# TD décodage trame GPS NMEA

## 1 Objectifs

Le but de ce TP est de manipuler par programmation, une trame GPS de type NMEA afin d'en extraire l'information de position et l'heure. Cette trame est transmise sur une liaison série sous forme d'une chaîne de caractères. Ce sera l'occasion idéale d'utiliser les fonctions permettant de manipuler les chaînes de caractères en langage C avec des pointeurs.

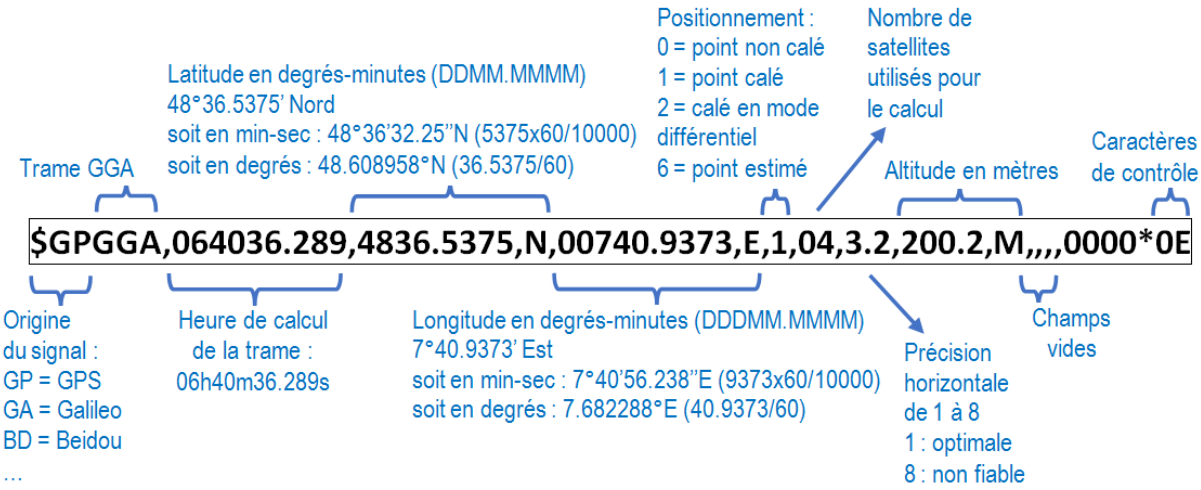
## 2 La réception GPS

Un récepteur GPS est capable de se géolocaliser grâce à la réception de signaux émis par des satellites géostationnaires. Le récepteur GPS détermine par calcul (**trilatération**) sa position. il génère une trame (une ligne de texte) regroupant plusieurs informations comme l'heure, la latitude, la longitude, l'altitude, etc.

Afin que tous les équipements GPS puissent tous être compatible, il faut que cette trame ait toujours la même forme !

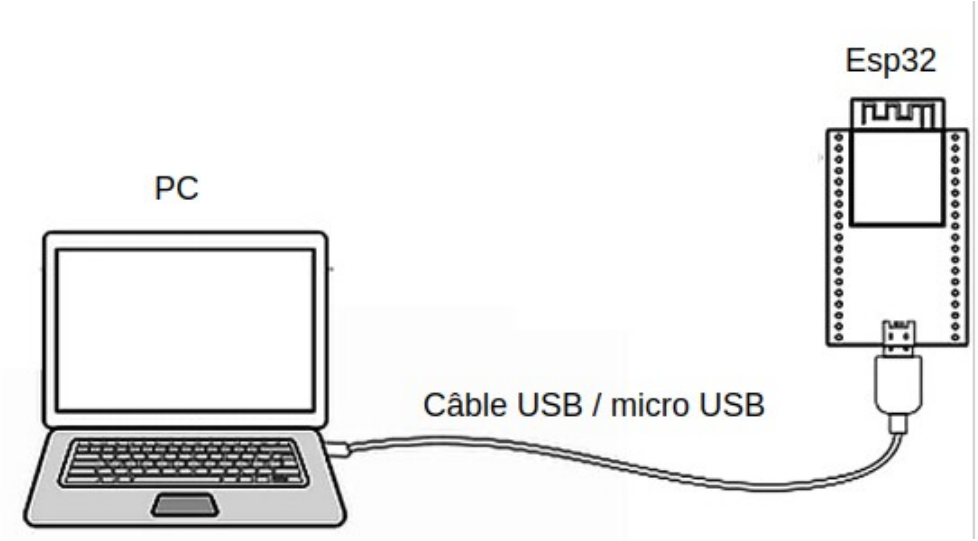
Ainsi, l'association (la NMEA : National Marine Electronics Association) a créé une norme dont la trame la plus utilisée aujourd'hui s'appelle la trame **GGA**,

Voici un exemple pour comprendre à quoi ressemble cette trame. On remarque que chaque donnée est séparée par une virgule et que les valeurs décimales utilisent le point :



# 3 Mise en œuvre du capteur GPS simulé avec l'esp32

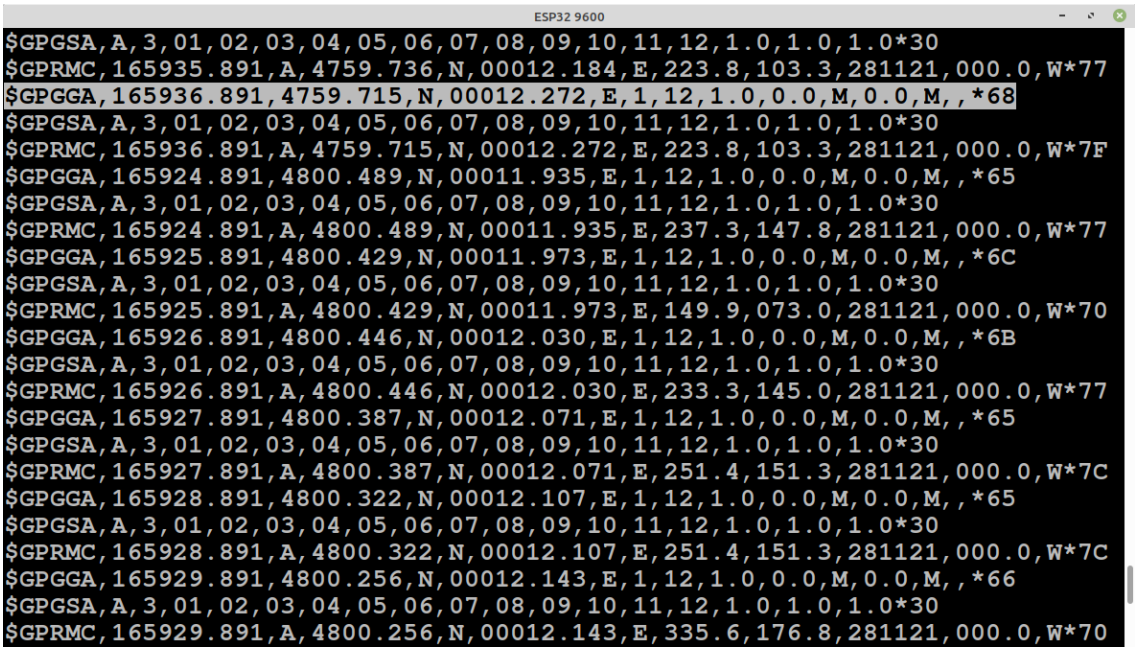
Il est peu pratique de mettre en œuvre un capteur GPS réel en classe car l'antenne du récepteur doit être placé près d'une fenêtre pour être en mesure de capter les signaux émis par les satellites. Aussi pour simplifier les manipulation nous utiliserons un ESP32 avec un programme de simulation NMEA. Le programme émet en continu des trames NMEA sur la liaison série.




# 4 Visualisation des trames émise avec Putty

Configurer Putty en **Serial** avec **9600** bauds **8** bits par caractère **1** bit de stop, pas de parité pas de contrôle de flux.

Envoyer le caractère **n** pour démarrer l'envoi des trames NMEA. Les trames NMEA sont envoyés en continu.



Il existe plusieurs trames correspondant à des besoins différents. Chaque trame possède une syntaxe différente. Nous nous intéresserons à la trame la plus utilisée pour connaître la position courante du récepteur : La trame **GGA**.

Une trame est constituée de **champs**. Les champs **sont séparés entre eux par des virgules**. Un champ peut être vide mais **la présence de la virgule est obligatoire**.

| Nom du champ | Exemple de valeur | Signification |
|---|---|---|
| Type de trame | $GPGGA | Indique qu'il s'agit d'une trame $GP (pour GPS) de type GGA |
| Heure | 064036.289 | Signifie que la trame a été envoyée à 6h40min36.289 s |
| Latitude Ex de valeur | 4836.5375 | Latitude en deg, min, sec. Les secondes doivent être convertit en base 60 : 5375/100*60 = 3225 , soit 32.25 s |
| Indicateur Latitude | N | N : Nord , S : Sud |
| Longitude | 00740.9373 | Longitude en deg,min,sec soit 7°40'56.238'' |
| Indicateur longitude | E | E : Est , W :Ouest |
| Positionnement | 1 | 0 = point non calé, 1 = point calé, 2 = point calé en mode différentiel, 6 point estimé |
| Nb de satellites | 4 | Nombre de satellites utilisés pour le calcul. La précision du positionnement dépend du nombre de satellites détectés |
| Précision | 3.2 | Dilution horizontale de la précision. Permet de connaître la fiabilité du calcul. 1 : Valeur optimale, 2 à 3 : excellente, 5 à 6 : bonne, supérieure à 8 : Mesure non fiable |
| Altitude | 200.2 | Altitude de l'antenne par rapport au niveau de la mer |
| Unité altitude | M | Altitude en mètres |
| Somme de Contrôle | *68 | Somme de contrôle de parité, un simple XOR sur les caractères précédents |

La somme de contrôle à la fin de chaque phrase est le OU EXCLUSIF (XOR) de tous les octets de la phrase à l'exclusion du premier caractère ($) et jusqu'au caractère avant l'étoile (*)

Site officiel : www.nmea.org

# 5 Programmation en langage C

## 5.1 Fonctionnalité du programme

Le programme à réaliser aura les fonctionnalités suivantes :

1. **Découpage de la trame** : Chaque champ de la trame pourra être extrait en fonction de sa position dans la trame. L'index 0 représente le champ n°1 et ainsi de suite
2. **Contrôle de la validité** la somme de contrôle calculée doit être la me que celle reçue.
3. **Affichage de la longitude, la latitude** : La longitude et la latitude devront être affichés sous forme degré décimaux.
4. **Affichage de l'heure d'envoi de la trame** : L'heure devra être affichée sous forme hh:mm:ss

## 5.2 Principe du décodage de trame

le principe du décodage consiste à découper la trame suivant la virgule pour la transformer en un tableau de chaîne de caractères, chaque élément du

Le prototype de la fonction sera le suivant:

```
int decouper(char trame[], char *champs[], int size)
```

**Algorithme proposé**

Fonction **Découper ()**
paramètre d'entrée : **trame** chaîne de caractères
paramètre de sortie : **champs** tableau de chaîne de caractères
paramètre d'entrée : **size** entier la taille maxi du tableau

## Schéma algorithmique

```
début
    ptr ← @trame[0]
    fin ← faux
    i ← 0

    faire
    |    tant que ( *ptr == ' ' )
    |    |    ptr ← ptr + 1
    |    fin tant que
    |
    |    si (*ptr == '\0') alors
    |    |    fin ← vrai
    |    sinon
    |    |    champs[i] ← ptr
    |    |    // parcours la trame à la recherche de la virgule suivante
    |    |    tant que ( *ptr ≠ ',' et *ptr ≠ '\0' )
    |    |    |    ptr ← ptr + 1
    |    |    fin tant que
    |    | // Remplace la virgule par '\0' et passage au champ suivant
    |    |    si ( *ptr ≠ '\0' ) alors
    |    |    |    *ptr ← '\0'
    |    |    |    ptr ← ptr + 1
    |    |    fin si
    |    |
    |    |    i ← i + 1
    |    fin sinon
    |
    tant que ( i < (size - 1) et pas fin)

    champs[i] ← NULL
    retourner i
fin
```

## Variables locales

**ptr** adresse d'un caractère de la trame

**fin** drapeau indiquant que la fin de trame est atteinte.

**i** entier contient le nombre de champ trouvé dans la trame

1. Coder en langage C la fonction **découper**
2. Proposer un algorithme pour calculer la somme de contrôle puis coder votre l'algorithme.
3. Coder une fonction pour vérifier la validité d'une trame.