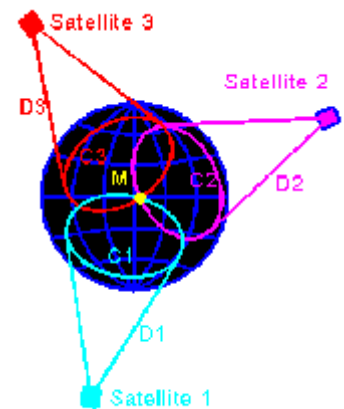




Capteurs - Géolocalisation

[Mise à jour le 4/6/2024]



1. Généralités

1.1 Géolocalisation

- La géolocalisation est un procédé permettant de positionner un objet, un véhicule, ou une personne sur un plan ou une carte à l'aide de ses coordonnées géographiques. Certains systèmes permettent également de connaître l'altitude (géolocalisation - dans l'espace - en 3D).
[Wikipédia](#)
- **Identifier** sa position sur une carte à l'aide de ses coordonnées GPS en allant sur coordonnees-gps.fr/

Quel GPS ?

Consulter le guide de sélection d'un GPS sur seedstudio.com

1.2 Trame NMEA 0183

- La norme **NMEA 0183** est une spécification pour la communication entre équipements marins, dont les équipements GPS. Elle est définie et contrôlée par la **National Marine Electronics Association (NMEA)**, association américaine de fabricants d'appareils électroniques maritimes, basée à Severna Park au Maryland (États-Unis d'Amérique).
[Wikipédia](#)

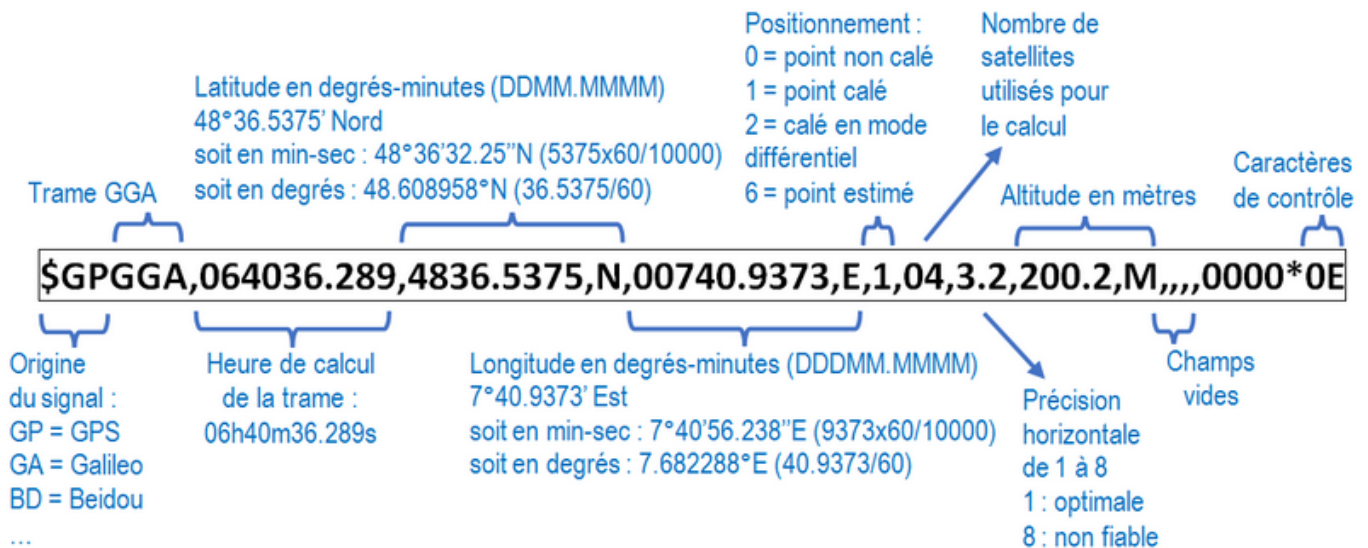
Structure

- Les **trames NMEA** sont codées au format ASCII et sont de la forme : **\$ <talker ID> <Trame type> [, <données> , <données>] * <checksum>**

Exemple : \$GPRMC,160906.308,A,4704.425,N,00224.374,E,,,040624,000.0,W*7D

- **Expérimenter**

- Générer une trame avec ce [Générateur](#) de trames NMEA en ligne (GPGGA, GPGSA et GPRMC à partir d'une position sur une carte).
- Décoder cette [trame](#) avec ce [Décodeur](#) de trames NMEA en ligne (informations de géolocalisation et affichage de la position sur une carte).

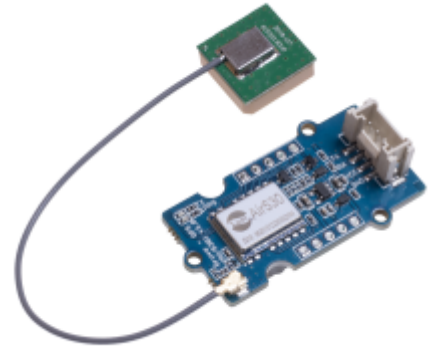


GNGGA

L'identifiant **\$GNGGA** est utilisé par certains récepteurs GNSS (souvent d'**origine chinoise** comme le GPS **Air530** ci-dessous) pour transmettre les données de position. Voir la [description des champs](#).

2. GPS

2.1 Grove (Air530)



2.1.1 Présentation

- **Source** : [wiki](#) Seeed studio

Module GPS économique basé sur un Air530 compatible Grove permettant de connaître sa position en temps réel. Ce module communique avec un microcontrôleur type Arduino ou compatible via une liaison série. Il comporte un circuit RTC permettant la sauvegarde de l'heure et de la date. L'utilisation de ce circuit nécessite une pile CR1220.

- **Distributeur** : [Gotronic](#)

- **Caractéristiques**

- Alimentation: 3,3 ou 5 Vcc
- Consommation: 60 mA maxi
- Interface: UART (9600 bauds)
- Sensibilité:
 - au démarrage: -160 dBm
 - sortie de veille: -162 dBm
 - acquisition: -166 dBm
- Précision:
 - distance: 2,5 m
 - vitesse: 0,1 m/s
- Dimensions interface GPS: 40 x 20 mm
- Compatible: GPS, Beidou, Glonass, Galileo, QZS, SBAS
- Température de service: -35 à 85 °C



- **Documentation** et **schéma**

- Voir le [wiki](#) Seeed studio.



- **Chronogrammes**

- Tramme GNGGA (.png et .scana) relevée avec un analyseur Ikalogic à télécharger : [ici](#)

2.1.2 Bibliothèques

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)

- *A partir du gestionnaire de bibliothèques de l'IDE Arduino, installer :*

TinyGPSPlus par Mikal Hart

1.0.3 installed

TinyGPSPlus provides object-oriented parsing of GPS (NMEA) sentences. NMEA is the standard format GPS devices use to report location, time, altitude, etc. TinyGPSPlus is a compact, resilient library that parses the most common NMEA 'sentences' used: GGA and RMC. It can also be customized to extract data from **any** compliant sentence.

A venir



2.1.3 Exemples de code

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)

- **Ressource** : [wiki](#) | [SoftwareSerial](#) | [Parser NMEA](#)

- **Exemple 1** pour visualiser les trames délivrée par le GPS



*.cpp

```
// Testé avec une carte Arduino Uno (2:RXD, 3:TXD)
// Interface série logicielle (asynchrone)
// 9600baud, 8, N, 1

#include <SoftwareSerial.h>

SoftwareSerial SoftSerial(2,3); // (2:RXD, 3:TXD) Arduino Uno
unsigned char buffer[64]; // tableau pour la réception de données via
```

```
le port série
int count = 0;           // compteur
void setup() {
  SoftSerial.begin(9600); // Réglage du débit de l'interface série
                           //logicielle
  Serial.begin(9600);    // Réglage du débit de l'interface série
                           //matérielle (moniteur série)
}

void loop() {
  if (SoftSerial.available()) // si disponible
  {
    while (SoftSerial.available()) // lire les données
    {
      buffer[count++] = SoftSerial.read(); // et les écrire dans le
      //tableau
      if (count == 64) break;           // tant qu'il n'est pas
      //plein
    }
    Serial.write(buffer, count); // écrire le tableau dans la console
    //Serial.write("\n");
    clearBufferArray(); // vider le tableau et
    count = 0;          // raz compteur
  }
  if (Serial.available()) // si les données sont
  //disponibles sur le port série matériel ==> les données proviennent d'un
  //PC ou d'un ordinateur portable
  SoftSerial.write(Serial.read()); // les écrire sur le port série
  //logiciel
}

void clearBufferArray() // fonction pour effacer le tableau
{
  for (int i = 0; i < count; i++) {
    buffer[i] = NULL;
  } // On efface le tableau avec la commande NULL
}
```

- **Exemple 2** de l'IDE Arduino pour obtenir la latitude et la longitude à partir des trames NMEA
Dans l'IDE Arduino, sélectionner : Fichier → Exemples → TinyGPSPlus → **DeviceExample**

Remarque

Effectuer les modifications suivantes dans le code de l'exemple

- Sur une **Arduino UNO** : RXPin = **2**, TXPin = **3**; GPSPbaud = **9600**;

A venir

2.2 Sparkfun Neo M9N (ublox)

From:
<https://webge.fr/dokuwiki/> - WEBGE Wikis

Permanent link:
<https://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:geolocalisation:geolocalisation&rev=1717521190>

Last update: **2024/06/04 19:13**

