



Développement d'un géonavigateur GPS

—

UV 2.4 - Projet de découverte des systèmes (2015-2016)

Sujet proposé par : Pierre Bosser et Fabrice Comblet

Sarah DELMAS

Joris TILLET

Sophie TUTON

Nicolas VEYLON

2018 Promotion

Mars 2016

Sommaire

I. Préambule	2
II. Première approche : Une étude bibliographique	2
II.1 Le port série	2
II.1.1 Le fonctionnement général.....	2
II.1.2 Les types de communications	2
II.1.3 Notre cas : le GPS G-STAR IV	3
II.2. Les origines, le fonctionnement, les applications du NMEA.....	3
II.2.1 Qu'est-ce que les NMEA ?.....	3
II.2.2. Les trois types de phases.	4
II.2.3 Les significations des données envoyées.....	5
II.3. Les origines, le fonctionnement, les applications du GNSS et GPS.....	5
II.3.1 Les trois segments.....	6
II.3.1.1 Segment spatial.....	6
II.2.1.2 Segment de contrôle.....	6
II.3.1.3 Segment utilisateur	6
II.3.2 Géodésie	7
II.3.3. Sources d'erreur.....	8
III. Travail réalisé et difficultés rencontrées.....	9
III.1. Travail réalisé	9
III.2. Difficultés rencontrées.....	10
IV. Répartition des tâches au sein du groupe et organisation du travail au cours de l'UV.....	11
V. Un aperçu des résultats obtenus à l'issue de ce projet	13
Conclusion.....	14
Bibliographie	15

I. Préambule

L'objectif principal de ce projet était de mettre en œuvre l'acquisition de données GPS et l'affichage du positionnement obtenu sur une carte. De plus, nous devons afficher une carte du ciel et y positionner les satellites avec lesquels le GPS communique. Nous sommes donc partis d'un GPS se connectant via un port série, et avons développé une application récupérant ces données, les traitant, et affichant les principales informations dans une interface graphique.

Ce rapport a été réalisé dans le but de présenter, dans un premier temps, sur quelles technologies sont basées le GPS et les données qu'il envoie, ainsi que leur fonctionnement. Plus loin est décrit le travail réalisé, et les difficultés apparues au cours du projet. Ensuite, la répartition du travail dans le groupe est détaillée. Finalement, avant la conclusion, des images présentent à quoi ressemble l'application finale.

II. Première approche : Une étude bibliographique

II.1 Le port série

II.1.1 Le fonctionnement général

Le port série sert à établir une communication entre un terminal (ordinateur), et un appareil communiquant, ici notre GPS. Pour se faire, les données sont envoyées bit par bit, contrairement aux ports parallèles. Ils sont faciles à utiliser et ne requièrent que peu de fils. Les débits réalisés sont compris entre 75 bits par seconde et 115 200 bits par seconde. On les utilise notamment pour communiquer avec des imprimantes, des souris, des modems... ou des GPS.

Théoriquement, il n'y a besoin que de deux lignes : une utilisée pour les données, et une ligne reliée à la terre. Seulement, des perturbations provoquant des erreurs obligent l'utilisation d'autres lignes pour faire des contrôles.

Pour établir la communication entre l'ordinateur et l'équipement, il faut définir le protocole utilisé, ce qui consiste à définir le débit de transmission, le codage utilisé, le découpage en tram, etc. La norme standardisant ce protocole s'appelle la norme RS-232. Elle permet d'éviter les problèmes dus aux différences existant entre les différents ports. En effet, l'arrangement des ports peut varier d'un équipement à un autre : certains utilisent un même pin (connecteur) pour envoyer et recevoir, alors que d'autres utilisent des pins différents et peuvent donc envoyer et recevoir simultanément (ce qui est donc plus rapide).

II.1.2 Les types de communications

Il existe deux types de communication :

- La communication synchrone : l'émetteur et le récepteur ont chacun une horloge qui sont synchronisées et qui donnent chacune l'intervalle de temps entre chaque bit envoyé ou reçu. S'il venait à avoir une différence entre les intervalles des deux horloges, la communication s'arrête avec une erreur.
- La communication asynchrone : la communication se fait par flux de données. Le début d'un flux de bits est signalé par un starter bit (bit de départ valant 0). Le récepteur reçoit alors les bits pendant un certain temps avant d'avoir un nouveau flux.

Un tel flux de bits se décompose ainsi :

1 bit de départ (qui vaut 0)

7 à 8 bits de données

1 bit de parité (optionnel)

1 ou 2 bits d'arrêt.

II.1.3 Notre cas : le GPS G-STAR IV

Lorsque l'on consulte la documentation fournie par le constructeur du GPS qu'on utilise, on peut avoir toutes les informations nécessaires sur le fonctionnement du port série de ce dernier. On apprend ainsi (voir Tableau 1) que le premier pin sert à l'alimentation électrique du GPS, les deux suivants sont ceux qui envoient les données à proprement parler, et le dernier pin est relié à la terre.

PIN Number(s)	Name	Type	Description
1	Vin	P	Main power supply to the BU-5353.
2,3	USB IF		USB IF to transmits channel for outputting navigation and measurement data to user's navigation software or user written software.
4	GND	P	Ground.
LED	RED		LEDOFF: Receiver switch off LED ON: Receiver switch on No fixed, Signal searching LED Flashing: Position Fixed

Tableau 1 : Caractéristiques du port série du GPS STAR IV. Extrait du manuel d'utilisation du GPS.

D'autre part, on trouve également dans le manuel sous quelle forme sont envoyées les données qui nous intéressent :

« **Set Baud Rate** 4800, **Data Bit**: 8, **Parity**: None, **Stop Bit**: 1, **Flow Control**: None »

On sait ainsi que les données sont transmises à une vitesse de 4800 baud par secondes (un baud étant un « symbole », pouvant contenir à lui tout seul plusieurs bits), que le codage d'une donnée se fait sur 8 bits, qu'il n'y a pas de bit de parité, et que le 1 est le bit d'arrêt. La communication est donc asynchrone, puisqu'il y a un bit d'arrêt, et que l'on a besoin de connaître la vitesse de communication (le baud rate) qui compense l'absence d'horloge interne au GPS.

Enfin, le GPS utilisé pour ce projet se connecte via un port USB. En effet, le port USB remplace de plus en plus les ports série aujourd'hui. Le port USB va alors s'interfacer comme un port série, et pourra servir de lien de communication entre le GPS et l'ordinateur.

II.2. Les origines, le fonctionnement, les applications du NMEA

II.2.1 Qu'est-ce que les NMEA ?

Les NMEAs, créés par la National Marine Electronics Association, dont ils tiennent leur nom, sont des standards pour des protocoles de communications entre instruments nautiques. Ils

permettent à ceux-ci de dialoguer avec une même « grammaire ». Il en existe deux, le 2000 qui n'est pas lisible, et le 0183, crée en Mai 1983, qui est celui auquel nous nous intéresserons pour la communication GPS.

Ces protocoles régissent quelles informations sont échangées par les instruments et dans quel ordre celles-ci sont transmises. Les messages NMEA 0183 sont sous forme de « phrase » commençant toujours par « \$ » et finissant par un retour chariot, Carriage Return, et un saut de ligne, Ligne Feed : <CR><LF>. Sans compter les « \$ », les CR et LF, les phrases font au maximum 80 caractères. Codés sur 8 bits, ils utilisent des caractères ASCII.

II.2.2. Les trois types de phrases.

Il existe trois types de phrase différentes : les phrases émettrices, les phrases propriétaires et les phrases interrogatrices.

Émettrice :	\$TTSSS,champs de données 1,champs de données 2, ...*XX<CR><LF>
Propriétaire :	\$PABC,champs de données 1,champs de données 2, ...*XX<CR><LF>
Interrogatrice :	\$TTLQ,BBB*XX<CR><LF>

TT	Identification en deux lettres de l'appareil nécessaire pour utiliser les données, ou de l'appareil interrogeant	SSS	Identification en trois lettres du contenu de la phrase
P	Identification en une lettre un message du propriétaire	MMM	Identification en trois lettres du propriétaire
Q	Identificateur de question (toujours en 5 ^{ème} position)	BBB	Question transmise toutes les 5 secondes, jusqu'à ce que la réponse arrive
LL	Identification en deux lettres de l'appareil questionné	*XX	Checksum en hexadécimal : XOR de tous les caractères inclus, entre « \$ » et « * »

Tableau 2 : Format d'une phrase.

Lors de la réception des données par le GPS, une partie des phrases peut être perdue. C'est pour ça qu'il existe le « checksum ». Chaque trame se termine par le caractère « * » suivi de deux autres caractères représentant un nombre en hexadécimal. Ce nombre est le XOR de tous les caractères inclus entre « \$ » et « * », comme précisé dans le tableau 2. Avant d'extraire les données d'une ligne reçue, il est donc facile de contrôler si cette dernière contient des erreurs, ou a des données manquantes. Si le checksum est vérifié, il y a de bonnes chances pour que la trame ne contienne aucune erreur. Nous avons donc fait l'hypothèse qu'une trame est correcte si et seulement si le checksum est valide.

II.2.3 Les significations des données envoyées

Le protocole NMEA 0183 est basé sur la norme RS-232, qui fonctionne grâce à un port série. Les ports série auxquels nous avons accès pour notre projet ne nous servent qu'à recevoir. De ce fait, nous ne sommes intéressés que par les messages émis et, plus précisément, par certains types de messages. Ils sont précédés de GP (Global Positioning), signifiant que les données envoyées par la suite sont des données de géolocalisation. Ces messages sont : les messages GSV donnant des données sur les satellites en vue, et les messages RMC et GGA, donnant la position du port série. Ci-dessous se trouve une explication plus détaillée du contenu des messages GSV, RMC et GGA (nous avons utilisés les informations indiquées en italique pour ce projet).

GSV : Satellites in view

:\$--GSV,1,2,3,4,5,6,...*XX

- 1) nombre de messages envoyés
- 2) numéro du message
- 3) nombre de satellite en vue
- 4) *numéro du satellite*
- 5) *élévation du satellite en degrés par rapport au port série*
- 6) *azimut en degrés entre le Vrai Nord et le port série, par rapport au Vrai Nord*
- XX) *Checksum*

RMC : Recommended minimum specific GPS/Transit data

:\$--RMC,1,2,3,4,5,6,,,,,*XX

- 1) l'heure (UTC en HeureMinutesSecondes.Centisecondes) de l'envoi du message
- 2) *l'état du satellite, « V » si il y a un problème dans les données, « A » autrement*
- 3) *la latitude du port série en degrés-minute*
- 4) *« N » si au nord, « S » si au sud*
- 5) *la longitude du port série en degrés-minute*
- 6) *« E » si à l'est, « W » si à l'ouest*
- XX) *Checksum*

GGA : Global Positioning System Fix Data

:\$--GGA,1,2,3,4,5,6,,,9,10,,,*XX

- 1) l'heure (UTC en HeureMinutesSecondes.Centisecondes) de l'envoi du message
- 2) *la latitude du port série en degrés-minute*
- 3) *« N » si au nord, « S » si au sud*
- 4) *la longitude du port série en degrés-minute*
- 5) *« E » si à l'est, « W » si à l'ouest*
- 6) la qualité de l'indicateur GPS,
- 9) l'altitude par rapport au niveau de la mer du port série
- 10) l'unité de l'altitude, « M » si en mètres
- XX) *Checksum*

II.3. Les origines, le fonctionnement, les applications du GNSS et GPS

Les systèmes GNSS (Global Navigation Satellite System) sont apparus dans le domaine civil depuis les années 2000. Le plus répandu est le GPS (Global Positioning System). Ces systèmes permettent de donner les coordonnées géographiques d'un récepteur grâce à un réseau de satellites émetteurs.

Le système se découpe en trois segments : les satellites, les stations de contrôle et le récepteur mobile. Ce système n'est pas d'une précision absolue du fait de différentes sources d'erreur. Cependant, il offre tout de même une précision suffisante pour les applications standards.

II.3.1 Les trois segments

II.3.1.1 Segment spatial

Il s'agit de satellites se trouvant en orbite autour de la Terre à une altitude de 20 200km sur l'orbite terrestre moyenne. Le nombre de satellites et leurs positions sont tels que les signaux d'au moins quatre satellites sont reçus par un quelconque récepteur, quel que soit sa position. En effet, il est nécessaire d'avoir au moins trois satellites pour obtenir 2 positionnements possibles. L'ajout d'un quatrième satellite est suffisant pour faire le choix entre ces deux positionnements, tout en synchronisant l'horloge. Cependant, du fait des erreurs de mesures (cf. partie III), il est usuel d'utiliser plus de 4 satellites pour une meilleure précision des coordonnées.

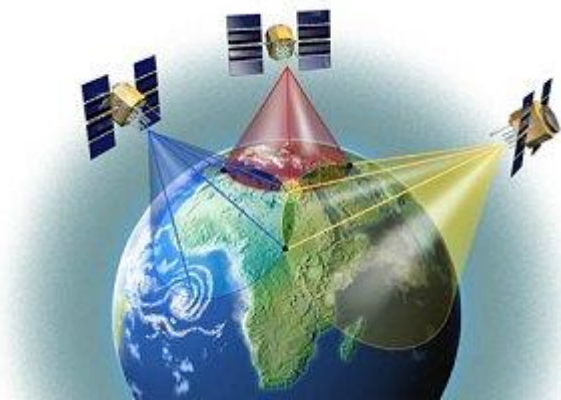


Figure 1: Triangulation
mayerwin.free.fr

Les satellites transmettent des ondes radios porteuses. Il existe deux fréquences d'usage : 1575.42MHz et 1227.60MHz. Chacune de ces porteuses porte un code qui contient les données nécessaires à la triangulation, notamment la position des satellites (éphémérides) ainsi que l'heure d'émission. Deux codes sont utilisés : le code d'acquisition brut (code C/A) civil, et le code précis (code P) réservé aux usages militaires.

II.2.1.2 Segment de contrôle

Il s'agit d'un réseau de stations au sol permettant de rectifier les trajectoires des satellites, d'assurer une synchronisation des horloges des satellites, et de minimiser les erreurs possibles (cf. partie III) nécessaire au bon fonctionnement du système de positionnement.

II.3.1.3 Segment utilisateur

Le rôle du récepteur est d'assurer la réception du signal et l'acquisition des données. Il calcule ensuite les coordonnées de son antenne par triangulation comme vu précédemment.

Deux méthodes de calcul sont possibles : l'une à partir du code C/A, l'autre à partir de la porteuse.

- **Méthode 1 :** Sachant que la distance entre le satellite et le récepteur est égale au temps de propagation de l'onde multiplié par la vitesse de la lumière (supposée constante), les données reçues des différents satellites mènent à la résolution d'un système donnant les coordonnées X, Y, Z et le temps de propagation de l'onde.

- Méthode 2 : La distance d entre le satellite et le récepteur peut se déduire du déphasage φ entre les deux signaux émis et réceptionnés, suivant la formule : $d = \lambda(\varphi + n)$, où λ correspond à la longueur d'onde supposée constante durant sa propagation et n le nombre d'oscillation durant celle-ci. Cette dernière grandeur est difficile à estimer et requiert « de lourds algorithmes » (Lahaye R, Ladet S, 2014).

Les deux méthodes sont suffisamment précises pour des applications standards. En effet, en théorie, la première méthode offre une précision de l'ordre du mètre (3 mètres pour le code C/A, et 30 centimètres pour le code P), tandis que la seconde de l'ordre du millimètre (entre 1,9 et 2,4 millimètres).

II.3.2 Géodésie

La position ainsi calculée nécessite un traitement pour être parfaitement exploitable. Il est usuel d'utiliser un système géodésique, c'est-à-dire sous forme de latitude, longitude et hauteur à la normale au sol. Les récepteurs convertissent ainsi les coordonnées géocentriques directement obtenues en d'autres formats, tels le WGS 84 (système mondial le plus courant) ou le RGF93 (système français).

Le passage du système de coordonnées RGF93 (x, y) au WGS84 (longitude, latitude) se fait par une série de calculs : translation métrique, rotation et mise à l'échelle. C'est ce qu'on appelle la projection Lambert 93. Il s'agit en fait d'une "projection conique conforme sécante" caractérisée par les paramètres suivants :

- *parallèles d'échelle conservée* : $L1 = 44^\circ \text{ N}$ et $L2 = 49^\circ \text{ N}$;
- *méridien central* : $G0 = 3^\circ \text{ E}$ (Méridien de Greenwich) ;
- *latitude origine* : $L0 = 4630' \text{ N}$;
- *coordonnées de l'origine* : $E = 700\,000 \text{ m}$ et $N = 6\,600\,000 \text{ m}$.

Pour résumer, pour passer des coordonnées cartographiques (x, y) aux coordonnées géographiques (lat, long) on utilise les coordonnées projetées (p, θ) sur le cône passant par les deux parallèles $L1$ et $L2$ (cf. figure ci-dessous) et calculées en utilisant les paramètres définis précédemment.

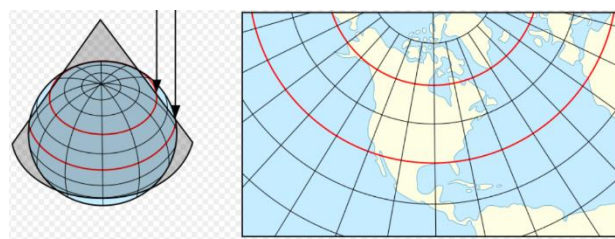


Figure 2: de RGF93 à WGS84 (en rouge : $L1$ et $L2$)
wikipedia.org

Ces différents systèmes sont dits géodésiques. Ils sont plus pertinents car ils prennent en compte les imperfections dans le champ de gravité terrestre. En effet, du fait de l'hétérogénéité du manteau terrestre ainsi que des reliefs, le champ gravitationnel n'est pas uniforme à la surface. La Terre n'est pas simplement un ellipsoïde, mais ce qu'on nomme un géoïde (cf figure ci-contre).

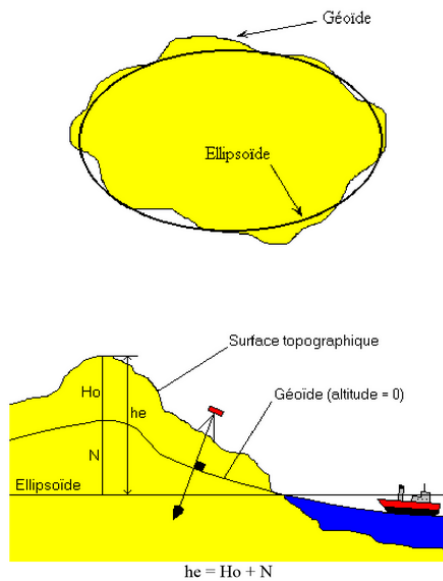


Figure 3: Géοide
80calcs.pagesperso-orange.fr

Il existe différents systèmes géodésiques selon l'endroit considéré, chacun utilisant un ellipsoïde qui sert de modèle pour le géοide et qui se confond parfaitement avec elle en un point considéré. Le système WGS 84 trouve son origine au centre des masses de la Terre et de sorte que l'axe Z soit confondu avec l'axe de rotation de celle-ci, d'où sa pertinence à l'échelle mondiale du fait que son ellipsoïde (IAG GRS 80) se rapproche le plus du géοide.

II.3.3. Sources d'erreur

Diverses sources d'erreur affectent la précision du système. Elles vont être détaillées par ordre décroissant de l'erreur causée.

○ Ionosphère :

Les ondes radio émises par les satellites sont déviées de leur trajectoire en traversant l'ionosphère, retardant ainsi aléatoirement le temps d'émission des ondes et engendrant une erreur de l'ordre de 4 mètres.

○ Désynchronisation des horloges

Le décalage d'une nanoseconde d'une horloge engendre une erreur de 30 centimètres. En pratique, les horloges sont désynchronisées d'environ 7 nanosecondes, engendrant alors une erreur de 2,1 mètres sur la position. Cette désynchronisation est notamment une conséquence d'une imperfection intrinsèque des horloges ainsi que des phénomènes de la relativité restreinte et générale.

○ Ephémérides

Un défaut sur la position des satellites due aux imprécisions des éphémérides peut engendrer une erreur d'environ 2 mètres.

○ Multi-trajets

L'onde récupérée par le récepteur peut avoir été réfléchiée une ou plusieurs fois au cours de son trajet, notamment en zone urbaine et montagneuse. Il survient alors un écho de l'onde par rapport à celle non-réfléchiée, c'est le multi-trajet. L'erreur engendrée est de l'ordre du mètre.

○ Effet Canyon

La célérité de l'onde dépend du milieu traversé. L'onde arrive ainsi en retard, ce qui fausse la mesure.

○ Troposphère

Les différences d'humidité et de pression dans la troposphère engendrent une modification de vitesse et de direction des ondes émises. L'erreur engendrée est d'environ 0,7 mètre. Il existe cependant des correcteurs intégrés à certains récepteurs, la correction étant faite grâce aux données météo.

- Imprécision du récepteur
- Le récepteur

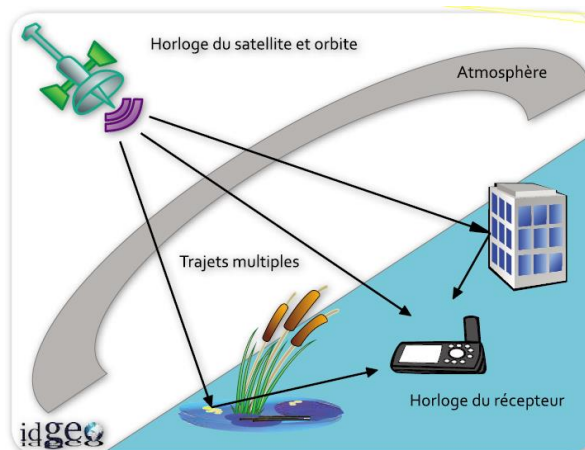


Figure 5: quelques sources d'erreurs
fr.slideshare.net/IDGEO/sig-et-gps-gnss-concepts-de-base

L'ensemble de ces erreurs cumulées font un total de l'ordre d'une dizaine de mètres.

III. Travail réalisé et difficultés rencontrées

III.1. Travail réalisé

La première des tâches qui devait être réalisée était l'acquisition des données, depuis le GPS via le port série. Pour se faire, nous avons dû utiliser la bibliothèque *pySerial*, qui permet de lire et écrire sur un port série (ici la lecture seule nous importait). Les données nécessaires pour communiquer correctement avec le GPS en utilisant le port série sont celles présentées dans la partie II.1.3.

Une fois les données du GPS récupérées, seules les trames correctes sont gardées. Pour savoir si une trame est correcte, ces critères de choix ont été sélectionnés :

- La ligne doit faire au moins 3 caractères.
- Elle doit contenir le caractère '*', qui précède obligatoirement le checksum.
- Elle doit vérifier le checksum (cf. II.2.2)

Puis les trames correctes sont formatées pour une meilleure exploitation future, c'est-à-dire découpées en utilisant les virgules comme séparateur. Ensuite, celles-ci sont triées selon leur nom (GGA, GSV ou RMC). En fonction de leur type, seules les informations que nous désirions gardées sont enregistrées (cf. II.1.3, données en italique) après avoir été converties en des données plus exploitables pour le futur affichage graphique (nous devons par exemple convertir les coordonnées en degrés-minutes vers des degrés).

Au début, nous écrivions n trames dans un fichier qui était ensuite exploité. Mais finalement, pour que notre application soit plus dynamique, nous avons opté pour une boucle qui englobe tout le reste du programme. Ainsi, à chaque tour de boucle, une nouvelle trame est lue puis traitée et enfin

affichée via l'interface graphique présenté plus loin. Le programme traite ainsi les données reçues par le GPS en temps réel.

Une fois les données ordonnées et traitées, elles sont appelées par des fonctions de conversion utilisant la bibliothèque *gdal* de Python et permettant d'obtenir un équivalent des coordonnées du GPS et des satellites en pixels. Cette dernière conversion utilise les coordonnées Lambert 93 de l'origine (ici le coin supérieur gauche) et le pas (par pixel), récupérés dans l'orthoimage grâce à la méthode *GetGeoTransform* de *gdal*. Connaissant ces données, nous pouvons trouver les coordonnées Lambert 93 de n'importe quel pixel de l'image, et vice-versa. Toutes ces conversions servent ensuite à l'affichage de la position sur une carte et de la constellation de satellites sur une interface graphique crée via la bibliothèque *Tkinter* de Python. Ainsi, le script utilise les données de géolocalisation pour cibler le GPS G-STAR sur une carte, et les données d'information sur les satellites pour cartographier ceux-ci sur une carte du ciel.

Pour l'interface graphique, nous avons d'abord cherché à créer deux fenêtres contenant respectivement l'orthoimage et la carte du ciel. Cette solution a été adoptée dans un premier temps, puis nous avons créé une interface composée d'une seule fenêtre et de deux onglets. Dans le premier onglet, on retrouve l'orthoimage et un bouton *START* qui permet de lancer le positionnement GPS et la carte du ciel. Après avoir cliqué une première fois sur *START*, la position se met à jour automatiquement. La carte du ciel se trouve dans le second onglet. A côté de celle-ci se trouve une zone de texte qui affiche le numéro et la position de chaque satellite.

III.2. Difficultés rencontrées

Plusieurs difficultés ont dû être surmontées. La première difficulté a été de faire communiquer le port série avec Python. Il manquait la librairie *pySerial* sur les ordinateurs. Puis Python n'avait pas les droits pour accéder au port série. Mais le problème a été résolu rapidement et les premières données ont pu être acquises depuis le GPS.

La seconde difficulté a été d'utiliser la bibliothèque conseillée (*gdal*) pour obtenir les informations détenues par une orthoimage. En effet, les données contenues dans l'orthoimage n'étant pas indiquées, nous pensions devoir extraire un *raster* de l'image. Mais après discussion avec un encadrant, nous avons su quelles données contenait l'orthoimage. Suite à cela, nous avons trouvé facilement comment extraire ces données et les utiliser pour connaître la position GPS de n'importe quel point sur la carte.

Nous avons eu du mal à gérer l'affichage, mais surtout la mise à jour d'un point superposé à une image dans une fenêtre *Tkinter*. Nous avons tout d'abord affiché le résultat dans une figure intermédiaire *matplotlib*, puis nous l'avons transféré dans l'interface graphique via l'objet *FigureCanvasTkAgg*. Mais l'image ne se mettait pas à jour. Nous avons résolu le problème en créant notre propre fonction *plot* utilisant la traditionnelle *matplotlib.pyplot.plot*. Nous avons été étonnés de voir que plusieurs personnes sur des forums ont eu recours à cette solution pour remédier aux problèmes de rafraîchissement des canvas *matplotlib FigureCanvasTkAgg*.

Pour l'affichage de la constellation nous avons testé la même méthode, mais en vain : la différence non négligeable est qu'il faut utiliser un objet « axe » de *matplotlib* pour pouvoir utiliser les figures polaires. Ces axes sont très mal gérés par *Tkinter*. Nous avons donc perdu beaucoup de temps sur cette phase. Pour pallier au problème, nous avons décidé de créer la carte du ciel dans un canvas à l'aide de méthodes permettant de dessiner des formes géométriques, et de positionner les points représentant les satellites de la même façon.

Un des points qui nous a pris le plus de temps fut la programmation de l'interface graphique, et donc entre autre l'apprentissage en autodidacte du module *Tkinter*.

IV. Répartition des tâches au sein du groupe et organisation du travail au cours de l'UV

La première des choses que nous devons faire était la recherche bibliographique. Nous nous sommes donc répartis cette première tâche de la façon suivante :

- Sarah s'est occupée du protocole NMEA,
- Joris a travaillé sur les ports série et la norme RS-232,
- Et Sophie et Nicolas ont mené des recherches sur ce qu'est la géodésie, et le fonctionnement d'un GPS plus généralement.

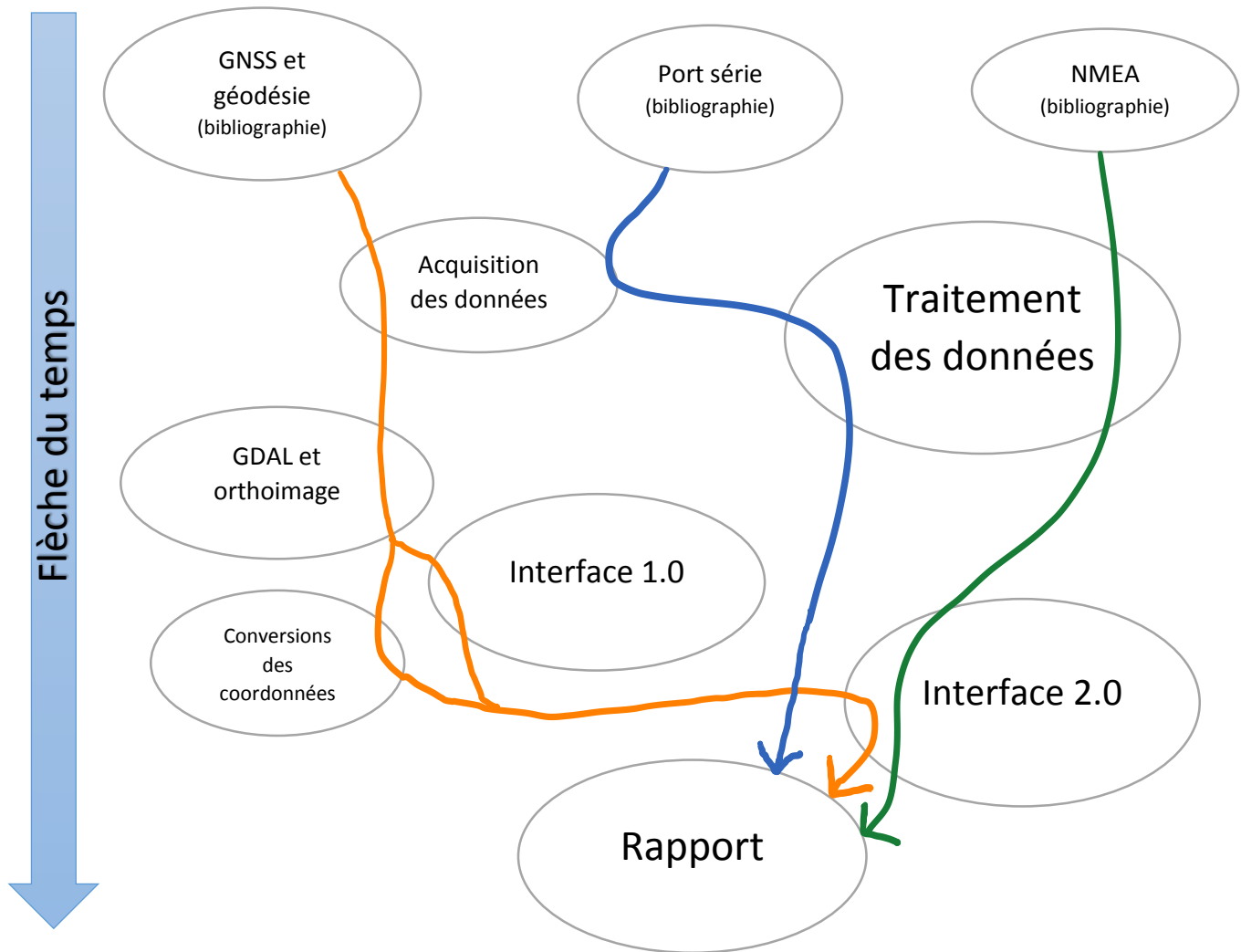
Cette tâche effectuée, nous avons continué le travail sur les points suivants :

- Nicolas et Sophie ont commencé à chercher un script Python susceptible de réaliser l'acquisition des données depuis le GPS branché en port série. Cette tâche a été confiée ultérieurement à Joris qui l'a mise au point.
- Sarah s'est attaquée au codage du script Python qui traitera les données reçues depuis le GPS pour en tirer uniquement les informations qui nous intéressent.

Une fois le script d'acquisition des données du GPS terminé, Joris a rejoint Sarah pour traiter les données, et Nicolas et Sophie se sont penchés sur l'aspect graphique du programme, et notamment sur comment utiliser l'orthoimage. De longues recherches ont été faites pour traiter celle-ci grâce à la librairie *gdal*. Ceci fait, Nicolas et Sophie ont poursuivi sur les algorithmes de conversion WGS84 → Lambert 93 → pixel de la carte, puis ils sont enfin retournés sur l'interface graphique et plus particulièrement sur le rafraichissement des canvas de *matplotlib*.

Une fois les algorithmes de traitement de données GPS réalisés, Joris s'est concentré sur un algorithme de vérification du checksum. Après moult recherches, il a compris le principe et a pu rajouter les quelques lignes de codes qu'il a écrit au programme pour que seuls les trames correctes soient traitées.

Sur la fin du projet, Nicolas et Joris se sont rejoints pour coupler l'acquisition et le traitement des données avec le script principal contenant l'interface *Tkinter*. Sarah a amorcé et a bien avancé le script d'une nouvelle interface munie d'onglets, qui a largement été remanié par Sophie et Nicolas pour donner ce qui est aujourd'hui le script principal final.



Organigramme de la répartition des tâches

Sophie et Nicolas

Joris

Sarah

La taille des bulles est proportionnelle au temps qui y a été consacré

V. Un aperçu des résultats obtenus à l'issue de ce projet

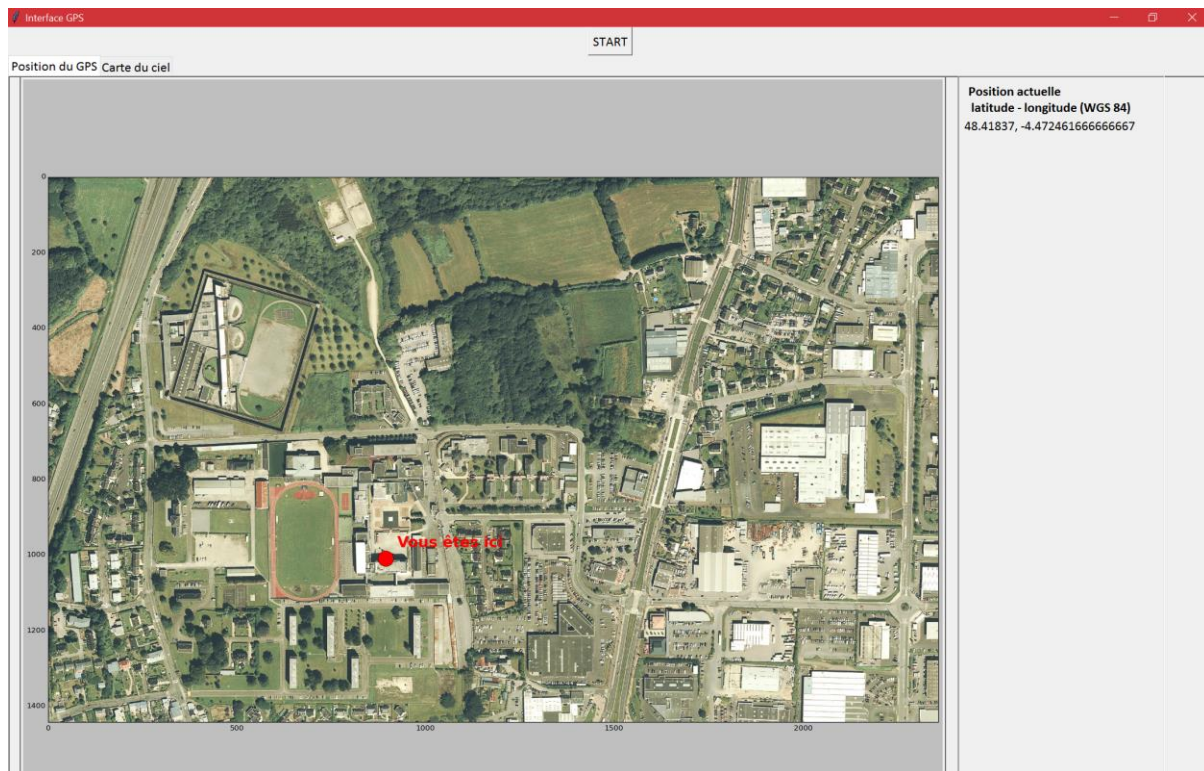


Figure 6: Capture d'écran de notre application en fonctionnement, lorsque l'on affiche la position du GPS sur l'orthoimage.

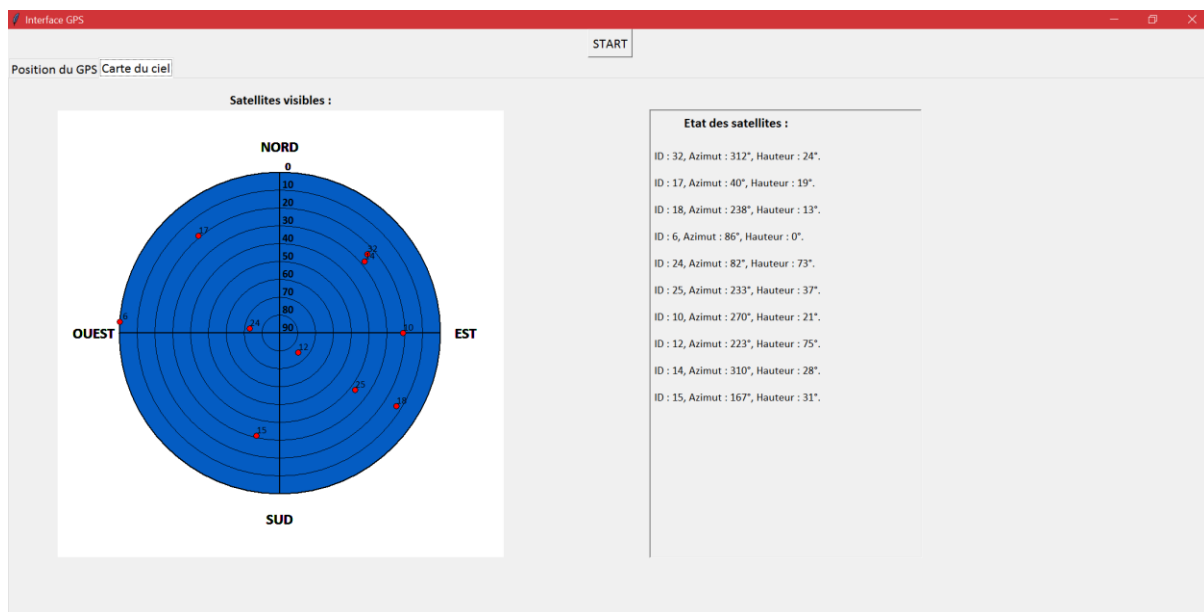


Figure 7: Capture d'écran de notre application en fonctionnement, lorsque l'on affiche la carte du ciel avec l'état des différents satellites en vue.

```

satellite 13 : 8°, 143° (élévation, azimut) 03/25/16 15:40:51
satellite 17 : 24°, 47° (élévation, azimut) 03/25/16 15:40:51
satellite 18 : 20°, 245° (élévation, azimut) 03/25/16 15:40:51
position gps : 48.41836333333333, -4.472441666666667 (WGS 84) 03/25/16 15:40:51
position gps : 48.41836333333333, -4.472441666666667 (WGS 84) 03/25/16 15:40:51
position gps : 48.41836333333333, -4.472441666666667 (WGS 84) 03/25/16 15:40:52
position gps : 48.41836333333333, -4.472441666666667 (WGS 84) 03/25/16 15:40:52
position gps : 48.41836333333333, -4.472441666666667 (WGS 84) 03/25/16 15:40:52

##### Nouvelle session : 03/25/16 15:44:03 #####

position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:07
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:08
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:09
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:09
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:09
satellite 24 : 79°, 58° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 12 : 66°, 218° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 15 : 39°, 167° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 25 : 29°, 229° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 10 : 24°, 278° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 17 : 23°, 46° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 32 : 17°, 314° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 13 : 7°, 144° ; (élévation, azimut) ; 03/25/16 15:44:10
satellite 18 : 19°, 244° ; (élévation, azimut) ; 03/25/16 15:44:10
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:10
position gps : 48.418348333333334, -4.472473333333333 ; (WGS 84) ; 03/25/16 15:44:11

```

Figure 8: Capture d'écran du fichier enregistré suite à une session sur notre application. Il contient l'historique des données reçues.

Conclusion

Finalement, le programme développé tout au long de ce projet remplit les objectifs fixés au départ : il affiche, dans une interface graphique, la position du récepteur sur une carte, et il représente les satellites en vus sur une carte du ciel. Cependant, beaucoup de pistes d'améliorations existent :

- Elargir la carte pour rendre le GPS utilisable ailleurs qu'à l'ENSTA Bretagne ;
- Proposer de tracer l'itinéraire en se servant de l'historique ;
- Afficher l'altitude du récepteur, l'état des satellites ;
- Utiliser un système de double tâches (multi-threading) pour optimiser l'affichage des informations et la réception simultanée des informations provenant du GPS.

Pour conclure, le travail réalisé durant ce projet nous a apporté plusieurs enseignements : nous avons acquis (ou approfondi) des connaissances dans les systèmes de géodésie, dans le fonctionnement d'un GPS, ou encore dans l'utilisation d'un port série. En outre, le travail ayant été réalisé en groupe, nous avons pu profiter des avantages à être plusieurs sur un projet pour avancer plus vite et rendre à la fin un travail complet et abouti.

Bibliographie

- Aurélien Jarno, « Le port série ». https://www.aurel32.net/elec/port_serie.php.
- Jean-Etienne Poirrier, « Port série ». <http://www.poirrier.be/~jean-etienne/notes/structordi/ports/>.
- « RS-232 ». *Wikipédia*, 8 février 2016. <https://fr.wikipedia.org/w/index.php?title=RS-232&oldid=123143261>.
- Glenn Baddeley, « GPS NMEA sentence information », 2001.
- Klaus Betke, « The NMEA 0183 Protocol », mai 2000, modifié août 2001.
- « Les satellites GPS », Consulté le 24 février 2016. <http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/divers/gps2.html>.
- « Communication avec un GPS : Protocole NMEA 0183 | Dossiers Techniques ». Consulté le 5 février 2016. <http://www.dossiers-techniques.fr/geolocalisation/communication-avec-un-gps-protocole-nmea-0183.html>.
- Glenn Baddeley, « GPS - NMEA sentence information », 2001. <http://aprs.gids.nl/nmea/>.
- Peter Bennett. « The NMEA FAQ », 15 septembre 1997. <http://www.kh-gps.de/nmea.faq>.
- Mark Roseman, « TkDocs - Tk Tutorial - Basic Widgets », 2007. <http://www.tkdocs.com/tutorial/widgets.html>.
- Jonathan Sawyer, « Python gdal -- Class or function equivalent to 'gdaltransform' », août 2009. <http://lists.osgeo.org/pipermail/gdal-dev/2009-August/021762.html>.
- Lahaye, R., & Ladet, S. (2014). Les principes du positionnement par satellite : GNSS. *Cahier des Techniques de l'INRA*, 918.
- Le RGF93 | Géodésie. (2016, février 7). Consulté 7 février 2016, à l'adresse <http://geodesie.ign.fr/index.php?page=rgf93>
- BRIAL, P., & SHAALAN, C. (2009). INTRODUCTION à la GEODESIE et au GEOPOSITIONNEMENT PAR SATELLITES (GPS). *Centre d'Etudes Alexandrines*, 29.