

Transmission des coordonnées GPS : trame NMEA

Cette activité s'inscrit dans la thématique **Localisation, cartographie et mobilité** du programme de SNT.

Elle est proposée par Luc Vincent, actuellement professeur de Physique-Chimie et d'ISN/ICN au lycée des Graves à Bordeaux.

Capacités attendues

- Décoder une trame NMEA pour trouver des coordonnées géographiques.

Introduction

Les différents composants d'un appareil électronique (ex : un téléphone mobile) communiquent par des protocoles normalisés. Ainsi, les puces GPS qui effectuent les calculs de positionnement envoient leurs résultats présentés suivant une trame normalisée : la trame NMEA 0183. Le développeur d'un dispositif souhaitant interroger un GPS sait qu'il pourra exploiter cette trame pour obtenir des informations de date et de position.

La norme NMEA 0183

La norme NMEA est le protocole de transmission des données GPS. Ces données sont transmises sous la forme de trames. Chaque trame commence par le caractère \$ et se compose de plusieurs éléments séparés par des virgules. Voici un exemple de trame :

```
$GPGGA, 064036.289, 4836.5375, N, 00740. 9373, E, 1, 04, 3.2, 200.2, M, , , ,  
0000*0E
```

Les deux premiers caractères correspondent à l'identifiant du récepteur : ici GP pour Global Positioning System. Les trois lettres suivantes correspondent à l'identifiant de la trame : GGA pour GPS Fix et Date. C'est la trame la plus courante.

Éléments de la trame GGA

Décomposons maintenant cette trame selon les premiers éléments qui la composent :

- **GP****GGA** : type de la trame
- **064036.289** : heure d'envoi de la trame, ici 06h 40min 36,289s (UTC)
- **4836.5375, N** : latitude Nord, ici 48°36,5375' (en DM, degrés minutes)
- **00740.9373, E** : longitude Est, ici 7°40,9373' (en DM également)
- **1** : type de positionnement (1 pour le positionnement GPS)
- **04** : nombre de satellites utilisés
- **3.2** : précision horizontale
- **200.2, M** : altitude, ici 200 mètres

Les notations DMS, DM et DD

Généralement, on exprime les coordonnées géographiques dans le système sexagésimal, noté DMS pour degrés, minutes, secondes. Par exemple 49°30'30" pour 49 degrés, 30 minutes et 30 secondes. Une minute d'angle vaut 1/60 degrés tandis qu'une seconde d'angle vaut 1/3600 degrés.

Il est également possible d'utiliser les unités DM (Degré Minute) ou DD (Degré décimal) :

- En DMS : 49°30'30"
- En DM : 49°30,5'
- En DD : 49,5083° (généralement avec quatre décimales)

1. Vérifier par un calcul que la latitude 48°36.5375' (DM) de la trame NMEA donnée en exemple en début d'activité correspond à 48°36'32.25" (DMS).

Sachant que $1' = 60''$ alors $0.5375' = 60 \times 0.5375 = 32.25''$.

Extractions des données contenues dans une trame avec Python

Voici une vue des résultats de quelques instructions Python obtenues depuis la console :

```
1 >>> ligne="nom,prenom,age,17"
2 >>> element=ligne.split(",")
3 >>> element
4 ['nom', 'prenom', 'age', '17']
5 >>> type(element)
6 <class 'list'>
7 >>> element[1]
8 'prenom'
9 >>> prenom=element[1]
10 >>> prenom[2:4]
11 'en'
12 >>> type(element[3])
```

```

13 <class 'str'>
14 >>> int(element[3])
15 17
16 >>> float(element[3])
17 17.0
18 >>>

```

2. D'après ces résultats, quelle instruction en python permet d'obtenir une liste nommée **attribut** à partir d'une chaîne de caractères nommée **trame** ?

```
trame = "$GPGGA, 064036.289, 4836.5375, N, 00740. 9373, E, 1, 04, 3.2,
        200.2, M, , , , 0000*0E"
```

Il faut saisir l'instruction : `attribut=trame.split(",")`.

3. Ecrire la fonction `tramePrefixes(trame)` qui reçoit une trame complète et renvoie l'identifiant du récepteur, c'est-à-dire les deux premières lettres du type de la trame (premier élément après le caractère \$). Sur la trame d'exemple, la fonction doit renvoyer "GP".

On teste ensuite la fonction avec la trame d'exemple :

```

deg PYTHON
>>> from gps import *
>>> tramePrefixes(trame)
'GP'
>>> |

```

```

1 def tramePrefixes(trame):
2     talkerId=trame[1:3]
3     return talkerId

```

4. Modifier cette fonction pour qu'elle renvoie le nom de l'équipement qui a émis la trame. On utilisera les correspondances suivantes :

- BD ou GB : Beidou
- GA : Galileo
- GP : GPS
- GL : GLONASS

On teste alors la fonction :

```
deg PYTHON
>>> from gps import *
>>> tramePrefixes(trame)
'GPS'
>>> |
```

```
1 def tramePrefixes(trame):
2     talkerId=trame[1:3]
3     if talkerId=="GD" or talkerId=="GB":
4         tramePrefixesValue="Beidou"
5     elif talkerId=="GA":
6         tramePrefixesValue="Galileo"
7     elif talkerId=="GP":
8         tramePrefixesValue="GPS"
9     elif talkerId=="GL":
10        tramePrefixesValue="GLONASS"
11    return tramePrefixesValue
```

5. Ecrire une fonction `ggaUtc(trame)` qui reçoit une trame complexe et renvoie l'heure en h, min, s.

```
deg PYTHON
>>> from gps import *
>>> ggaUtc(trame)
06
'06 h 40 min 36.289 s'
>>> |
```

```
1 def ggaUtc(trame):
2     #on transforme la trame en liste
3     attribut=trame.split(',')
4     #on sélectionne l'élément qui correspond à l'heure
5     time=attribut[1]
6     utc=time[:2]+" h "+time[2:4]+" min "+time[4:]+" s"
7     return utc
```

6. Ecrire une fonction `ggaLat(trame)` qui reçoit une trame complète et renvoie la latitude convertie en DMS.

```
deg PYTHON
>>> from gps import *
>>> ggaLat(trame)
'48 D 36 M 32.25 S '
>>> |
```

```
1 def ggaLat(trame):
2     attribut=trame.split(',')
3     lat=attribut[2]
4     lat_deg=lat[:2]
5     lat_min=lat[2:4]
6     lat_sec=str(float(lat[4:])*60)
7     latDMS=lat_deg+" D "+lat_min+" M "+lat_sec+" S "
8     return latDMS
```

A retenir

Pour définir sa position sur la Terre, on utilise le plus souvent les coordonnées géographiques : la latitude, la longitude et l'altitude. Pour la latitude et la longitude on rencontre trois notations :

- Degré Minute Seconde (DMS)
- Degré Minute (DM)
- Degré décimal (DD)

Les récepteurs "GPS" fournissent la localisation sous une forme normalisée facilement décodable, par exemple selon le protocole NMEA 0183 (National Marine Electronics Association).