

TD Communication dans les systèmes informatiques: Réception de trames GPS :

Le système GPS (Global Positioning System) est composé d'une mire de satellites en orbite haute autour de la terre. Ces satellites transmettent des messages sur les bandes de fréquences de 1.57542 GHz (L1 signal) and 1.2276 GHz (L2 signal) à une vitesse de 50 bit/s. Chaque message de 1500 bits contient une information temporelle (date/heure GPS), l'éphéméride (orbite du satellite) et "l'almanach" qui contient des informations de statut du satellite.

Un récepteur GPS à besoin de recevoir l'information en provenance d'un minimum de 3 satellites afin de trianguler sa position sur la surface de la terre. Le récepteur peut alors transmettre à l'application la position calculée ainsi que des informations temporelles (date/heure). Le format le plus répandu pour la transmission de ces données entre le récepteur GPS et un autre appareil est le format NMEA (National Marine Electronic Association) souvent transmis au moyen d'une liaison série asynchrone ASCII.

Le format NMEA définit un ensemble de trames qui contiennent chacune des informations de navigation (positionnement, vitesse, date/heure ...). Une trame NMEA est préfixée par le caractère «\$» suivit du nom de la trame sur 5 caractères (GPRMC, GPGBA, GPRMA ...) et d'un ensemble de données séparés par des virgules et terminée par le caractère saut de ligne ('\n' soit 0x0d en hexadécimal).

La trame **GPRMC** (**G**lobal **P**ositionning **R**ecommended **M**inimum **C**) se décompose comme suit :

\$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh suivi de \n\r

Exemple : \$GPRMC, 225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68\n\r

N°	Nom/Fonction	Exemple
	Délimiteur début de trame '\$' et nom de la trame	\$
1	Heure universelle (UTC) du "fix"	22 :54 :46
2	Statut du récepteur, A si le GPS fournit une information valide (fix), V sinon	A
3	Latitude en degrés/minutes (voir note1).	49°16.45'
4	(N) Nord ou (S) Sud	N (nord)
5	Longitude en degrés/minutes (voir note1).	123°11.12
6	(E) Est ou (W) Ouest	W (Ouest)
7	Vitesse au sol en nœuds	0.5 (nœuds)
8	Cap en degrés	54,7°
9	Date	19novembre 1994
10	Variation magnétique du cap en degrés	20.3°
11	(E) Est ou (W) Ouest	E (Est)
...	Peut contenir d'autres informations	
	Délimiteur fin de trame '*'	*
	Somme de contrôle (XOR de tous les octets transmis entre \$ et * EXCLUS, affichée en hexadécimal sur deux caractères ASCII)	0x68
...	Après la trame, il peut y avoir des caractères supplémentaires, ou d'autres trames	\n\r

Note 1 : Le format utilisé pour la latitude est le suivant (ddmm.mmmmm) avec un nombre de chiffres variables (de 0 à 7) après le point décimal. (dd) code l'angle en degrés (90°max) et

(mm.mmmm) code les minutes en décimal (soixantièmes de degrés). Le format utilisé pour la longitude est le suivant (dddmm.mmmm). (ddd) code l'angle en degrés (180°max)

Note 2 : Le standard NMEA indique qu'il peut y avoir d'autres champs entre le 11° et la somme de contrôle. Nous n'essaierons pas de les décoder ici, mais calculerons la somme de contrôle en les utilisant si ils sont présents.

Les champs entre virgules peuvent ne pas être renseignés par le récepteur. La longueur de cette trame est donc de l'ordre de 75 caractères ASCII en considérant le format fourni. Le récepteur peut transmettre cette trame à la fréquence de 1, 5 ou 10hz avec des vitesses de transmission série allant de 9600bps à 115 200 bps.

Objectif de l'exercice : On souhaite réaliser un programme permettant de décoder à la volée (sans stockage, aussi appelée stream parsing) les trames GPRMC d'un module récepteur GPS relié à une carte microcontrôleur via un port de communication série asynchrone à une fréquence d'une trame par seconde, au format 8N1 9600 Bds en utilisant le moins de mémoire RAM possible (~20 octets) pour le stockage des données. Le programme est structuré comme suit :

-une tâche de fond ou boucle principale,

-une fonction de gestion d'interruption pour la réception des données sur la liaison série, qui écrit dans une FIFO de taille 5 octets,

- une fonction d'affichage appelée si la trame reçue est valide

La fonction de parsing des données GPS est appelée à chaque exécution de la boucle principale. Cette fonction consomme les données depuis la FIFO (des caractères ascii) et remplit au fur et à mesure la structure suivante :

```
struct gps_data {  
    unsigned char valid ; //trame valide  
    float time ; //champ time de la trame: heure UTC  
    float lat ; //champ latitude de la trame  
    float lon ; //champ longitude de la trame  
    unsigned char received_checksum; // valeur du checksum reçu  
};
```

1) Proposer une MAE pour parser un champ float de la trame.

2) Implémenter en C cette fonction avec le prototype suivant : **char parseFloatField(char c, float * val, unsigned char * count, unsigned char * dp_count)**

Lorsque la fonction a terminé de parser le champ (lorsqu'un caractère ',' est reçu), elle renvoie la valeur 1 si la trame est valide et -1 si la trame n'est pas valide. Tant que la champ n'est pas entièrement décodé, la fonction renvoie 0.

3) Définir la structure générale du parser (MAE) exécuté par la fonction **parseGPS** dont le prototype est le suivant : **void parseGPS (char c, struct gps_data * gpsP) ;**

Lorsque la fonction a terminé de parser la trame, elle renvoie la valeur 1 si la trame est valide et -1 si la trame n'est pas valide. Tant que la trame n'est pas entièrement décodée, la fonction renvoie 0;

4) La fonction **char parseHexField(char c, unsigned char * val, unsigned char * count)** est supposée disponible et vous devrez la réaliser en TP.

5) Les latitudes/longitudes peuvent être représentés dans un repère orienté Nord Est, (Latitude positive quand orientée Nord, Longitude positive quand orientée Est). Compléter la structure du parser pour traiter l'orientation des latitudes longitudes.

6) Compléter la MAE pour le calcul et la vérification du checksum de la trame

7) Donner l'algorithme du programme principal (dont le rôle est d'afficher sur le terminal la position courante ainsi que l'heure ; ces données étant transmises par le récepteur GPS). Traduire en C cet algorithme.