Mise en œuvre d'un récepteur GPS sur Raspberry Pi

Sommaire :

I - Introduction

L'objectif de ce document est de mettre en œuvre un récepteur **GPS** sur port **USB** utilisant le standard **NMEA** (**N**ational **M**arine & **E**lectronics **A**ssociation) dans sa version **NMEA-0183**, pour l'émission de ses données.

Sous ce standard, toutes les données sont transmises en mode série asynchrone sous forme de trames de caractères **ASCII**.

Le protocole utilisé est **19200 bits/s série asynchrone**, **sans parité**, **8 bits de données** et **1 bit de stop**.

II - Mise en œuvre du récepteur GPS

II.1. Introduction

Lorsqu'on branche le récepteur **GPS** sur le port **USB** du **Raspberry Pi**, le port attribué au récepteur **GPS** est **/dev/ttyUSB0** ou **/dev/ttyACM0**. On peut alors visualiser les trames envoyées par le récepteur **GPS** en exécutant la commande **cat /dev/ttyUSB0** ou **cat /dev/ttyACM0**.

A la mise sous tension du récepteur **GPS**, celui-ci émet la **trame NMEA** suivante :

$GPGGA,142333.00,,,,,0,00,99.99,,,,,,*62

En fonctionnement normal, le récepteur **GPS** émet la **trame NMEA** suivante :

$GPGGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M,,*42

II.2. GPSD

GPSD est un démon jouant le rôle de passerelle entre le récepteur **GPS** et des clients réseaux souhaitant recevoir les coordonnées **GPS**. Le démon écoute sur le port **2947**. De plus amples informations sur **GPSD** sont disponibles à l'adresse **https://gpsd.io/**.

A l'aide de la commande ci-dessous, on peut installer sur le **Raspberry Pi** les paquets nécessaires à l'utilisation de **GPSD** :

sudo apt install gpsd gpsd-clients python-gps libgps-dev

Il faut alors ouvrir le fichier **/etc/default/gpsd** et modifier la ligne :

DEVICES="/dev/ttyACM0" (si le récepteur GPS est sur /dev/ttyACM0)

A l'aide de la commande ci-dessous, on relance le démon **GPSD** :

sudo service gpsd restart

A l'aide de la commande ci-dessous, on peut alors valider le bon fonctionnement du démon **GPSD** :

cgps -s

II.3. Application cliente pour GPSD

On peut développer une application cliente pour **GPSD** permettant de récupérer les coordonnées envoyées par notre récepteur **GPS**. Pour développer cette application, il faut installer la librairie **libgps-dev** et compiler l'application avec la commande **g++ -o example example.cpp -lgps -lm**.

Le code de l'application est le suivant :

```
#include <libgpsmm.h>
#include <unistd.h>      // for sleep()
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
gpsmm gps_rec("localhost", DEFAULT_GPSD_PORT);
if (gps_rec.stream(WATCH_ENABLE|WATCH_JSON) == NULL) {
        cerr << "No GPSD running.\n";
        return 1;
        }
for (;;)
        {
        struct gps_data_t* newdata;
        if (!gps_rec.waiting(50000000)) continue;
        if ((newdata = gps_rec.read()) == NULL)
                {
                cerr << "Erreur Read()\n";
                return 1;
                }
```

```
        else
                {
                if (newdata->set & STATUS_SET) printf("Status = %d\n", newdata->status);
                if (newdata->set & TIME_SET) printf("Time = %lf\n", newdata->fix.time);
                if (newdata->set & LATLON_SET) printf("Latitude = %lf et Longitude = %lf\
                n",newdata->fix.latitude, newdata->fix.longitude);
                if (newdata->set & ALTITUDE_SET) printf("Altitude = %lf\n", newdata-
                >fix.altitude);
                if (newdata->set & MODE_SET) printf("Mode = %d\n", newdata->fix.mode);
                if (newdata->set & SATELLITE_SET) printf("Satellites in view = %d\n", newdata-
                >satellites_visible);
                }
                sleep(5);       // Pause de 5 secondes
                }
 return 0;
}
```