

# GPS

---

## I) Les composants

Voici donc la liste des composants utilisés pour ce projet :

- Une Arduino Uno
- Puce GPS : Un GPS avec antenne intégré. Pour ce projet, j'utiliserai la librairie TinyGPS++ qui propose de nombreuses options pour gérer le GPS
- Des fils : male-femelle ou male-male : Afin de réaliser nos branchements

## II) Le GPS : Fonctionnement

Le GPS fonctionne à **9600 bauds** et envoie une **trame** du type **NMEA** qui ressemble à ce qui suit :

```
$GPRMC,194509.000,A,4042.6142,N,07400.4168,W,2.03,221.11,160412,,A*77
```

Les données GPS au format **NMEA** sont de données textuelles contenant de très nombreuses informations. Il est possible d'extraire ces données pour recueillir des informations comme :

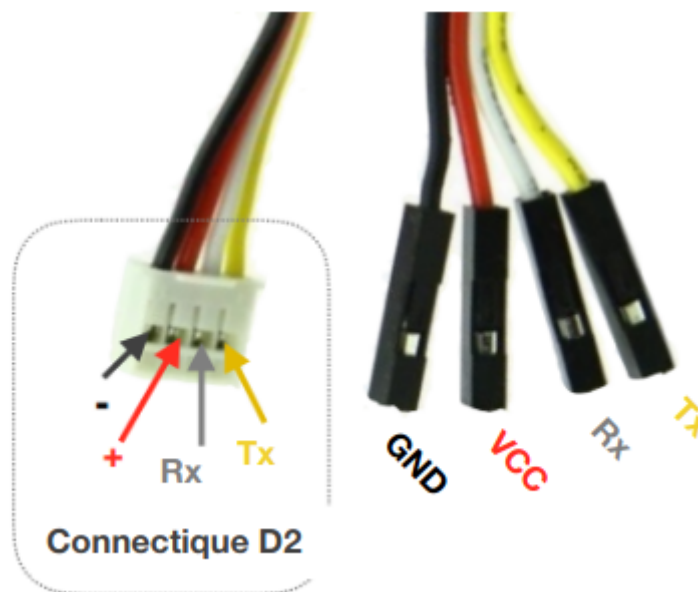
- Temps (heure...)
- Date
- Longitude
- Latitude
- Altitude
- Vitesse estimée (vitesse de déplacement)
- Type de fix (en fonction de nombre de satellites)

Ne vous inquiétez pas si vous ne comprenez pas le fonctionnement précis de cette trame, car la librairie **TinyGPS++** va s'occuper pour nous de nous traduire tout ceci en donnée exploitable et compréhensible.

### III) Montage des composants

- Tous les composants utilisés auront besoin d'être connecté au 5V (ou 3.3V) et au GND pour fonctionner.
- Voilà ce qui a de commun à chaque composant, passons maintenant aux spécificités du GPS : Le GPS utilise 4 pins, les 2 VCC et GND et ensuite TX et RX. Ces deux dernières sont respectivement le signal de transmission et celui de réception.

Prise		R.T.C.
Prise D2	Noir	GND -
	Rouge	VCC +
	Blanc	Rx
	Jaune	Tx



- La connexion entre le module GPS et l'Arduino UNO utilise une liaison série de type UART pour la transmission des données TX du module GPS vers RX de l'Arduino avec une vitesse de 9600 bauds. En d'autres termes, pour que votre Arduino puisse recevoir les données, vous devez brancher le TX du module GPS sur du RX de l'Arduino.
- Il faut relier les quatre broches du module GPS et l'Arduino comme indiqué ci-dessous :

Arduino UNO	NEO-6M
5V	VCC
GND	GND
TXD	RXD
RXD	TXD

## IV) Code

- Le code de mon programme de test est simple puisqu'il se contente de créer une liaison série logicielle entre le module GPS et l'Arduino pour récupérer les données GPS envoyées par le module qui sont ensuite affichées dans la console série.
- Nous allons voir un programme minimal décodant les trames **NMEA** reçues afin d'afficher la latitude, la longitude, l'altitude, la vitesse, le nombre de satellites captés, la date et l'heure. Nous utiliserons pour cela la bibliothèque **TinyGPS++**.
- Broches que j'utilise pour la liaison série logicielle entre l'Arduino UNO et le module GPS : **RX=2** et **TX=3**
- Vitesse de transmission série, 9600 bauds :
  - pour la liason Arduino UNO – Module GPS
  - pour le moniteur série

```
// Insertion bibliothèque TinyGPS
#include <TinyGPS++.h>

// Déclaration des variables pour le GPS
char data;
double latitude;
double longitude;
double alt; //altitude
double vitesse;
unsigned long nbre_sat;

// Pour afficher la date
int annee;
byte mois, jour, seconde, minute, heure; // heure GMT
String date = "";

// Création de l'objet GPS
TinyGPSPlus gps;
```

```

// Déclaration des variables pour la liaison série entre Arduino UNO
et GPS
#define RX 2 // Affectation des broches pour la liaison série
logicielle
#define TX 3 // de l'Arduino

#include <SoftwareSerial.h> // Insertion de la bibliothèque serie
logicielle
SoftwareSerial GPS(RX, TX); // Création de l'objet GPS pour la
liaison série

                                // entre l'Arduino et le module GPS

void setup() {
//Initialisation des liaisons série
  Serial.print("Initialisation liaisons series");
  GPS.begin(9600); // initialisation de la liaison série du GPS
pour reception données
  Serial.begin(9600);
}

void loop() {
  while (GPS.available()) {
    data = GPS.read();
    Serial.print(data);
    gps.encode(data);
    if (gps.location.isUpdated()){
      latitude = gps.location.lat();
      longitude = gps.location.lng();
      alt = gps.altitude.meters();
      vitesse = gps.speed.kmph();
      nbre_sat = gps.satellites.value();

      annee = gps.date.year();
      mois = gps.date.month();
      jour = gps.date.day();

      heure = gps.time.hour();
      minute = gps.time.minute();
      seconde = gps.time.second();

      date =String(jour) + "/" + String(mois) + "/" +
String(annee) + " " + String (heure) + ":" + String (minute) + ":" +
String (seconde);

```

```
        Serial.println(" ");
        Serial.println("----- DONNEES GPS DECODEES -----
---");
        Serial.print("Latitude="); Serial.println(latitude);
        Serial.print("Longitude="); Serial.println(longitude);
        Serial.print("Altitude (m) ="); Serial.println(alt);
        Serial.print("Vitesse (km/h)=");
Serial.println(vitesse);
        Serial.print("Nbre de satellites=");
Serial.println(nbre_sat);
        Serial.print("Date="); Serial.println(date);

        delay(10000);
    }
}
}
```