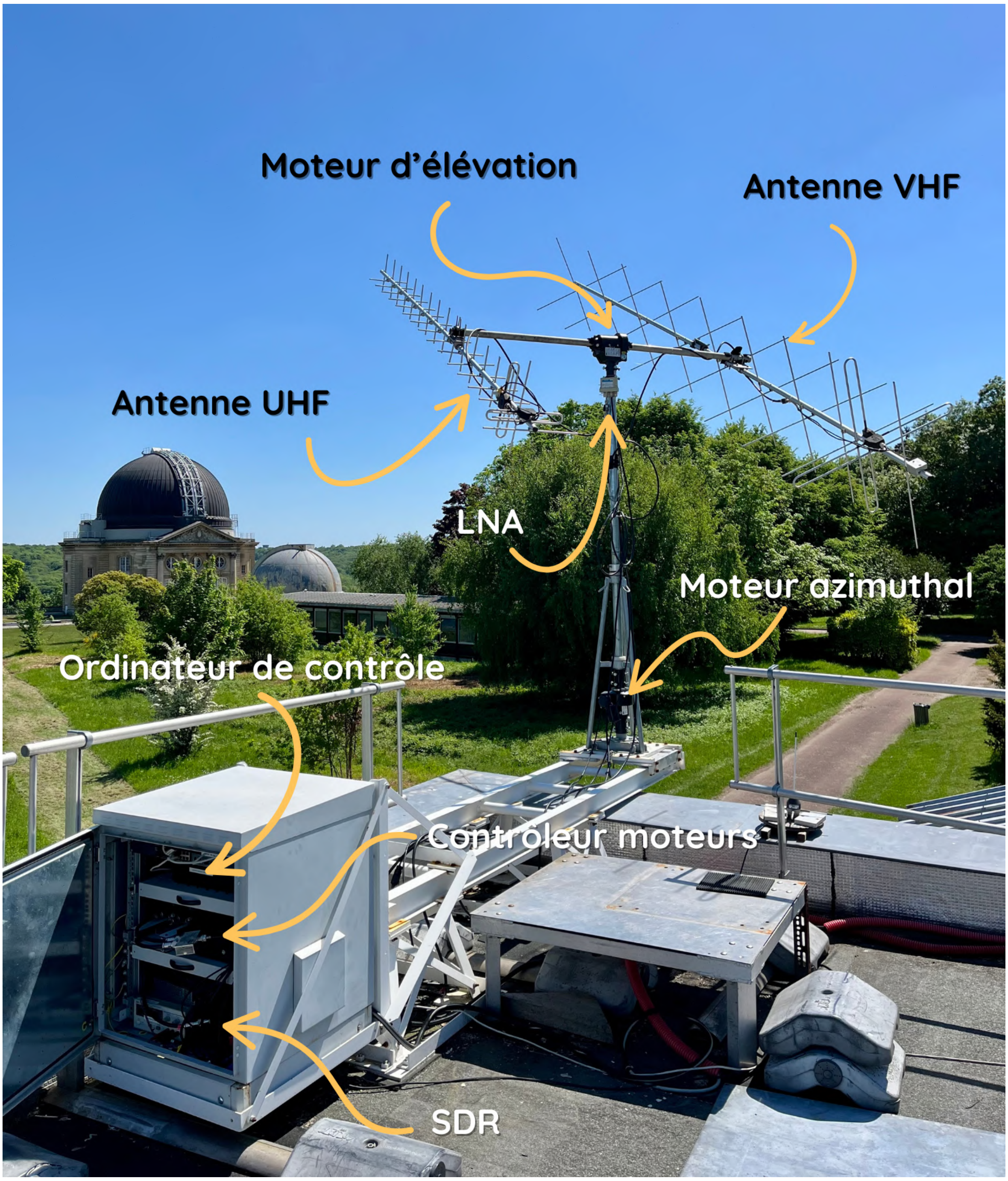


Figure 4 : Nouvelle implementation de la station SSOL

Antenne VHF + Antenne UHF :
Deux antennes directionnelles couvrant respectivement les bandes VHF (Very High Frequency) et UHF (Ultra High Frequency). Elles permettent la réception (et éventuellement la transmission) des signaux radio émis par les satellites opérant dans ces plages de fréquences.

LNA (Low Noise Amplifier) :
Amplificateur à faible bruit positionné à proximité de l'antenne. Il a pour rôle de renforcer les signaux faibles reçus dans la bande UHF tout en minimisant l'ajout de bruit électronique, améliorant ainsi le rapport signal/bruit (SNR).

Moteur azimuthal + Moteur d'élévation :
Ensemble de moteurs contrôlant l'orientation de l'antenne selon deux axes : l'azimut (rotation horizontale) et l'élévation (inclinaison verticale). Ils assurent le suivi dynamique des satellites pendant leur passage.

Contrôleur moteur :
Dispositif électronique permettant d'interpréter les commandes logicielles et de piloter précisément les moteurs d'azimut et d'élévation pour orienter l'antenne vers la position attendue.

SDR (Software Defined Radio) :
Récepteur radio flexible qui numérise le signal analogique reçu et permet un traitement logiciel de modulation/démodulation. Il offre une grande adaptabilité pour écouter différentes fréquences et protocoles.

Ordinateur de contrôle :
Ordinateur central coordonnant l'ensemble du système. Il exécute les logiciels de planification, de suivi des satellites, de contrôle moteur, de traitement du signal SDR et d'enregistrement des données.

b. Objectif de stage

Le volet matériel étant déjà en place, mon stage a consisté à repenser et réimplémenter toute la partie logicielle, indispensable pour permettre le redémarrage fiable et durable de la station sol. C'est dans ce cadre que mon objectif de stage a été défini :

Reconstruction de l'architecture logicielle de la station sol

---

c. Exigences de travail

Pour éviter les précédents déboires liés à l'utilisation, à la maintenance et à la mise à jour du segment logiciel de la SSOL, les contraintes suivantes ont été identifiées :

Tableau 1 : Tableau des exigences

ID	Titre	Description
E1	Compatibilité multiplateforme	L'installation ne doit pas dépendre du système d'exploitation de l'utilisateur ou de celui de l'ordinateur de contrôle, qu'il s'agisse de Linux, macOS ou d'un autre environnement.
E2	Facilité de maintenance	Chaque composant logiciel doit être isolé et indépendant afin de faciliter les opérations de maintenance.
E3	Installation reproductible	L'architecture logicielle doit pouvoir être facilement reconfigurée ou réinstallée, quel que soit l'environnement.
E4	Fiabilité, sécurité et robustesse	Le système doit être stable, résistant aux erreurs et sécurisé sur le long terme.

Ces contraintes constitueront un cadre de référence pour orienter et motiver les décisions de développement exposées dans les parties suivantes.

V.1 Méthodologie globale

Dès le début de mon stage, j'ai choisi d'adopter une approche structurée, progressive et cohérente. J'ai en effet souhaité éviter l'écueil, pourtant tentant, de me confronter trop rapidement aux aspects pratiques de la mission sans avoir, au préalable, acquis une compréhension suffisante du contexte et des outils mobilisés.

a. Découverte et compréhension du contexte du projet

La première phase de mon stage a été consacrée à l'analyse du contexte global du projet, à la fois sur les plans organisationnel et technique. Il s'agissait notamment de retracer l'historique de la station sol (SSOL) de l'Observatoire de Paris et de comprendre les travaux déjà réalisés en amont. Ce stage constituait ma première immersion concrète dans le domaine des communications spatiales et radio. J'ai ainsi dû me familiariser avec un nouvel environnement de travail, un vocabulaire technique spécifique, ainsi qu'avec les différents acteurs et pratiques propres à ce domaine. Le travail réalisé durant cette phase a principalement consisté en l'analyse de rapports antérieurs, la lecture de documentation technique et la découverte sur site des installations matérielles de la station.

b. Montée en compétence sur les outils et technologies

Une fois le cadre général assimilé, j'ai pris en main les différents outils liés à la station sol, notamment la mise en fonctionnement des moteurs, la configuration du SDR, ainsi que la prise en main du réseau SatNOGS. Cette phase d'expérimentation pratique, couplée aux contraintes définies par les objectifs de la mission, m'a permis d'identifier plusieurs choix technologiques pertinents pour atteindre ces objectifs. Une fois ces choix arrêtés, j'ai consacré du temps à une formation approfondie sur plusieurs logiciels et outils indispensables à l'implémentation de la solution retenue, tels que Docker, Gpredict, SDR++ et GitLab. Le travail mené durant cette étape s'est traduit par la réalisation de tutoriels, la participation à des workshops et de l'entraînement sur des exemples pratiques de programmation.

c. Implémentation concrète de solutions

Fort de cette compréhension, j'ai pu entamer l'implémentation pratique des solutions logicielles : réorganisation de l'architecture logicielle, développement d'un environnement Docker, automatisation du processus d'installation et tests de la solution proposée. Le travail réalisé durant cette phase consistait principalement en du développement en Python et Bash. En raison du nombre limité de ressources disponibles sur la mise en place de la solution, j'ai pu m'appuyer sur le soutien des ingénieurs de CENSUS ainsi que sur un forum communautaire d'entraide dédié au réseau SatNOGS.

d. Itérations, tests continus et pérennisation

Tout au long du développement, des phases régulières de test, d'évaluation et d'ajustement ont permis d'identifier les points de fragilité et d'améliorer progressivement la solution. Ce travail itératif m'a également permis de documenter mon travail pour en assurer sa prénnité.

V.2 Choix et solutions techniques retenues

Le travail réalisé au cours de ce stage s'est appuyé sur trois décisions techniques majeures, visant à répondre efficacement aux exigences identifiées précédemment (E1, E2, E3 et E4). Ces choix structurants sont les suivants :

- L'adoption du client SatNOGS comme composant principal de gestion des opérations de la station
- Le recours à un playbook Ansible pour automatiser le processus d'installation
- L'utilisation de conteneurs Docker pour garantir portabilité et isolation de l'environnement logiciel.

Chacun de ces choix sera présenté et justifié dans les sections suivantes, en lien direct avec la problématique du projet.

a. Utilisation du client SatNOGS

Le client SatNOGS est un logiciel open source développé par la communauté Libre Space Foundation (LSF). Il constitue le cœur opérationnel d'une station sol souhaitant se rendre visible et accessible sur le réseau SatNOGS. Son rôle principal est de piloter automatiquement les différents éléments de la station, notamment les moteurs de poursuite, le SDR (Software Defined Radio), et les séquences de décodage, selon un planning d'observations défini par le réseau SatNOGS. Il se connecte à une interface web distante pour récupérer les observations programmées, puis exécute localement chaque session d'écoute radio, en assurant le suivi de la trajectoire du satellite, la réception et l'enregistrement des signaux, et enfin leur envoi vers une base de données centralisée.

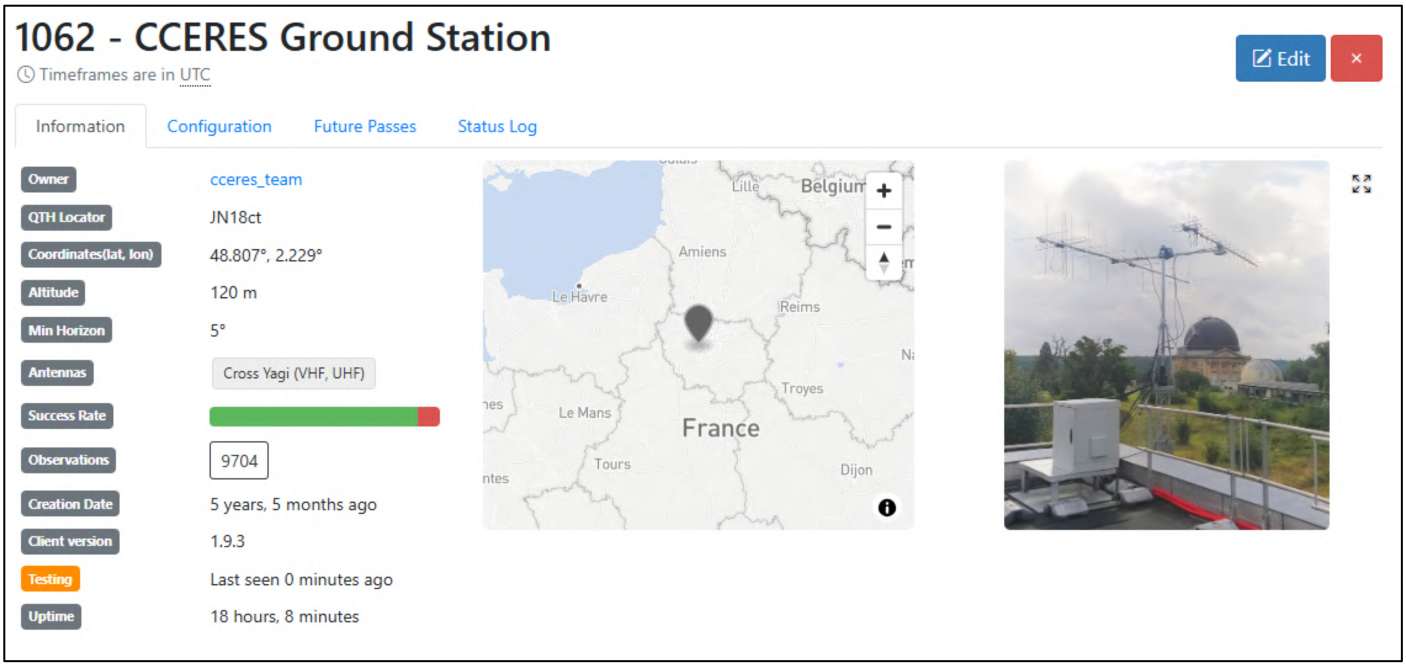


Figure 5 : Fiche descriptive de la station sol sur le réseau SatNOGS

Étant donné que le réseau SatNOGS est aujourd'hui le plus grand réseau mondial de stations sol open source dédiées à l'écoute de satellites en orbite basse, le choix de positionner la SSOL sur ce réseau et donc d'utiliser le client SatNOGS pour son exploitation me semblait cohérent avec les objectifs :

- **O1 :** *La station sol doit constituer une infrastructure visible et reconnue au sein de la communauté nanosatellite, et contribuer activement au développement du secteur scientifique régional*
- **O3 :** *La station doit également être capable de recevoir les signaux émis par d'autres nanosatellites contribuant ainsi à une démarche collaborative d'écoute*

Tout en satisfaisant l'exigence :

E4 : *Le système doit être stable, résistant aux erreurs, et sécurisé sur le long terme.*

<u>Stabilité :</u> Le client SatNOGS est un outil open source maintenu activement par une large communauté, avec de nombreuses installations opérationnelles à travers le monde. Son code est régulièrement mis à jour, testé et utilisé dans des environnements variés, ce qui garantit une grande fiabilité.

<u>Résistance aux erreurs :</u> Le client est conçu pour fonctionner dans des conditions réelles avec des mécanismes de journalisation, de surveillance des processus et de redémarrage automatique en cas d'erreurs ou d'échecs de communication, ce qui en fait une solution résiliente.

<u>Sécurité et pérennité :</u> Utilisé à grande échelle dans un cadre collaboratif, le client bénéficie d'une veille de sécurité continue. Son intégration dans un projet open source garantit la traçabilité des évolutions, la transparence du code et la possibilité d'être maintenu sur le long terme indépendamment d'un fournisseur.

Mon utilisation du client SatNOGS a été avant tout opérationnelle (je reviendrai plus en détail sur les résultats dans la partie VI. RÉSULTATS). Il s'agissait principalement de programmer des sessions d'écoute et d'observation de satellites, puis de collecter les données obtenues. Je n'ai pas apporté de modifications au code source du client, pour plusieurs raisons : d'une part, sa complexité technique, d'autre part le fait qu'il répondait parfaitement aux besoins fonctionnels de notre station. Enfin, il était essentiel de préserver l'intégrité du client afin de garantir la conformité de la SSOL avec les standards du réseau SatNOGS, condition indispensable pour assurer sa visibilité et son intégration officielle au sein de ce réseau collaboratif.

Si l'utilisation du client ne présente généralement pas de difficulté, son installation, en revanche, peut s'avérer plus complexe à mettre en œuvre.

b. Utilisation et modification du playbook ansible de LSF

Il existe en effet de nombreuses méthodes pour installer le client SatNOGS. Le choix de la méthode la plus adaptée dépend principalement de l'infrastructure matérielle disponible (ordinateur de contrôle, carte Raspberry Pi ...) ainsi que de l'usage prévu de la station sol. Dans mon cas, j'ai opté pour l'utilisation d'un playbook Ansible dont je vais décrire le fonctionnement et justifier le choix.

Un playbook Ansible est un fichier de configuration écrit en YAML qui permet d'automatiser des tâches d'administration système de manière reproductible et cohérente. Il décrit étape par étape les actions à exécuter sur une ou plusieurs machines, comme l'installation de logiciels, la configuration de services ou la mise à jour du système en s'appuyant sur une approche déclarative. L'intérêt principal réside dans la simplicité d'écriture, la clarté des instructions et la possibilité de rejouer exactement les mêmes opérations sur n'importe quel environnement. On comprend donc que l'utilisation d'une telle méthode participe à la vérification des exigences :

- **E2 :** *Chaque composant logiciel doit être isolé et indépendant afin de faciliter les opérations de maintenance.*
- **E3 :** *L'architecture logicielle doit pouvoir être facilement reconfigurée ou réinstallée, quel que soit l'environnement.*

Une autre raison majeure qui m'a conduit à choisir cette méthode d'installation est la disponibilité d'un playbook publié par la communauté LSF, permettant d'automatiser l'installation du client SatNOGS. Cela m'a permis de profiter d'un travail existant, bénéficiant ainsi d'une base fiable et éprouvée. Toutefois, contrairement au client SatNOGS qui, une fois installé, ne nécessite généralement pas de modifications, le playbook a dû être adapté pour répondre aux spécificités de mon projet. En particulier, j'ai dû le modifier afin de l'intégrer pleinement dans un environnement Docker, qui constitue un élément central de l'architecture logicielle que j'ai mis en place.

c. La mise en place d'un environnement Docker

Pour répondre entièrement à mon objectif de stage, il reste donc une exigence à prendre en compte :

E1 : *L'installation ne doit pas dépendre du système d'exploitation de l'utilisateur ou de celui de l'ordinateur de contrôle, qu'il s'agisse de Linux, macOS ou d'un autre environnement.*

Parmi les exigences définies, celle-ci est sans doute la plus déterminante, car elle a directement motivé la refonte de l'installation de la station sol (SSOL). Lors de précédentes utilisations, notamment pour les communications avec le nanosatellite PicSat, la station s'appuyait sur plusieurs logiciels de contrôle et d'exploitation, chacun nécessitant un système d'exploitation spécifique. Cette hétérogénéité rendait l'installation complexe, fragile et difficilement reproductible. L'objectif de la nouvelle architecture était donc de rendre l'ensemble de la station sol entièrement indépendante du système d'exploitation de l'utilisateur, afin de garantir une plus grande portabilité, simplicité et pérennité. Pour ce faire, l'utilisation d'une architecture Docker (5) a été choisie.

Docker est une plateforme logicielle qui permet de créer, déployer et exécuter des applications dans des environnements appelés conteneurs. Un conteneur est un ensemble autonome et léger qui regroupe tout ce dont une application a besoin pour fonctionner : son code, ses bibliothèques, ses dépendances et ses configurations. Contrairement aux machines virtuelles, les conteneurs ne nécessitent pas de système d'exploitation complet pour chaque application, ce qui les rend beaucoup plus rapides à lancer et moins gourmands en ressources. L'un des principaux avantages de Docker est qu'un conteneur s'exécute de manière identique quel que soit le système hôte, que ce soit sous Linux, macOS ou Windows. Cela garantit une portabilité et une reproductibilité optimales. Dans le cadre de la station sol, cette technologie permet donc de s'affranchir des spécificités de chaque système d'exploitation, tout en assurant une installation standardisée, fiable et facilement partageable.

L'utilisation de l'outil Docker repose sur trois élemnts principaux :

- **Dockerfile** : C'est un fichier texte contenant un ensemble d'instructions décrivant comment construire une image Docker. Il définit notamment le système de base à utiliser, les logiciels à installer, les fichiers à copier, et les commandes à exécuter.
- **Image Docker** : Une image est un modèle figé et immuable, généré à partir d'un Dockerfile. Elle contient tout le nécessaire pour exécuter une application (système de fichiers, dépendances, configurations …) Elle peut être partagée et utilisée pour créer des conteneurs identiques sur n'importe quelle machine.
- **Conteneur Docker** : Un conteneur est une instance en cours d'exécution d'une image. Il représente l'environnement isolé dans lequel l'application s'exécute. Plusieurs conteneurs peuvent être lancés à partir d'une même image, chacun fonctionnant de manière indépendante.

Le fonctionnement général peut être résumé ainsi :

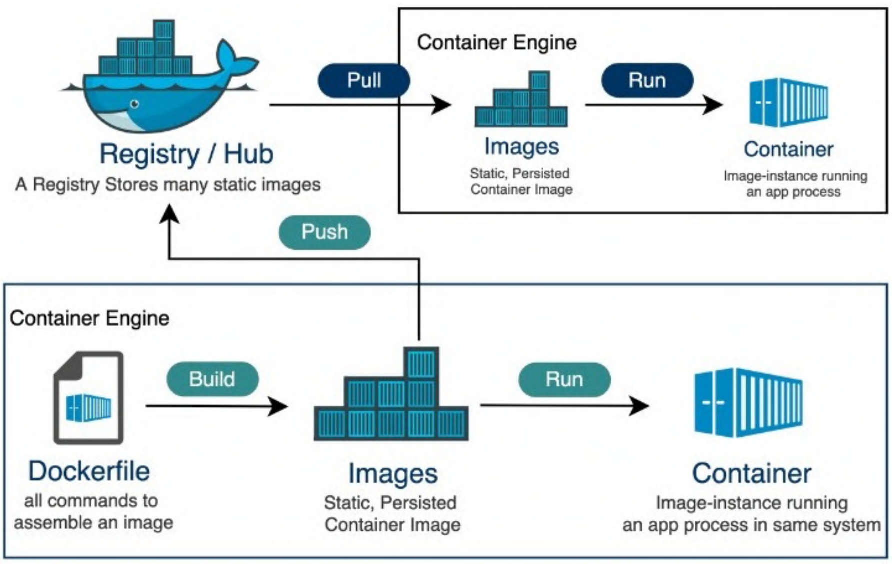


Figure 6 : Schéma de principe de fonctionnement de Docker

Le cœur de mon travail durant ce stage a donc consisté à imaginer et programmer une architecture logicielle basée sur Docker permettant d'installer un client SatNOGS en utilisant un playbook Ansible.

V.3 Déploiement et mise en œuvre des solutions

a. Déploiement automatisé de l'architecture logicielle

J'ai choisi de bâtir l'ensemble de mon travail sous forme de projet GitLab car ce format me semblait être le plus adapté car déjà utilisé au sein de l'équipe CENSUS, il pourra être facilement mis à jour et facilement partagé. L'ensemble du processus d'installation repose sur un script automatisé permettant à l'utilisateur d'être guidé dans le déploiement de la station sol.

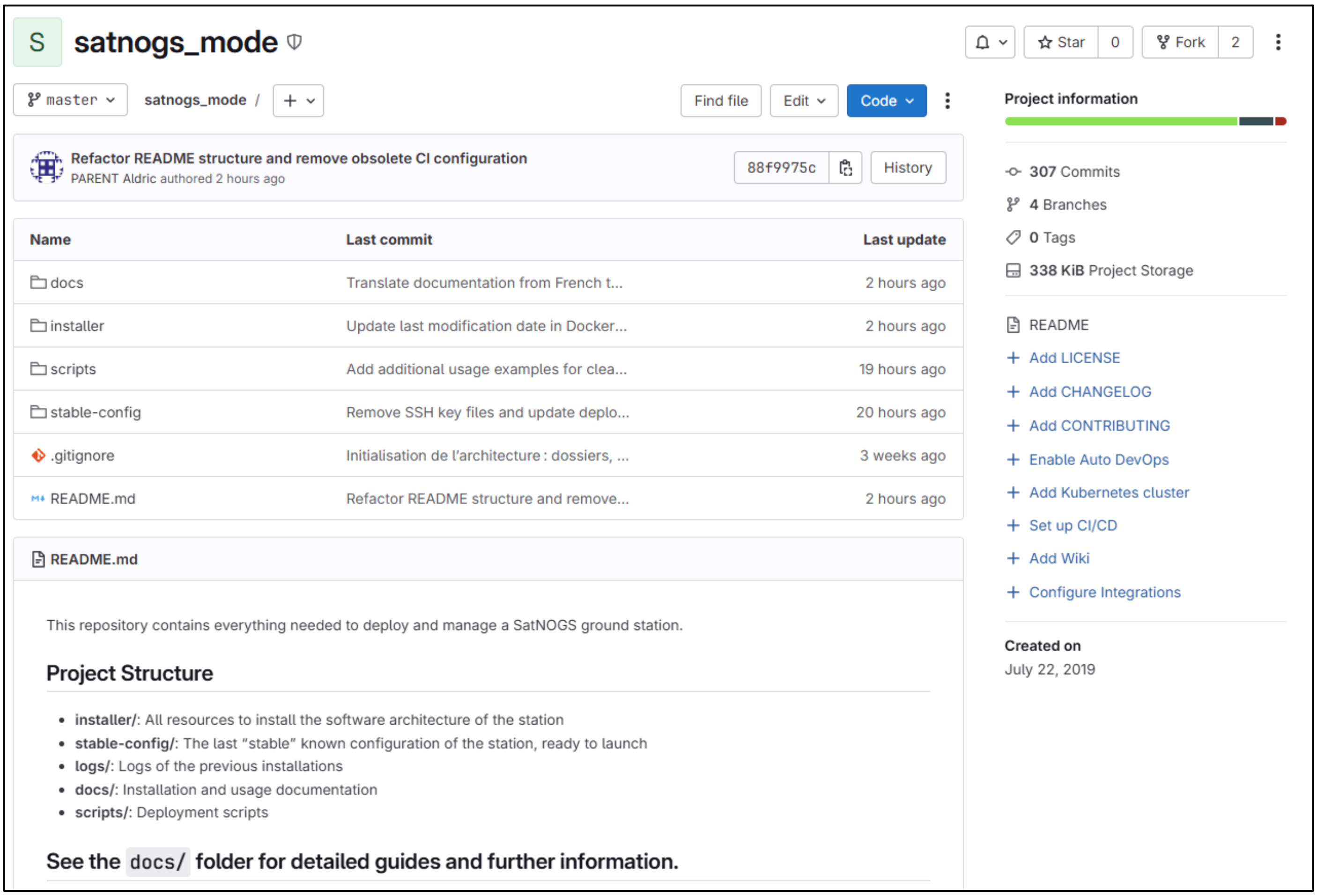


Figure 7 : Page principale du projet GitLab

Vous trouverez en annexe de ce rapport la documentation technique décrivant plus en détail la structure et le fonctionnement de la station sol et plus précisément, l'architecture et le fonctionnement du projet GitLab (p23→p28 de la documentation technique).

e. Rédaction d'un guide utilisateur

À l'issue du développement et de la mise en place du projet GitLab, j'ai rédigé plusieurs guides utilisateurs visant à accompagner efficacement le déploiement de l'architecture logicielle. Ces documents ont pour but d'expliquer les étapes essentielles, de faciliter la prise en main des outils, et de guider l'utilisateur pas à pas tout au long du processus d'installation et d'utilisation.

L'utilisation de ce projet GitLab offre à l'utilisateur plusieurs avantages majeurs :

- **Une installation et un paramétrage automatisés** d'un client SatNOGS, indépendamment de la configuration matérielle choisie. Cette indépendance est rendue possible grâce à l'utilisation de conteneurs Docker, qui encapsulent l'environnement logiciel nécessaire.
- **Une maintenance simplifiée et un processus de test sécurisé**, grâce à la conservation, au sein du projet, de la dernière configuration logicielle stable. En cas de dysfonctionnement du segment logiciel lors de tests, l'utilisateur peut restaurer rapidement cette version stable à l'aide du script automatisé fourni.

Vous trouverez en annexe de ce rapport, au sein de la documentation technique, les guides utilisateurs élaborés pour la mise en place du projet GitLab (les numéros de page indiqués correspondent à ceux de la documentation technique) :

- P58 : Launch of the configuration script
- P60 : Installation of SatNOGS client from scratch
- P63 : Reuse of the last stable configuration
- P65 : Save of the current stable configuration

VI.1 Redémarrage opérationnelle de la station sol

L'ensemble du travail mené en amont sur l'infrastructure matérielle ainsi que la solution logicielle que j'ai fournie à l'équipe CENSUS m'a permis de redéployer la station sol de l'Observatoire de Paris - PSL sur le réseau SatNOGS. Le principal résultat de ce travail ne réside toutefois pas uniquement dans la remise en service de la station, mais dans sa réintégration dans une architecture moderne, basée sur des conteneurs Docker, et déployée de manière entièrement automatisée.

Cette nouvelle configuration a rendu possible la validation concrète du fonctionnement de la station en permettant de recevoir des signaux issus de plusieurs satellites lors de leurs passages au-dessus de Meudon.

Une fois le client SatNOGS installé et configuré sur la station sol, l'ensemble des opérations est mené depuis l'interface web du réseau SatNOGS. Le processus d'observation suit alors les étapes suivantes :

- Sélection d'un satellite parmi les satellites passant prochainement au-dessus de la station sol :

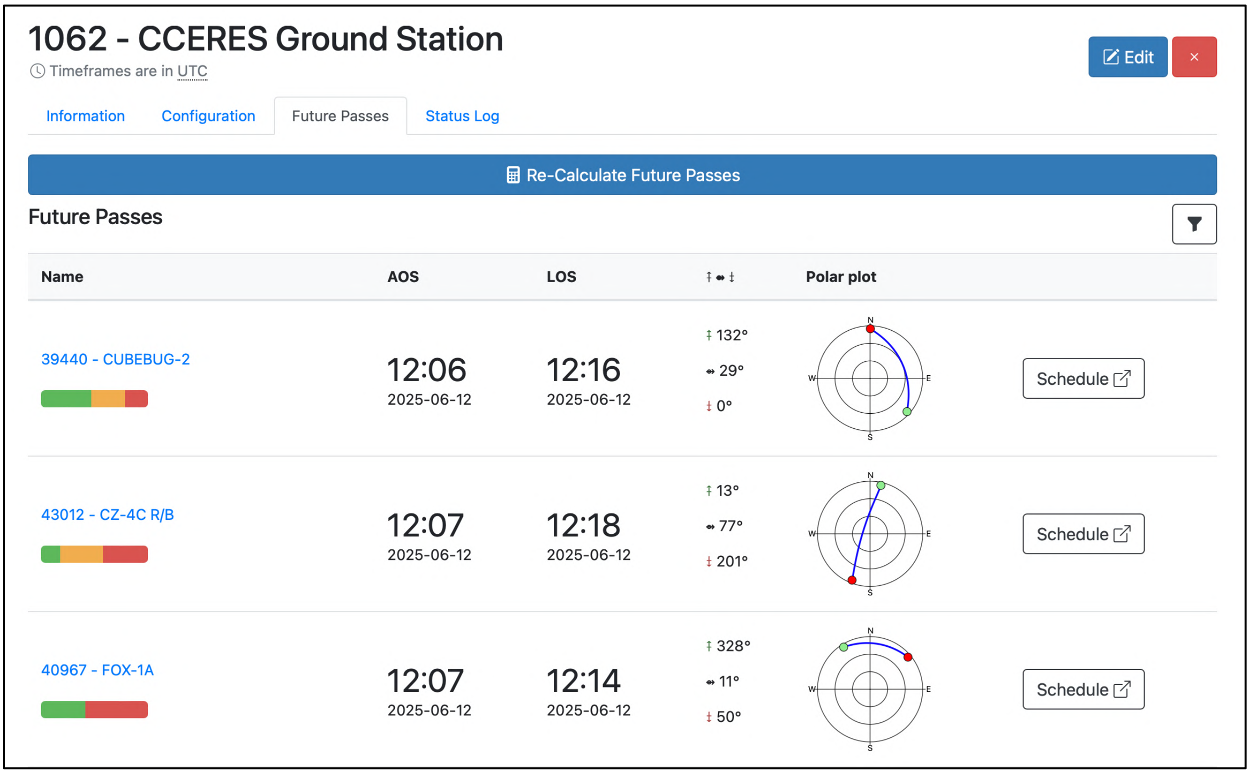


Figure 8 : Menu de sélection du satellite à observer

- Une fois le satellite sélectionné un identifiant est attribué à notre observation et une page de gestion de l'observation est ouverte. On peut alors y lire les informations principales de l'observation à venir :

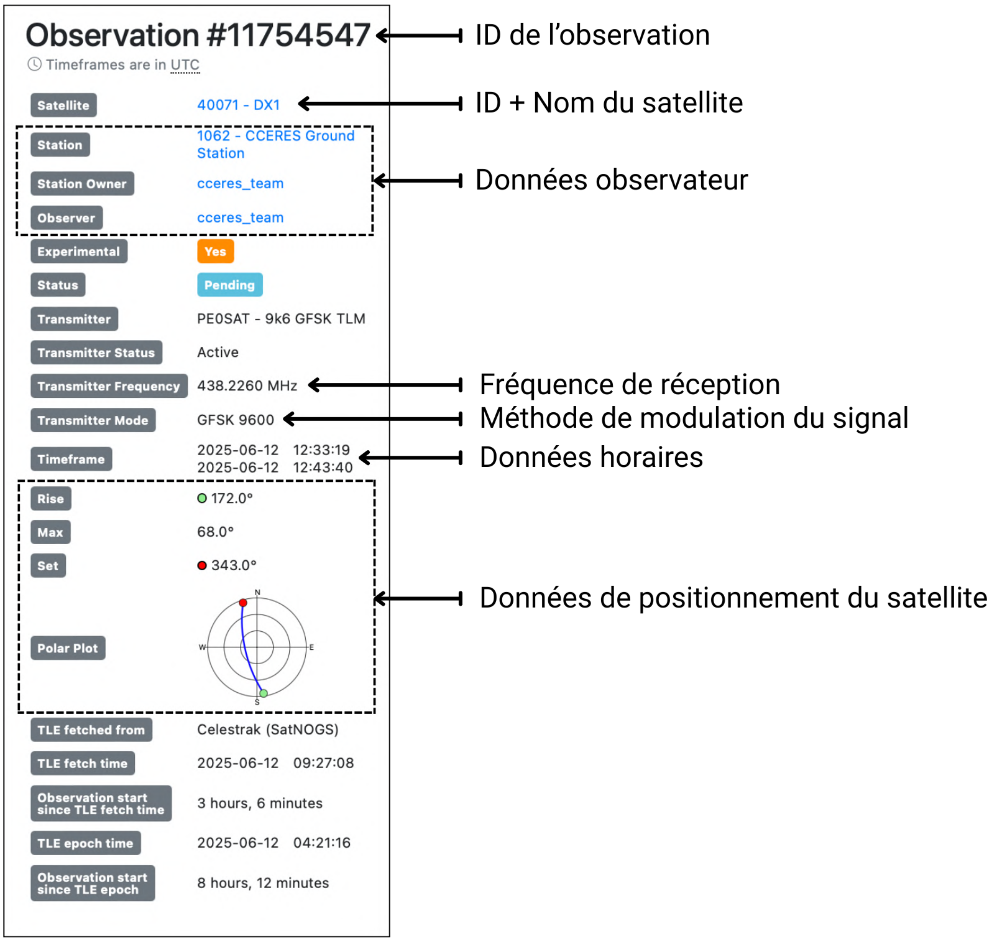


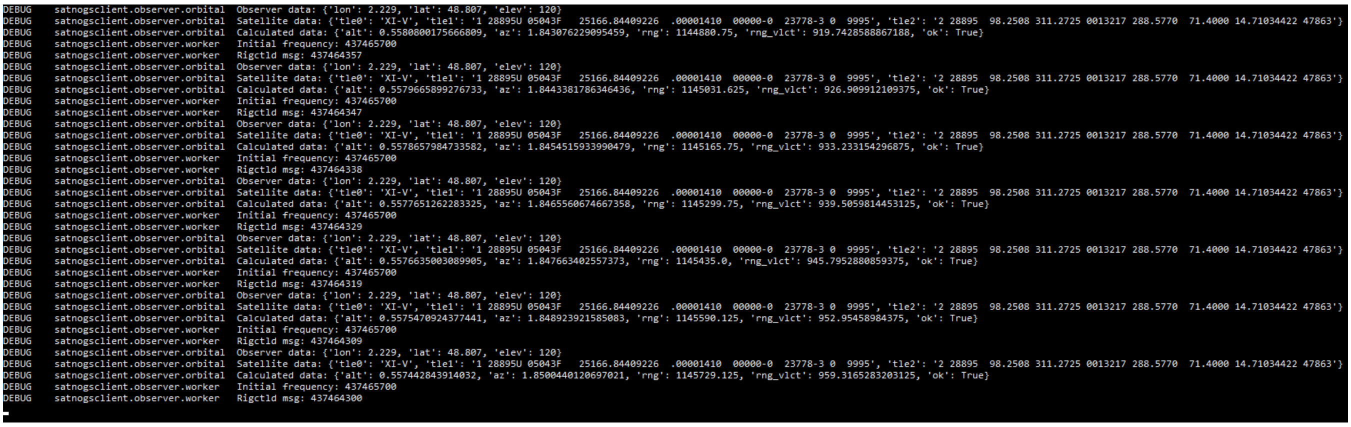
Figure 9 : Page de gestion de l'observation

L'ensemble de ces informations va permettre de configurer le matériel hardware de la station. Ce paramétrage (préparation du suivi, réglage de la bonne fréquence d'écoute …) sera effectué automatiquement par le client SatNOGS (c'est justement l'intérêt d'utiliser un client SatNOGS).

- Lorsque le satellite apparaît dans le champ couvert par la station sol, le suivi et l'écoute s'activent. Les antennes vont alors suivre la position du satellite dans le ciel.

Figure 10 : Mouvements des antennes au cours d'une observation

- Lors d'une observation, il est possible de suivre en direct le journal d'activité du client SatNOGS. J'ai développé une procédure permettant d'avoir accès en temps réel à celui-ci. Ce journal est notamment utile pour comprendre d'éventuels problèmes d'observation.



```
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5580800175666809, 'az': 1.843076229095459, 'rng': 1144880.75, 'rng_vlct': 919.7428588867188, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464357
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5579665899276733, 'az': 1.8443381786346436, 'rng': 1145031.625, 'rng_vlct': 926.909912109375, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464347
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5578657984733582, 'az': 1.8454515933990479, 'rng': 1145165.75, 'rng_vlct': 933.233154296875, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464338
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5577651262283325, 'az': 1.8465560674667358, 'rng': 1145299.75, 'rng_vlct': 939.5059814453125, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464329
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5576635003089905, 'az': 1.847663402557373, 'rng': 1145435.0, 'rng_vlct': 945.7952880859375, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464319
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.5575470924377441, 'az': 1.848923921585083, 'rng': 1145590.125, 'rng_vlct': 952.95458984375, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464309
DEBUG    satnogsclient.observer.orbital  Observer data: {'lon': 2.229, 'lat': 48.807, 'elev': 120}
DEBUG    satnogsclient.observer.orbital  Satellite data: {'tle0': 'XI-V', 'tle1': '1 28895U 05043F   25166.84409226  .00001410  00000-0  23778-3 0  9995', 'tle2': '2 28895  98.2508 311.2725 0013217 288.5770  71.4000 14.71034422 47863'}
DEBUG    satnogsclient.observer.orbital  Calculated data: {'alt': 0.557442843914032, 'az': 1.8500440120697021, 'rng': 1145729.125, 'rng_vlct': 959.3165283203125, 'ok': True}
DEBUG    satnogsclient.observer.worker   Initial frequency: 437465700
DEBUG    satnogsclient.observer.worker   Rigctld msg: 437464300
```

Figure 11 : Journal d'activité lors d'une observation

- Une fois l'observation réalisée, les données écoutées/observées seront accessibles directement sur l'interface web. Pour chaque observation menée, le spectre de fréquences brut de l'écoute est fourni. En plus de cela, si le constructeur/exploitant du satellite a fourni au service SatNOGS un module de décodage, un fichier de données décodées peut être accessible. Des exemples de résultats bruts/décodés seront donnés ci-dessous.

La configuration matérielle de la station sol de l'Observatoire de Paris - PSL (présence de deux antennes) permet l'écoute de satellites dans deux plages de fréquences :

- VHF : Very High Frequency (*30 MHz < freq < 300 MHz)*
- UHF : Ultra High Frequency (*300 MHz < freq < 3 GHz)*

Je présenterai dans ce rapport deux exemples d'observations réussies (une dans chaque plage).

a. Exemple de résultat en VHF

Pour la bande VHF, j'ai choisi de présenter une écoute d'un satellite NOAA (National Oceanic and Atmospheric Administration). Ce sont des satellites météorologiques en orbite basse, principalement dédiés à l'observation de la Terre et à la collecte de données environnementales. Ils sont particulièrement intéressants dans le cadre des communications radioamateurs et des stations sol, car ils émettent des signaux radio forts dans la bande VHF. De plus, le module de décodage est implémenté sur le réseau SatNOGS et permet d'obtenir une image prise par le satellite. Cela rend le résultat plus concret et intéressant.

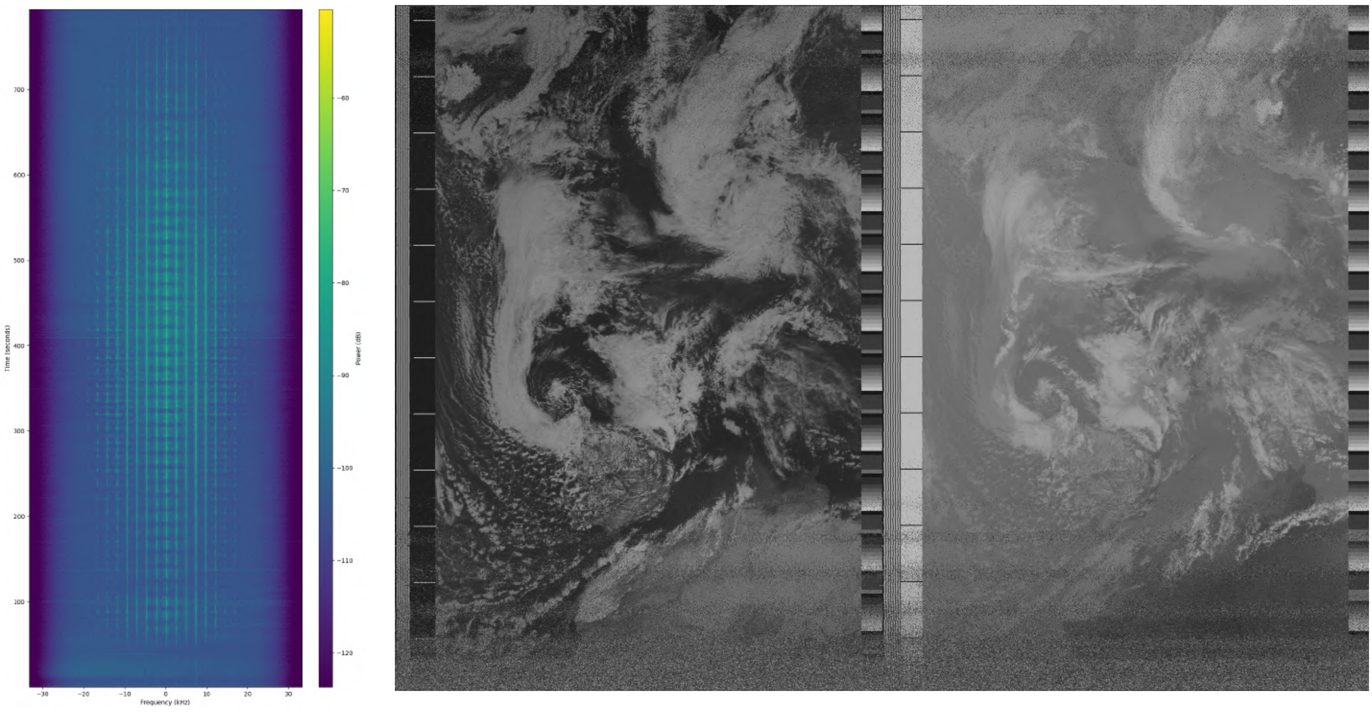
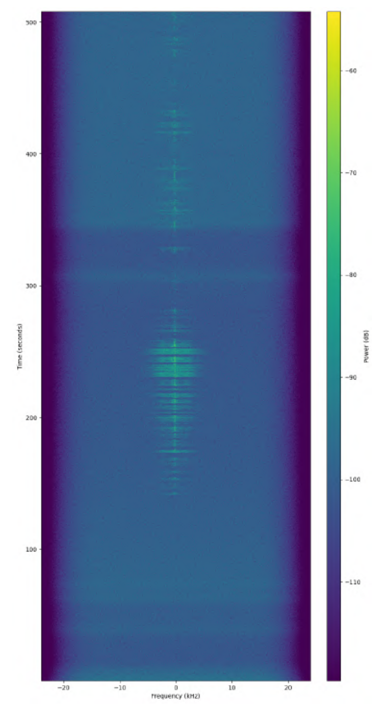


Figure 12 : Signal NOAA en fréquences (à gauche) et signal décodé (à droite)

b. Exemple de résultat en UHF

Pour ce qui est de la bande UHF, un bon exemple est les signaux transmis par l'ISS (International Space Station). En effet, ces derniers sont puissants (chose plutôt rare pour des signaux UHF) et de bonne qualité. De plus, comme c'était le cas pour les satellites NOAA, le réseau SatNOGS peut décoder certains signaux envoyés par l'ISS. C'est par exemple le cas de celui que je présente dans ce rapport qui permet d'obtenir un extrait audio de conversations.



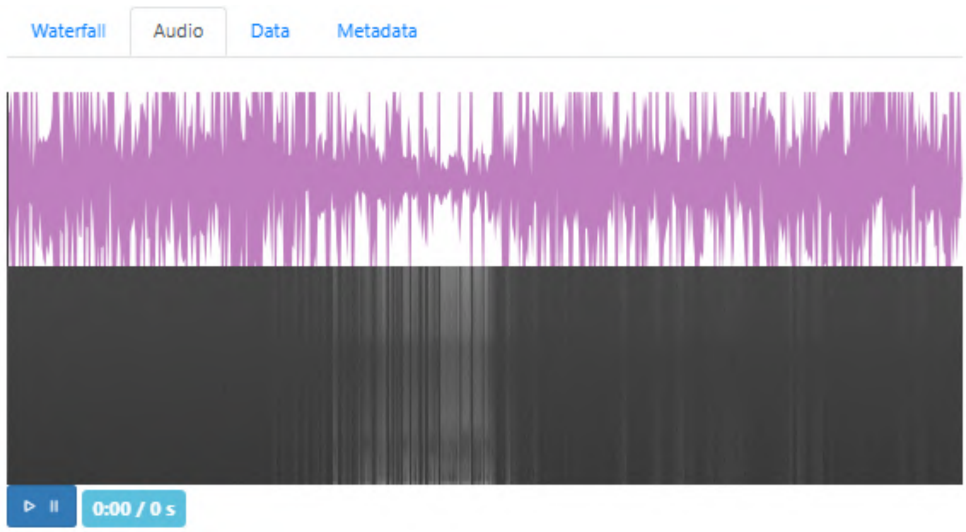


Figure 13 : Signal ISS en fréquences (à gauche) et signal décodé (en haut)

[Iss_audio_record]

Bien que satisfait des résultats obtenus, ce stage a également été marqué par de nombreuses difficultés, parfois indépendantes des aspects logiciels de ma mission. La résolution de ces points bloquants s'est néanmoins révélée particulièrement formatrice. Elle m'a conduit à développer des méthodes rigoureuses pour diagnostiquer les problèmes rencontrés, identifier leur origine et y apporter des solutions adaptées. C'est pourquoi il me paraît pertinent de revenir sur certaines de ces difficultés dans ce rapport.

VII.2 Problèmes et solutions apportés

a. Isolation et indépendance des environnements Docker

Comme évoqué précédemment, l'un des principaux avantages de l'utilisation de conteneurs dans le développement logiciel réside dans l'isolation des environnements qu'ils créent. En effet, un conteneur Docker est, par nature, indépendant du système d'exploitation hôte, assurant ainsi une portabilité et une reproductibilité accrues. Dans le cadre de ce projet, cette indépendance était particulièrement recherchée, car elle permettait de déployer un client SatNOGS avec des contraintes matérielles et logicielles minimales.

Cependant, mon cas d'utilisation présentait une spécificité importante : le client SatNOGS devait interagir avec des périphériques physiques connectés directement à l'ordinateur de la station sol. Il a donc été nécessaire d'établir certaines liaisons entre l'environnement du conteneur et celui de l'hôte. C'est notamment le cas du contrôleur moteur, utilisé pour orienter les antennes et pour la clé SDR (Software Defined Radio), qui permet la réception des signaux satellites à la bonne fréquence. Ces deux dispositifs doivent être directement accessibles par le client SatNOGS pour assurer le bon fonctionnement de la station.

Heureusement, l'architecture de Docker permet ce type d'interactions spécifiques, notamment via le partage de périphériques entre l'hôte et le conteneur. Toutefois, la mise en œuvre de telles liaisons s'est révélée à la fois complexe et délicate, car elle nécessite une compréhension approfondie du fonctionnement interne des conteneurs, ainsi que de leurs interactions avec le système hôte.

J'ai donc consacré beaucoup de temps à la configuration et à la sécurisation de ces connexions, avec pour objectif de les rendre aussi robustes et pérennes que possible. Il était essentiel que ces liaisons ne soient pas remises en cause à chaque évolution de l'architecture logicielle ou à chaque redéploiement du système. Ce travail m'a ainsi permis d'acquérir une meilleure maîtrise de la gestion fine des ressources matérielles dans un environnement conteneurisé.

b. Architecture logicielle inédite

Une autre difficulté majeure que j'ai rencontrée tient au caractère inédit et pionnier de ma mission. Si l'installation d'un client SatNOGS "classique" est aujourd'hui bien documentée et relativement courante, son intégration au sein d'un conteneur Docker représentait, en revanche, une approche entièrement nouvelle. À ma connaissance, aucune documentation officielle ni exemple public n'existent à ce jour pour ce type d'implémentation.

Cette absence de références m'a contrainte à naviguer entre deux univers techniques, Docker d'un côté, SatNOGS de l'autre, en croisant leurs documentations respectives afin de surmonter les problèmes rencontrés. Ce travail d'exploration constante, bien que stimulant, s'est souvent avéré complexe et chronophage, notamment en l'absence de solutions toutes faites ou de retours d'expérience similaires.

J'ai néanmoins pu m'appuyer sur une ressource précieuse : la communauté de la Libre Space Foundation (LSF), en charge du développement et de la maintenance du client SatNOGS. Très active et accessible, cette communauté dispose d'un forum dédié, sur lequel je me suis investi afin d'échanger autour des problématiques techniques rencontrées durant le projet. Grâce à cet espace collaboratif, j'ai pu bénéficier de l'expérience et des conseils de radioamateurs expérimentés, certains impliqués depuis de nombreuses années dans le déploiement et le fonctionnement de stations sol. Leurs retours m'ont permis d'orienter plus efficacement mes recherches et de résoudre plusieurs blocages techniques.

c. Dysfonctionnement de la partie UHF

Le principal problème rencontré au cours de mon stage concernait une disparité marquée dans les performances d'observation selon la bande de fréquence utilisée. En effet, alors que les observations de satellites émettant en VHF donnaient d'excellents résultats, celles des satellites émettant en UHF étaient, quant à elles, très faibles voire totalement absentes dans la majorité des cas. J'avais alors réussi à remettre en fonctionnement la station sol avec une nouvelle architecture logicielle cependant, les résultats en UHF n'étaient pas satisfaisants.

La principale difficulté a alors été d'identifier la source exacte du problème. Plusieurs causes potentielles pouvaient l'expliquer, rendant le diagnostic particulièrement complexe. De plus, le fait d'obtenir malgré tout certains signaux UHF, bien que faibles, a contribué à brouiller l'analyse. Cela rendait le dysfonctionnement moins évident à caractériser : il ne s'agissait pas d'une absence totale de réception, mais d'une dégradation partielle et irrégulière, parfois compensée par l'intensité d'émission de certains satellites.

J'ai alors essayé de mettre en place une approche structurée et méthodique pour trouver la source du problème. Je suis parti de ce premier constat : lors de l'utilisation précédente de la station sol (communication avec PicSat), les signaux UHF étaient bien détectés, or la différence majeure entre la précédente implémentation et la nouvelle concerne l'aspect logiciel, j'ai donc orienté mes recherches dans cette direction.

- <u>Un problème venant de mon architecture logicielle</u>

Dès le début de mes investigations, j'ai pu écarter l'hypothèse d'un problème lié à l'installation du client SatNOGS, puisque la réception des signaux en VHF s'effectuait sans difficulté. Par ailleurs, aucun dysfonctionnement n'était signalé dans le journal d'activité du client. Mon attention s'est alors portée sur un paramètre clé de la configuration de la station sol : le gain d'amplification des signaux. En effet, j'ai suspecté un mauvais réglage du gain, ce dernier ayant un impact plus marqué sur la réception en UHF, naturellement plus faible que celle en VHF.

Le système de gain, associé au SDR, repose sur trois valeurs à renseigner dans le client, chacune correspondant à un niveau d'amplification différent au cours du traitement du signal. Ces paramètres dépendent fortement de deux facteurs : l'environnement radioélectrique (niveau de bruit ambiant) et la configuration matérielle spécifique de la station sol.

Après avoir mené de nombreuses recherches et sollicité les conseils de la communauté LSF, j'ai effectué plusieurs séries de tests en modifiant ces valeurs. Toutefois, aucun des jeux de paramètres testés n'a permis d'améliorer de manière significative la qualité de réception des signaux UHF.

Mes soupçons se sont alors orientés vers un problème de configuration matérielle. Bien que le matériel utilisé ait déjà fait ses preuves par le passé, la station sol ayant été entièrement démontée, déplacée puis remontée, un dysfonctionnement matériel ne pouvait être exclu. Dans le cadre de cette analyse, j'ai pu écarter les composants communs aux chaînes de réception VHF et UHF. Mon attention s'est donc portée plus précisément sur la partie UHF du système, notamment sur l'hypothèse d'un câble coaxial défectueux.

- Altération d'un câble coaxial

Un câble coaxial exposé durablement aux conditions extérieures peut subir divers types de dégradations susceptibles d'altérer la qualité du signal transmis. Les variations de température, l'humidité, les rayons UV ainsi que l'oxydation des connecteurs peuvent provoquer des pertes de signal, des interférences ou une atténuation accrue. De plus, des microfissures dans l'isolant ou une infiltration d'eau peuvent modifier les caractéristiques d'impédance du câble, entraînant une dégradation significative de la réception, notamment dans les bandes de fréquences les plus sensibles comme l'UHF.

Afin de vérifier l'intégrité des câbles coaxiaux suspects, j'ai réalisé deux types de tests à l'aide d'un multimètre :

1. **Test de continuité du conducteur central** : j'ai mesuré la résistance électrique entre les deux extrémités du conducteur central (le cœur du câble). Une faible résistance indiquait que le conducteur n'était pas rompu.

2. **Test de continuité de la gaine conductrice** : j'ai effectué la même mesure entre les deux extrémités de la tresse métallique (la gaine extérieure). Là encore, une faible résistance permettait de confirmer la continuité du blindage.

3. **Test d'isolation entre cœur et gaine** : j'ai ensuite mesuré la résistance entre le cœur du câble et la gaine à une même extrémité. Une résistance très élevée (idéalement infinie) indiquait une bonne isolation, confirmant l'absence de court-circuit interne.

J'ai également réalisé un test dynamique afin d'évaluer la qualité de transmission du câble. Pour cela, j'ai injecté un signal de type créneau (signal carré) à une extrémité du câble à l'aide d'un générateur de fonctions, puis j'ai observé le signal à l'autre extrémité à l'aide d'un oscilloscope.

Deux paramètres principaux ont été mesurés :

1. **L'amplitude du signal en sortie**, comparée à celle du signal injecté, afin de détecter une éventuelle atténuation due à des pertes résistives ou à une dégradation du câble.
2. **Le temps de montée (Tr90)**, c'est-à-dire le temps nécessaire au signal pour passer de 10 % à 90 % de son amplitude maximale. Une augmentation significative de ce temps, par rapport à un câble de référence, peut indiquer une dégradation de la bande passante du câble, souvent causée par le vieillissement ou une détérioration de la structure interne.

Pour réaliser une mesure de référence, je me suis appuyé sur un câble neuf provenant du même lot que ceux utilisés pour la station sol, mais conservé en intérieur, dans le bureau, à l'abri des intempéries.

Le tableau suivant présente mes résultats :

Tableau 2 : Vérification des câbles coaxiaux

ID	Description	Resistance			Vamp (V)	Tr90 (ms)
		Coeur-Coeur (Ohm)	Gaine-Gaine (Ohm)	Coeur-Gaine (Ohm)		
C0	Câble labo	0,20	0,20	NA	1,93	4
C1	SDR --> Switch	0,20	0,20	NA	1,95	4,00
C2	Bias Tee --> LNA	0,20	0,20	NA	1,93	4,00
C3	LNA --> Antenna	0,20	0,20	NA	1,95	4,00

Les valeurs obtenues m'ont permis d'écarter l'hypothèse d'une éventuelle altération des câbles coaxiaux. J'ai donc poursuivi mes investigations en me concentrant sur un autre élément matériel spécifique à la réception UHF : le Low Noise Amplifier (LNA).

- <u>Dysfonctionnement du Low Noise Amplifier</u>

Le LNA (Low Noise Amplifier) est un amplificateur placé aussi près que possible de l'antenne afin de renforcer les signaux faibles dès leur réception, tout en introduisant un minimum de bruit supplémentaire. Il est particulièrement utile dans les bandes de fréquence comme l'UHF, où les signaux provenant des satellites sont souvent très atténués en raison de la distance, des pertes dans les câbles et des interférences environnementales. En amplifiant le signal avant qu'il ne subisse des pertes significatives, le LNA améliore le rapport signal/bruit global du système, ce qui permet d'obtenir des observations plus fiables et exploitables.

Dans un premier temps, j'ai vérifié si le LNA était correctement alimenté. En l'absence de bouton d'alimentation ou de témoin lumineux indiquant son fonctionnement, cette vérification n'était pas triviale. De plus, l'alimentation du LNA ne se fait pas par un câble dédié, mais via une tension injectée directement dans le câble de transmission du signal à l'aide d'un dispositif appelé bias-tee. Après avoir réglé le générateur externe sur la tension appropriée, un premier signe rassurant a été observé : un courant a bien circulé au moment du raccordement du LNA, indiquant qu'il recevait l'alimentation attendue.

Cependant, le fait que le LNA soit alimenté ne me permettait pas de conclure quant à son bon fonctionnement. J'ai alors eu l'idée suivante : si je parvenais à visualiser en temps réel les signaux reçus par l'antenne, je pourrais évaluer concrètement l'impact de la présence du LNA sur ces signaux. Sur les conseils d'un membre de la Libre Space Foundation, j'ai découvert le logiciel SDR++ (6), qui permet précisément de visualiser en direct le spectre des signaux transitant par le SDR. Après m'être formé à son utilisation, j'ai entrepris de l'intégrer à mon architecture logicielle. Cette tâche ne s'est pas avérée simple en raison de la structure en conteneurs Docker et de l'isolation stricte entre les environnements. J'ai néanmoins réussi à faire fonctionner l'outil, ce qui m'a permis dans un premier temps de visualiser le niveau de bruit radio ambiant, avec ou sans LNA activé :

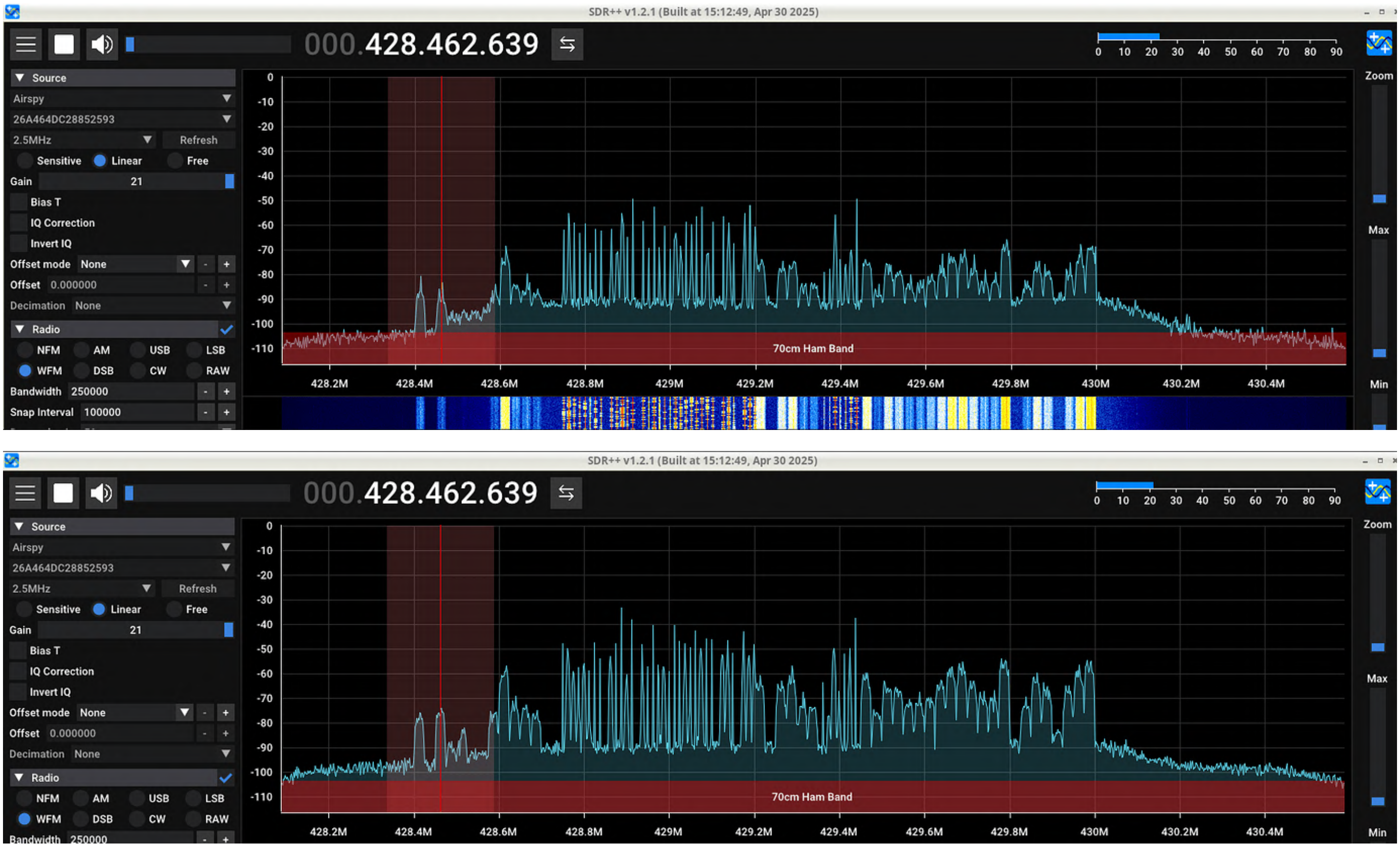


Figure 14 : LNA off (en haut) et LNA on (en bas)

Ce premier test m'a permis de montrer que le LNA a bien un impact sur les signaux. En effet, on peut voir qu'en l'absence du dispositif, aucun pic au-delà de -50dB n'est observé, tandis qu'après son activation, certains pics sont amplifiés jusqu'à -30dB.

Cependant, bien que ces résultats soient a priori satisfaisants, ils ne m'ont pas pleinement convaincu. En effet, les tests menés jusqu'alors portaient uniquement sur l'amplification du bruit radio ambiant, et non sur celle de signaux effectivement émis par des satellites. Par ailleurs, les mesures précédentes concernaient une bande de fréquence comprise entre 428 MHz et 430 MHz, alors que, selon la fiche technique du constructeur, le LNA présente des performances optimales dans la plage 430–440 MHz, qui correspond précisément à la bande UHF d'intérêt pour notre application. Il m'a donc semblé nécessaire d'observer le comportement du système lors du passage effectif d'un satellite afin de pouvoir évaluer concrètement l'impact du LNA sur la réception réelle des signaux.

Pour ce faire, j'ai dû retravailler l'architecture logicielle de la station. Il n'était en effet plus possible de m'appuyer sur l'utilisation du client SatNOGS pour assurer le suivi des satellites pour ce test, car ce dernier ne permet pas de visualiser en direct les signaux reçus par le SDR, un SDR ne pouvant être utilisé que par un seul processus à la fois. Heureusement, un logiciel open source, GPredict, (7) a été développé à destination des radioamateurs souhaitant suivre des satellites de manière plus flexible, sans dépendre du client SatNOGS. Je me suis donc formé à l'utilisation de ce nouvel outil, puis j'ai adapté et modifié en conséquence ma configuration Docker afin d'intégrer GPredict dans mon environnement logiciel.

Cela m'a permis de prendre conscience d'une chose importante : le LNA n'avait aucun impact sur les signaux satellites dans la plage UHF 430 MHz – 440 MHz. J'avais donc réussi à identifier une source potentielle du dysfonctionnement. Je souhaitais désormais caractériser plus précisément le comportement du LNA. En effet, bien que ce dernier semblait a priori fonctionner, le dernier test réalisé laissait entrevoir un comportement possiblement dégradé.

Suite à une discussion avec mon superviseur de stage, il m'a été conseillé de prendre contact avec une équipe du LIRA, spécialisée et équipée pour répondre aux problèmes de radiofréquence. Après leur avoir exposé en détail la problématique rencontrée, l'équipe m'a proposé d'apporter le LNA afin de procéder à une caractérisation complète de celui-ci à l'aide d'un analyseur de réseau vectoriel (VNA, pour Vector Network Analyzer). Cet appareil permet de mesurer précisément les performances en fréquence du dispositif, notamment son gain, son facteur de réflexion et sa bande passante effective.

Les mesures effectuées à l'aide du VNA ont permis de confirmer que le préamplificateur (LNA) s'allume correctement, comme en témoigne la consommation de courant observée. Le dispositif est également capable d'amplifier un signal. Toutefois, plusieurs anomalies ont été mises en évidence :

- Le pic de gain mesuré se situe autour de 420 MHz, alors que la fiche technique du LNA indique une bande de fonctionnement optimale entre 430 MHz et 440 MHz, soit une plage fréquemment utilisée pour les communications satellites en UHF.
- Le gain maximal mesuré est de seulement 9 dB à 420 MHz, ce qui est très en dessous du gain attendu de 26 dB (d'après la documentation technique).
- Aux fréquences correspondant à la majorité des satellites UHF (435–437 MHz), le gain chute à environ 5 dB, ce qui est insuffisant pour une amplification efficace dans le cadre des observations visées.

- Enfin, aucune variation du gain n'a été observée lorsque la tension d'alimentation du LNA a été modifiée entre +8 V et +15 V, alors qu'un gain compris entre 22 dB et 26 dB est normalement attendu sur cette plage de tension.

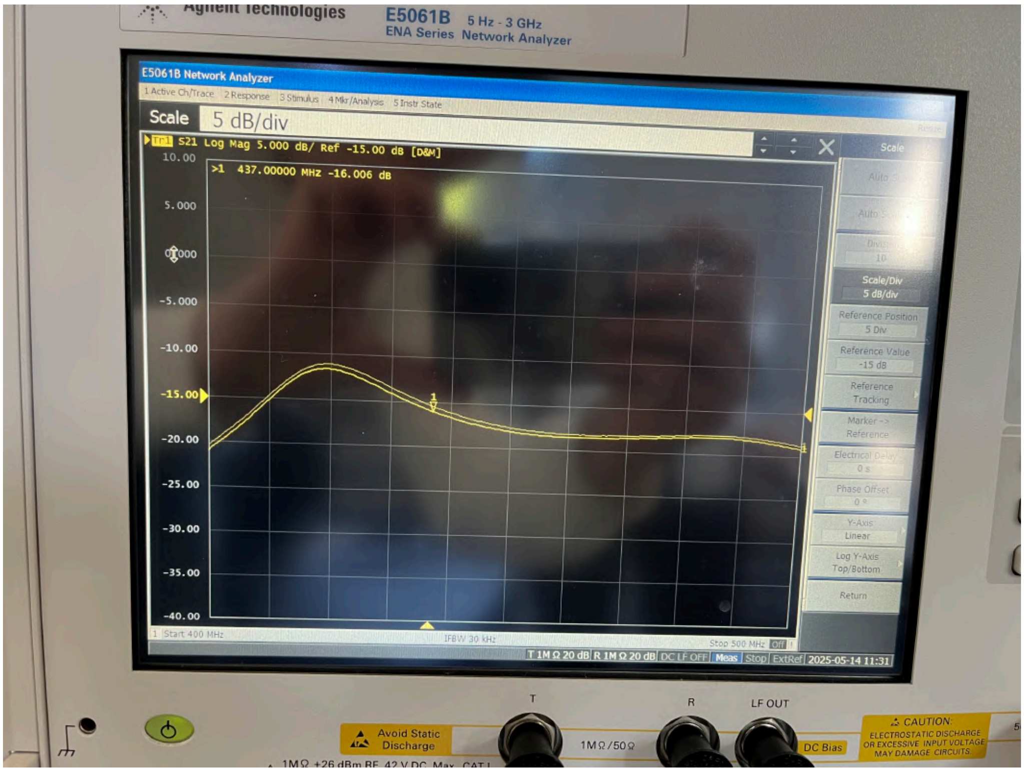


Figure 15 : Test au VNA du LNA

Cette analyse approfondie m'a permis de confirmer que le LNA était défaillant et constituait la principale cause des signaux faibles ou dégradés observés dans la bande UHF. L'origine exacte de ce dysfonctionnement reste difficile à établir, mais une hypothèse probable réside dans la dégradation progressive des composants internes due à une exposition prolongée à des conditions extérieures, parfois rigoureuses. Pour confirmer cette piste, j'ai procédé au démontage du boîtier afin d'inspecter la carte électronique et les composants : leur état laissait effectivement supposer une altération significative liée au vieillissement et aux intempéries.

Une nouvelle phase de résolution du problème s'est alors ouverte : la recherche et l'acquisition d'un nouveau LNA adapté aux besoins actuels et futurs de la station sol. Ce composant étant conçu pour durer dans le temps et rarement remplacé, il m'a semblé essentiel de prendre en compte les objectifs d'usage à long terme, notamment en lien avec le développement du projet NANOSATGRID, visant à créer un réseau interconnecté à grande échelle de stations sol. Le marché propose une large variété de LNA, aux performances variables en termes de gain, de bande passante couverte et de bruit introduit. J'ai ainsi mené une étude comparative des différentes options disponibles, que j'ai synthétisée dans une présentation destinée à l'équipe de CENSUS en charge de la station sol.

Cette présentation m'a permis d'exposer plusieurs solutions possibles, d'en discuter les avantages et les inconvénients, et de proposer différents scénarios d'évolution en fonction du choix retenu.

Tableau 3 : LNA envisageables et caractéristiques

	HP MVV 432	MVV 432-VOX	MINI 70	SP 7000
Noise Factor	0.4 – 0.7 dB	0.8 – 1.2 dB	0.7 – 1.0 dB	0,9 dB
Gain	~10–20 dB	~22-26 dB	~12–25 dB	~10–20 dB
TX Power (VOX)	750 W	350 W	150 W	100 W
TX Power (PTT)	Up to 5 kW PEP	500 W	Not specified	500 W
Insertion Loss	< 0.02 dB	< 0.5 dB	≤ 0.5 dB	0.3 dB
Power Supply	13.5 V via multipolar patch cord	13.5 V, ~200 mA (remote via coax or separate)	13.5 V, ~100 mA (remote via coax or separate)	12-14 V, ~250 mA (remote via coax or separate)
RF Connectors	7/16 standard sockets	N-type sockets	N-type sockets	N-type sockets
Price	on request	259€	168€	~320€

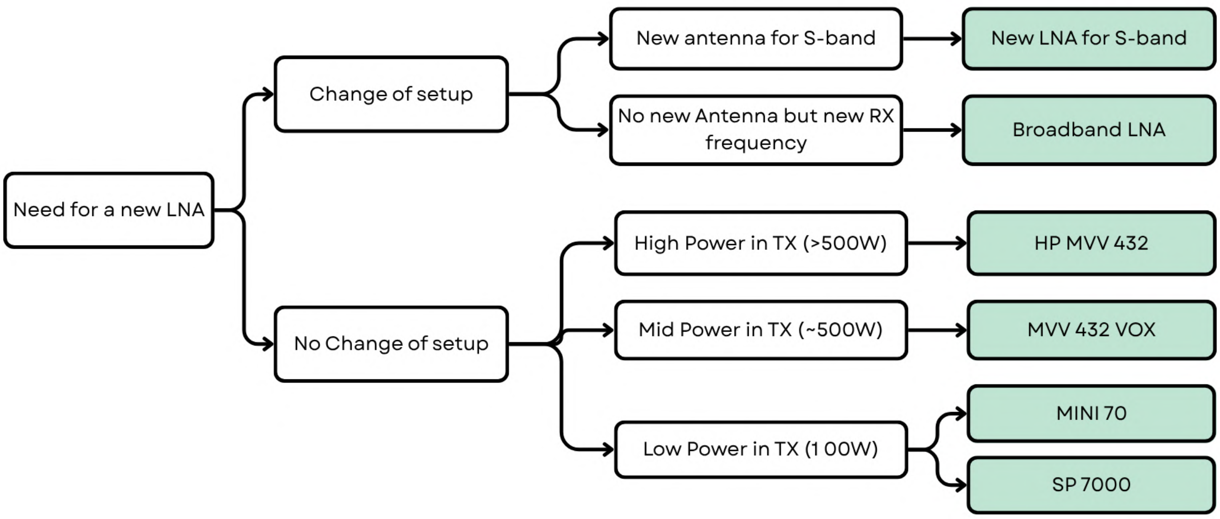


Figure 16 : arbre de décision pour le remplacement du LNA

À l'issue de ce travail de comparaison, nous avons choisi un nouveau LNA, jugé adapté aux exigences techniques et aux perspectives d'évolution de la station sol. J'ai donc entamé les démarches de commande. Toutefois, le fabricant m'a informé qu'en raison de la rupture de certaines pièces, la commande ne pourrait avoir lieu avant un délai de deux mois. Je ne pourrai donc malheureusement pas être présent pour constater directement les effets de ce remplacement sur la qualité des signaux reçus, mais je resterai en contact avec l'équipe afin de suivre les retours et l'évolution du système.

VII. CONCLUSION

Au terme de ce stage, j'ai contribué à la remise en service et à la modernisation complète de la station sol du centre spatial CENSUS de l'Observatoire de Paris – PSL. En reprenant d'abord le travail matériel engagé précédemment, j'ai pu concentrer mon travail sur la refonte de l'architecture logicielle qui devait permettre de répondre aux quatre exigences définies en amont : portabilité multiplateforme, maintenance facilitée, reproductibilité des installations et robustesse sur le long terme. Mon choix d'installer le client SatNOGS grâce à un playbook Ansible au sein de conteneurs Docker a permis de déployer la station de façon entièrement automatisée, fiable et indépendante du système hôte et donc de satisfaire les objectifs et exigences de ma mission.

J'ai pu valider cette nouvelle configuration par la réception de signaux VHF et UHF, montrant la bonne interopérabilité des outils développés et la facilité de maintenance offerte par la nouvelle architecture. L'étude approfondie du dysfonctionnement identifié sur la chaîne UHF, et plus particulièrement du LNA, m'a permis d'imaginer et de mettre en place une approche méthodique de résolution de problèmes : du diagnostic logiciel initial, aux tests matériels (câblage, bruit, caractérisation VNA), en passant par l'intégration et la formation à de nouveaux outils (SDR++, GPredict). Si le LNA original s'est avéré défectueux, la sélection du nouveau LNA et le lancement de sa commande m'ont conduit à intégrer une réflexion à plus long terme sur l'évolution de la station sol et plus particulièrement sur le projet NANOSATGRID.

J'ai ainsi pu, en m'appuyant sur les travaux réalisés précédemment, concevoir une solution logicielle adaptée aux besoins actuels de la station sol, tout en anticipant d'éventuels développements futurs de cette dernière.

VIII.1 Mon expérience de stage

J'ai sincèrement apprécié mon stage au sein de CENSUS. Dès le début, quand M. Segret m'a présenté la mission, le sujet m'a tout de suite plu : je n'avais encore jamais travaillé ni étudié les communications spatiales, et c'était l'occasion de découvrir un domaine totalement nouveau pour moi. Ce qui m'a aussi beaucoup attiré était l'ambition du projet : contribuer à rendre l'Observatoire de Paris plus autonome dans ses capacités d'accès à l'espace. J'ai de même particulièrement aimé l'approche orientée "système" : bien comprendre les objectifs, les traduire en exigences claires, puis réfléchir à une solution concrète à tester et valider sur le terrain.

Un autre point que j'ai grandement apprécié a été l'équilibre entre travaux théoriques, travaux pratiques et travaux de programmation. J'avais conclu un précédent rapport de stage par un léger regret à propos de la mission que j'avais trouvée trop informatique (essentiellement de la programmation Python) et pas assez tangible. Ici, ce n'a pas été le cas, si j'ai adoré les aspects « programmation » de mon stage et j'étais content de pouvoir voir concrètement les résultats produits par mon travail lors d'observation de satellites.

Enfin, étant de nature très curieuse et toujours avide d'apprendre, ce stage m'a permis d'explorer de nombreux domaines qui me seront, j'en suis sûr, utiles dans mon futur parcours professionnel. J'ai notamment pu me former au développement logiciel dans un environnement conteneurisé, une compétence de plus en plus recherchée aujourd'hui. J'ai également découvert une nouvelle façon de structurer un projet à travers l'utilisation de GitLab, que je ne connaissais pas auparavant. Cette approche m'a beaucoup intéressé, même si je sais que j'ai encore beaucoup à apprendre pour la maîtriser pleinement.

VIII.2 Ma poursuite d'étude

Ce stage a, sans aucun doute, renforcé ma volonté et ma motivation à travailler à terme en tant qu'ingénieur système dans le domaine du spatial. Au-delà de ma mission à proprement parler, l'environnement professionnel dans lequel j'ai été immergé durant ces six mois m'a permis de mieux comprendre le rôle et les activités d'un ingénieur système. En effet, l'équipe CENSUS est composée de plusieurs ingénieurs système, ayant des parcours et expériences différents ce qui a grandement enrichi les échanges que j'ai pu avoir avec ces derniers. Cette mission a ainsi conforté mon choix de poursuivre mon cursus académique au sein du M2 OSAE (Outils et Systèmes de l'Astronomie et de l'Espace), une formation qui m'attire depuis un certain temps car elle me permettra de découvrir, comprendre et pratiquer les concepts de l'ingénierie système directement appliqués au secteur spatial. Je suis impatient de débuter cette nouvelle formation et je suis convaincu que ce stage de M1 sera un atout pour ma réussite dans cette nouvelle aventure !

VIII.3 Mon stage de fin d'étude

Avec un peu de recul sur mon parcours académique, je réalise que mes expériences de stage dessinent un cheminement progressif et complémentaire. Mon premier stage, réalisé en première année d'école d'ingénieur, m'a plongé dans l'environnement dynamique d'une startup. Le second, effectué en deuxième année, s'est déroulé dans un laboratoire de recherche et portait sur un sujet à dominante théorique. Enfin, ce dernier stage, bien qu'également réalisé au sein d'un laboratoire scientifique, portait davantage sur une mission d'ingénierie concrète et appliquée. Pour compléter ce parcours, j'aimerais grandement pouvoir effectuer mon stage de fin d'études dans l'industrie, et plus précisément au sein d'un grand groupe du secteur spatial (CNES, ArianeGroup, Airbus, etc.). Une telle expérience me permettrait d'avoir un aperçu des pratiques, méthodes et organisations propres au monde industriel, et de comparer ces approches à celles que j'ai déjà pu découvrir. Cela me donnerait aussi les clés pour mieux définir le type de poste et d'environnement professionnel dans lequel je pourrais m'épanouir pleinement et être heureux !

IX. RÉFÉRENCES

IX.1 Figures

IX.2 Tableaux

IX.2 Bibliographie

1. **SCHWORER, G.** *PicSat Ground Segment Software Description.* 2017.

2. **Michele, CIOCCHETTA.** *Station au sol - Manuel d'utilisation.* 2017.

3. **MULUMBA, Alvin Michael.** *Meudon Ground Station "SSOL".* 2024. CENSUS-2024-166.

4. **Silla, DJIBRIL DRAMANE Souwédou.** *Modernisation de la station sol UHF/VHF du CENSUS.* 2021.

5. **Foundation, Libre Space.** SatNOGS home page. *SatNOGS.* [En ligne] https://satnogs.org.

6. **Inc, Docker.** Docker home page. *Docker.* [En ligne] https://www.docker.com.

7. **Inc, SDR.** *SDR++ User guide.* 2022.

8. **Alexandru, OZAEC Csete.** *Gpredict User Manual.* 2017.

X.1 Acronymes

- **PSL** : **P**aris **S**ciences et **L**ettres
- **SSOL** : **S**tation **SOL**
- **GS** : **G**round **S**tation
- **SatNOGS** : **Sat**ellite **N**etworked **O**pen **G**round **S**tation
- **LSF** : **L**ibre **S**pace **F**oundation
- **CENSUS** : **CE**ntre pour les **N**anosatellites en **S**ciences de l'**U**niver**S**
- **LIRA** : **L**aboratoire d'**I**nstrumentation et de **R**echerche en **A**strophysique
- **SDR** : **S**oftware **D**efined **R**adio
- **VHF** : **V**ery **H**igh **F**requency
- **UHF** : **U**ltra **H**igh **F**requency
- **LNA** : **L**ow **N**oise **A**mplifier
- **VNA** : **V**ector **N**etwork **A**nalyser
- **CNES** : **C**entre **N**ationale d'**E**tudes **S**patiales
- **ISS** : **I**nternational **S**pace **S**tation
- **NOAA** : **N**ational **O**ceanic and **A**tmospheric **A**dministration
- **SUTS** : **S**ciences de l'**U**nivers et **T**echnologies **S**patiales
- **OSAE** : **O**utils et **S**ystèmes de l'**A**stronomie et de l'**E**space

X.2 Documentation Technique

Vous trouverez, ci-après, la documentation technique de la station sol de l'Observatoire de Paris – PSL. Une première version avait été rédigée par un précédent stagiaire de M2. J'ai pour ma part relu, corrigé et complété ce document afin qu'il reflète fidèlement l'état actuel de la station, intégrant l'ensemble des développements réalisés durant mon stage.

Elle se compose de deux parties principales :

- **une spécification technique** du système, couvrant les aspects matériels et logiciels de la station sol

- **un guide d'utilisation** détaillé, destiné à accompagner l'exploitation opérationnelle de la station.

MEUDON GROUND STATION “SSOL”

Abstract

The updated technical specifications of the SSOL ground station in Meudon, Paris Observatory include maintenance procedures and operation instructions. Current versions of the hardware and software systems are described and justified to meet the ground station requirements. Furthermore, a detailed functional analysis is carried explaining the design process of the entire system. SSOL ground station shall operate in two modes namely: “SATNOGS” mode for only reception and “CENSUS” mode for both transmission and reception.

Author:

Aldric PARENT – CENSUS Intern, M1 at Observatoire de Paris – PSL (Paris, France).

Approval:

Boris SEGRET - Supervisor, CTO CENSUS

Revision	Date	Major Changes
3.x	06-MAY-25	Review of document and update by Aldric PARENT
2.x	08-AUG-24	Review of previous documents and update by Alvin Michael Mulumba: • C²ERES 2019-87, 14/08/2019, *Jean-Baptiste Jarnoux* • “Sécurité de la station Sol de Meudon”, 3/05/2019, *J-B. Jarnoux* • CENSUS 2019-85, v1.5, 30/08/2021, *Souwedou Silla Djibril Dramane.*

Table of Content

Table of Procedures

I. - Introduction

I.1. - *Acronyms*

GS	Ground station.
PDU	Power Distribution Unit
SATNOGS	Satellite Networked Open Ground Station
SSOL	Ground station name
TC	Telecommands
TLE	Two Line Element (orbital parameters)
TM	Telemetry
TNC	Terminal Node Controller

I.2. - *References*

[1] ANFR (2018), Modeling of radio sites and security perimeters for the public, France.

[2] Decree No. 2002-775, Official Journal of the French Republic, INDI0220135D, 2002.

[3] Rasila, T., Ojala, A. (2024). National Regulation of Satellite Ground Stations: A Global Comparison. In: Ojala, A., Baber, W.W. (eds) Space Business. Palgrave Macmillan, Singapore. https://doi.org/10.1007/978-981-97-3430-6_8

[4] Gpredict software, [Accessed 28-10-2024] [Online] Available: https://oz9aec.dk/gpredict/

[5] GQRX software, [Accessed 28-10-2024] [Online] Available: https://www.gqrx.dk/

[6] GNU Radio software, [Accessed 28-10-2024] [Online] Available: https://www.gnuradio.org/

I.3. - Background

The ground station at the Meudon site was initially designed for communication with our PicSat nanosatellite. PicSat was a space mission developed at LIRA (ex-LESIA, one of the departments at Observatoire de Paris | PSL) from 2014 to 2017 to observe the exoplanetary transit of ***Beta Pictoris b.*** This ground station project was financed by CENSUS as a key asset in the development of scientific nanosatellites in France. The ground station is located at Observatoire de Paris | PSL in Meudon.

This document specifies the up-to-date status of the SSOL ground station. This ground station has had two different set-up designs in 2019 and 2021 respectively. Each new phase has seen development from the previous set-up both in hardware equipment and software configuration. The re-design of this ground station is derived from the challenges that were remaining from the previous versions. This document also describes the functional and technical specifications for the SSOL ground station to improve the effectiveness of its use and management.

The ground station is made available to the **SATNOGS network** which is a global management interface that operates multiple ground stations remotely. An observer with access to the SSOL ground station can listen to multiple satellites on the public communication database of the SATNOGS network. SSOL shall also transmit to other nanosatellites in the Low Earth Orbit (LEOs) launched or managed by the Observatory and its partners.

I.4. - Heritage

Development and (re-)installation of Meudon ground station was inspired by the challenges discovered during set-up and operation of the ground station:

- Poor location of the ground station. The previous model of the ground station was directly cable-stayed to a grass floor, out in the open with less controlled restrictions that compromised the security of the hardware equipment. Its low height also affected the quality of satellite observations due to poor line of sight and propagation concerns.
- The previous model of the ground station had low rigidity and poor stability that was not appropriate for regular maintenance.
- Lack of adequate protection mechanisms increased the possibilities of mechanical damage or results inaccuracies due to environmental factors.
- Incompatibility of the two ground station operation modes, **CENSUS** and **SATNOGS** with Debian and Ubuntu evidenced by failure of both modes to work with same Linux distribution.
- Regular occurrences of hardware component and system failure multiplying mechanical challenges.

I.5. - Project objectives

The top-level functional requirements for the development of the SSOL ground station are as follows:

- **R1**: The ground station shall be a visible asset in the nanosatellite community at large and the development of science sector of the region.
- **R2**: The ground station shall communicate with the Observatory's or partners' launched nanosatellites to aid in operating these satellites.
- **R3**: The ground station shall listen to other nanosatellites on the SATNOGS network.
- **R4**: NANOSATGRID *(next developments of the ground segment)* shall expand CENSUS' ability to operate partners' nanosatellites by integrating Meudon ground station to a broader network of partners' ground stations.

I.6. - Constraints / Limitations

- Like any emitting facility, regulations as in [3] apply to SSOL.
- CENSUS department shall have full authority over the SSOL ground station operations, in coordination with the specific missions of tracked satellites.
- The ground station in Meudon shall be used for satellite operations in Low Earth Orbit, but the future NANOSATGRID may expand to satellite operations for deep space.
- The Meudon forest is a hindrance to satellite observations below 30° elevation in certain directions and should therefore be considered in system design, operation processes and analysis.

II. - Functional analysis

This functional analysis is carried out based on the higher-level functional requirements stated above in Chapter I.

The SSOL ground station system has the following top level functions:

- Operate nanosatellites from Meudon.
- Contribute as a visible asset in the nanosatellite community.

Each top level function is considered an individual system, the first one being represented by System A and the second one as System B.

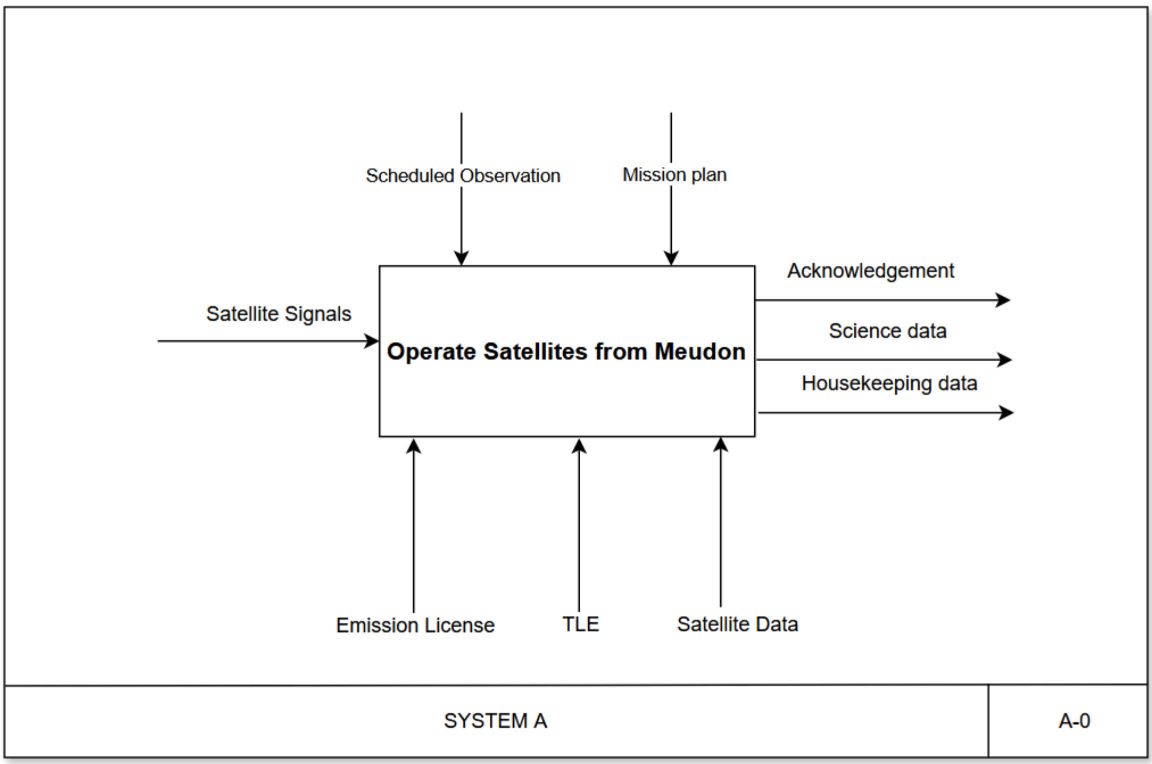


Figure 1: Top-Level Context Diagram for System A

In Figure 1, the function, *“**Operate Satellites from Meudon**”* requires the space mission needs for these launched nanosatellites as an input, and is supported by the satellite data sheet containing general information of the satellite, payload information, platform systems, launch, environmental information, the TLE data that leads to computation of passes during tracking and the ground station software and hardware components that support satellite operations. The system A is triggered by the satellite mission plan and the scheduled satellite observations on the SSOL ground station by the system users. Finally the function outputs science data of the detected satellites from scientific instruments, cameras, payload measurements and their housekeeping data that includes information about systems, power levels, temperatures, battery status, health to be analyzed for different science research or applications or missions.

System A: Operate nanosatellites from Meudon.

This system is broken down into three sub-functions as follows:

A1: Listen to nanosatellites.

A2: Interpret telemetry data from satellites.

A3: Send telecommands to CENSUS’ and partners’ satellites.

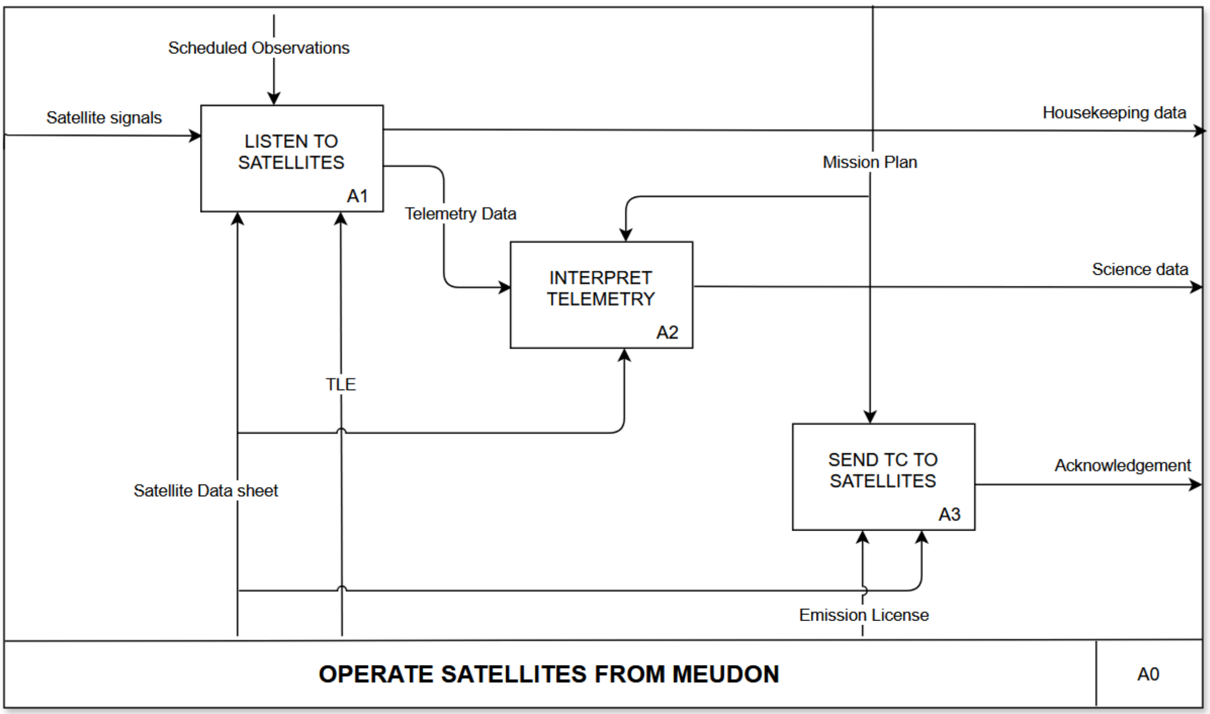


Figure 2: Functional Analysis of System A0

The signals from the detected nanosatellites are the starting point for Function A1 which is triggered by a scheduled observation on the SATNOGS network or on the GS command interface for other satellites. This function is supported by a data sheet containing the technical specifications and operational information of the satellite, satellites’ TLE that gives us orbital information that is turn computed by GPredict software to give satellite passes over the Meudon ground station location, and ground station hardware and software that receives the signals from the satellites and demodulate it into housekeeping data together with telemetry data. The telemetry data in its raw format becomes an input for Function A2 for interpretation. A2 requires decoding software to transform it into science data usable for research, and the data sheet containing the detected satellites’ specifications that guide analysis. It is also triggered by the mission timeline that outlines the satellite's objectives and operational modes thus providing a foundation for the analysis of telemetry data. Function A3 is triggered by the timeline of the specific mission plan to produce a satellite acknowledgment. A3 requires resources like the satellite operational information contained in the data sheet, GS hardware for telecommand encoding and emission, software for encoding and transmission support to the target satellite.

The functions of the sub-systems of System A are further broken down and analyzed in detail.

Sub-system A1: Listen to nanosatellites

This sub-system is broken down into three sub-functions as follows:

A11: Prepare to listen to nanosatellites.

A12: Detect nanosatellite signals.

A13: Demodulate satellite signals.

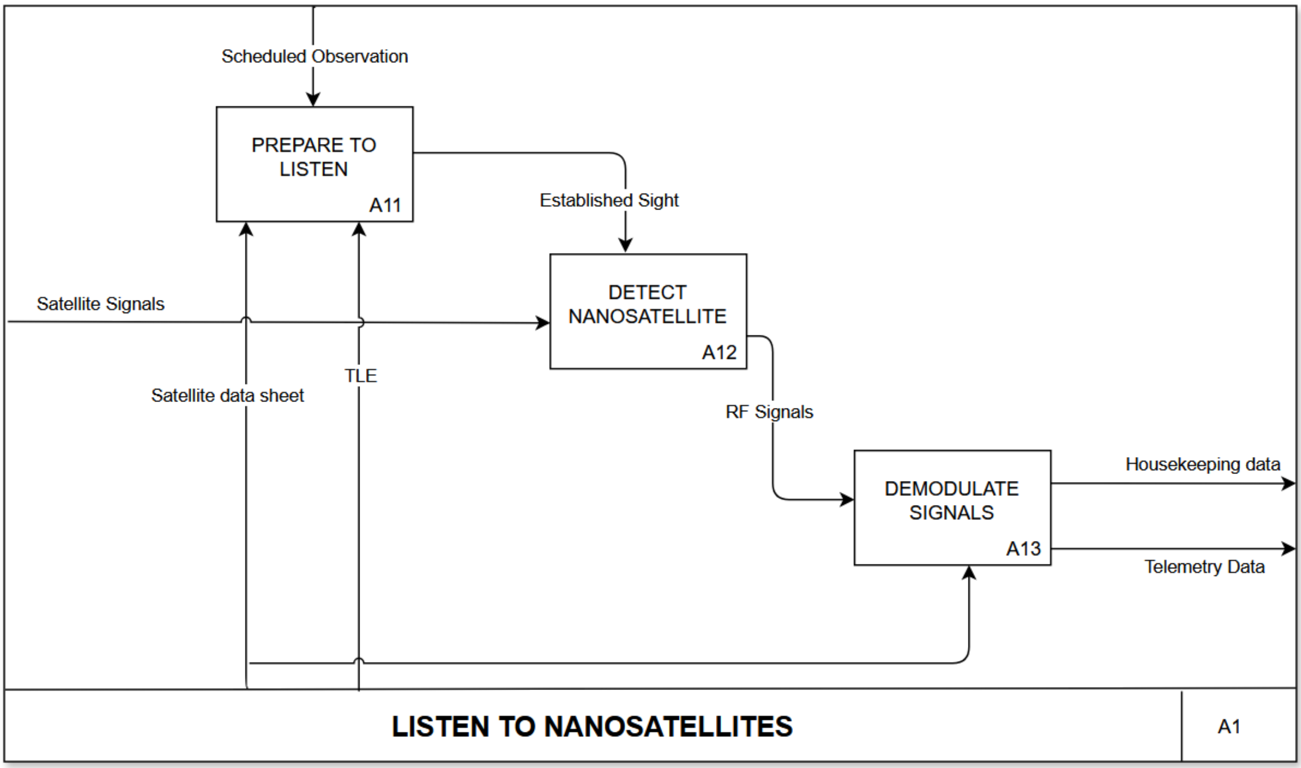


Figure 3: Functional Analysis of System A0

Function A11 is triggered by a scheduled observation in the SATNOGS network or on the GS command interface for other satellites. This function is supported by a data sheet containing all the satellite specifications, satellites' TLE that gives orbital information which is computed by GPredict software to give satellite passes over the Meudon ground station location, and antennas in conjunction with rotors that adjust their orientation to target the nanosatellites in orbit, and rotctld software that enables the antennas to turn. The output of A11 is the confirmation of communication with targeted satellites that triggers Function A12 whose input is the signals from various nanosatellites. A12 is supported by the UHF & VHF antennas that receive the satellite signals and ROTCTLD software that enables the antennas to turn to the specific direction. The RF signals feed Function A13 that is supported by an SDR that filters the wanted signal based on the information in the data sheet and SATNOGS client software or GQRX or GNU radio that perform the filtration, signal conversion and data processing for use to produce science and housekeeping data consumed by science researchers.

NB: Sub-system A2 shall not be detailed.

Sub-system A3: Send telecommands to Observatory satellites and partner satellites.

This sub-system is broken down into three sub-functions as follows:

A31: Prepare telecommands to be sent to the satellites.

A32: Prepare to transmit the telecommands to the satellites.

A33: Transmit telecommand packets to satellites.

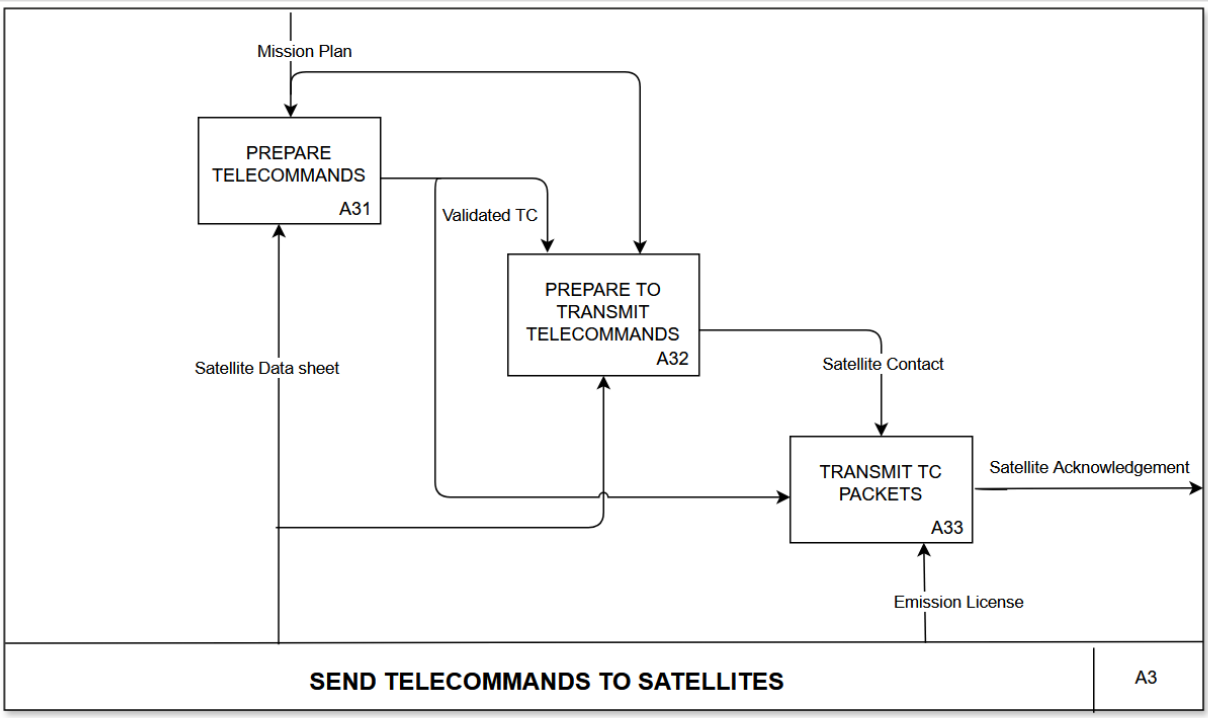


Figure 4: Functional Analysis of System A3

Function A31 is triggered by the timeline of the specific mission plan to produce telecommands in the actual format detectable by the satellite. The sequence and contents of the telecommands are defined by the satellite design. A31 requires resources such the satellite data sheet containing the satellite specifications, GS hardware for telecommand encoding. The complete execution of Function A31 triggers execution of Function A32 that is also controlled and determined by the timeline of the mission plan. A32 also requires satellite data sheet for complete identification and its orbital information, GPredict software for computation of satellite passes and satellite tracking, antennas with rotors to adjust position to target the satellite and a Kenwood transceiver that when configured confirms communication with the satellite. This established contact of the satellite observed on the command software (PicSAT) through a two-way communication with the satellite on-board computer triggers the execution of Function A33 that needs the validated telecommands for execution. An emission operator license is required for compliance regulation, the Kenwood Transceiver that modulates the telecommands into RF signals sent by antennas to the satellite, GS computer Linux operating software to ensure commands transmission and telemetry decoding software that confirms successful execution of telecommands based on acknowledgment data.

System B: Contribute as a visible asset in the nanosatellite community.

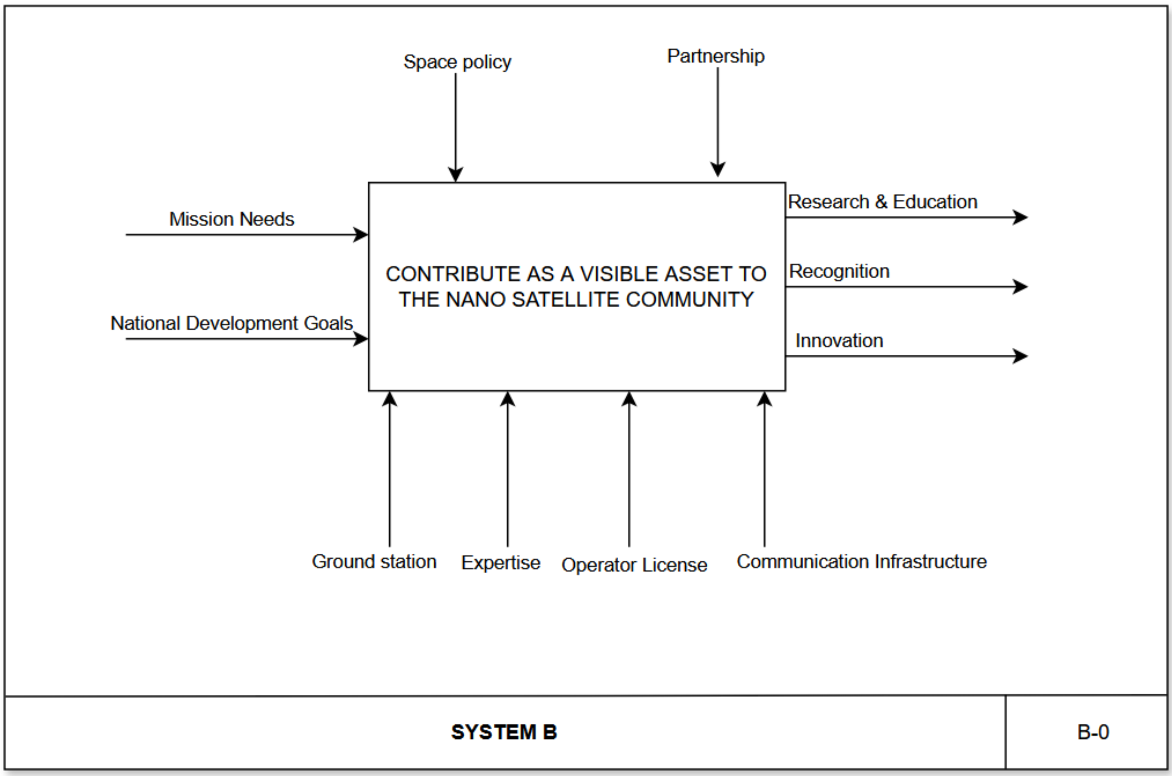


Figure 5: Top Level Context Diagram of System B

The function, *"**Contribute as a visible asset in the nanosatellite community**"* uses the space mission needs for the launched nanosatellites by the Observatory and its partners as well as the national objectives for development of the industry. The system function is triggered by the national policies concerning space operations and partnership with other public institutions, space agencies and regional satellite operators. It is supported by the Meudon ground station hardware and software to provide nanosatellite data for research and development, a satellite operator license, human expertise (satellite engineers, data scientists, astronomers and science researchers) that analyze and disseminate the satellite data to the community and communications infrastructure (website, social media, broadcast TV, symposiums) for publicity of ongoing projects and research towards the community at large. This system thereby highlights research & education of the scientists, students and the general public, the recognition of the Paris Observatory's engagement in the community and innovation in the nanosatellite industry for science missions.

NB. System B has not been detailed in this functional analysis.

III. - Technical Specifications

The Meudon ground station is described in two groups: **hardware** and **software** components.

III.1. - Hardware specifications

Overview

The current ground station takes the form of a pylon connected to two cross wire antennas (VHF and UHF) on opposite ends of a horizontal bar. The pylon spire structure is mounted onto a sliding platform structure attached to the electronic cabinet and a pulley system connected to the ground (Fig.6).

Figure 6: SSOL in Meudon (from webcam, optical distortions). Here the pylon spire structure is detached from the pulley system.

The pylon is then connected by an elevation rotor at the top and an azimuth rotor at the bottom. Both rotors are linked via wires to their controller device placed in the electronic cabinet. Both antennas are each connected to a corresponding surge fuse to protect them from lightning using coaxial cables that run from the respective antenna feed. The UHF antenna has an LNA amplifier attached under it on the pylon to boost the weak received signals. The coaxial cables are plugged in the electronic cabinet to their respective Bias-Tees. The VHF coaxial cable is approximately 10 m long while the UHF coaxial cable is approximately 8.5 m long all including a 10% margin to prevent cable intertwining during antenna motion. *[Refer to Figure 6].*

The mechanical support structure of the ground station is robust to resist wind and provide stability for the antennas (rubber shock-absorbents shall be added). This consists of a telescopic mast ended by a pylon spire with a triangular base, a sliding platform to use for maintenance operations and a stable mounting system set-up containing a pulley to provide stability. There are steel plates on the roof of the building that distribute the load of the structure. The structure is attached to the electronic cabinet that counter-weights the sliding platform. The antennas are raised to an altitude of 7 m above the ground.

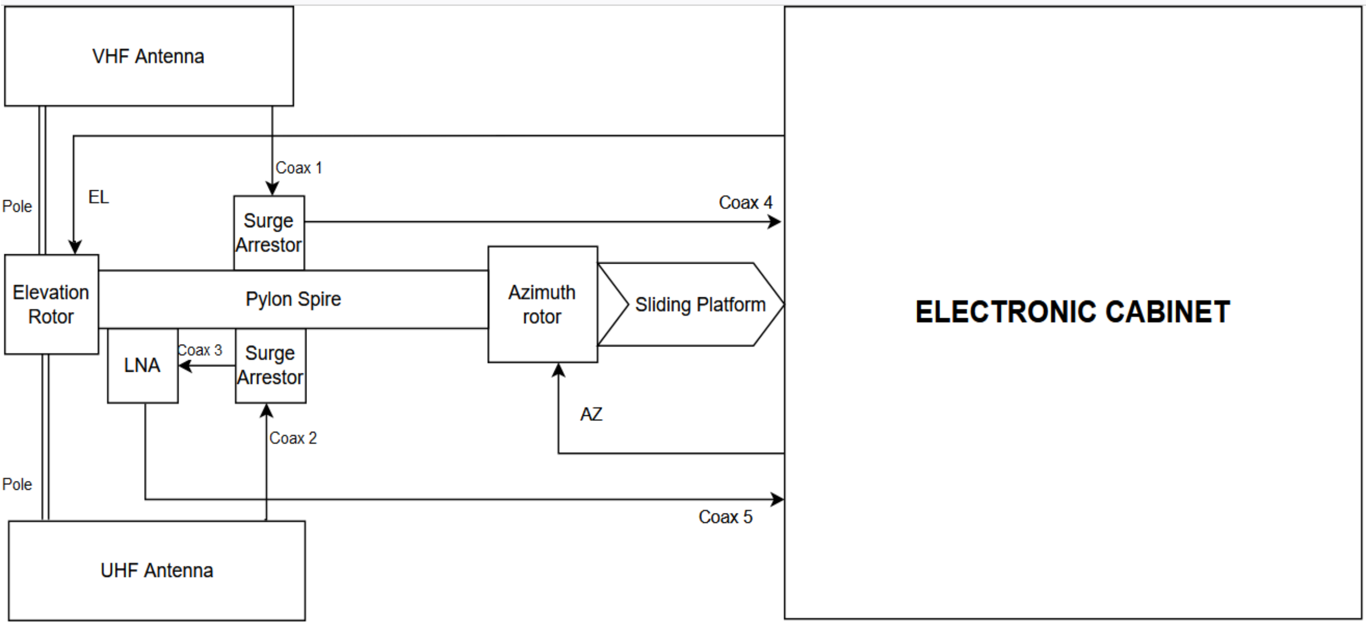


Figure 7: Block diagram of macroscopic view of the exterior ground station hardware at the roof top excluding the pulley

Electronic cabinet.

The electronic cabinet as shown in Figure 8 below houses the smaller and more delicate hardware components and provides the interfaces between the hardware and software systems. It also houses the network and electrical power supply connections.

Antennas

The two crossed Yagi antennas positioned at the top of the pylon each have a different frequency band. The UHF antenna measures 1.4 m in length and is of type Wimo WX-7036, 2×18 with 18 elements per antenna cross component has for optimization of high gain and directional accuracy. The VHF antenna measures approximately 4 m and is of type Wimo WX-220, 2×10 with 10 elements per antenna cross component. The ground station shall transmit (uplink) via the VHF antenna of frequency 145 MHz (wavelength of 2 m) and receive (downlink) via the UHF antenna of frequency range 436.5 MHz (wavelength of 70 cm). The antennas have been physically arranged to have a circular polarization. The antennas have multiple elements for directional gain of approximately 15 dBi. All the connections [*as shown in Figures 6 and 7]* to the surge arrestors, LNA(for the UHF antenna) and the RF devices in the electronic cabinet are made using coaxial cables. Finally the antennas are located at the top of a pylon spire and in an open space to ensure a clear line of sight for the incoming and transmitted signals to the various satellites passing over the ground station.

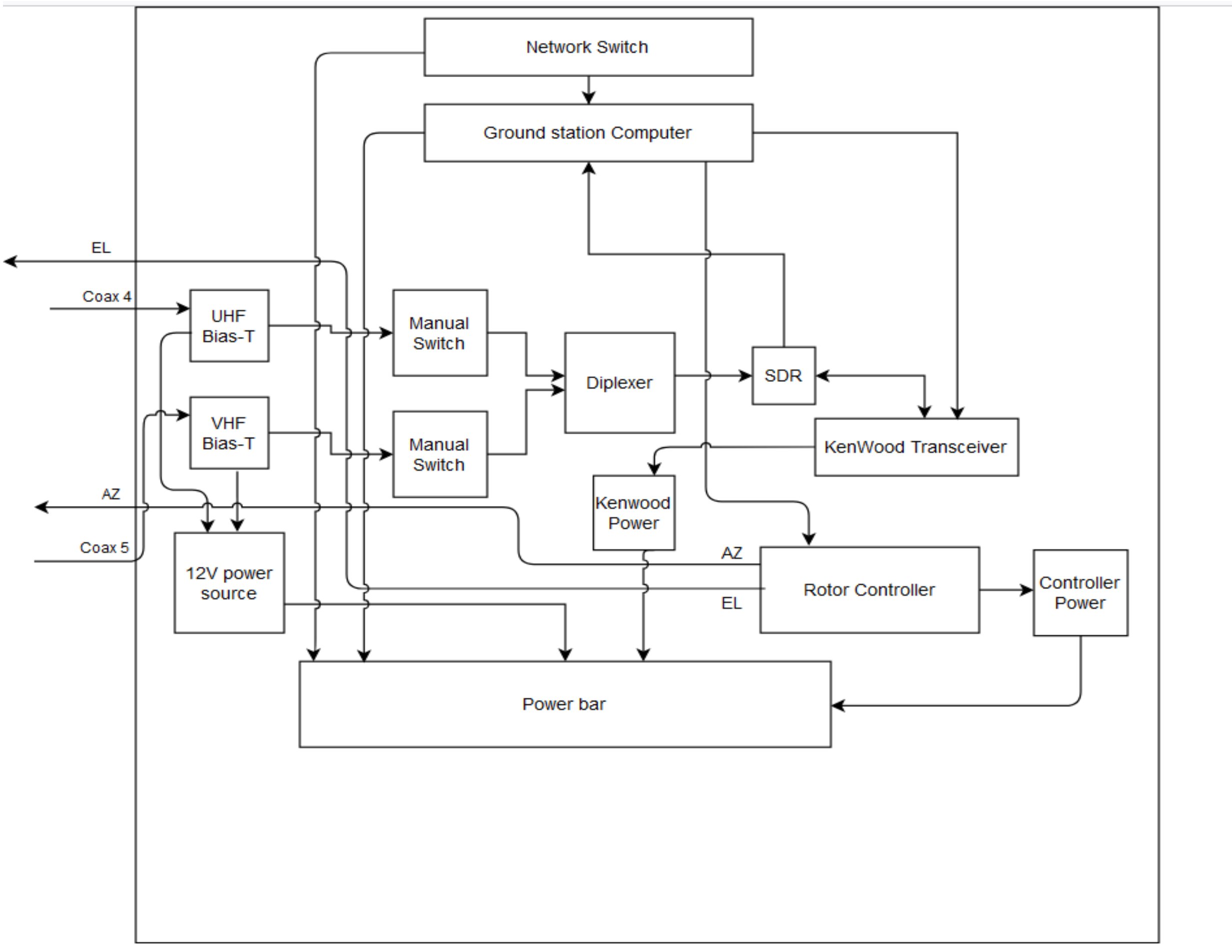


Figure 8: Interior of Electronic cabinet showing all device connections

Figure 9: Azimuth rotor

Azimuth rotor

The azimuth rotor, located at the base of the pylon, manages the azimuth adjustment of the antenna-pylon setup. It manages the clockwise and anti-clockwise rotation of the directional antennas according to the predicted direction of the satellite and has an angular range from -180 to 540 ° (2 full turns). These movements are controlled by the MD-03 controller (see below). The azimuth rotor used on the ground station is of SPID-RAU type, light version corresponding to a similar manufacture of the controller and demonstrates reduced mechanical complexity.

Elevation rotor

The elevation rotor, mounted on top of the pylon spire and connected to a horizontal bar. It manages the vertical direction in which both antennas are pointing, according to the predicted direction of the satellite and has an angular range from -180 to 180 degrees. These movements are controlled by the SPID MD-03 controller (see below). The elevation rotor used on the ground station is of SPID-RAEL type corresponding to a similar manufacture of the controller and demonstrates a gear drive reduction for holding position in the wind.

Figure 10: Elevation rotor

Pre-amplifier

Figure 11: LNA amplifier

The role of the pre-amplifier is to amplify the received signal from the antenna before it is processed. The MVV-432VOX LNA model was used because it allows powering of the amplifier via the coaxial cable and adjustment of the gain. The current model of the pre-amplifier provides a gain of 10 dB at 10 V and 16 dB at 15 V. This amplifier is connected to the UHF antenna, as close as possible below it, and attached to the pylon.

Surge arresters

Surge arresters prevent the propagation of lightning along the coaxial cables, which could damage the entire ground station. Each antenna has a surge arrester connected below it, and electrically linked to the pylon, also connected to earthing wires that grounded.

Figure 12: Surge arrestor

MD-03 controller

An antenna rotor controller is responsible for controlling azimuth and elevation rotors. This controller, located in the electronic cabinet is of type SPID MD-03 requests voltage of 15 V DC with 2 RS232 serial ports, 2 USB serial ports. It has a simple graphical user controller interface that allows manual operations. The controller’s power supply (220 V→15 V) is plugged to the power bar. The control is configured into the ground station software (Docker) where it interacts with the GPredict application and SATNOGS during satellite tracking and antenna positioning.

Figure 13: Rotor controller

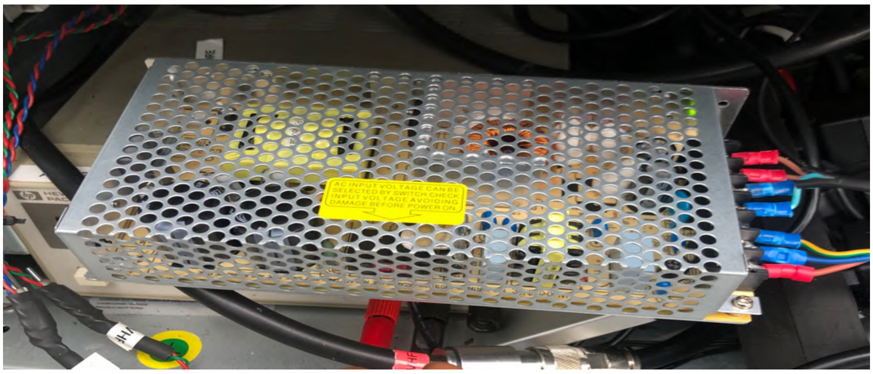


Figure 14: Power supply for Controller

Diplexer

A diplexer allows the UHF and VHF coaxial cables to share a common communications channel. It combines the two to a common output leading to the SDR [*see Fig 14 below*]. The different frequencies on the two communication channels coexist without interfering with each other. The diplexer allows the simultaneous transmission and reception of the RF signals via the same antennas (bi-directional device). Different signals can travel the diplexer with minimum interference.

Figure 15: Diplexer

Bias-Tee

A **bias tee** is a three-port network device used to insert DC power into each antenna. In the electronic cabinet, each antenna is connected to a Bias Tee to the antenna port (”ANT” end). The second port of the Bias Tee (“12 V” end) is connected to the power (DC). The third (“TRCV”) is connected to a manual switch (see below) and finally to the SDR. The FSP 435 12V model was chosen for compatibility with the MVV-VOX pre-amplifier.

Figure 16: Bias - Tee

Manual switches

The manual switches select the antenna mode configuration for either transmission, reception or both. This is possible because they have the ability to block one band and only operate in the other. The manual switches link the Bias-Tees to the Diplexer. Diamond Antenna CX-210N Heavy Duty Wideband Coaxial switches are used to provide very low loss and high isolation. They are rated DC up to 3 GHz including HF frequencies.

Figure 17: Manual switch

The Transceiver

The Transceiver role is to transmit and receive signals from and to the antennas. The transceiver is of type Kenwood-TS-2000X. It modulates the signal commanded by the computer for transmission through the antennas. The Kenwood contains all mode operation from HF to 1.2 GHz amateur radio band with dual high speed Digital Signal Processing (DSP) units and an independent sub-receiver for the 144 MHz and 430 MHz bands to allows the radio to receive/send on both bands simultaneously. The transceiver will be used for transmission of telecommands to satellites in the CENSUS mode. The Kenwood Transceiver is connected to an SDR (shown below) that enables it to transmit and receive data like a network device with digital capabilities and then to SSOL computer all located in the electronic cabinet.

Figure 19: Transceiver

Figure 18: Transceiver Power Supply

The SDR

Software-Defined Radio (SDR) is a radio communication device allowing the modulation (in transmission) or demodulation (in reception) of a radio signal. The SDR plays the role of analog/digital converter in reception. For this connection, SDR Airspy R2 is used in SATNOGS mode because it offers a continuous reception range from 24 to 1800 MHz, making it suitable for analysis and reception of a large number of VHF and UHF signals. It is also compatible with all open source solutions based on GNU Radio. The SDR connects the antennas to SSOL computer through the diplexer thus transforming the incoming RF signals to electric signals for reception (and electric signals to RF signals for transmission purposes, but not used for transmission in SATNOGS mode).

Figure 20: SDR device

Ground Station Computer

The ground station computer, SSOL, coordinates the communication between a satellite and all software and hardware equipment. Its responsibilities include satellite tracking hardware control, data processing, and network operations, enabling the station to communicate with satellites, demodulates the received data or command the transceiver.

Figure 21: Ground Station Computer

Ground station Power bar – This device is used to power devices in the electronic cabinet remotely. It consists of numerous ports that can house many devices. In the current electronic cabinet connection, it powers the GS computer, MD-03 Controller, the Kenwood Transceiver and the power source amplifier for complete remote control of the ground station. The settings of the power bar and its connection are done using Telnet commands. It is remotely accessed to manage power supply of a particular device or reboot any powered device and can be managed using a web server.

Figure 22: Power bar in electronic cabinet

All electronic devices in the electronic cabinet require to plugged in a source of power. The table below describes the respective power supplies for all the devices

Device	**Power supply**
GS Computer	a.c supply—12V/5A inlet
Controller	a.c supply—12V/5A inlet
Transceiver	SPS-250-II power supply
Bias – Tees	d.c supply—12V inlet

Network Switch

The network switch connected the GS computer to the Observatory Network (LAN), operates at the data link layer directing data to all connected devices based on their MAC addresses.

Figure 23: Network Switch

III.2. - Software specifications

The software shall facilitate the ground station in two different modes namely SATNOGS mode and CENSUS mode.

III.2.a. General specifications of the software system

The user can manually connect to SSOL via the Observatory network using ssh connection, or via the use of remote desktop software NoMachine (an alternative of TightVNC) while connected to the Paris Observatory network. Remote access leads a user to two different computers as defined below.

Computers	Identifier	Location
SSOL	*cceres-sol*	Roof Top of Building 7
Radio room	*cceres-station*	Building 15

Below are the login credentials that provide access for different parts of the software system:

Access name	Login name	Password
Ground station Computer	*cceres*	cceres2019
Radio room	*cceres*	cceres2019
SATNOGS network database	*cceres_team*	Cceres2019
Ground Station – Power Bar	*cceres*	cceres2019

The software system shall operate using a virtual environment containerized application software called **Docker.** Docker shall run on Linux DEBIAN operating system for the SSOL computer. Docker shall allow the operation of CENSUS and SATNOGS modes simultaneously and can be switched upon request of the user using an interactive user interface on the SSOL computer. (*see User Manual in section IV.1*).

III.2.b. SATNOGS MODE

SatNOGS (Satellite Networked Open Ground Station) is an open-source, global network of ground stations designed primarily for receiving, monitoring, and decoding signals from satellites, particularly low-Earth orbit (LEO) satellites such as CubeSats and amateur radio satellites. The project is developed and maintained by the Libre Space Foundation (LSF) as part of the broader open-source space ecosystem.

The SATNOGS mode of the SSOL shall exclusively be used for reception purposes on the SATNOGS global network. Operations in this mode require the following concepts :

SatNOGS Network
This web-based coordination platform is an “orchestrator” : it schedules observations across a network of ground stations. Users can schedule observations either on their own ground station or on other participants' stations. All observation results are automatically uploaded to the platform for access and sharing.

Schedule and upload of results of SSOL observations is made on the SSOL SatNOGS network page accessible using the above login credentials on the website : *https://network.satnogs.org/*

SatNOGS Database
Public repository that stores and organizes decoded satellite transmissions. It allows users to access, analyze, and share received data, supporting research, education, and satellite mission operations.

SatNOGS Ground Station
Physical hardware setup that receives satellite signals. For SSOL it includes :

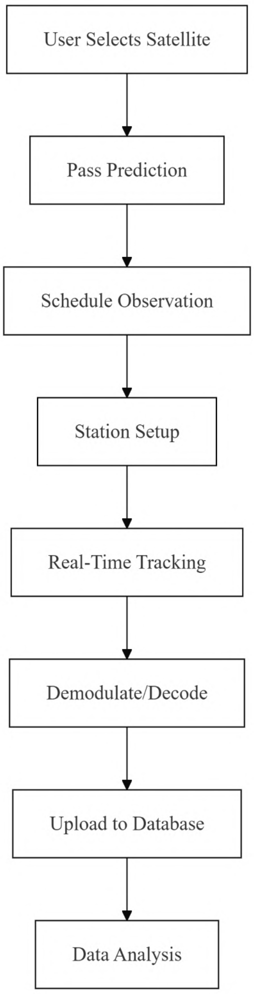
- Two antennas, one in VHF and one in UHF
- coaxial cables
- Low Noise Amplifier (LNA) to amplify UHF signals
- Controller + rotators for tracking satellites
- Software-defined radio (SDR) for signal reception
- Computing device running the SatNOGS client software to control operations.

For SSOL, see hardware specifications above

SatNOGS Client
Software component that operates on the ground station's local hardware. It handles real-time tasks such as satellite tracking with the antenna, signal reception, and command transmission. The client interfaces with the SatNOGS Network to receive observation schedules, then controls the SDR (Software-Defined Radio) and rotator to capture satellite signals during scheduled passes. Once data is received, the client processes and uploads it to the SatNOGS Database for public access. Built as an open-source application, it ensures compatibility with various hardware configurations while maintaining automation and reliability

For SSOL, the SatNOGS client is running in a container named cceres_*satnogs* host on the GS computer

SatNOGS Observation Workflow :



1. Satellite Selection & Pass Prediction
The operator selects a satellite via the SatNOGS Network interface. The system computes pass times using SGP4 propagation with TLEs, determining visibility windows for the ground station.

2. Automated Scheduling
The observation is scheduled either manually or via the network's collision-avoidance algorithm, assigning it to the station's SatNOGS Client.

3. Pre-Pass Hardware Configuration
The client configures the SDR to the satellite's frequency (e.g., 437 MHz UHF) and modulation method (e.g., GFSK). The rotator aligns the antenna to the initial azimuth/elevation.

4. Signal Acquisition & Tracking
During the pass, the client adjusts the rotator via PID control to maintain pointing accuracy while the SDR streams IQ data. Signal lock is verified via SNR thresholds.

5. Decoding & Data Processing
The signal is demodulated and decoded if a decoded protocol is provided

6. Data Submission & Crowdsourcing
Decoded and raw telemetry is uploaded to the SatNOGS DB via API. The network aggregates data from multiple stations for redundancy.

Installation of the software architecture in SatNOGS_mode :

The software architecture of the SSOL can be installed by cloning the project repository hosted on GitLab:

https://gitlab.obspm.fr/cceres/ground_station/satnogs_mode

Access to the repository requires the user to be part of the GitLab projet. Please contact your supervisor to have access to the project

The entire installation process is based on an automated script that leverages Docker and Ansible to install the client. All Docker images required during installation are stored locally in the container registry of the GitLab project (navigate to **Deploy → Container Registry → satnogs_mode**).

The Docker images used during the installation process are organized into two main categories:

1 - Custom images, entirely developed in-house and specifically tailored to our ground station and its Docker-based implementation. These images are essential and will be used regardless of the installation options selected.

Name	Author	Description	Docker file
installer_container	CENSUS	Image used for remote installation of the SatNOGS client	In the GitLab project (installer → main)
target_container	CENSUS	Image used to setup an environment ready to support the SatNOGS client	In the GitLab project (installer → target)

2 - Generic images, provided by the broader SatNOGS community. These images have not been modified for our specific use case but are necessary for the general operation of the SatNOGS client. Their usage may vary depending on the selected installation configuration.

Name (of the image)	Author	Description	Docker file
ansible	LSF	Ansible Docker image used for provisioning Libre Space Foundation infrastructure	https://gitlab.com/librespacefoundation/ops/docker-ansible
hamlib	LSF	Hamlib Docker image that includes Hamlib Amateur Radio Equipment Control libraries and utilities.	https://gitlab.com/librespacefoundation/docker-hamlib
satnogs-client	LSF	SatNOGS Ground Station Client software. This package automates the reception of signals from satellites for the satnogs.org network	https://gitlab.com/librespacefoundation/satnogs/satnogs-client
satnogs-config	LSF	SatNOGS client system configuration utility.	https://gitlab.com/librespacefoundation/satnogs/satnogs-config

The retrieval and deployment of these images are fully automated by the installation script, ensuring a seamless and reproducible setup process.

It is important to note that all Docker images (custom or generic) are stored internally within a private GitLab project. As a result, any updates made by the Libre Space Foundation to their official images will **not** be automatically reflected in the operation of the ground station.

If such updates are to be tested and integrated, they must be manually fetched and pushed to the project's container registry. This ensures that the system remains stable and changes are explicitly controlled.

The configuration script used during the installation offers three modes of operation:

1 – Full installation of the software architecture from scratch
This option allows the user to perform a complete remote installation of the software architecture from the Main computer. Once the SatNOGS client is installed, an interactive menu will assist the user in configuring the ground station, including hardware specifications, station location, gain settings, and other relevant parameters. (see *User Manual, procedure INS 4.0 + INS 4.1).*

2 – Reinstallation of the latest stable and working configuration
This option enables the user to reinstall the most recent stable configuration of the ground station, previously saved in the GitLab project. It is particularly useful for reverting to a functional setup after conducting tests or experimenting with new configurations. (see *User Manual, procedure INS 4.0 + INS 4.2).*

3 – Save the current station configuration as the new stable version
This option is available only if a SatNOGS client is already running on the ground station. It allows the user to save the current parameters and configuration as the latest known stable version in the GitLab project (as a new commit to the project). This saved configuration will then be used by default if option 2 is selected in future installations. (see *User Manual, procedure INS 4.0 + INS 4.3)*
In the GitLab project, the user can access the latest stable configuration parameters at :
/satnogs_mode/stable-config/stable_config.txt

If the user wishes to consult a previously saved stable configuration, it is available in the GitLab project under:
/satnogs_mode/stable-config/past/

Additionnal features :
- for every configuration run by the user, a log file is created and is accessible at :
  /satnogs_mode/logs/deploy_{..}.log
- the configuration script can be launch with options :
  - -- help : show the help menu
  - --clean-param : reset ssh connections parameters
  - --clean-logs : clean old logs
  - --clean : clean old logs and reset ss connections parameters

Architecture of the GitLab projet

```
SATNOGS_MODE/
├── docs/                        # Project documentation
│   ├── ARCHITECTURE.md          # Architecture overview
│   └── INSTALL.md               # Detailed installation procedure
├── installer/                   # SSOL installation resources
│   ├── main/                    # Ressources for the installer Docker container
│   │   ├── satnogs-ansible/     # Ansible playbooks
│   │   ├── ssh_keys/            # SSH keys for connecting to the GS computer
│   │   ├── docker-compose.yml   # Installer container Docker Compose file
│   │   ├── Dockerfile           # Installer Docker image
│   │   └── entrypoint.sh        # Main entrypoint script
│   ├── target/                  # Ressources for the target Docker container
│   │   ├── docker-compose.yml   # Target Docker Compose file
│   │   └── Dockerfile           # Target Docker image
│   └── variables/               # Variables for ssh connection
│       └── .env                 # Variables for ssh connection
├── logs/                        # Execution and installation logs
├── scripts/                     # Scripts
│   └── deploy-full.sh           # Main automated deployment script
├── stable-config/               # Stable configurations
│   ├── past/                    # History of stable configurations
│   └── stable_config.txt        # Last stable configuration
├── .gitignore                   # Files and folders ignored by Git
└── README.md                    # General project overview
```

Operations in Satnogs_mode :

Once the client is properly installed and configured, all operations are carried out through the SatNOGS Network web interface. From there, the user can monitor the ground station’s status, schedule observation passes, and analyze the results.

Additionally, the user can access the SatNOGS client logs by following the procedure : *MANUAL 9.0*

Auto scheduling

The Two Line Element (TLE) on the SATNOGS network regularly updates which affects the forecast passes of satellite and hence the status of observations, leading to failure if old TLE information of the satellite is used. However, it is not a good idea to schedule passes for more than two days in advance and is also a tedious process to keep scheduling observations every other day.

To avoid manual scheduling of satellite on the SATNOGS Network, a particular code named "Auto-scheduler" is deployed to prioritize particular satellites to be tracked. This code automates satellite selection, observation scheduling and planning dynamically real time based on updated TLE.

III.2.c. CENSUS MODE

For the CENSUS mode, the GPredict + GQRX + GNURadio pipeline is used for operation of the ground system software, in order to receive from and transmit to partners' satellites.

GPredict

GPredict is a real-time tracking software for satellites in orbit that provides efficient and detailed predictions of future satellite passes based on the satellite's TLE and hence orbital information [4]. The ground track of the satellite is presented on a map with several tables and graphs. The intended satellite to be observed is selected by the user and GPredict performs orbital calculations to provide the real-time satellite passes over the ground station location and also performs a Doppler shift correction to aid demodulation by the SDR.

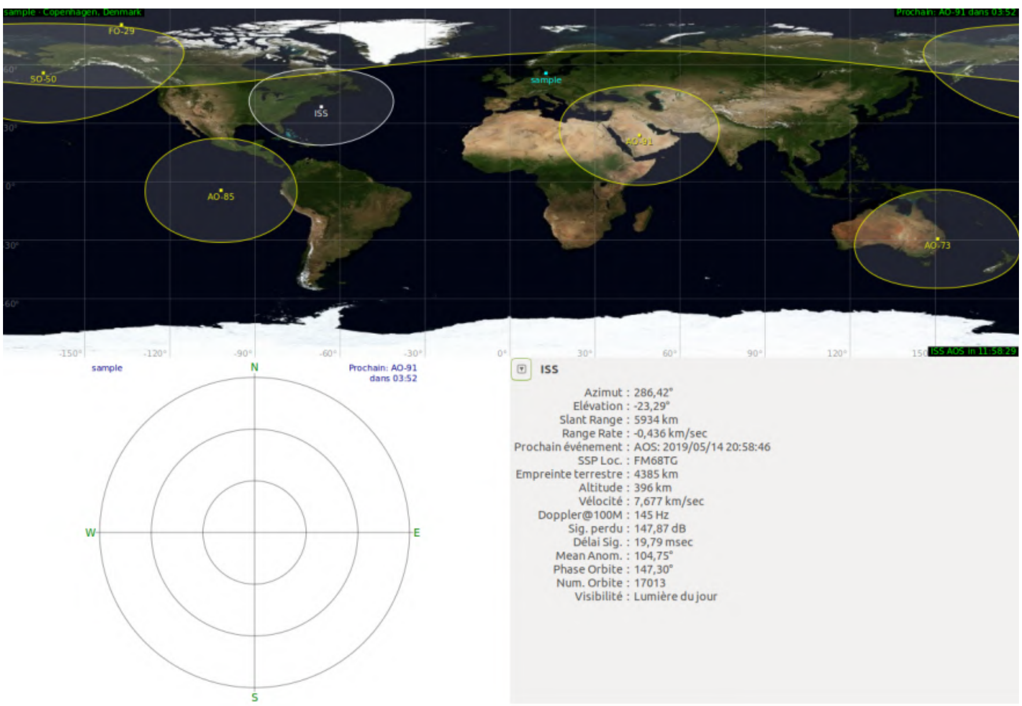


Figure 24: GPredict software interface

GQRX

GQRX is an SDR software that allows frequency, gain and I/Q balance (polarization) adjustment [5]. It provides signal display through spectrum, waterfall and audio outputs. GPredict software is linked to the GQRX software for frequency correction (Doppler effect) of the incoming signals from the antenna or the telecommands to be sent to the satellite. The displayed signals are then transferred to another software, GNURadio, for further processing.

GNURadio

GNURadio is a software for digital signal processing that performs filtering, correction, analysis and detection of the received signal [6]. GNURadio contains easy-to-use reusable blocks in a GUI, offers scalability and provides an extensive library of standard algorithms. Since GNURadio is connected to GQRX, a user can combine existing blocks into a high-level flow graph in order to receive digitally modulated signals from the SDR for decoding and GNURadio will automatically process the data.

PicTalk software

PicTalk software is provided as a Virtual Machine (VM) that can be opened via VMWare or any virtual box manager. This is a combined "all in one" software (GPredict, GQRX, GNURadio) that enables reception, decoding of satellite telemetry and scientific data. PicTalk is highly adaptable because of its ease of use, but is not suitable for all satellites. It supports both real-time interactions and automated scheduling for satellite operations. PicTalk software shall be used as alternative to the three software combination pipeline described above for reception of signals. It possesses a graphical interface that allows its users to interact with the satellite systems, send commands, and view telemetry data during reception. Commands input by the user are checked for validity and scheduled for transmission to the satellite.

PicSat ground segment

This is a specialized software that was built specifically to send telecommands to the PicSat satellite. It was configured to communicate with the on-board computer of the satellite that could interpret the commands received. It possesses a window that displays the sending process of telecommands, acknowledge of receipt by the satellite and finally confirmation of task execution. It makes use of an encryption key that protects it from any eavesdroppers to the satellite and unwanted noise. The satellite receiving the telecommands is recognized by its identification number that was already per-configured in the PicSAT software.

IV. - User manual

All the operations that involve the software requires a stable internet connection for successful execution. Users shall choose either of the modes using a graphical interface on the SSOL computer.

The following procedure must be followed at all times for all users accessing the SSOL software:

Procedure 1.0: **Logging into the SSOL software system – Command-line interface**		
No.	**Procedure purpose**	**Ref**
1,1	In case you are on-site at the Paris Observatory in Meudon, connect your computer device to the Observatory Network LAN through use of an Ethernet cable. *The icon for Internet settings at the bottom task bar menu should reflect this addition*.	
1,2	In case you are off-site (remote), connect your computer device to the Observatory Network VPN given to you by the Network administrator. *Contact the Network for any details.*	
2	In your command-line interface, use ssh to connect to the ground-station computer Type `ssh cceres@cceres-sol.obspm.fr`	
3	Enter the password for ***cceres-sol*** as defined above in Section III.2a	Section III.2a
4	A successful login should bring you this below: `cceres@cceres-sol ~ %`	
5	You can now type any commands of your choice	

IV.1. - SATNOGS MODE

SATNOGS network is designed as an open source participatory project based on the users operating a ground station and is accessed via a web page for all of the network users. A user needs to login to the SATNOGS network using SSOL ground station credentials. (See Section III.2a) SSOL is named **CCERES ground station** with an ID of **1062** in the SATNOGS network. A successful login on your dashboard shall appear as in **Figure 23** below.

The dashboard displays the location details of the ground station (coordinates, altitude above ground level), QTH locator, operation mode (***Offline, Testing or Online***), total number of scheduled observations by the ground station since its creation, scheduled future observations, status logs and the types of antennas used for the ground station. The ground station shall remain in **Offline** mode (*a state where the ground station is not noticeable by the SATNOGS database*) or **Testing** mode (available for planning but not visible) whenever the station is not available due to technical reasons or priority to CENSUS mode. Otherwise, it shall switch to **Online** mode (*a state where the ground station can fully carry out its operations*) when all testing and verification of the entire system has been successfully completed.

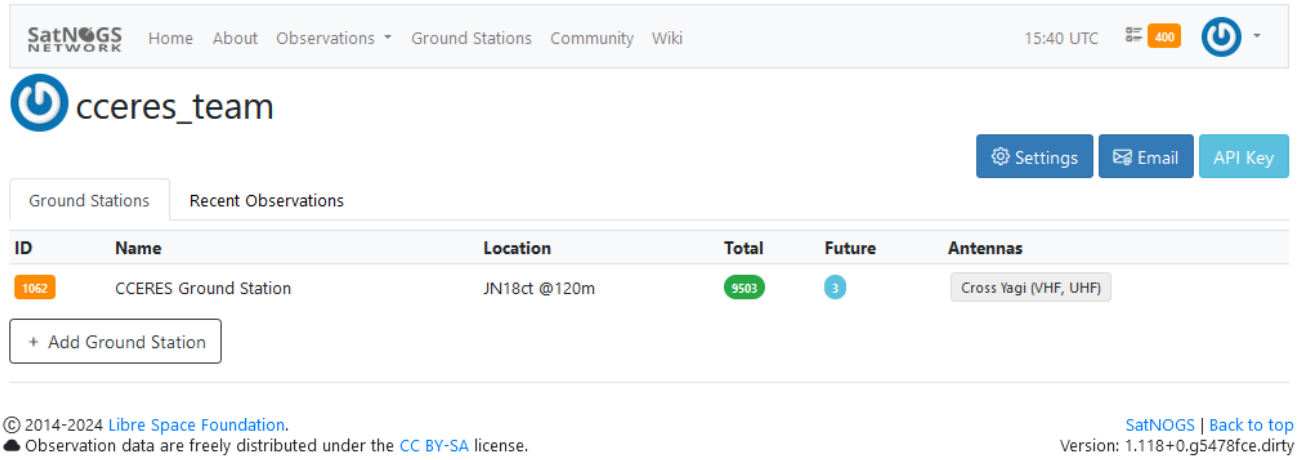


Figure 25: SATNOGS dashboard for SSOL

Below is a list of the procedures for the use of SATNOGS mode. This is done using the SATNOGS website interface.

Manual 2.0: **SATNOGS mode Operations – Website interface**		
No.	**Procedure purpose**	
1	After the login is completed as shown above, on the same dashboard, click “Observations” option and thereafter “Schedule” option.	
2,1	Select a satellite that you intend to observe by typing its name or ID in the search bar.	
2,2	This satellite must be observable by the CCERES ground station. Confirmation is done by the Station ID 1062 showing up among the available stations.	
3,1	Adjust the “Start Time” and “End Time” to your desired time duration for observations. Click “Calculate”.	
3,2	If they are no available passes for this duration, adjust the time duration until you find suitable satellite passes.	
4	Select your best suited passes and click “Schedule”.	
5	After the observation time has expired, click “Observations” option and thereafter click the “Browse” option.	
6	Select CCERES ground station as the ground station.	

7,1	All the results from the scheduled observations shall be returned for your analysis. Click on each result for your analysis.	
7,2	The observation result may be of a particular status as explained below.	
8	The display results in different forms such as audio files, hexadecimal text files or images as transmitted by the satellites.	
9	The observation result has a digits-identifier associated with an URL for future reference and also gives more information about the satellite pass. (*See Figure 24 below)*	

- The status of an observation is given by its color,
  - **Green** for successfully processed observations by the SATNOGS network, a signal was received and demodulated. However this doesn’t confirm successful tracking of the satellite by the antennas nor does it detail the content of demodulation to be accurate.
  - **Black** for observations that failed due to a malfunction of the ground station hardware
  - **Orange** for observations awaiting examination or whose signals have been partially demodulated.
  - **Blue** for future (planned) observations.
  - **Red** color corresponds to a “bad” observation on which no signal is visible and demodulation could not be performed for that signal.

For every satellite pass over the SSOL ground station that is scheduled on the SATNOGS network, the path of satellite is displayed on a polar plot as in Figure 26, highlighting the starting point and its angle and as well the end point for the satellite pass duration. [236 degrees to 72 degrees] for the satellite path shown. The antennas then have to adjust their orientation along this path in order to download the satellite signal.

The results displayed after an observation are in form of a waterfall accompanied with a data audio file containing the necessary frequencies as transmitted by the satellite. A successful download of a satellite signal displays images or coded written instructions in Hexadecimal format that can be translated to into ordinary alphabetical or numerical text. The waterfall as shown below in Figure 26 shows spikes representing the detected signal. A blank waterfall indicates no demodulation of the signal. The waterfall is displayed according to the power of the received signal.

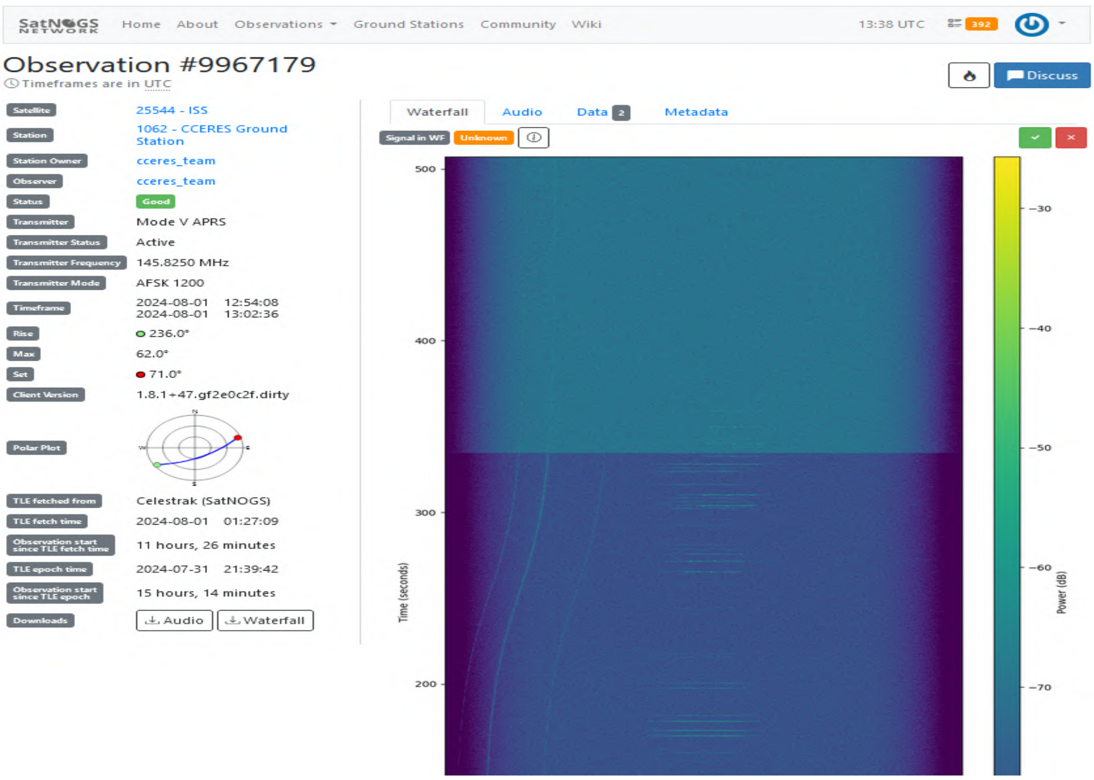


Figure 26: ISS successful observation

In order to access the Auto-scheduler option of scheduling observations for the ground station, below is the procedure that must be followed.

Manual 3.0: Auto Scheduler Operations – Command Line interface		
No.	**Procedure purpose**	**Output**
1	Open the command-line interface and type, `ssh cceres@cceres-sol.obspm.fr` Log in the Observatory Network. (*See section III.2a for login details*)	
2	Access the ground station directory, `cd station-1062`	
3,1	Pull the SATNOGS Auto-Scheduler Docker image: `docker pull registry.gitlab.com/librespacefoundation/satnogs/satnogs-auto-scheduler/satnogs-auto-scheduler:master`	
3,2	Run Auto scheduler container (as defined in III.2b) consisting this image, passing in required environment variables.	

	`docker-compose up -d`	
4,1	Run the following command to schedule observations: Specify your intended duration and any extra options. `python schedule_single_station.py -s <1062> -d <DURATION> -r <0> <EXTRA_OPTIONS>`	

In order to control the power supply remotely using the power bar, below is the procedure that must be followed.

Manual 4.0: **Power bar Manual – Website interface**		
No.	**Procedure purpose**	
1	In your browser and type, `http://cceres-sol-alim.obspm.fr` Log in the SSOL account on the Schneider Electric Web Application. (*See section III.2a for login details*)	Section III.2a
2	Check the information on the dashboard (PDU section) corresponds to the SSOL ground station. The Up time should correlate with time the station has been in operation. In case of any errors, edit settings using existing documentation.	
3	Check for any information that might need attention in the “*Active Alarms*” and “*Recent Device Events*” on the Home dashboard. In case of an immediate alarm or fault, troubleshoot using the Maintenance Procedure in Section V.	Section V.1. Page 48
4	To find out the number of power outlets the power bar has, click “*Control*”, followed by “*RFDU*” and then “*Outlet*”. You will see a list of outlets and their corresponding names.	
5	All the outlet power states should be “*On*” for normal working conditions. “*” Indicates a pending state change.	
6,1	You can change the state of an outlet, Select the specific outlet, select the control action to be performed and then Click “*Next*”.	
6,2	You shall receive a confirmation alert indicating and validating the details of your change request. Click “*Apply*” to save changes	
7	Go back to the Outlet Control dashboard. The state of that outlet shall reflect as changed.	
8	To check for the Power bar logs, Click “*Logs*” on the Home dashboard and explore the different options present.	

IV.2. - CENSUS MODE

This mode can only be accessed by authorized personnel connected to the Observatory network to listen to and send instructions to particular satellites deployed by the Observatory for scientific missions.

In the CENSUS mode, the user has to separately operate the three software in the **GPredict+GQRX+GNURadio combination** or PicTalk software.

Below are the procedure manuals of the CENSUS MODE for an authorized user from the initial request of a scheduled observation to analysis of the data results obtained for interpretation.

Manual 5.0: GPredict Operation – Website Interface		
No.	**Steps to follow**	
1	Open the GPredict software on your computer	
2	Go to the option "*File*", add a new module and name it according to your project.	
3	Select satellites to track, click "*add*" and then "*ok*".	
4,1	Once the module is created, you can monitor the selected satellites based on their information e.g azimuth & elevation, passes and signal information.	
4.2	To track a satellite pass, select your module and observe the satellites orbiting in real time.	
5	Go to Pass Predictions, Click the "*Predict*" button to find out the time and duration of the satellite pass over the ground station.	
6	Click "*Start Tracking*" when the satellite is in view to follow its live path.	
7	To predict future passes, Click the "*Predict*" button and choose a satellite with a future time range.	
9	In "*Preferences*", you can also add and configure your rotor (ROTCTL).	
10	After tracking, export your satellite pass predictions to a file and save it.	
11	Periodically update your TLE data to ensure you're tracking satellites accurately.	

Manual 6.0: GQRX Operation – Website Interface

Prerequisite

- Airspy (SDR) documentation manual

No.	Steps to follow	
1	Open the GQRX software on your computer.	
2	In the “*Input Configuration window*”, select your SDR device. (*See Section III.1 and Figure 18 for details*)	
3	Set device-specific settings e.g sample rate according to the SDR documentation.	
4,1	Configure the audio back end and output device to the system’s default audio output.	
4,2	Use the volume slider to control the audio output level.	
5	Set the RF Gain slider based on the signal strength.	
6	Configure the frequency, modulation mode and bandwidth for the signals to be detected.	
7,1	Click the "*Play*" button to start receiving signals. The spectrum display will show real-time frequency signals in a graphical view.	
7,2	Adjust the waterfall and spectrum display optimally according to your preference.	
7,3	You should hear the signal through your speakers or headphones.	
8	Save received data and the configuration used for future sessions.	
9	Tune to a known frequency (e.g., a local FM station) to test that everything is working correctly.	
10	If you're receiving weak signals, try optimizing gain, frequency and bandwidth for better signal clarity. [*Step 6*]	

Manual 7.0: **Gnu radio Operation – Website interface**		
No.	**Steps to follow**	
1	Open the Gnu radio software on your computer.	
2,1	In the Gnu radio application, create a simple flow graph in the software.	
2,2	To achieve this, on the workspace follow through these steps below; • Add a signal source and set the parameters of the desired signal to be demodulated. • Add an audio sink to output the signal to your computer sound card so you can hear it. • Connect the output of the signal source to the input of the audio sink block. • Save and run your flow graph. You should now hear the output signal through your speakers or headphones.	
5	You can always add various blocks to the flow graph for more complex signal processing tasks e.g filters, modulators, demodulators.	
6	Once the flow graph is created, it can be run in real time by clicking the "**Play**" button. You can hence process live signals.	
7	Adjust any signal parameters according to the SDR to observe any real-time changes in the output. Ensure the UDP port number is correct [7352]. It can be changed by double clicking on the UDP source box.	
8	You can save your flow graph as `a.grc` file or generate python code for future use.	
9	If something doesn't work as expected or in case of poor quality results, troubleshoot and optimize through block parameters to ensure they are correctly set and compatible.	
10	Once you are done, you can close the GNU Radio Companion.	

- It is imperative that the port from which the signal is received by GNU Radio corresponds to the GQRX output port.
- After the flow graph has been generated on GNU Radio, a terminal must then opened for proper display of the decoded frames if the signal received is sufficiently clear and powerful.

IV.3. - Manual hardware operations

Manual 8.0: **Camera (Live feed) Operation**

Prerequisites

- GS computer or remote desktop.
- Camera

No.	Procedure purpose	Ref
1	At the ground station, Plug the camera cable in to the GS computer and the Ethernet cable to the camera to connect to stable internet.	
2	Using a computer, Run Manual 1.0 to log into the Observatory network.	Section III.2a
3	Go to your browser and enter this link “`http://ssol0.obspm.fr/`	
4	You will access a live feed of the ground station.	
5	Click the camera picture to adjust the feed appearance. Use the buttons below the picture and on the right for optmisation.	
6	Save your intended files to a folder of your choice.	

IV.4. - Manual SatNOGS operations

Manual 9.0: **Display the logs of the SatNOGS client**

Prerequisites

- Having a SatNOGS client running on your GS computer (procedures INS 4.1 + INS 4.2 or INS 4.3)
- Bash terminal

Duration : 5 minutes

No.	Procedure purpose	Ref
1	Open a Bash terminal on your Main computer	
2	Enter the command : `% ssh -t cceres@145.238.0.229 'docker exec -it cceres_satnogs bash'`	
3	You will see the message : " `cceres@145.238.0.229's password:` " Use the credetials above and enter the password provided for cceres user	
4	You will then have access to the CLI of the cceres_satnogs container	
5	To view the SatNOGS client logs in real time, use the following command : `% docker logs -f satnogs_satnogs-client`	
6	You are now able to see the logs of you client	

Note :

The richness and verbosity of the logs depend on the station's configuration settings.

V. - Maintenance

V.1. - Scheduled Maintenance

The maintenance of SSOL ground station is a critical recurring process that is performed on a scheduled basis or in case of an emergency. The scheduled maintenance shall take place **every 3 months** and shall be carried by one or two individuals (as mentioned for each procedure), including supervision of trained staff of the Paris Observatory. The past maintenance are reported in the maintenance log book in appendix.

WARNING: All these maintenance procedures should be carried taking into consideration of the solar observation processes that take place at the ground level of Building 7 (just adjacent to the ground station) to avoid interference of their operations and safeguard the human body from damage.

Procedure (ID)	Description
M1.1	Un-mounting of the antenna spire
M1.2	Re-mounting of the antenna
M2.1	Azimuth Calibration
M2.2	Elevation Calibration
M3	Routine hardware check
M4	Power Bar functionalities

Procedures M1.1 and M1.2 are carried out during the scheduled maintenance plans, upgrade or replacement of external hardware components and during a trouble shooting session in case of failures.

Procedures M2.1 and M2.2 are carried out during the scheduled maintenance plans and re-installation of the hardware system.

Procedure M3 is carried out during the scheduled maintenance plans and during a trouble shooting session in case of hardware system failure.

Procedure M4 is carried out during the scheduled maintenance plans and during a trouble shooting session in case of power bar software system failure.

Procedure M1.1: **Un-mounting of the antenna spire**

Prerequisites

- Two individuals.
- Screw keys with a large variety of different radii. [9mm, 18mm, 19mm]
- Heavy duty gloves, safety boots, protective clothing, helmet, sunglasses
- Small storage container for the screws.
- Two-way communication devices e.g talk-walkies

Duration: 12 minutes

No.	Steps to follow	Tooling
1	At the ground station antenna area, check that the area is clear of any unnecessary objects on the roof or on the ground floor, that could injure, damage or be damaged before working.	
2	Organize all the screw keys needed for this maintenance procedure.	
3	Ensure the pylon spire has nothing hindering smooth movement of the pulley.	
4	With **1** operator on the roof and **1** on the ground, the ground operator must turn the pulley anticlockwise to move the pylon spire downwards until it gets into contact with the white square mounting base.	Helmet
5	Roof operator to adjust the position of the pulley and the triangular base, while ground operator is pulling down, in order to ensure proper alignment of the holes and the screws during the pulley movement.	
6	Ensure safety of any persons during movement of the pulley. Each person but the operators must be at least 1m away from the pulley or a mounting system during works.	
7	Using a screw key of radius 19mm, loosen one of the three storage screws on the triangular base of the spire. Store the free unused screw components carefully.	Small container
8	Using the same screw key of radius 19mm, tighten one of the three attachment screws between the triangular base and the movable white square mounting base.	
9	Repeat steps 7 and 8 successively for the remaining two screws of each type one pair combination at a time.	
10	Using a hanger screw key, loosen the two white alignment screws on the sides of	Small

	the white square mounting base. Store these screws carefully.	container
11	The antenna spire structure is then ready for moving along the sliding platform. It should be mobile.	
12	Move the antenna spire structure to the extreme left towards the electronic cabinet.	
13	The antenna is now available for maintenance.	

NB: During the tightening and loosening of the screws in Steps 7-9, the pylon spire should never be left unattached at any moment for safety reasons.

Procedure M1.2: **Re-mounting of the antenna**

Prerequisites

- Two individuals.
- Screw keys with a large variety of different radii. [9mm, 18mm, 19mm]
- Heavy duty gloves, safety boots, protective clothing, helmet, sunglasses
- Small storage container for the screws.
- Two-way communication devices e.g talk-walkies

Duration: 12 minutes

No.	Steps to follow	Tooling
1	At the ground station antenna area, check that the area is clear of any unnecessary or dangerous objects before working.	
2	Organize all the screws needed for this maintenance procedure.	Small container
3	Move the antenna–pylon structure to the extreme right away from the electronic cabinet.	Helmet
4	Ensure the pylon spire has nothing hindering smooth movement of the pulley.	
5	Verify alignment of the triangular base of the pylon with the white square mounting base. This can be checked by ensuring all the screws fit within their respective holes.	
6	Using a screw key of radius 19mm, loosen one of the three attachment screws between the triangular base of the pylon and the movable white square mounting base. Store the free unused screw components carefully.	Small container
7	Using the same screw key of radius 19mm, tighten one of the three storage screws back in their storage position to connect the pylon spire to the pulley.	
8	Repeat steps 6 and 7 successively for the remaining two screws of each type one pair combination at a time.	
9	With the help of another partner at the ground level, turn the pulley clockwise to move the pylon spire upwards until it gets into contact with the white square mounting base.	Helmet
10	Using a hanger screw key, tighten the two white alignment screws on the sides of	

	the white square mounting base.	
11	The pylon must be free of the white square mounting base.	
12	With the help of another partner at the ground level, they must turn the pulley clockwise to move the pylon spire more upwards to the desired height.	Helmet
13	The antenna–pylon structure must be at a height such that the VHF antenna does not interfere with the rail barrier protecting the electronic cabinet.	
14	Ensure the tools used are back in their initial storage and put away in a safe area.	
15	Ensure that the mounting area is clean and clear of any drainage.	
16	The antennas are now successfully re-mounted.	

NB: During the tightening and loosening of the screws in Steps 6-8, the pylon spire should never be left unattached at any moment for safety reasons.

Procedure M2.1: **Azimuth Calibration**		
Prerequisites • Updated map extraction of the area (Hard copy is preferred) • N-S Hand held compass. • Two individuals • SPID MD-03 controller configured to communicate with the rotors. • Two-way communication devices e.g talk-walkies • Heavy duty gloves, safety boots, protective clothing, helmet, sunglasses • Prerequisites of M1.2 & M1.1 **Duration**: 45 minutes		
No.	**Steps to follow**	**Output**
1	At the ground station, open the electronic cabinet and pull out the black MD-03 controller.	
2,1	Record the Azimuth (A1) and Elevation (E2) values by reading the controller screen.	
2,2	Ensure the Azimuth numerical calibration is the 0-360 degrees range.	
2,3	The acronym for the azimuth value on the screen should be A1 and must not have A+.	
2,4	In case, the A+ is appearing on the screen, Adjust it back to A by continuously pressing the left hand adjustment control button on the controller menu.	
3	Adjust on the controller, the Azimuth value to 90 degrees and the elevation values to zero degrees (0)	
4	Observe the direction the antenna focal plane is facing at this value. Record its value.	[Angle]
4.1	The direction of the focal plane is that which is at the extreme opposite end of the focal plane (black focus feed) on the antenna. (*Refer to Figure 25*)	
5	Run Procedure M1.1 for this step to un-mount the antenna from the pulley structure.	
6,1	Adjust the antennas to face East.	

6,2	Gently unscrew the 8 connections between the azimuth elevator and the vertical pole. Use screw keys of radius 17mm.	
6,3	Manually rotate the vertical pole connecting to the antennas through a quarter turn and ensure the focal plane accurately faces East	
6,4	Measure the length of the coaxial cables connecting the rotors to the electronic cabinet to include a 30% acceptable margin for flexibility of two full turns of the pylon-spire.	
6,5	Identify a landmark /an observation feature in the East as a reference point. Use a map for validation and precise location of this feature.	
6,6	The reference point is the tree as seen in Figure 26 (end of blue line)	
6,7	By the use of a compass and adjust the vertical pole connecting to the antennas to properly face East. Record the angle.	[90]
6,8	Re-screw back the 8 connections tightly using screw keys of radius 17 mm.	
7	Run Procedure M1.2 for this step to mount the antenna back on the pulley structure.	
8	Verify the azimuth value on the controller screen.	[90]
9	Place the black MD-03 controller back in the electronic cabinet and close it.	
10	Ensure the entire set-up looks the same way you found it before you leave. In case this is not so, follow through the procedures again from the top to identify any error made. *(Refer to Figure 25)*	

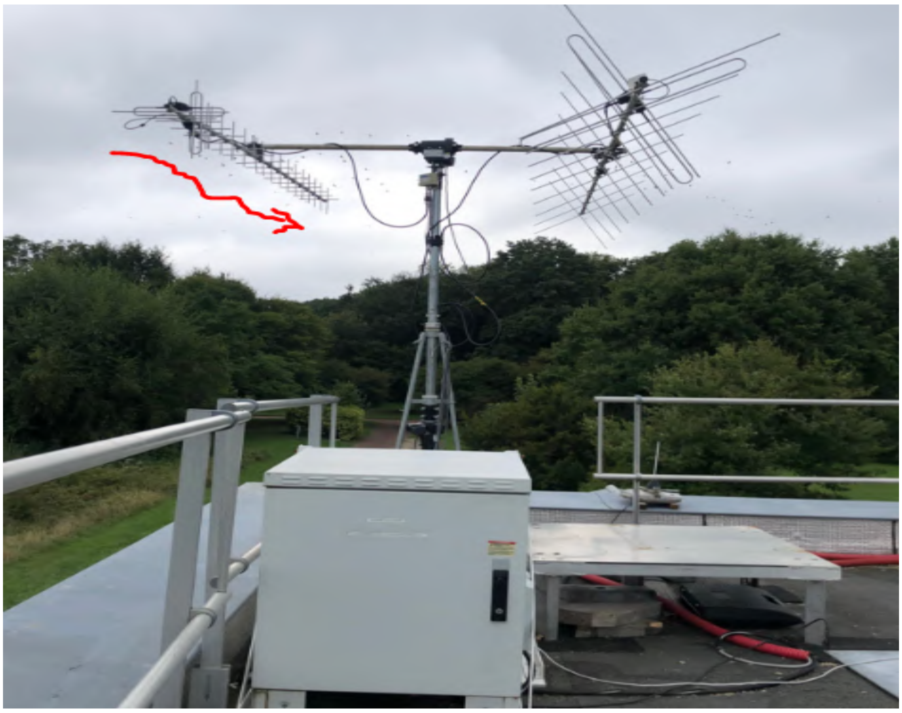


Figure 27: Antennas on pylon-spire + mounting platform. Arrow showing the direction of the focal plane.

Figure 28: Map showing reference point in the East.

Procedure M2.2: **Elevation Calibration**

Prerequisites

- Updated map extraction of the area (Hard copy is preferred)
- N-S Hand held compass (Physical /Online)
- SPID MD-03 controller configured to communicate with the rotors.
- Two-way communication devices e.g talkie-walkies
- Heavy duty gloves, safety boots, protective clothing, helmet.
- Two individuals
- Horizontal Level
- Prerequisites of M1.2 & M1.1

Duration: 45 minutes

No.	Steps to follow	Output
1	At the ground station, open the electronic cabinet and pull out the black MD-03 controller.	
2,1	Record the Azimuth (A1) and Elevation (E2) values by reading the controller screen.	
2,2	Ensure the Elevation numerical calibration is the 0-180 degrees range.	
3	Adjust on the controller, the Azimuth value to thirty degrees (30) and the elevation value to zero degrees (0)	
4	Observe the direction the antenna focal plane is facing at this value. Record its value.	[Angle]
4.1	The direction of the focal plane is that which is at the extreme opposite end of the focal plane (black focus feed) on the antenna. (*Refer to Fig 27)*	
5	Run Procedure M1.1 for this step to un-mount the antenna from the pulley structure.	
6,1	Gently unscrew the 8 connections between the elevation rotor and the horizontal pole connecting the two antennas. Use screw keys of radius 17mm.	
6,2	Manually rotate the horizontal pole to ensure it is horizontally straight (0° elevation). Confirm this with a horizontal level and using the ground as your reference.	
6,3	Identify a landmark /an observation feature in the 30° azimuth and 0° elevation	

	direction as a reference point. Then use a map for validation and precise location of this feature.	
6,4	The reference point is the top of the ground telescope as seen in Figure 29. (Blue circle)	
6,5	By the use of a compass and a horizontal level, adjust the horizontal pole connecting to the antennas to align with the reference point of the land mark. Record the angle.	[0]
6,7	Re-screw back the 8 connections tightly using screw keys of radius 17 mm.	
7	Run Procedure M1.2 for this step to mount the antenna back on the pulley structure.	
8	Verify the elevation value on the controller screen.	[0]
9	Place the black MD-03 controller back in the electronic cabinet and close it.	
10	Ensure the entire set-up looks the same way you found it before you leave. In case this is not so, follow through the procedures again from the top to identify any error made. *(Refer to Figure 25)*	

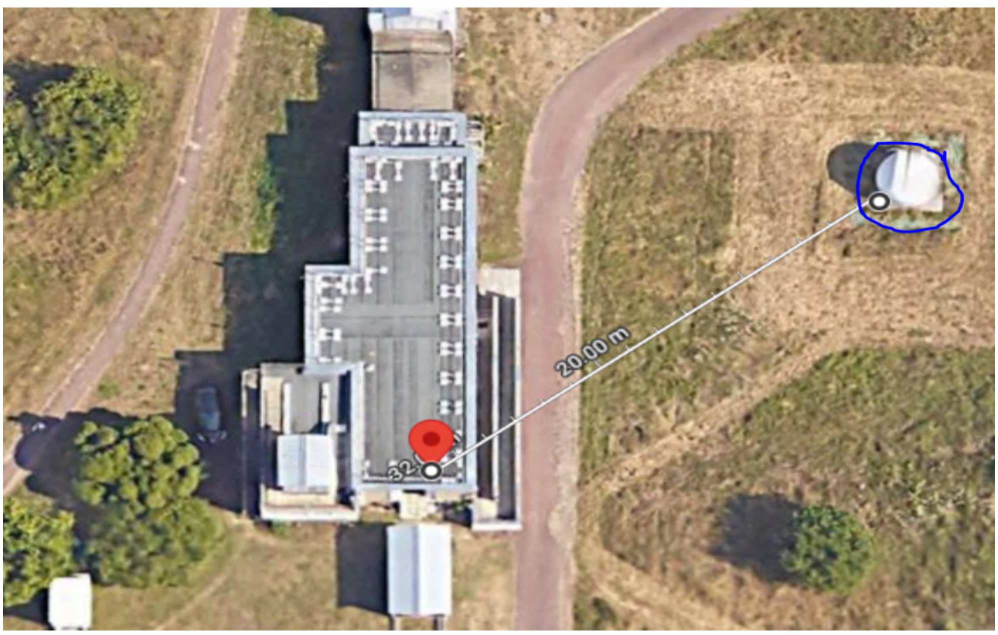


Figure 29: Map showing the antenna and reference point (Blue circle)

Procedure M3: Routine check of the Ground station.

Prerequisites

- Two individuals
- Screw keys with a large variety of different radii. [3 -20mm]
- Heavy duty gloves, safety boots, protective clothing, helmet.
- Anti-corrosion spray
- Multi-meter
- Blow dryer
- Grease/ Lubricant
- Clean piece of cloth / Towel
- Weatherproof tape, Insulating tape
- Screwdriver
- SSOL installed camera at the ground station

Duration: 30 -120 minutes

No.	Steps to follow	Tools
1	At the ground station antenna area, check that the area is clear of any unnecessary or dangerous objects before working.	
2	For the antennas, check for visible damage, such as cracks, bends, or rust. Check for corrosion on metal parts and ensure that the antenna is securely mounted. *Take a picture for record purposes and planning for replacement.*	Camera
3	Wipe down the antenna and mounts to remove dust, bird droppings, or other debris that could interfere with reception.	Towel
4	For the cables and wires, check for any damaged cables and secure connectors. Look for signs of water ingress, especially in outdoor cables. *Cover any open parts with weatherproof tape.* *Take a picture for record purposes and planning for replacement. Replace immediately if damage is of a high degree.*	Camera Tape
5	For the pylon spire, ensure the bolts are securely tightened. Use a screw key of 17mm to tighten loose ends. Check for any corrosion or bending on the pylon. *Take a picture for record purposes and planning for replacement.*	Screw key
6	For the mounting platform, check for any friction in the sliding of the pylon base.	Lubricant

	Apply some lubricants on the contact parts.	
7,1	Inspect the surge arrestors for visible damage. Verify they are still in working condition by measuring the grounding resistance of the station by using a multi-meter. (Resistance of less than **5 ohms** is optimal)	Multimeter
7,2	Set the multi meter to **resistance (ohms)** mode. Place one probe on the center pin of the antenna connector and the other probe on the surge arrestor to measure resistance.	Multimeter
7,3	The multimeter should show infinite resistance (open circuit) for a good condition.	
7,4	Ensure that the surge arrestors are still properly linked to the ground by use of an earthing wire. *Replace the earthing wire if not connected properly or tighten any loose connections.*	
8,1	Inside the electronic cabinet for the power connections, check for worn out, damaged power cables, overheated cables or exposed wires. *Take a picture for record purposes and planning for replacement. Replace immediately if damage is of a high degree. Close any open connections.*	Camera
8,2	Use a multi meter to measure if the power supply voltage and current for each component matches the equipment’s requirements. Set the multi meter to **voltage and current** mode. (one mode at a time) Place the multi-meters on opposite electric poles of the equipment. Read the values on the screen and compare with manufacturer’s specifications.	Multimeter
8,3	Dust off the electronic cabinet interior and its equipment. Ensure ventilation paths are clear for cooling.	Towel
9,1	Treat any metal components with anti-corrosion spray if in a high-humidity or coastal area.	
9,2	Ensure the electronic cabinet is locked keeping the electronic components safe from animals’ destruction and theft.	
10	Verify that all the battery backup systems are fully charged and functional. You may test this by temporarily unplugging the main power source.	
11	Check if the surveillance camera is still intact to connection cables and look out for any breakages on the screen.	
12	Ensure the entire set-up looks the same way you found it before you leave. In	

	case this is not so, follow through the procedures again from the top to identify any error made. *(Refer to Figure 19)*	

Procedure M4: **Power bar functionalities.**

Prerequisites

- Computer connected to Internet
- Multi-meter
- Clean towel/Cloth
- Electrical device to act as a dummy Load

Duration: 30 minutes

No.	Steps to follow	Tools
1	Open the door of the electronic cabinet at the ground station and locate the power bar at the extreme back.	
2	Using a clean towel or cloth, wipe the power bar clean of any dust.	
3,1	Inspect for any visible signs of wear, such as cracks, burns, or frayed cables. Damaged units should be taken out of service.	
3,2	Check that each outlet is clean and undamaged. Loose or damaged outlets should be repaired. o ensure solid contact with the equipment plugs.	
3,3	Verify that the PDU's power cables are properly managed and secured. Loose cables should be replaced.	
4	Use a multi meter to confirm that each outlet on the power bar delivers the expected voltage of the respective device.	[Volts]
5	On your personal computer, login in to the remote power connection. [Schneider Electric's management interface] (See section III.2a)	
6	At the interface, check and install any updates to improve functionality.	
7	Test and verify the PDU's network connection. Go to "Tests" then "Network Test", Input your desired LED Blink duration.	
8,1	Test and verify the PDU's remote access and control of the outlet device.	

8,2	At the dashboard, Click the “Control”,then “RPDU” then “Outlet” Switch on and off any outlet of your choice as you observe the response of the respective devices.	
9	On the Schneider Electric dashboard, monitor current per outlet or phase. Click the “Configuration”,then “RPDU” then “Outlet” Switch	
10	Test the load by connecting a dummy load to simulate the PDU’s maximum rated load. This can be verified on the Home Page of the Schneider Electric dashboard on your device.	
11	Confirm that the surge protection is operational.	
12	Simulate a power outage to verify the PDU resumes normal operation without requiring manual intervention.	
13	Check the logs of maintenance records, store and save test results, date and any issues found.	
14	Schedule the next maintenance for the next 3 months.	

V.2. - Installation

In case of any system vandalism or system failure or system replacement, re-installation of different components has to take place immediately for continuous operation of the SSOL ground station and efficient system function performance.

V.2.a. Hardware installation

Procedure INS1.0: **Configuring the controller**		
Prerequisites • SPID MD-03 Rotor Controller and its power supply • GS computer or remote desktop. • SPID MD-03 Controller Manual • USB serial cable • ROTCTL & ROTCTLD list of commands (available online) **Duration**: 30 minutes		
No.	**Steps to follow**	
1	At the ground station, open the electronic cabinet and neatly place the black MD-03 controller inside there.	
2	Connect the controller power cable to a free power port in the cabinet and switch on the controller.	
3	Using the SPID MD-03 controller manual (Pages 1-28), configure all the settings on the controller to get it functioning properly.	
4	After fully setting up the controller, connect one end of the serial cable to the serial port on the back of the SPID MD-03 controller marked as "COM" and the other end to the serial port on the GS computer.	
5	Access the GS computer through a screen interface connected to the electronic cabinet or from your personal device via remote desktop.	
6	You must be connected to the Paris Observatory network to access the GS computer files.	
7	Check for the serial port on the computer. Type the following Linux command in a command line terminal to list available serial devices.	

	`%ls /dev/ttyUSB*` The USB-to-Serial adapter shall appear as `/dev/ttyUSB0.`	
8	Confirm that the SPID MD-03 controller communicates with the GS computer via the serial port. `%ls -l /dev/serial/by-id/` The device name should turn up in the terminals. [usb-FTDI_FT232R_USB_UART_AQ03MO0U]	
10	Install Hamlib to control the rotors. `%sudo apt-get install hamlib-utils`	
11	Test the connection to the Spid MD-03 by using ROTCTL to control the rotor via Hamlib: `%rotctl -m 903 -r /dev/ttyUSB0 -s 57600 -vvv`	
12	Once the software is configured, test the connection, send commands via the terminal and verify that the controller responds correctly. (*See Procedure T1*)	
13	If the rotor is not responding, repeat steps 1-12 and *refer to testing procedures T1 and T6* for troubleshooting.	

MD-01

Figure 30: Front view of the controller

Back

MD-01

Figure 31: Back view of the controller

Procedure INS2.0: **Organization of electronic cabinet**

Prerequisites

- Cutter and pair of pliers
- Screw drivers
- Labelled electronic cabinet equipment. (Refer to Table 1 for reference);
- Cabinet Set-up plan
- Label marker
- Plastic tightening straps
- Power extension

Duration: 20 minutes

No.	Steps to follow	Tooling
1,1	Make a list of all the equipment that will be housed in the electronic cabinet as shown in Figure 7 and print it on a piece of paper. [*Refer to Section IIIa*].	
1,2	Organize the equipment into categories based on their functionality, such as power components, communication equipment, networking devices and proximity to their connections.	
1,3	Plan the layout so that frequently accessed components are easily reachable.	
2	At the ground station, unlock and open the electronic cabinet to access its components.	
3	Organize the equipment properly according to the layout plan designed.	
4	Ensure each equipment that needs to be plugged into power has a port on the power bar	
5	Label all cables, wires and equipment in English and make sure the labels are visible and can be read to avoid confusion and make future maintenance easier	
6,1	As you arrange the cabinet components, use appropriate length of wires to avoid tangles and reduce signal loss. Minimize them.	
6,2	Use trays, ties, and velcro strips to organize the wires and cables. Keep power and data cables separated to reduce any electromagnetic interference.	
6,3	Insulate any bare or damaged wires.	

6,4	Arrange the equipment to maintain proper airflow. Leave space between devices if possible to allow air to circulate.	
7	After installation, test all components to ensure proper operation. Verify that power, network, and communication systems are functioning as expected.	

V.2.b. Software installation

The software installation process starts with Docker software because it is the main operating system for SSOL ground station operations in both SATNOGS and CENSUS modes. The procedures gives a step by step breakdown of installing Docker software on the Ground station computer.

Procedure INS 3.0: **Installation of Docker software**

Prerequisites

- GS computer or remote desktop.
- Linux Operating system
- 64-bit version of Debian 12
- Software packages such as Python 3.6+

Duration: 15 minutes

No.	Steps to follow	Output

1	Before you install Docker, make sure you meet the prerequisites concerning security implications and firewall incompatibilities.
	`sudo apt-get update`
	`sudo apt-get install ca-certificates curl`	

2	Run the following command to uninstall all conflicting packages: `for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done`	

3	Add Docker’s Official GPG Key
	`sudo install -m 0755 -d /etc/apt/keyrings`
	`sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc`
	`sudo chmod a+r /etc/apt/keyrings/docker.asc`	

4	Set up Docker's `apt` repository.	

	`echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`	
5	Install the Docker packages using the following command. `sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`	

6	Start and Enable Docker
	`sudo systemctl start docker`
	`sudo systemctl enable docker`	

7	Verify that the installation is successful by running the hello-world image: `sudo docker run hello-world`	[Hello World]
8	You have now successfully installed and started Docker Engine.	

After the successful installation and testing of Docker software, you can launch the configuration script using the following procedure :

Procedure INS 4.0: **Launch of the configuration script**

Prerequisites

- GS computer or “remote computer”
- Main computer or “office computer”
- Access to a Bash terminal on the Main computer (Linux/MacOS or WSL2/Git Bash on Window)
- Docker installed and running on both computer (GS and Main)
- openssh-client package installed on both computer
- git installed on the Main computer

Duration: 5 minutes

No.	**Steps to follow**	**Ref**
1	Make sure Docker is running on your machine before proceeding. On Windows : launch the Docker Desktop app On Linux : `% sudo systemctl start docker`	
2	Remove previous installation projet :	

	`% rm -rf ~/satnogs-client`	

3	Create a directory to hold the GitLab project and data for your SatNOGS client installation
	`% mkdir -p ~/satnogs-client`
	`% cd ~/satnogs-client`	

4	Clone the GitLab project satnogs_mode :
	`% git clone https://clone-repo:glpat-fdnzaD5P__UU7TSaf7zC@gitlab.obspm.fr/cceres/ground_station/satnogs_mode.git`
	Access the project :
	`% cd satnogs_mode`	

5	Launch the automated installation script `% ./scripts/deploy-full.sh`	

6	You should see the message :
	"
	`Which type of installation do you want to perform?`
	`1) Full installation (from scratch)`
	`2) Reuse the last working configuration`
	`3) Save the current configuration as the new stable configuration`
	"	

At this stage, you have several options depending on your needs :

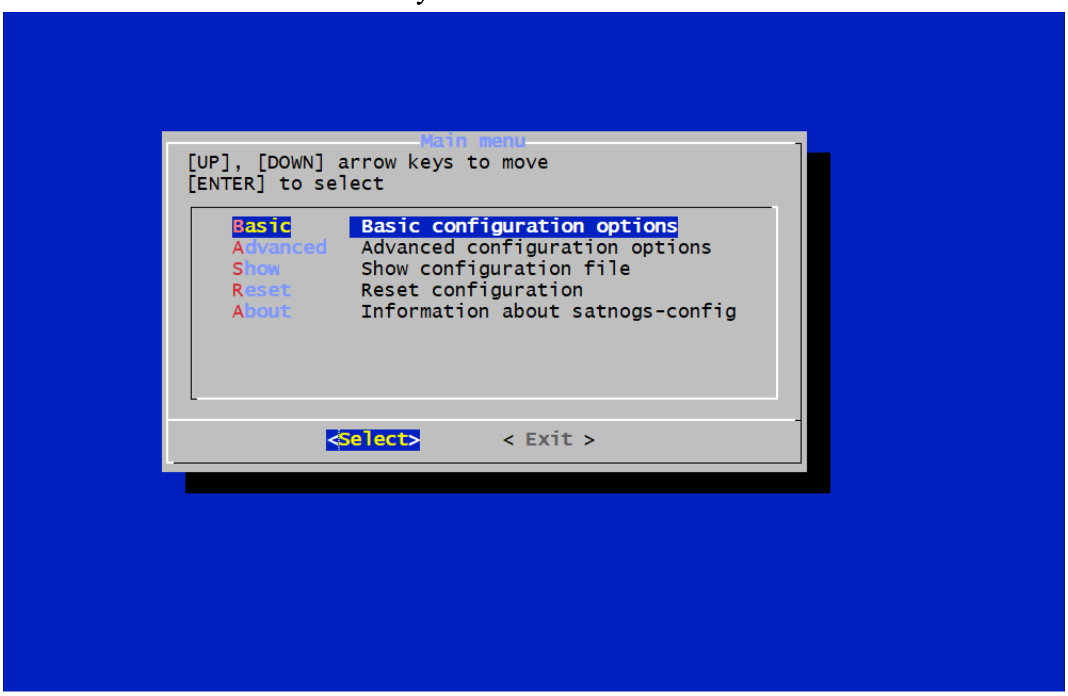
- You wish to install or reinstall the SatNOGS client from scratch → Refer to the user manual, procedure INS 4.1.
- You want to reinstall the most recent stable and functional configuration of the ground station → Refer to the user manual, procedure INS 4.2.
- You already have a SatNOGS client running and want to save the current station configuration as the latest stable reference → Refer to the user manual, procedure INS 4.3.

Procedure INS 4.1: **Installation of SatNOGS client from scratch**		
Prerequisites • procedure INS 4.0 **Duration**: 10-25 minutes		
No.	**Steps to follow**	**Ref**

1	In your terminal you should have the following message (at the end of procedure INS 4.0).
	"
	`Which type of installation do you want to perform?`
	`1) Full installation (from scratch)`
	`2) Reuse the last working configuration`
	`3) Save the current configuration as the new stable configuration`
	"
	If not, please go through procedure INS 4.0
		INS 4.0

2	After :
	"
	`Your choice [1/2/3]:`
	"
	Enter the command :
	`% 1`	

3	Enter the IP of the remote computer in which you want to install the client. For GS computer enter : `% 145.238.0.229`	
4	Enter the computer port used for ssh connection with the GS computer : `% 22`	
5	Enter the user that will be used for ssh connection with the GS computer : `% cceres`	
6	Enter the user password for ssh connection with the GS computer : `% cceres2019`	
7	At this point you should see :	

	“ ===== Deployment started on Tue Jun 10 11:23:57 2025 ===== [WARNING][2025-06-10 11:23:57] File .env not found. Creating a default file... [SUCCESS][2025-06-10 11:23:57] Default .env created. TARGET_IP: 145.238.0.229 SSH_PORT (default 22): 22 SSH_USER: cceres USER_PASSWORD: cceres2019 Configuration summary: ➤ Target computer IP : 145.238.0.229 ➤ SSH user : cceres ➤ SSH port : 22 ➤ Project directory on target computer : /stanogs_mode ➤ Station configuration parameters : /home/cceres/config_file ➤ Installation mode : New ground station configuration Confirm and continue? [y/N]: “	
8	If you have the exact same message, you can confirm : % y If it’s not the same message, please enter : % N And go back to step 3 of the procedure	
9	The installation process will then start automatically. It may take between **5 and 15 minutes** depending on your computer’s performance. During this time, you will see **logs for each step** of the process displayed in the terminal. You have to wait until you see that screen :  	

10	You can now proceed with the configuration of the station. (the last stable known parameters are provided in the appendix).	
11	Once you finished the configuration, select “exit”	
12	Wait for the end of the installation process. It may take between **2 and 5 minutes.**	

13	Once the message
	"
	`SatNOGS station provisioned successfully!`
	`Connection to 145.238.0.229 closed.`
	"
	appears, the installation of the SatNOGS client on the GS computer is complete.	

Procedure INS 4.2: **Reuse of the last stable configuration**

Prerequisites

- procedure INS 4.0

Duration: 10-25 minutes

No.	Steps to follow	Ref

1	In your terminal you should have the following message (at the end of procedure INS 4.0).
	
	"
	`Which type of installation do you want to perform?`
	`1) Full installation (from scratch)`
	`2) Reuse the last working configuration`
	`3) Save the current configuration as the new stable configuration`
	"
	
	If not, please go through procedure INS 4.0
		INS 4.0

2	After :
	
	"
	`Your choice [1/2/3]:`
	"
	
	Enter the command :
	`% 2`	

3	Now you have two cases :
	- If the connection with the GS computer was already setup in the project, you will see the message :
	"
	`===== Deployment started on Wed Jun 11 14:47:16     2025 =====`
	
	`Configuration summary:`
	`   ➢ Target computer IP                    : 145.238.0.229`
	`   ➢ SSH user                              : cceres`
	`   ➢ SSH port                              : 22`
	`   ➢ Project directory on target computer    : /stanogs_mode`
	`   ➢ Station configuration parameters      : /home/cceres/config_file`
	`   ➢ Installation mode                         : Redeployment of the last stable and working configuration`
	
	"
	
	And you can confirm :
	`% y`
		INS 4.1

<table>
<tr><td></td><td>
- If it's the first time you are making a connection with the GS computer, you will be asked to enter the connection parameters (if you need help on how to enter these parameters, please refer to INS 4.1 (steps 3 to 7). At the end you will see the message :

"
<pre>
===== Deployment started on Wed Jun 11 14:48:04   2025 =====

[WARNING][2025-06-11 14:48:04] File .env not found. Creating a default
file...
[SUCCESS][2025-06-11 14:48:04] Default .env created.
TARGET_IP: 145.238.0.229
SSH_PORT (default 22): 22
SSH_USER: cceres
USER_PASSWORD: cceres2019

Configuration summary:
   ➤ Target computer IP   : 145.238.0.229
   ➤ SSH user   : cceres
   ➤ SSH port   : 22
   ➤ Project directory on target computer   : /stanogs_mode
   ➤ Station configuration parameters   : /home/cceres/config_file
   ➤ Installation mode   : Redeployment of the last
stable and working configuration
</pre>
"

And you can confirm :
<code>% y</code>
</td><td></td></tr>
<tr><td>4</td><td>The installation process will then start automatically. It may take between 5 and 15 minutes depending on your computer's performance.
During this time, you will see logs for each step of the process displayed in the terminal.</td><td></td></tr>
<tr><td>5</td><td>Once the message
"
<pre>
SatNOGS station provisioned successfully!
You can now run 'satnogs-setup' to configure the station.
Connection to 145.238.0.229 closed.
</pre>
"
appears, the re-installation of the latest stable SatNOGS client on the GS computer is complete (you don't need to run satnogs-setup).</td><td></td></tr>
<tr><td>6</td><td>If you want to access the parameters used for that latest stable, enter :

<code>% cat ./stable-config/stable_config.txt</code></td><td></td></tr>
</table>

Procedure INS 4.3: **Save the current station configuration as the new latest stable configuration**

Prerequisites

- procedure INS 4.0
- having a SatNOGS client running on your GS computer

Duration: 10-25 minutes

No.	Steps to follow	Ref

1	In your terminal you should have the following message :
	
	“
	`Which type of installation do you want to perform?`
	`1) Full installation (from scratch)`
	`2) Reuse the last working configuration`
	`3) Save the current configuration as the new stable configuration`
	“
	If not, please go through procedure INS 4.0
		INS 4.0

2	After :
	
	“
	`Your choice [1/2/3]:`
	“
	
	Enter the command :
	`% 3`	

3	You will see the message :
	“
	`===== Deployment started on Wed Jun 11 14:47:16     2025 =====`
	
	`Configuration summary:`
	`➢ Target computer IP : 145.238.0.229`
	`➢ SSH user : cceres`
	`➢ SSH port : 22`
	`➢ Project directory on target computer : /stanogs_mode`
	`➢ Station configuration parameters : /home/cceres/config_file`
	`➢ Installation mode : Redeployment of the last stable and working configuration`
	
	“
	And you can confirm :
	`% y`	

4	Once you see : “ ✅ `New stable configuration saved.`	

	" It means that your actual configuration has been saved in the project as the latest stable configuration.	
5	If you want to access the parameters saved for that latest stable configuration, enter : `% cat ./stable-config/stable_config.txt`	
6	If you want to access the previous parameters saved for that latest stable configuration , enter : `% ls ./stable-config/past/` And choose one to display	

After the successful installation and testing of Docker software, for the CENSUS mode where PicTalk software is used, the three software components as defined in Section III.2a (GPredict, GQRX and Gnu radio) of that combination are then installed according to the procedures below.

Procedure INS 5.1: **Installation of GPredict Software**

Prerequisites

- GS computer or remote desktop.
- Linux Operating system
- Docker installed on your computer
- Software packages such as Python 3.6+, Hamlib 4.0+ Python binding

Duration: 15 minutes

No.	Steps to follow	Ref
1	Before installation, update your system to ensure the latest versions. `% sudo apt update`	
2	Install GPredict via APT `% sudo apt install Gpredict`	
3	Once installed, start GPredict by running the following command in the terminal: `% Gpredict`	
4	In the GPredict software, set up the ground station-1062 by entering location name, latitude, longitude, altitude and save the settings.	Manual 3.0
5	Configure GPredict to update the TLE data by downloading the latest files from the internet.	Manual 3.0
6	For any errors, troubleshoot using Testing procedures in Section V.II	Procedure T7

Procedure INS 5.2: **Installation of GQRX software**

Prerequisites

- GS computer or remote desktop.
- Linux Operating system
- Docker installed on your computer
- Software packages such as Python 3.6+, Hamlib 4.0+ Python bindings

Duration: 20 minutes

No.	Steps to follow	Output
1	Before installation, update your system to ensure the latest versions. `% sudo apt update`	
2	Install GQRX via APT `% sudo apt install gqrx-sdr`	
3	Once installed, start GQRX by running the following command in the terminal: `% gqrx`	
4	Ensure GQRX has detected the SDR device by confirming it has been configured.	Manual 4.0
5	Set the input device as the connected SDR.	Manual 4.0
6	Tune into a known frequency to test and ensure everything is working properly.	Procedures T4 & T7

Procedure INS 5.4: **Installation of GNU radio**

Prerequisites

- GS computer or remote desktop.
- Linux Operating system
- Docker installed on your computer
- Software packages such as Python 3.6+, Hamlib 4.0+ Python bindings

Duration: 20 minutes

No.	Steps to follow	Ref
1	Before installation, update your system to ensure the latest versions. `% sudo apt update`	
2	Install gnu radio via APT `% sudo apt install gnuradio`	
3	Once installed, start **GNU Radio** by running the following command in the terminal: `% gnuradio-companion`	Manual 5.0
4	You can start now developing SDR applications and obtaining signal processing flowgraphs using the graphical tool GNU Radio Companion (GRC).	Manual 5.0

In the case when the software system has totally failed/crashed, or when there is need of an entire upgrade of the software system, re-installation is the next step. Below is a step by step procedure for a re-installation of the software system.

Procedure INS6.0: **Re-installation of the SSOL software system**

Prerequisites

- GS computer or remote desktop (e.g. No Machine)
- Fully installed GS hardware system as illustration in Figures 7 and 8.
- Python versions 3.7 – 3.10 package for SATNOGS mode, 3.9 and 3.10 for CENSUS mode (custom) installations.
- Hamlib versions 4.0 – 4.3 package
- GPSD 3.22+ Python Bindings
- ATLAS 3.10+ Library
- HDF5 1.10+ Library
- A bootable USB drive with latest Ubuntu ISO, Debian 12 ISO

Duration: 90-120 minutes

No.	Steps to follow	Ref
1	Ensure all the necessary hardware for the ground station has been correctly set up, tested and installed according to Figures 7 and 8.	Section III.1
2,1	Access the SSOL computer by: **EITHER** going to the ground station physically where you connect a screen monitor to the GS computer. (*See Fig 20*)	
2,2	**OR** Logging in remotely via No Machine software.	Section III.2a
3	Ensure a stable internet connection for the GS computer for downloading necessary files and packages by checking if the Ethernet cable at the ground station is properly connected and same applies to your personal computer in case of a remote connection.	
4,1	Remove all existing software and configuration files from the computer by deleting any installations from Control Panel or through use of the command line terminal.	

4,2	• For SATNOGS software
	`% sudo apt purge satnogs-client satnogs-setup`
	• For Gpredict
	`% sudo apt remove --purge gpredict`
	• For GQRX
	`% sudo apt remove --purge gqrx-sdr`
	• For GNU radio
	`% sudo apt remove --purge gnuradio`
	• For Docker
	`% sudo systemctl stop docker`
	`% sudo apt remove --purge -y docker-ce docker-ce-cli containerd.io`
	`% sudo rm -rf /var/lib/docker`
	`% sudo rm -rf /var/lib/containerd`
	`% sudo apt autoclean -y`	

6	Ensure that you back up any important data from the Ubuntu system or any shared partitions.	
7,1	Clean the existing Ubuntu on the ground station computer by deleting the partitions used by the Ubuntu system and reformatting the disk.	

7,2	Repair the Bootloader by running the following commands in a terminal
	`% bootrec /fixmbr`
	`% bootrec /fixboot`
	`% bootrec /rebuildbcd`	

7,3	Insert the bootable USB drive with the latest Ubuntu ISO and into your computer and restart the computer.	
7,4	Once the system boots from the USB, you can re- install Ubuntu.	
7,5	After installation, you'll be prompted to remove the USB drive and the system will restart into your newly installed Ubuntu.	
8,1	Insert the bootable USB drive with the Debian 12 ISO again and into your computer and restart the computer.	
8,2	When the system boots, you'll see the Debian installation menu and start the installation process.	
8,3	After installation, remove the USB drive and the system will boot into your new Debian 12 installation.	

8,4	To install the required dependencies in Debian 12 run: `$ xargs -r -a packages.debian apt-get install`	
9	Update the system by typing this in the command line terminal `% sudo apt update && sudo apt upgrade -y`	
10,1	Install python 3.7 by typing this in the command line terminal. `% sudo apt install python3.7`	
10,2	Verify the installation `$ python3.6 --version`	

11,1	Install Hamlib 4.0 by typing this in the command line terminal.
	`% sudo add-apt-repository ppa:hamlib/hamlib`
	`% sudo apt install hamlib-utils libhamlib-dev`	

11,2	Verify the installation `$ rigctl --version`	
12,1	Install GPSD on your computer `% sudo apt install gpsd gpsd-clients`	
12,2	Install Python Bindings for GPSD `$ pip install gps`	
12,3	Verify the Installation `$ python3 -c "import gps; print(gps.__version__)"`	
13,1	Install ATLAS and related packages `% sudo apt install libatlas-base-dev`	
13,2	Verify the Installation `% ldconfig -p | grep atlas`	
14,1	Install the HDF5 library `% sudo apt install libhdf5-dev`	
14,2	Verify the Installation `$ h5cc -showconfig | grep "HDF5 Version"`	
15	Configure the controller to the ground station computer according to *Procedure INS1.0*	(Section V.3)
16	Re-install Docker software on the ground station computer according to *Procedure INS 3.0*	(Section V.3)

V.3. - Testing

Testing of hardware is done at component and system level for both hardware and software systems. Below are the most common procedures to be used in the testing of the SSOL ground station.

Hardware – On site Functional Testing

These are the tests that have to be performed on the hardware system of the ground station to ensure optimum functional performance of all components. These procedures are carried out on-site after installation/re-installation or in case of a detected malfunction in the software operation process.

Procedure (ID)	Description
T1	Antenna rotation
T2.1	Signal Acquisition Testing – Antennas
T2.2	Signal Acquisition Testing – SDR

Procedure T1.0: **Antenna rotation**

Prerequisites

- Controller manual from manufacturer.
- Screw keys of 19mm.
- Flexible gloves, safety boots, protective clothing, helmet.
- Small storage container for the screws.

Duration: 30 minutes

No.	Steps to follow	Tooling
1	At the ground station, open the electronic cabinet and access the black MD-03 controller inside there.	
2	Ensure the controller is ON. The green screen should be showing digits and words. (*See Figure12*)	
3.1	Test the azimuth rotation for full range of motion by pressing the left and right triangular buttons on far-left on the controller. [Minimum value = -360, Maximum value = +540]	
3.2	The antennas should turn right during the right pressing while they should turn left during the left pressing.	
3.3	The elevation **"E2"** value on the screen should increase in value after the right	

	pressing while it should decrease in value after the left pressing.	
4.1	Test the elevation rotation for full range of motion by pressing the top and bottom triangular buttons on far-left on the controller. [Minimum value = -180, Maximum value = +180]	
4.2	The antennas should turn facing up during the top pressing (increase of elevation) and should turn facing down during the bottom pressing (decrease of elevation).	
4.3	The azimuth **“A1”** value on the screen should increase in value after the top pressing and should decrease in value after the bottom pressing.	
5	For steps 3.1 – 4.3, confirm that the rotor moves smoothly without unusual internal mechanical sounds. If it does, perform mechanical and electrical inspection.	
6	Press the “S” button to test how to change configuration.	
7	For any more further testing of the controller abilities, use the SPID MD-03 controller manual (Pages 1-28).	
8	Ensure the controller is still “ON” and close the cabinet door.	

NB: Ensure sufficient margin of cable length to prevent all the power and RF cables from intertwining with each other during the antenna rotation (Steps 3.1 – 4.3).

The antennas should not interfere with any protection barriers or the mounting platform during antenna rotation. (Steps 3.1 – 4.3).

Procedure T2.1: Signal Acquisition of Antennas

Prerequisites

- Spectrum analyzer
- Handheld transmitter – Walkie talkie
- Mobile phone with internet
- Small storage container for the screws.
- Screen interface for the Ground station computer.

Duration: 20 minutes

No.	Steps to follow	Tooling
1	At the ground station, visually inspect the antennas and its cables to ensure they are not damaged from pressure, jamming, surface attrition on their full length, no-loose attachments, orientations.	
2	Note the target satellite frequency value in the UHF/VHF bands you are testing as described above in Section III.1a.	Section III.1a.
3	Connect a spectrum analyzer to the antenna system output at the SDR(receiver) to measure received signal strength at those frequencies by disconnecting the coaxial feed line and connecting it directly to the spectrum analyzer's input.	
4,1	Using the walkie-talkie set to a known frequency from a measured distance of 2m, generate a signal to be observed by the antennas.	
4,2	Observe the satellite signals on the analyzer and assess received quality.	
5,1	On the SATNOGS dashboard at your computer, look for any scheduled passes for that specific time period. (*See Manual 1.0*)	
5,2	Observe the live satellite signals on the analyzer during a scheduled pass to assess the received signal strength and quality.	
6,1	For steps 4.2 and 5.2, signals should appear with peaks at the expected satellite frequencies. Confirm that the signal strength is above the noise level.	
7,1	Generate a weak test signal at the test frequencies being used through a mobile phone hot-spot from a distance of 10m.	

7,2	Measure the ground station’s ability to pick up the weak signal and verify reception quality.	
8	Check that the received signal strength matches expectations based on distance and transmission power. Compare manufacturer specifications for the antenna.	

Procedure T2.2: Signal Acquisition of SDR

Prerequisites

- Handheld transmitter – Walkie talkie
- Mobile phone with internet
- Spectrum analyzer
- Screen interface for the Ground station computer with GQRX installed.

Duration: 20 minutes

No.	Steps to follow	Ref
1	At the ground station, open the electronic cabinet and locate the SDR.	
2	Ensure the SDR is properly connected to the antennas through the RF device combination in the electronic cabinet. (*See Figure 8*). Inspect the cable connections and ensure that they are free of any physical damages or tears.	
3	Ensure GQRX is installed on the GS computer to work with your the Airspy SDR (*See Manual 4.0 above).*	Manual 4.0
4,1	In GQRX, set the SDR to ensure accurate frequency tuning and select target UHF, VHF frequencies.	
4,2	Enter the satellite’s frequency into GQRX. Confirm the SDR is on the same frequency range as the expected satellite pass.	
5	Using the walkie-talkie set to the UHF and VHF frequency one after the other from a measured distance of 2m, generate a signal to be observed by the antennas.	
OR(6)	On the GS computer for the SATNOGS mode, schedule any satellite passes for that specific time period. (*See Manual 1.0*)	
7	As the satellite comes into range, observe the waterfall display in GQRX for the signal peak.	
8	Look for a clear signal line on the GQRX waterfall display corresponding to the satellite signal frequency. Optimize the gain and SNR for the clear reception for of signals with low and high SNR.	
10	Enable demodulation in GQRX to convert the received radio frequency (RF) signals into audio or data. Confirm the demodulated signal’s quality by observing the data.	

11	Listen for the audio output from the satellite signals through the speakers. Observe the quality of the audio.	
12	Analyze the audio files and review the decoded data for errors that may indicate alignment or reception issues.	
13	**For weak signals,** optimizing the gain settings, moving the SDR closer to the antenna output, or ensuring the LNA (if used) is functioning correctly. Adjust the frequency offset or correct for Doppler manually in GQRX or use tracking software if the SDR or antenna alignment drifts.	
14	Repeat Steps from 5 to 13, track multiple satellite passes to confirm consistent performance across different satellites.	
15	Close the electronic cabinet.	

Software – Remote (Offsite) Testing

The following tests have to be performed for the software system during troubleshooting in case of failure to achieve the expected observation data, or failed scheduled software system processes. These tests may or may not be carried out on site.

Procedure (ID)	Description
T3	Software updates
T4	Rotor connection & Satellite Tracking
T5	Ground station software Testing
T6	Software system Health Check

Procedure T3: **Software updates testing – Command line interface**

Prerequisites

- Screen interface for the Ground station computer
- Ground station computer

Duration: 30 – 90 minutes

No.	Steps to follow	Output
1	Log in to the SSOL GS computer. (See Section III.2a).	
2	Backup existing software configurations before any changes are made to the software.	
3	Confirm that each software component can connect to its respective hardware. (*Details described at the end of this procedure*) Use ping for IP-connected devices while for serial connections, verify with commands like dmesg or lsusb.	
4	Check the software logs for activities such as command history, signal data, errors correctly to specified locations. Type `ls` in your terminal for these locations to find out respective contents.	
5	Send any commands to any software of your choice to check responsiveness, positioning accuracy with the corresponding devices.	

6	Simulate hardware malfunctions or power interruptions to observe software responses and recovery by momentarily cutting internet supply, power supply and unplugging hardware devices.	
7	Restore and save the original configuration if any modifications were made during testing.	

Notes

The software to hardware mapping is defined as follows:

Software	Hardware
ROTCTL, GPredict	Controller
RIGCTL, GQRX	SDR
Gnu Radio	SDR
SATNOGS- CLIENT	Computer
Schneider Electric Application	Power Bar
GPredict, GQRX	KenWood Transceiver

Procedure T4: **Rotor connection & Satellite Tracking**

Prerequisites

- GPredict, Hamlib program, SATNOGS-client installed on your computer,
- ROTCTL & ROTCTLD list of commands (available online)
- Spid MD-03 controller manual

Duration: 20 minutes

No.	Steps to follow	Ref
1	Log in to the SSOL GS computer via command line interface (See Section III.2a).	
2	Access the ground station directory on the device `% cd station-1062`	

2	On your command line interface, ensure rotctld, GPredict and SATNOGS-client are installed by checking for their respective installation status.
	`%Rotctld:         Rotctld --version`
	`%GPredict:        gpredict --version`
	`%satnogs-client: which satnogs-client and docker ps`	

3	Crosscheck that SATNOGS configuration file (`station.env`) has been defined as per MD-03 controller and SATNOGS documentation for the rotor parameters. To check: `% cat station.env` To edit: `% nano station.env`	Controller Manual
4	Run the following command to ensure the rotor controller is recognisable by the computer. `%lsusb`	

5	Run the following commands to ensure the rotor controller is communicating with the computer via serial port USB0
	`%ls -l /dev/tty*`
	`% ls -l /dev/ttyUSB0`	

6,1	Run the rotctld program in the terminal to ensure the control is successfully configured and is communicating with the computer. (*See INS1.0*) `% rotctl -m 903 -r /dev/ttyUSB0 -s 57600 -vvv`	Installation 1.0

6,2	The rotctld window should open on your terminal for a successful connection. You can now try any command/option. `% docker --rotctld`	
6,3	Repeat step 4.1 in the *station-1062-rotctld-1* Docker container. `% docker exec -it station-1062-rotctld-1 bash`	
6,4	Check for any connection errors for the rotctld program running. `% sudo docker logs -f station-1062-rotctld-1`	
7	In the ROTCTLD docker container, send commands to the rotors to move to specific azimuth and elevation angles to test response. (*See list of ROTCTL commands*) `% docker --rotctld`	ROTCTL commands
8	Confirm that the rotor moves to the commanded position by observing the antenna movement on the live camera. If not, run Procedure T1.0 to check the connection, power, and software settings.	Manual 8.0 Procedure T1
9	Select a satellite currently in range and plan a pass in order to achieve real-time tracking.	Manual 3.0
10	As GPredict tracks the satellite, observe on the live feed if the rotor moves in sync with the satellite’s azimuth and elevation coordinates.	Manual 8.0
11	In case tracking does not start, verify port configurations in both rotctld and the tracking software. `% ps aux | grep rotctld`	

12	Restart rotctld and GPredict if any issues persist
	`sudo systemctl restart rotctld`
	`pkill gpredict`
	`gpredict`	

13	Perform tracking with multiple satellite passes to ensure consistent functionality and record the rotor’s performance for several passes to confirm accuracy and reliability of the setup.	
14	For any calibration issues, run *Procedures M2.1 and M2.2* to perform the rotor calibration as defined.	Procedures *M2.1* and *M2.2*

Procedure T5: **Ground station software Testing.**

Prerequisites

- Personal computer connected to the Internet.

Duration: 60 minutes

No.	Steps to follow	Ref
1	Log in to the SSOL GS computer via command line interface (See Section III.2a).	Section III.2a
2	Run Procedure T5 to update all the ground station software. Ensure that even the dependencies are updated.	Procedure T5
3	Crosscheck that SATNOGS configuration file (`station.env`) has been defined as per MD-03 controller and SATNOGS documentation for the rotor parameters. To check: `% cat station.env` To edit: `% nano station.env`	

4	Test network connectivity with SATNOGS network by typing in the command line interface from the SATNOGS client docker container.
	`% docker exec -it station-1062-satnogs_client-1 bash`
	`root@c7f5cb04bbb9:/# ping network.satnogs.org`	

5	Run this command in the SATNOGS container to ensure the radio controller is configured properly and can communicate with the SDR by running a RIGCTLD test. `root@c7f5cb04bbb9:/# rigctl -m 903 -r 4532 -s 57600 f`	
6	Run Procedure T6 to ensure that rotctld is communicating with the SATNOGS software moving the antennas accurately.	Procedure T6
7	Schedule a Test Observation using the command line interface or SATNOGS network website interface. Ensure the scheduled observation appears in your client logs.	Manual 1.0
8	After the observation, confirm that audio and data files are created in the directory of your choice by running `/var/lib/satnogs-client/data`	

9	Check that observation metadata is correctly populated in the SATNOGS dashboard and has been uploaded.	Manual 1.0
10	Run `% sudo docker logs station-1062-satnogs_client-1` to review any errors in the logs.	
11	For any issues are found in the logs, adjust the parameters in `station.env` and restart the SATNOGS client container to test new configurations. To edit: `% nano station.env` Restart : `% sudo docker restart station-1062-satnogs_client-1`	
12,1	If errors persist, consider reaching out to the SATNOGS support forum, LibreSpace Community for any assistance to your specific error.	
12,2	Type https://community.libre.space/t in your browser. Login with SSOL credentials to the cceres team dash board.	Section III.2a
13	If all solutions fail, consider re-installing the entire software system.	Procedure INS 6.0
14	Run a few satellite passes over a day to validate consistency and stability.	Manual 1.0

Procedure T6: Software system Health Check

Prerequisites

- Personal computer

Duration: 5 minutes

No.	Steps to follow	Output
1	Log in to the SSOL GS computer. (See Section III.2a).	
2	Type `apt list` in your command line interface to ensure that required dependencies and libraries are installed.	
3	Confirm that each package or library is at a compatible version. For specific applications, run a dependency management too For python use: `pip list` For software services use: `[service] --version` For docker: `docker ps` `docker inspect <container_id>`	
4	Type `top or htop` in your command line interface to monitor CPU, Memory, and disk usage.	
4	Use `df -h` to check available storage and `dmesg` to identify hardware-related messages that may indicate disk issues.	
5	To analyze network latency and speed, which can impact software performance. Run `ping -c 10 example.com` Observe the `min, avg` and `max` round-trip time values.	
6	Run `systemctl status [service]` to ensure all the software services (Docker, SATNOGS, GPredict, GQRX, GNU radio) are active and running without errors.	
7	Confirm any background applications are running smoothly and without high CPU or memory spikes by running. `% ps aux | grep [service]`	
8	Use `journalctl [service]` to review system logs, especially for services critical to system operation.	

9	Confirm firewall rules are in place and only necessary ports are open by running these processes. Check status: `% sudo ufw status` Check status: `% sudo firewall-cmd --state` Disable firewall: `% sudo systemctl stop firewalld`	

VI. - Safety and legal compliance

VI.1. - Radiation precautions

Satellite signals are made of electromagnetic waves that can cause harm to individuals. Therefore all satellite ground stations need to follow radiation precautionary measures.

A transmitting antenna produces an electric field whose power must be regulated for safety [1]. The goal is to define a security perimeter that is within the legal limit of an acceptable electric field power, therefore a need for calculation of a safety distance. The current VHF antenna has a gain of 12.3 dB, the UHF antenna has a gain of 14 dB. The regulations concerning radio communications state the value of the electric field produced by an antenna outside the security perimeter must not exceed 28.84 V/m for a transmission frequency of 440MHz [2]. The calculated safety distance for the rear lobe of the VHF antenna at a power of 110 W is 0.65m while that of the rear lobe of the UHF antenna at the same power is 1.51m. These calculations were carried out for a maximum power of 110 W supplied to the antenna as provided by the Kenwood.

In addition, the antenna is located at a height 7.5 m (which falls within the acceptable antenna height range (5m -20m) above the ground, implying a far distance from the electric field around the antennas and fully complies with the legal regulations concerning public exposure.

VI.2. - Lightning & Grounding

The facility requires grounding and protection against lightning.

- Proper grounding of the antennas was done during installation by a qualified expert (Direction de l'Immobilier et de la Logistique) and inspection thereafter during maintenance.
- lightning arresters are placed directly under each antenna before any other equipment running all the way to the ground to protect the entire ground station from lightning.
- Potential equalization is performed by connecting a grounded mast to the antennas and all outgoing coaxial cables.

VI.3. - Mechanical assembly precautions

The following precautions have been implemented for the hardware installation of the ground station to ensure safety of all involved persons and the equipment itself;

- There is clear identification of a security perimeter to close off the immediate adjacent areas around the ground station. The ground station is stationed on the roof a building whose access is constrained with a lock at all times.
- The antenna is requested to a height approximately 7.5m during emission above the ground to avoid any related harm to property, humans and life.
- Presence of tension struts to balance the equilibrium between electronic cabinet, the sliding platform, pylon spire and antenna weight.

- Allowable length margin in the cables and wires leading from the electronic cabinet to the antenna has been provided to relax tension in the wires and prevent repeated intertwining during the rotation motion by the rotors.
- Presence of a raised horizontal flat plate that will facilitate and ease the maintenance of the pylon spire and antenna at a higher altitude from the roof surface.

VI.4. - Compliance requirements

SSOL ground station is compliant with the all space and radio frequency laws and regulations of France and Europe, namely:

- A satellite operator can choose to establish their network of ground stations or utilize services from a ground station service provider.
- A ground station operator shall have a license for carrying out satellite operations and sharing satellite data to the general public.
- A ground station operator shall obtain radio permission to transmit signals (uplink). The Meudon observatory shall obtain a permanent license to cater for all its emission tests and operations. For installation purpose, CENSUS is supported by the former operator of Picsat who is granted a license..

VII. - Appendix

VII.1. - Last "stable" configuration parameters of the station (12/06/2025).

enable_iq_dump: true
hamlib_utils_rot_enabled: true
hamlib_utils_rot_opts: -m 903 -r /dev/ttyUSB0 -s 57600
satnogs_antenna: RX
satnogs_api_token: d87809bcaae3bc2a5de9b83ddfc4819affef1979
satnogs_gain_mode: Settings Field
satnogs_log_level: DEBUG
satnogs_other_settings: LNA=11,MIX=7,VGA=11
satnogs_rig_ip: 127.0.0.1
satnogs_rot_baud: '57600'
satnogs_rot_enabled: true
satnogs_rot_model: ROT_MODEL_SPID_MD01_ROT2PROG
satnogs_rot_threshold: '1'
satnogs_rx_samp_rate: 2.5e6
satnogs_soapy_rx_device: driver=airspy
satnogs_station_elev: '120'
satnogs_station_id: '1062'
satnogs_station_lat: '48.807'
satnogs_station_lon: '2.229'
udp_dump_host: 172.18.0.1
udp_dump_port: '49049'

VII.2. - Maintenance log book

ID	Operator	Date	Description of the maintenance