

E6.2 Téléscope : Fiche Technique Installation et utilisation du GPS BU-353

Table des matières

E6.2 Téléscope : Fiche Technique Installation et utilisation du GPS BU-353.....	1
Contexte :.....	1
Spécifications :.....	2
Utilisation GPS BU-353S4 :.....	3
Format des trames NMEA (RAPPEL) :.....	4
Préparation à l'import des coordonnées sur l'IHM Web.....	5
Dépannage du GPS.....	7

Contexte :



Le receveur GPS BU-353S4 est mis à disposition du projet télescope afin de pouvoir récupérer et actualiser les coordonnées géographiques de l'utilisateur. Ces informations permettront en plus des coordonnées équatoriales permettent le calcul des coordonnées horizontales.

Spécifications :



11. BU-353S4 SPECIFICATIONS

Electrical Characteristics (Receiver)	
GPS Chipset	SiRF STAR IV GSD4e
Frequency	L1, 1575.42 MHz
C/A Code	1.023 MHz chip rate
Channels	48
Sensitivity	-163dBm
Accuracy	
Position Horizontal	<2.5m 2D RMS SBAS Enable
Velocity	0.1m/sec 95% (SA off)
WAAS	Enabled for North America products (USGlobalSat, Inc)
Datum	
Datum	WGS-84
Acquisition Rate	
Hot start	8 sec., average (with ephemeris and almanac valid)
Warm start	35 sec., average (with almanac but not ephemeris)
Cold start	35 sec., average (neither almanac nor ephemeris)
Reacquisition	0.1 sec. average (interruption recovery time)
Protocol	
GPS Protocol	Default: NMEA 0183 (Secondary: SiRF binary) SiRF binary >> position, velocity, altitude, status and control
GPS Output Data	NMEA 0183 MEA0183 V3.0 protocol, and supports command: GGA, GSA, GSV, RMC, VTG, GLL v2.2 (VTG and GLL are optional)
GPS transfer rate	Software command setting (Default : 4800,n,8,1 for NMEA)
Dynamic Condition	
Acceleration Limit	Less than 4g
Altitude Limit	18,000 meters (60,000 feet) max.
Velocity Limit	515 meters/sec. (1,000 knots) max.
Jerk Limit	20 m/sec**3
Temperature	
Operating	-40°~ 80°C
Storage	-40°~ 85°C
Humidity	Up to 95% non-condensing
Power	
Voltage	5V ±5%
Current	60mA typical
Physical Characteristics	
Dimension	2.32" x 1.65" x 0.82" (59mm x 47mm x 21mm)
USB Cable Length	60" (152 cm)
Low Noise Amp	
Amplifier Gain w/out cable	27 dB Typical
Filtering	-25dB (+100 MHz)
Output VSWR	2.0 Max.
Voltage	DC 3 ~ 5.0V
Current	15mA max @ 5VDC

Due to continuous product improvements, all specifications are subject to change without notice.

Utilisation GPS BU-353S4 :

1. Connexion à la raspberry depuis le réseau domiris : ssh [pi@172.20.81.221](ssh://pi@172.20.81.221)
2. Création d'un dossier dédié à la gestion du gps :
`mkdir /home/pi/gps`
3. Installation de l'installateur de package python :
`sudo apt-get install python3-pip`
4. Installation du module serial pour communiquer avec le gps :
`sudo python3 -m pip install pyserial`
5. Création d'un programme python selon les spécifications `nano /home/pi/gps/gps.py` :

```

GPS transfer rate                                     Software command setting (Default : 4800,n,8,1 for NMEA)
"""placé dans /home/pi/gps/"""
import serial
gps = serial.Serial("/dev/ttyUSB0", timeout=None, baudrate=4800, xonxoff=False, rtscts=False, dsrdtr=False)

while True:
    line = gps.readline()
    print(line)

```

Après lancement du script : `sudo python3 /home/pi/gps/gps.py`

```

pi@rpi-telescope:/$ sudo python3 /home/pi/gps/gps.py
b'$GPGGA,105009.092,0.00,M,0.0,M,0.000*50\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'
b'$GPRMC,105009.092,V,,,,,,,,,170223,,N*4E\r\n'
b'$GPGGA,105010.092,0.00,M,0.0,M,0.000*58\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'
b'$GPRMC,105010.092,V,,,,,,,,,170223,,N*46\r\n'
b'$GPGGA,105011.092,0.00,M,0.0,M,0.000*59\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'
b'$GPRMC,105011.092,V,,,,,,,,,170223,,N*47\r\n'
b'$GPGGA,105012.092,0.00,M,0.0,M,0.000*5A\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'
b'$GPRMC,105012.092,V,,,,,,,,,170223,,N*44\r\n'
b'$GPGGA,105013.092,0.00,M,0.0,M,0.000*5B\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'
b'$GPGSV,3,1,12,12,69,235,23,15,16,172,31,19,36,061,17,42,00,000,*7E\r\n'
b'$GPGSV,3,2,12,25,53,057,,24,46,105,,28,31,055,,31,18,289,*78\r\n'
b'$GPGSV,3,3,12,03,12,245,,13,12,155,,22,11,267,,32,09,326,*75\r\n'
b'$GPRMC,105013.092,V,,,,,,,,,170223,,N*45\r\n'
b'$GPGGA,105014.092,0.00,M,0.0,M,0.000*5C\r\n'
b'$GPGSA,M,1,,,,,,,,,,,,,*12\r\n'

```

On reçoit des trames GGA, RMC et GSV. Les coordonnées peuvent ne pas s'afficher comme ci dessus si des obstacles sont rencontrés (exemple d'un plafond métallique, d'un tunnel). **Il sera alors nécessaire de sortir le GPS dans un lieu ouvert.**

Format des trames NMEA (RAPPEL) :

LA TRAME GGA :

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

123519 = Acquisition du FIX à 12:35:19 UTC

4807.038,N = Latitude 48 deg 07.038' N

01131.324,E = Longitude 11 deg 31.324' E

1 = Fix qualification : (0 = non valide, 1 = Fix GPS, 2 = Fix DGPS)

08 = Nombre de satellites en poursuite.

0.9 = DOP (Horizontal dilution of position) Dilution horizontale.

545.4,M = Altitude, en Mètres, au dessus du MSL (mean sea level) niveau moyen des Océans.

46.9,M = Correction de la hauteur de la géoïde en Mètres par rapport à l'ellipsoïde WGS84 (MSL).

(Champ vide) = nombre de secondes écoulées depuis la dernière mise à jour DGPS.

(Champ vide) = Identification de la station DGPS.

***42** = Checksum

Non représentés **CR et LF**.

LA TRAME GSV :

\$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75

2 = Nombre de trames GSV avec les données complètes.

1 = Trame 1 de 2 trames (jusqu'à 3 trames)

08 = Nombre de satellites visibles (SV).

01 = N° d'identification du 1er Satellite.

40 = Élévation en degrés du 1er Satellite.

083 = Azimut en degrés du 1er Satellite.

46 = Force du signal du 1er Satellite (Plus grand=meilleur)

(Cette séquence se répète jusqu'à 4 satellites par trames.

On peut donc avoir jusqu'à 3 trames GSV dans une transmission (12 satellites).)

***75** = checksum

non représentés les **CR et LF**

Ce qui explique la **limitation à 12 satellites** des GPS grand public.

LA TRAME RMC :

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

225446 = Heure du Fix 22:54:46 UTC

A = Alerte du logiciel de navigation (A = OK, V = warning (alerte)

4916.45,N = Latitude 49 deg. 16.45 min North

12311.12,W = Longitude 123 deg. 11.12 min West

000.5 = vitesse sol, Noeuds

054.7 = cap (vrai)

191194 = Date du fix 19 Novembre 1994

020.3,E = Déclinaison Magnétique 20.3 deg Est

***68** = checksum obligatoire

Non représentés **CR et LF**

Ces données "minimales", sont le plus souvent utilisées dans les programmes de navigation-GPS simples.

Préparation à l'import des coordonnées sur l'IHM Web

Les données suivantes devront être récupérés pour l'IHM Web et peuvent être fournis par la trame RMC :

- Date
- Longitude
- Latitude
- Heure

Celle ci est la 3ème reçue par le GPS, on modifie donc le programme python pour qu'il puisse récupérer la 3ème ligne.

""""placé dans /home/pi/gps/""""

```
import serial
```

```
try:
```

```
    gps = serial.Serial("/dev/ttyUSB0", timeout=None, baudrate=4800, xonxoff=False, rtscts=False, dsrdtr=False)
    for i in range(3): # se rend directement à la 3eme ligne
        result = gps.readline()
    print(result)
```

```
except:
```

```
    print("ERREUR GPS")
```

On affiche bien uniquement les trames RMC :

```
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105456.932,V,,,,,060323,,N*42\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105536.932,V,,,,,060323,,N*45\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105553.927,V,,,,,060323,,N*42\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105608.927,V,,,,,060323,,N*4F\r\n'
pi@rpi-telescope:~ $
```

Note : Si on lance le programme deux fois dans une courte intervalle, le résultat alternera une trame RMC avec une trame GSV ou GGA

```
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105407.927,V,,,,,060323,,N*42\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPGSV,3,1,10,25,35,353,32,13,57,115,,15,45,325,,29,42,197,*79\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105411.927,V,,,,,060323,,N*45\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPGSV,3,1,10,25,35,353,32,13,57,115,,15,45,325,,29,42,197,*79\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPRMC,105426.927,V,,,,,060323,,N*41\r\n'
pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
b'$GPGSV,3,1,10,25,35,353,32,13,57,115,,15,45,325,,29,42,197,*79\r\n'
```


Code final afin de ne récupérer que la trame RMC :

```

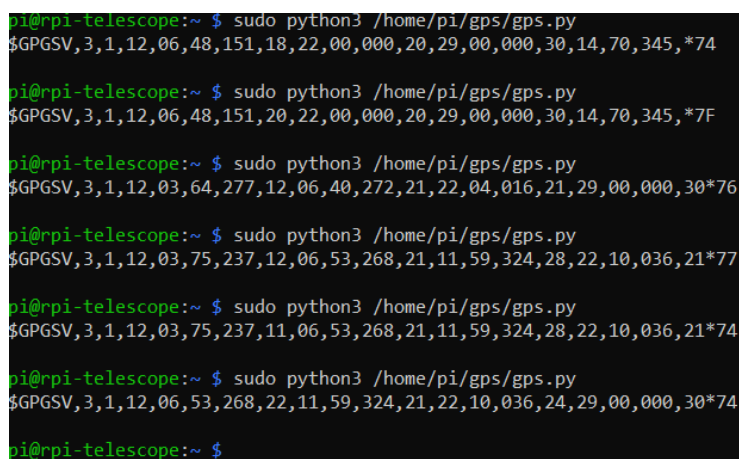
"""placé dans /home/pi/gps/"""

import serial

try:
    gps = serial.Serial("/dev/ttyUSB0", timeout=None, baudrate=4800, xonxoff=False, rtscts=False, dsrdtr=False)

    done = False
    while (done != True):
        result = gps.readline()
        if (result[:6] == b"$GPRMC"): # attend de recevoir une trame RMC pour sortir de la boucle
            done = True
        result = result.decode("utf-8") # décodage de l'objet octet pour l'affichage en chaîne de caractère
        print(result)
except:
    print("ERREUR GPS")

```



```

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,06,48,151,18,22,00,000,20,29,00,000,30,14,70,345,*74

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,06,48,151,20,22,00,000,20,29,00,000,30,14,70,345,*7F

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,03,64,277,12,06,40,272,21,22,04,016,21,29,00,000,30*76

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,03,75,237,12,06,53,268,21,11,59,324,28,22,10,036,21*77

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,03,75,237,11,06,53,268,21,11,59,324,28,22,10,036,21*74

pi@rpi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
$GPGSV,3,1,12,06,53,268,22,11,59,324,21,22,10,036,24,29,00,000,30*74

pi@rpi-telescope:~ $

```

Figure 1: On isole ici la trame GSV par exemple

Le site utilisera un accès non-root pour exécuter le fichier. Afin d'éviter l'erreur de permissions d'accès à /dev/ttyUSB0, on ajoute notre utilisateur pi à ce groupe.

On trouve que le groupe de /dev/ttyUSB0 est *dialout*
`stat /dev/ttyUSB0`

On ajoute l'utilisateur pi à ce groupe
`sudo usermod -a -G dialout pi`

En rajoutant `echo exec("whoami");` dans php on obtient l'utilisateur www-data, on l'ajoute également pour laisser le serveur accéder au gps.
`sudo usermod -a -G dialout www-data`

Un problème rencontré si le package python serial est installé en même temps que pyserial :
`python3 -m pip uninstall serial`

Dépannage du GPS

Lors d'une autre séance, après démarrage de la Raspberry, le GPS ne semble pas être détecté. Après rebranchement sur le même port toujours pas détecté. Branchement sur un autre port USB et GPS détecté.

Après lancement du programme de récupération du GPS, celui ci ne semble pas être supporté.

```
pi@pi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
Traceback (most recent call last):
  File "/home/pi/gps/gps.py", line 5, in <module>
    gps = serial.Serial("/dev/ttyUSB0", timeout=None, baudrate=4800, xonxoff=False, rtscts=False, dsrdtr=False)
  File "/usr/local/lib/python3.9/dist-packages/serial/serialutil.py", line 244, in __init__
    self.open()
  File "/usr/local/lib/python3.9/dist-packages/serial/serialposix.py", line 336, in open
    self._update_dtr_state()
  File "/usr/local/lib/python3.9/dist-packages/serial/serialposix.py", line 713, in _update_dtr_state
    fcntl.ioctl(self.fd, TIOCMIBIS, TIOCM_DTR_str)
OSError: [Errno 5] Input/output error
```

Redémarrage et même procédure mais impossible de détecter le GPS sur un autre port USB.

```
pi@pi-telescope:~ $ sudo python3 /home/pi/gps/gps.py
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/dist-packages/serial/serialposix.py", line 322, in open
    self.fd = os.open(self.portstr, os.O_RDWR | os.O_NOCTTY | os.O_NONBLOCK)
FileNotFoundError: [Errno 2] No such file or directory: '/dev/ttyUSB0'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/pi/gps/gps.py", line 5, in <module>
    gps = serial.Serial("/dev/ttyUSB0", timeout=None, baudrate=4800, xonxoff=False, rtscts=False, dsrdtr=False)
  File "/usr/local/lib/python3.9/dist-packages/serial/serialutil.py", line 244, in __init__
    self.open()
  File "/usr/local/lib/python3.9/dist-packages/serial/serialposix.py", line 325, in open
    raise SerialException(msg.errno, "could not open port {}: {}".format(self._port, msg))
serial.serialutil.SerialException: [Errno 2] could not open port /dev/ttyUSB0: [Errno 2] No such file or directory: '/dev/ttyUSB0'
```

Test du gps sur un autre équipement sur Windows 10. Installation du driver : GPS toujours non reconnu. Test avec un autre GPS : fonctionnel sur l'équipement Windows et sur la Raspberry pi.

Bilan de la panne :

GPS HS et remplacement par un nouveau GPS du même modèle fonctionnel

Note :

Les futures problèmes rencontrés peuvent être résolus en déplaçant le GPS dans un lieu à ciel ouvert, si cela ne fonctionne pas, redémarrer la Raspberry.