Thème Géolocalisation

Le protocole NMEA

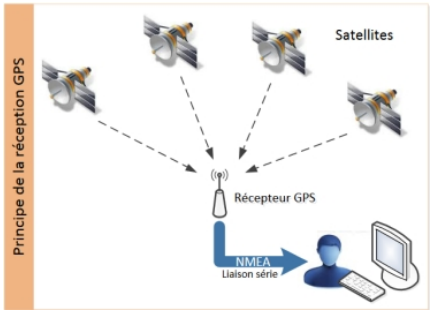
Nous avons utilisé Géoportail et OpenStreetMap. Mais comment le logiciel de carte numérique connaît-il les positions GPS ?

En informatique un protocole correspond à un ensemble de normes permettant à différents périphériques informatique de dialoguer entre eux en réseau. Les Normes de ce protocole sont définies et contrôlées par la National Marine Electronics Association (NMEA), association américaine de fabricants d'appareils électroniques maritimes.

Il faut distinguer deux Normes NMEA :

- le NMEA 0183 qui a l'avantage d'être « lisible » mais qui est basée sur une communication série assez lente, peu performante et plus complexe à mettre en réseau.
- Une plus récente le NMEA 2000 qui n'est pas directement « lisible » mais qui est basée sur un vrai réseau basé sur un « CAN bus » principalement issu de l'industrie des transports, beaucoup plus rapide et simple à installer en réseau

Une trame NMEA est donc une suite de caractères contenant des informations de géolocalisation comme :



- La latitude, la longitude.
- L'altitude
- Le nombre de satellites
- L'heure, la date…

Il existe plus d'une trentaine de trames GPS différentes. Le type d'équipement est défini par les deux caractères qui suivent le $. Le type de trame est défini par les caractères suivants jusqu'à la virgule.

Ex : $GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,,0000*0E

Les deux premiers caractères après le signe $ identifient l'origine du signal. Les principaux préfixes sont :

- BD ou GB - Beidou
- GA - Galileo ;
- GP - GPS ;
- GL - GLONASS.

Le préfixe GN est utilisé dans le cas de signaux mixés GPS + GLONASS.
Chaque trame a sa syntaxe propre, mais selon le cas elles peuvent ou doivent se terminer, après le *, par un système de contrôle qui permet de vérifier que la trame n'a pas été endommagée avant sa réception. La trame GGA est très courante car elle fait partie de celles qui sont utilisées pour connaître la position courante du récepteur GPS.

Traduction de la trame :

- **$ GPGGA** : Type de trame
- **064036.289** : Trame envoyée à 06 h 40 min 36 s 289 (heure UTC)
- **4836.5375,N** : Latitude Nord :48°36.5375' (DDM) = 48,608958° (DD) = 48°36'32.25" (DMS)
- **00740.9373,E** : Longitude Est :7°40.9373'(DDM)= 7,682288° (DD) = 7°40'56.238 " (DMS)
- **1** : Type de positionnement (le 1 est un positionnement GPS)
- **04** : Nombre de satellites utilisés pour calculer les coordonnées
- **3.2** : Précision horizontale ou HDOP (Horizontal dilution of precision)
- **200.2,M** : Altitude 200,2, en mètres
- **,,,,,0000** : D'autres informations peuvent être inscrites dans ces champs
- ***0E** : Somme de contrôle de parité, un simple ou exclusif sur les caractères entre $ et *OE

→A l'aide de Python nous allons tester quelques trame GPS :
$GPGGA,145702.00,5545.0900,N,03737.2311,E,1,07,0.19,122,M,,,,0000*0E
$GPGGA,145802.28,2224.0335,S,14302.3136,E,1,04,2.2,477,M,,,,0000*0E
$GPGGA,143102.05,5112.06711,N,10227.6746,O,1,04,3.1,500,M,,,,0000*0E
Voici le script à modifier à loisir pour déterminer les villes localisées par les trames :

```
# Code à utiliser avec le langage Python
    import webbrowser
     zoom='16' # Faire des essais avec différentes valeurs
     latitude=55.75025 # Expliquer comment vous trouvez  ce résultat
     longitude=37.6205 # Expliquer comment vous trouvez  ce résultat
     webbrowser.open("https://www.openstreetmap.org/#map=" + zoom + " / " + str ( latitude )
      + "/ " + str ( longitude))        # Observer dans l'URL les paramètres
```

Déterminer les latitudes et longitudes en DMS, puis exécuter le script pour chaque trame. Quels sont les lieux donnés par les trames ?

→Déterminez le lieu ou la photo a été prise :

La liste des données Exif (Exchangeable Image file format) intégrées par vos appareils photos numériques regroupe foule d'informations parmi lesquelles : La date et l'heure à laquelle la photo a été prise, les coordonnées GPS….

Enregistrer la photo sur votre ordinateur puis trouver les informations de la photo (clic droit détails). Où a été prise la photo?
C'est grâce à ce genre d'informations que des personnes ayant commis des actes délictueux (comme par exemple photographier les sujets du Bac 2019 !) ont pu être retrouvées.

CALCULS D'ITINERAIRES ET GRAPHES – PYTHON

Itinéraire : chemin à suivre pour se rendre d'un point géographique A à un point géographique B.

Comme vous avez pu le constater quand vous avez travaillé sur Open Street Map, il est possible de définir les voies de communication (les routes). La base de données OpenStreetMap contient donc les données des routes répertoriées à ce jour. En utilisant ces données, il est possible de développer des outils capables de calculer des itinéraires routiers.

Il suffit de renseigner votre lieu de départ, votre lieu d'arrivée puis l'application web calcule votre itinéraire.
Ce calcul d'itinéraire repose sur des algorithmes, par exemple l'algorithme de Dijkstra qui permet d'obtenir le plus court chemin entre deux points.

Nous avons réalisé un script Python qui permet de construire un graphe et de déterminer le plus court chemin avec l'algorithme de Dijsktra :

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import pylab

G = nx.Graph()

G.add_edges_from([('A', 'B'),('E','H'),('H','I'), ('F','I'), ('G','I')], weight=40)
G.add_edges_from([('A','C')], weight=75)
G.add_edges_from([('C','F'), ('B','E')], weight=100)
G.add_edges_from([('A','D')], weight=90)
G.add_edges_from([('D','G')], weight=120)

edge_labels=dict([((u,v,),d['weight'])
                  for u,v,d in G.edges(data=True)])

pos=nx.circular_layout(G)
#commande pour faire des arêtes avec indication de poids
# commande qui utilise pylab
nx.draw_networkx_edge_labels(G,pos,edge_labels=edge_labels)
#commande pour tracer le graphe
nx.draw(G,pos, node_color = 'yellow',with_labels=True, node_size=1500,edge_color='black',edge_cmap=plt.cm.Reds)
pylab.show()

print("Le trajet le plus court",(nx.dijkstra_path_length(G,'A','I')))
print("Le trajet le plus court",(nx.dijkstra_path(G,'A','I')))
```

Vous voulez vous rendre à la plage de Sainte Marie Du Lac en partant de Saint-Dizier en Bus.
La tableau suivant donne les liaisons possibles avec leur temps de parcours et leur distances :

Liaison	Distance	Temps
Saint-Dizier/Hallignicourt	6km	9 min
Hallignicourt/Ambrières	4 km	6 min
Ambrières/Landricourt	7 km	9 min
Landricourt/Sainte Marie du lac	8km	14 min
Saint-Dizier/Valcourt	4,5 km	7min
Valcourt/Eclaron	5 km	7 min
Eclaron/Sainte-Livière	3,5km	4 min
Sainte-Livière/Sainte Marie Du lac	8,9 km	13 min
Laneuville au pont/Ambrières	1,8 km	2 min

En adaptant le script précédent :
1. Déterminer le parcours le plus rapide
2. Déterminer le parcours le moins long