

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 60, les systèmes de positionnement par satellite 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.

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 aux données temps réel, ouverture du menu de prise de points ou validation.

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

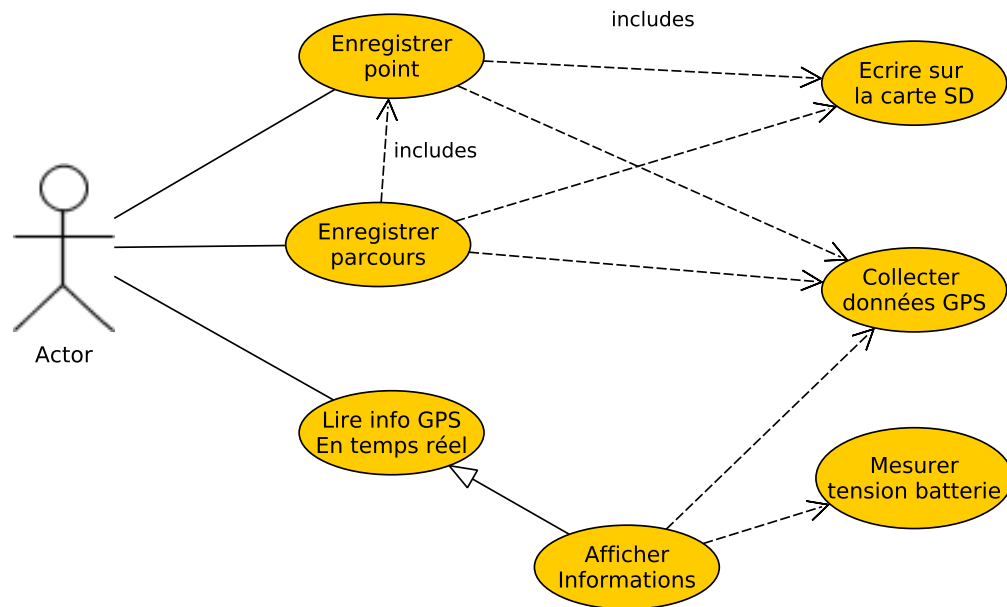


FIGURE 1 – Diagramme de cas d'utilisation

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

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 présents 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. Si toutefois la curiosité poussait le lecteur à vouloir en savoir plus, le code source est disponible à l'adresse suivante : <https://github.com/Vanell/IF23-Project>.

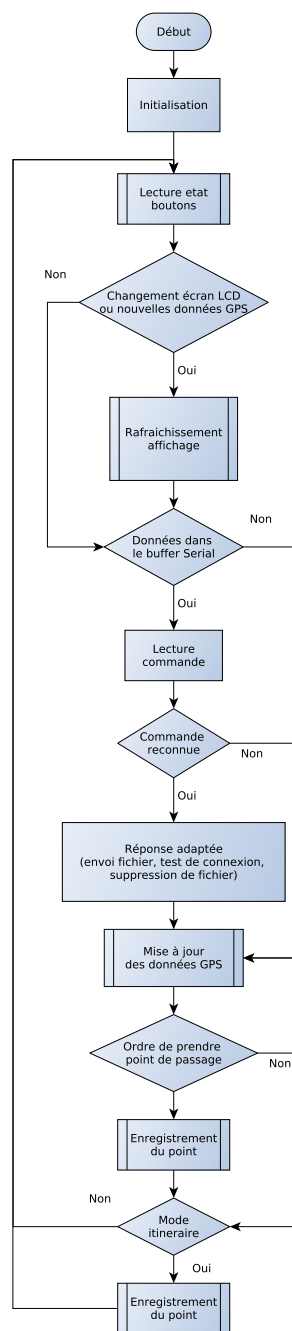


FIGURE 2 – Boucle principale

4 Traitement et interprétation des données

Conclusion