

Projet IF23
Etude d'un système de géolocalisation

Geoffrey Gaillard et Thomas Jeantet

15 Juin 2016

Table des matières

1	Fonctionnalités du système GPS	3
2	Utilisation du système	4
3	Description fonctionnelle	5
3.1	Bibliothèque GPS	5
3.2	Bibliothèque d’affichage	5
3.3	Bibliothèque SD	5
4	Traitement et interprétation des données	7

Avant-Propos

Depuis la première mise en service à la fin des années 60, les systèmes de positionnement par satellites n'ont cessé d'être améliorés. Leur utilisation a grandement facilité la navigation, qu'elle soit terrestre, maritime, aérienne ou parfois même spatiale. Dans le cadre de ce projet, nous avons pu programmer un système GPS embarqué et étudier la précision de ce système de positionnement. Ce système GPS est constitué d'une Arduino Micro Pro, d'un récepteur GPS, d'un écran LCD, d'une carte mémoire micro SD et de 4 boutons. L'intégralité du code source est disponible à l'adresse suivante : <https://github.com/Vanell/IF23-Project>.

1 Fonctionnalités du système GPS

Notre programme permet au boîtier, outre de récupérer les données GPS, d'afficher sur l'écran LCD les informations suivantes :

- Latitude
- Longitude
- Nombre de satellites visibles
- HDOP
- Altitude
- Vitesse
- Date
- Heure GPS
- Tension de la batterie
- Autonomie en heures
- Pourcentage de charge de la batterie

De plus, l'utilisateur peut accéder à un menu pour prendre un point ou lancer la prise d'itinéraire. En parallèle de la programmation du système GPS, nous avons développé un programme en python qui permet de :

- Récupérer les données enregistrées sur la carte SD via le port série
- Mettre les données en forme et leur appliquer une correction Lambert 93
- Calculer les précisions relatives à une mesure
- Gérer les fichiers présents sur la carte SD
- Exporter les données en format .csv et en .gpx

2 Utilisation du système

L'interface utilisateur se découpe en deux parties : l'affichage des données GPS en temps réel, et le menu de prise de points. Les deux boutons de gauche servent à naviguer dans les affichages, tandis que les deux boutons de droite servent à passer d'un type d'affichage à l'autre. L'utilisation des quatre boutons, de gauche à droite, est :

Monter dans le menu, descendre dans le menu, affichage des informations relatives à l'autonomie ou retour au données temps réel, ouverture du menu de prise de points ou validation.

Pour une vision plus claire de l'interface, voir le diagramme de cas d'utilisations en figure 1.

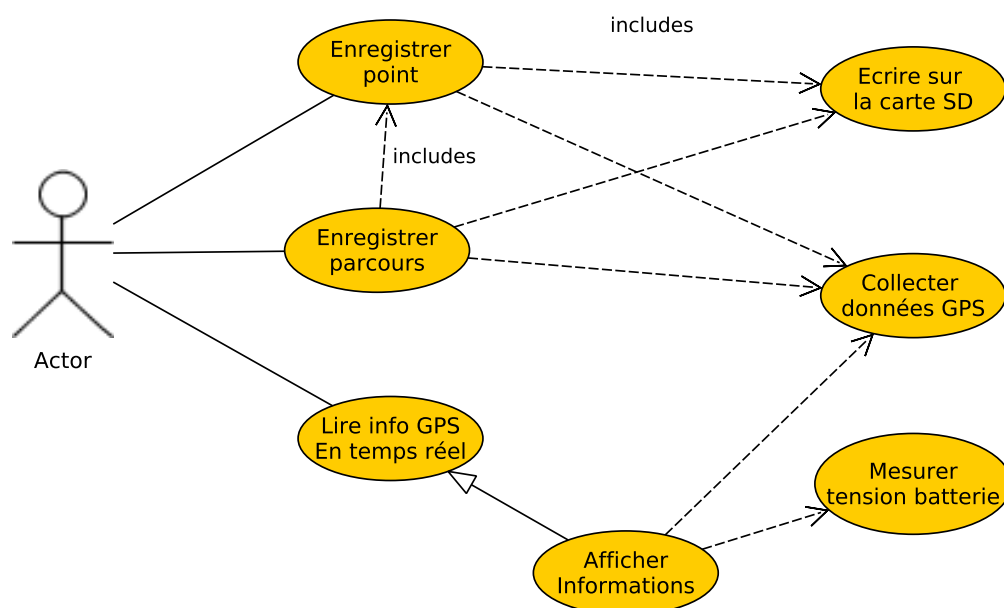


FIGURE 1 – Diagramme de cas d'utilisation

3 Description fonctionnelle

Pour nous faciliter le travail de développement, nous avons utilisé des bibliothèques de fonctions déjà existantes, dont voici la liste exhaustive :

- TinyGPS, bibliothèque de traitement de données brutes GPS
- SoftwareSerial, pour effectuer la liaison serial avec le recepteur GPS
- avr/pgmspace, permettant de stocker des variables dans la mémoire flash de l'arduino
- Bounce2, pour éviter de capter les rebonds des boutons poussoir
- LiquidCrystal, facilitant l'utilisation de l'écran LCD
- SD, bibliothèque native d'Arduino permettant de créer des objets "File" et de les enregistrer

Lors du développement, nous avons décidé de découper notre programme en plusieurs "bibliothèques" de fonctions Arduino, chacune orientée sur un aspect du programme du système.

3.1 Bibliothèque GPS

Cette bibliothèque de fonctions nous permet de définir les variables globales liées au GPS (comme un booléen indiquant si les données GPS ont changé), ainsi que la fonction de mise à jour des coordonnées GPS. Cette dernière récupère toutes les données fournies par la bibliothèque TinyGPS, et les stocke dans un tableau.

3.2 Bibliothèque d'affichage

L'affichage sur l'écran LCD est géré par trois fonctions : la première permet de détecter l'appui sur un bouton, et détermine quel bouton a été pressé. La deuxième incrémente ou décrémente des valeurs dans un tableau pour savoir quelles données afficher. Enfin, la troisième fonction lit les valeurs dans ce tableau, et met les données de l'écran à jour. C'est également dans cette fonction que l'on pourra décider de prendre un point ou un itinéraire.

3.3 Bibliothèque SD

A partir de la bibliothèque native d'Arduino permettant d'écrire et lire sur une carte SD, nous avons codé des fonctions permettant d'écrire toutes les données GPS instantanées sur un fichier ou de lire un fichier et d'écrire son contenu sur le port série. Nous avons aussi prévu de lister les fichiers présent sur la carte SD, mais cette option n'a pas pu être implémentée faute de place sur l'Arduino.

En figure 2 se trouve l'algorithme de la fonction *loop*. Les algorithmes des bibliothèques que nous avons développé ne sont pas détaillées ici, car il sont relativement simples et ne présentent pas grand intérêt.

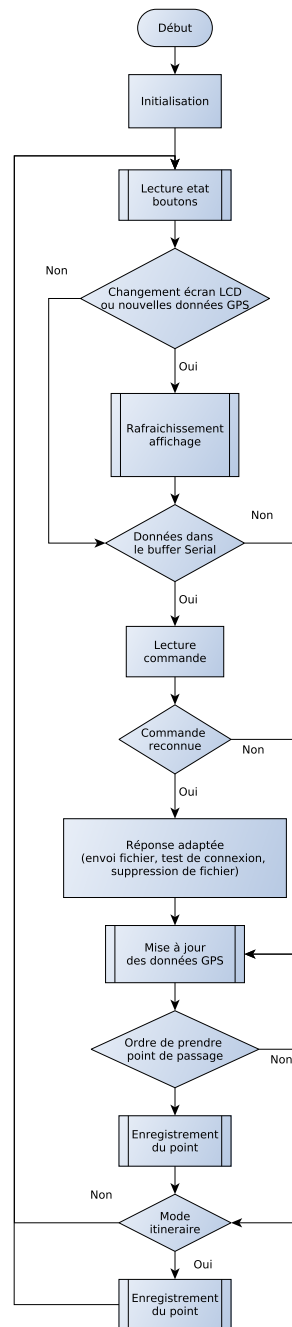


FIGURE 2 – Boucle principale

4 Traitement et interprétation des données

Nous avons effectué 4 séries de mesures utilisables, chacune dans un lieu différent. Le choix de ces endroits nous a semblé crucial car les mesures prises nous permettront d'estimer les paramètres influençant la précision du GPS. Pour mieux comprendre l'interprétation des résultats, voici une brève description des lieux :

- Appartement au 1er étage d'un immeuble : Batiments en béton autour, prise de points à la fenêtre. Mesure de nuit.
- Terrasse de la salle associative de l'UTT : Dans un recoin, entre deux murs métalliques de l'UTT. Météo ensoleillée.
- Voiture à l'arrêt dans le parking de l'UTT : environnement métallique moins important que la terrasse. Météo ensoleillée.
- Colline dégagée. Météo ensoleillée puis orageuse.

Observons tout d'abord la variance et l'écart type des mesures dans un repère local (figures 3, 4, 5 et 6). Le premier critère frappant est les valeurs énormes des écarts types sur y et x lors des mesures en terrasse. Ceci peut s'expliquer assez facilement par l'environnement autour du récepteur : Ayant des surfaces métalliques de chaque côté, le signal GPS peut rebondir un grand nombre de fois tout comme un faible nombre de fois, altérant grandement la précision. C'est ce phénomène qui peut également expliquer les variances élevées lors de la prise de points dans l'appartement, avec une moins grande influence car le béton réfléchit moins les ondes électromagnétiques que le métal.

Un deuxième résultat surprenant est la variance en hauteur de la mesure sur la colline. Nous nous attendions à obtenir les variances les plus faibles sur cette mesure. Cependant une justification est possible : Au cours de la mesure, qui a duré approximativement 8 heures, il s'est mit à pleuvoir fortement. Une intempérie de ce type a sûrement modifié drastiquement et durablement la vitesse de propagation du signal GPS et ainsi causé cet écart sur la mesure d'altitude. Cependant, les variances Nord-Sud et est-ouest semblent peu affectées par les intempéries puisqu'elles sont proches de celles de la mesure dans la voiture.

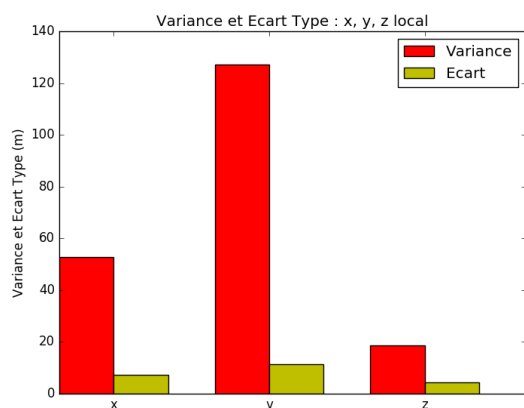


FIGURE 3 – Appartement

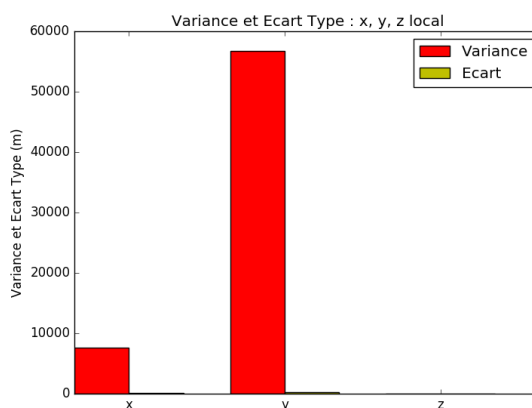


FIGURE 4 – Terrasse

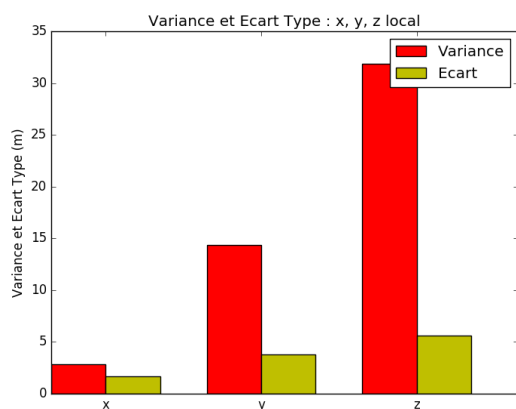


FIGURE 5 – Voiture

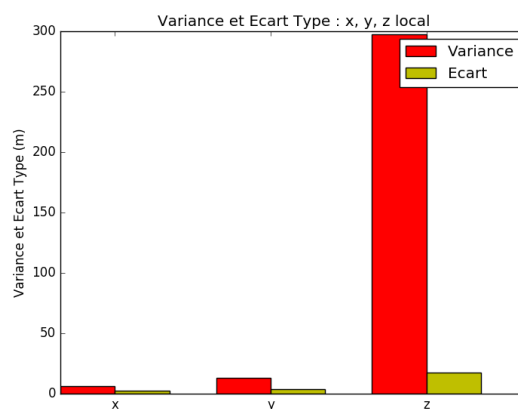


FIGURE 6 – Colline

Pour mettre en valeur les effets d'avoir plus de 4 satellites pour affiner la position, nous avons calculé les écarts types de chaque mesure en fonction du nombre de satellites visibles (figures 7 à 10). En observant les graphiques en figure 7 et 9, il semble en effet qu'un plus grand nombre de satellites visibles permet d'améliorer la précision. Les mesures représentées en figures 8 et 10 en revanche ne montrent pas de résultats flagrants de ce côté là. Cependant, comme dit plus haut, ces mesures ont pu être faussées par les intempéries ou le milieu extérieur.

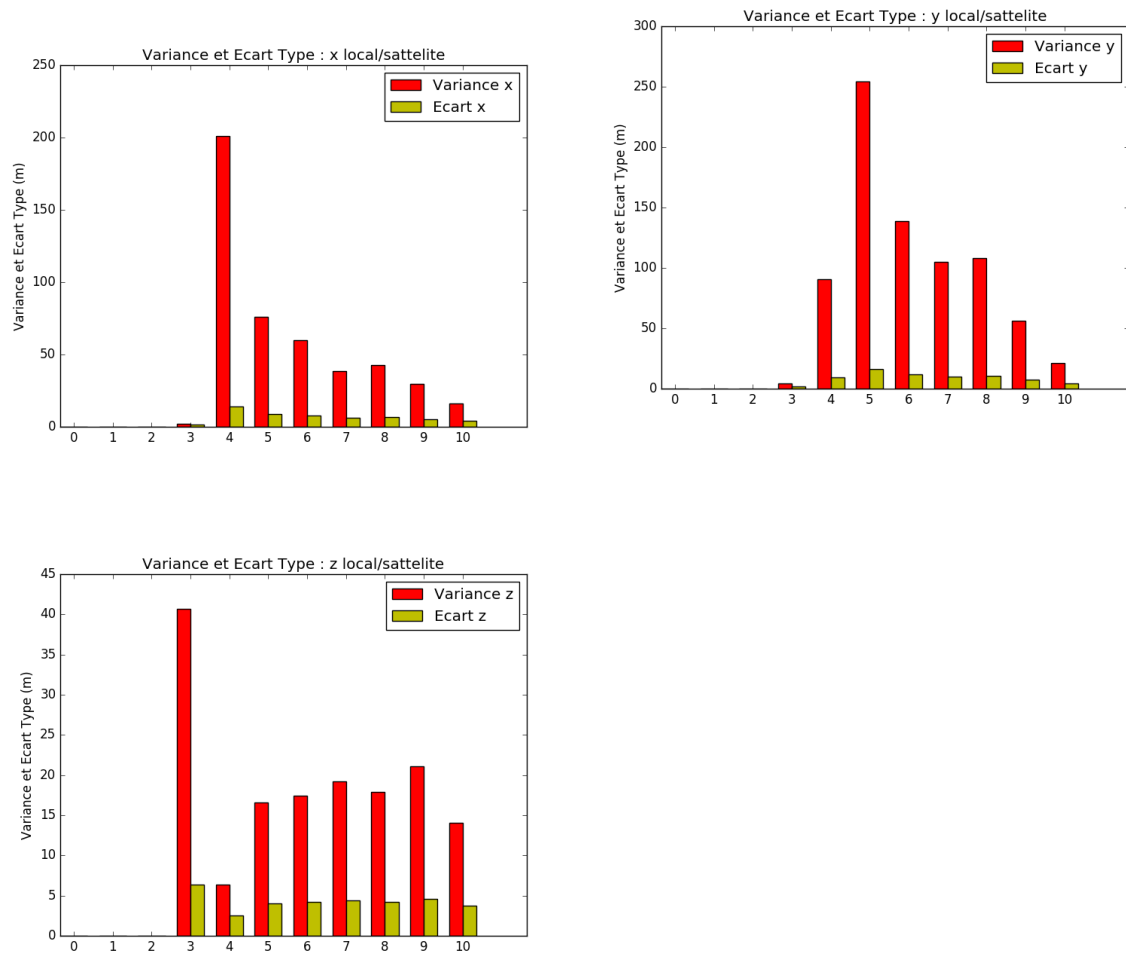


FIGURE 7 – Variances et ecart types sur x, y et z, mesure en appartement

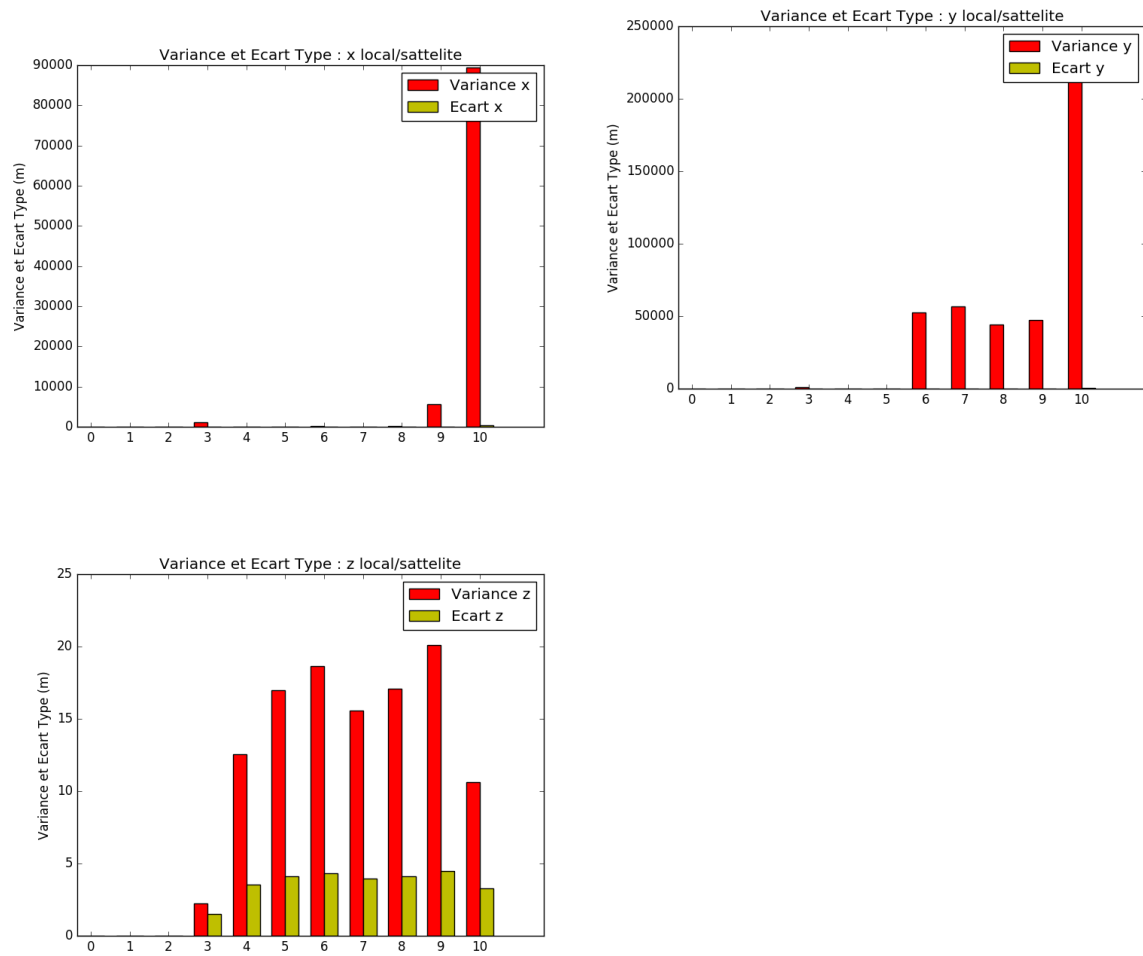


FIGURE 8 – Variances et ecart types sur x, y et z, mesure sur la terrasse

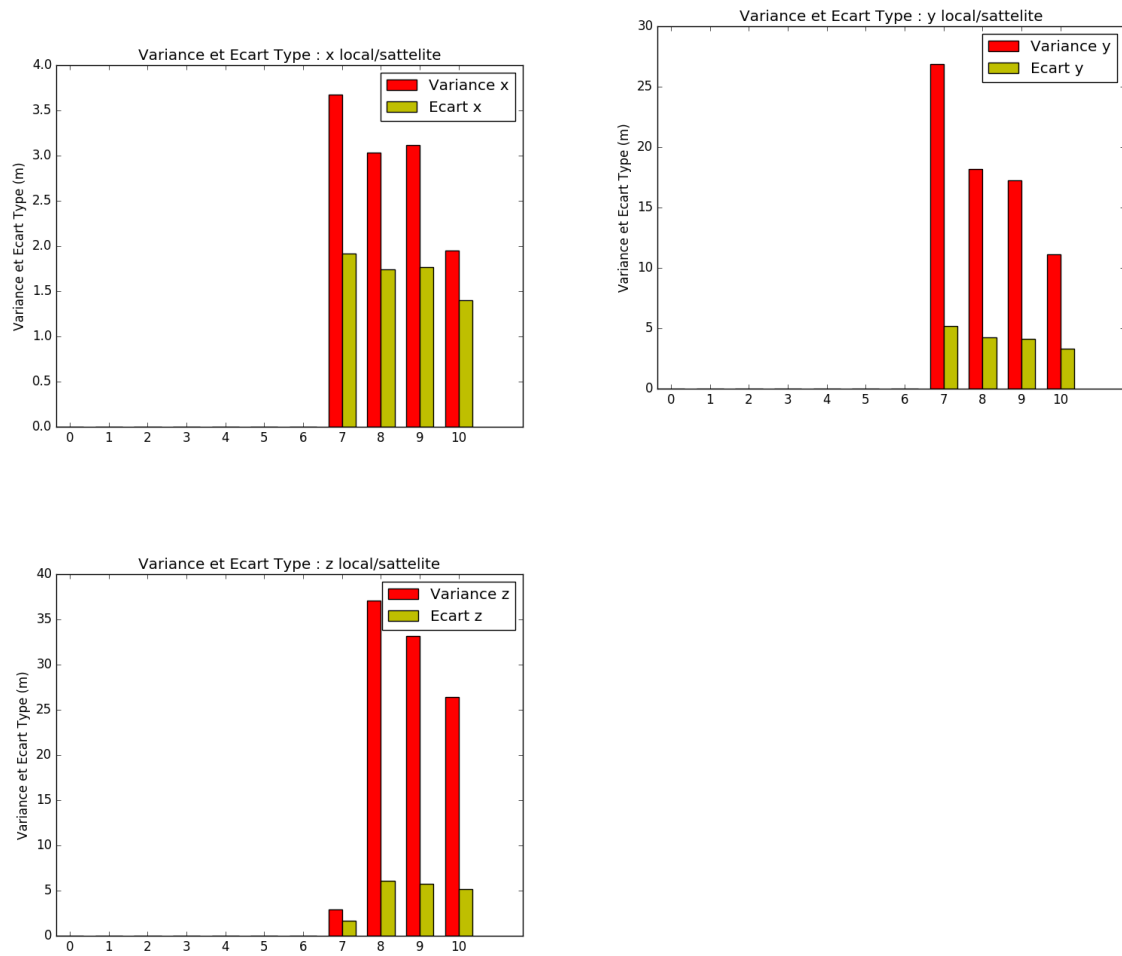


FIGURE 9 – Variances et ecart types sur x, y et z, mesure dans la voiture

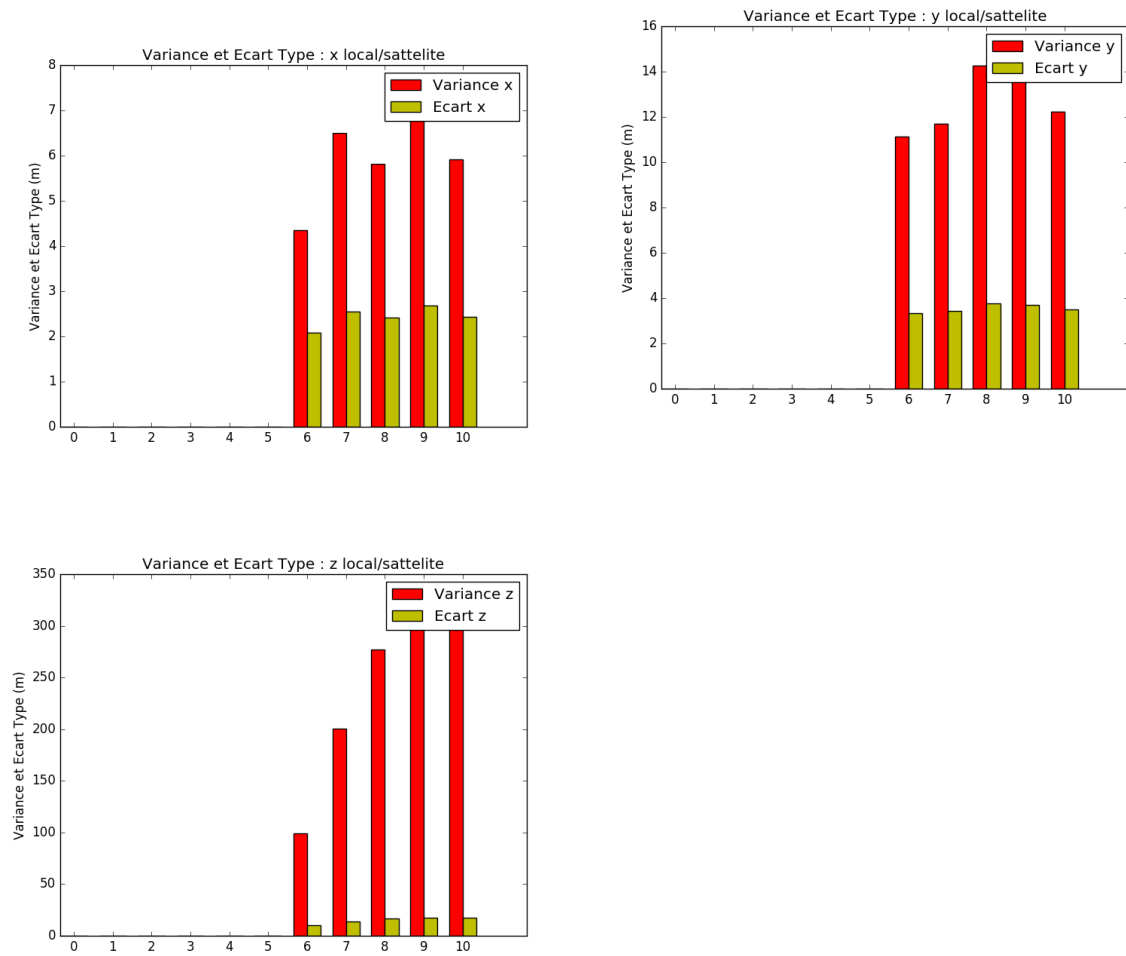


FIGURE 10 – Variances et écart types sur x, y et z, mesure sur la colline

Enfin, pour essayer de corriger les mesures de notre récepteur, nous nous sommes intéressés à la correction dite “Lambert 93”, qui est une approximation non pas plane mais cônica de la Terre, et qui permet de corriger la latitude. Cette correction a des effets sur toutes les coordonnées cartésiennes, et donc sur les coordonnées locales, car la latitude est utilisée pour la conversion de chaque composante des coordonnées géographiques en coordonnées géodésiques. Pour calculer cette correction, nous avons utilisé un algorithme proposé par l’IGN (Institut National de l’Information Géographique et Forestière). Une fois la correction de Lambert appliquée, on obtient les graphes en figures 11 à 14. On peut se permettre de mettre en doute l’exactitude de ces résultats cependant, car une précision aussi importante semble très improbable. Malheureusement nous n’avons pas réussi à trouver d’autres algorithmes exploitables pour effectuer une correction “Lambert 93”.

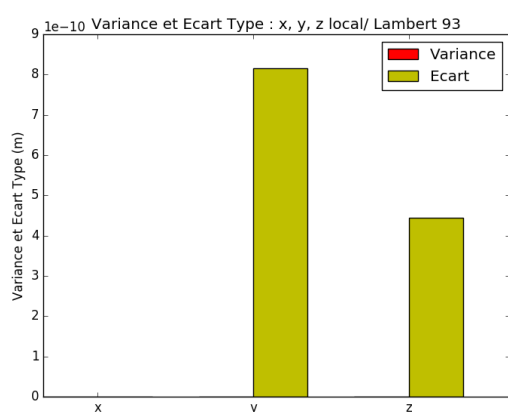


FIGURE 11 – Appartement

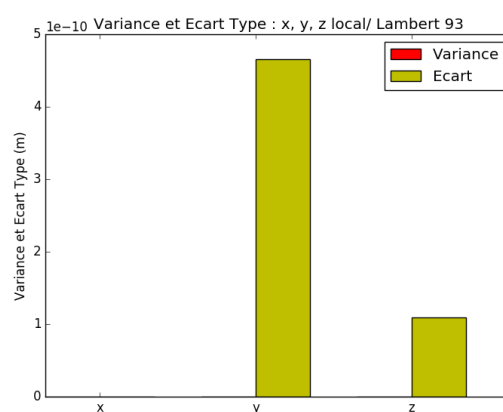


FIGURE 12 – Terrasse

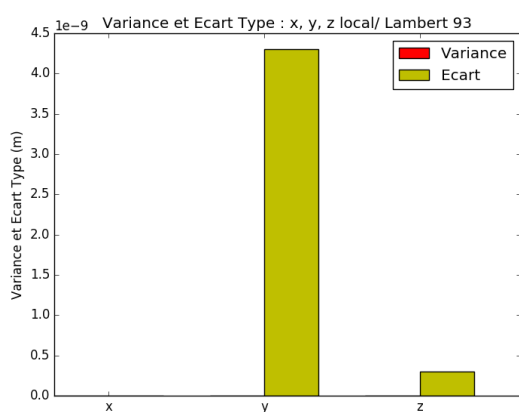


FIGURE 13 – Voiture

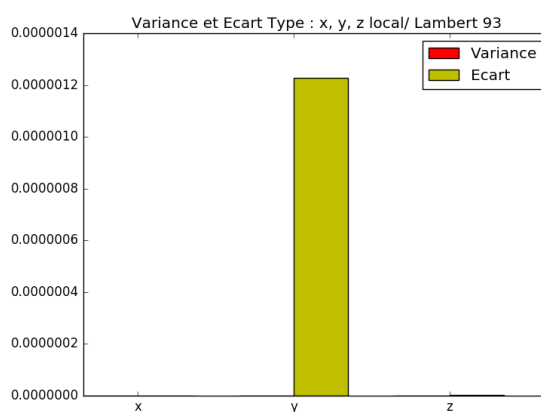


FIGURE 14 – Colline

Conclusion

Au cours de ce projet nous avons pu constater les problématiques de la programmation embarquée, notamment concernant les ressources disponibles, ainsi qu'estimer les effets (parfois surprenants) de l'environnement sur un récepteur GPS. Il nous a également permis de chercher par nous-même des méthodes pour améliorer cette précision, malgré le manque de cohérence de celui que nous avons implémenté.