

# Concevoir une application de géolocalisation

---

Travaux pratique sur l'exploitation des mesures de positionnement d'un récepteur GNSS



# Objectifs

- Développer une application utilisant des données de géolocalisation
- Prototyper diverses fonctionnalités:
  - TOPOGRAPHIE - Relever des points fixes
  - DATA – afficher des données de points (attributaires et position)
  - TRACE – Relever un parcours et calculer ses caractéristiques
  - EMPRISE – Relever une surface et calculer son aire
  - SIG - Afficher les points sur une cartographie statique
  - TRACKING - Afficher des points sur une cartographie dynamique
  - NOTIFIER – Déclencher une action lorsqu'on se géolocalise dans une zone
  - Etc...

## Terminal d'exploitation



## Terminal d'exploitation



## Terminal d'exploitation



# Développement et Tests sur PC

- Pour faciliter le test et le debug, on reste sur vos machines.
- Utilisation du framework .Net 3.5
- Programmation en langage C#
- Utilisation de librairies:
  - Pour gérer les ports série
  - Pour afficher une cartographie dynamique
  - Autres



# Récepteur GNSS

## CARACTERISTIQUES TECHNIQUES PYX



**CARACTERISTIQUES PHYSIQUES**

Dimensions	18cm x 6,5cm
Poids	325 g
Interface I/O	1. Port: USB-C (Emulation de deux ports série 2 et fonction TCP/IP) 2. Bluetooth 5.0 3. Sortie configurable (NMEA, RTCM, etc.) 4. Fréquence des mesures jusqu'à 100Hz 5. Alimentation USB-C
Caractéristiques Environnementales	Température : -20°C to + 60 °C Humidité : 5% - 95% (sans condensation) IP 68 étanche à l'eau, à la poussière et au sable. Chute : 2 m Vibration : MIL-STD-810G Autonomie : > 10h en mode TERIA et TERIASat
Alimentation Electrique	Alimentation Externe DC : USB- C - 5V - 2A
Certification	CE, RoHS

**CARACTERISTIQUES GNSS**

Levé cinématique en temps réel ( RTK) <sup>1,2,3</sup>	Précision horizontale 0,6 cm+ 0,5 ppm Précision verticale 1,0 cm+ 1 ppm Temps d'initialisation 7s
Positionnement DGSS en temps réel <sup>1,2,3</sup>	Précision horizontale : 40 cm Précision verticale : 70,0 cm
SBAS( WAAS/EGNOS/MSAS/GAGAN) <sup>1,2,3</sup>	Précision horizontale : 60,0 cm Précision verticale : 80,0 cm
Post-traitement & enregistrement interne des données	Rinex jusqu'à 100Hz
Signaux GNSS	448 signaux GPS : L1C/A, L1C, L1 PY, L2P, L2P, L5 GLONASS : L1C/A, L2CA, L2P, L3 CDMA Beidou : B1, B1C, B2a, B2, B3 Galileo : E1, E5a, E5b, E5 AltBoc, E6 QZSS : L1 C/A, L1C, L2C, L5, L6 NAVIC : L5 SBAS : EGNOS, WAAS, GAGAN, MSAS, SDCM (L1, L5) L-Band – TERIASat
Firmware	Mises à jour gratuites des logiciels (USB-C)

**Compatible TERIASat.**  
Surveillance et atténuation des interférences AIM+ (bande étroite, bande large, brouilleurs de signaux).  
IONO+ atténuation avancée de la scintillation.  
APME+ : estimateur de trajets multiples a posteriori pour l'atténuation des trajets multiples de code et de phase.  
LOCK+ : un système de suivi plus robuste en cas de chocs mécaniques importants ou de vibrations.  
Surveillance autonome de l'intégrité des récepteurs RAIM+.

**Compatibilité**

Tous les collecteurs de données et les applications sous les systèmes d'exploitation Android et Windows.

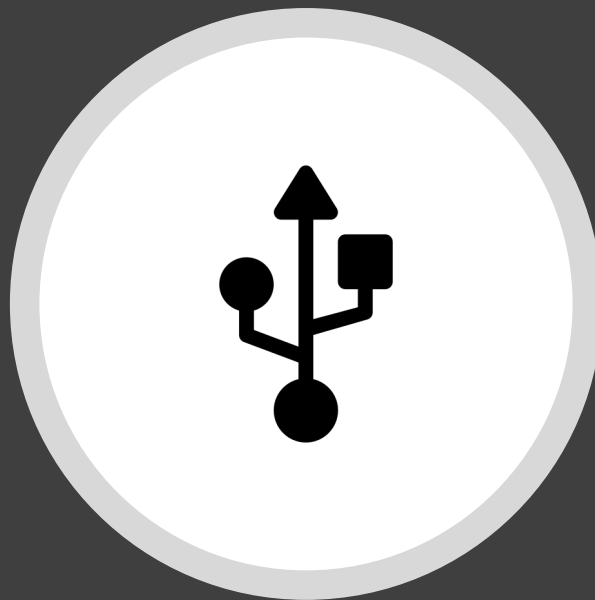
**PERFORMANCE TERIA EN TEMPS RÉEL**

RESEAU TERIA <sup>1,2,3</sup>	Précision horizontale 1,0 cm Précision verticale 2,0 cm Temps d'initialisation 5s
TERIASat SERVICE <sup>1,2,3</sup>	Précision horizontale 2,5 cm Précision verticale 5,0 cm Temps d'initialisation 15s

<sup>1,2</sup> La précision et les spécifications TTFF peuvent être affectées par les conditions atmosphériques, les trajets multiples du signal, la géométrie spatiale des satellites, la disponibilité de ceux-ci et la qualité des corrections.

<sup>3</sup> Les valeurs de performance supposent des conditions de ciel ouvert, en suivant les procédures recommandées dans le manuel du produit. La précision est exprimée en RMS. Les zones à trajets multiples, les valeurs PDOP élevées et les périodes de conditions atmosphériques sévères peuvent dégrader les performances.

**Fabriquée en France**



Liaison Terminal - GNSS

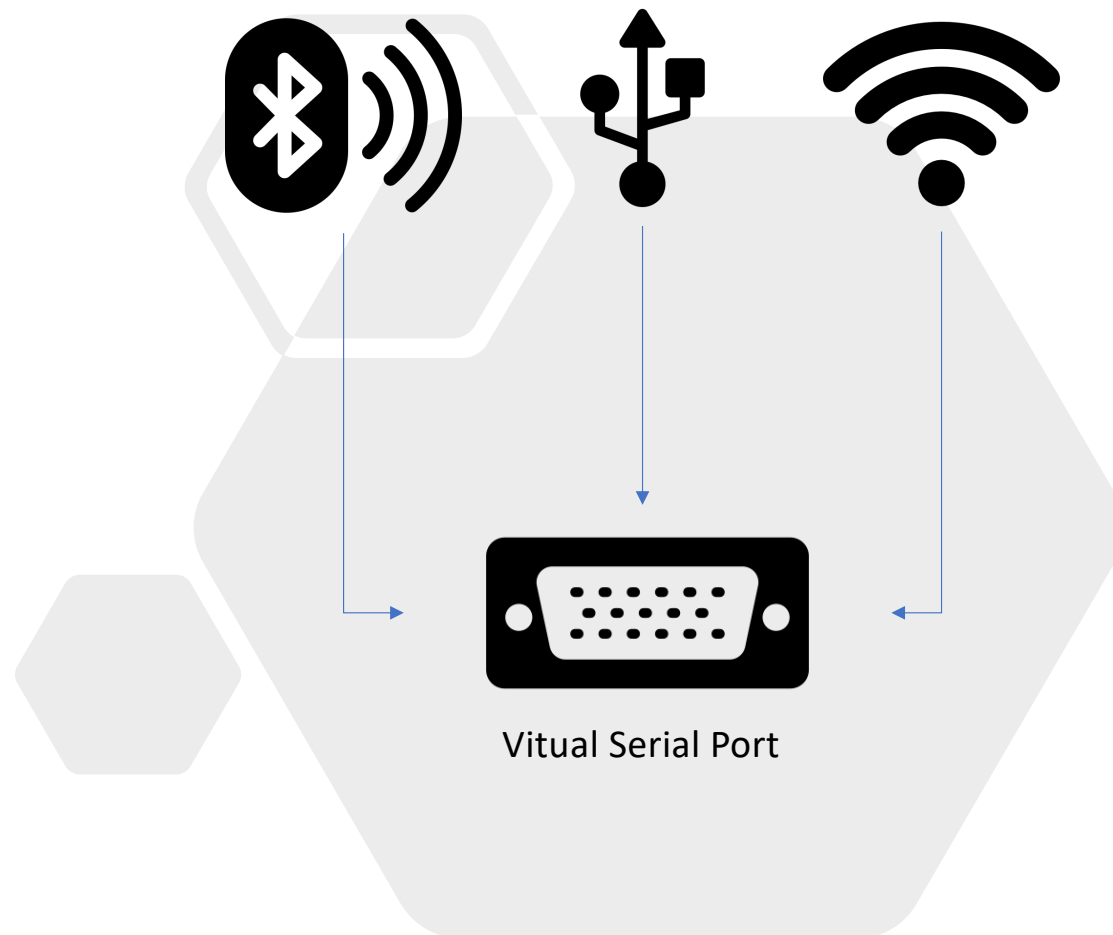


Peu importe...

# SERIAL PORT

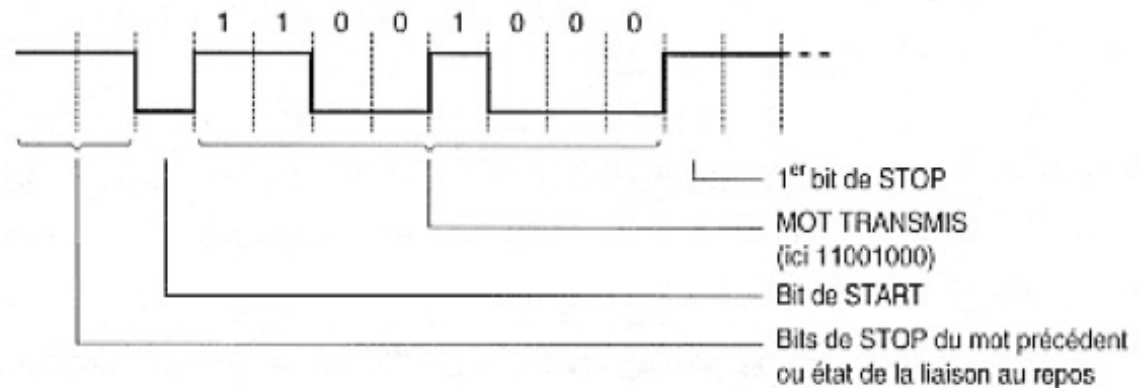
Nous écouterons **un port série**  
(Port COM) émulé par le système  
d'exploitation du terminal

La carte Bluetooth/Wifi/USB se  
charge de recevoir et envoyer les  
données au GNSS



# Rappels sur le port série

- Les bits des données sont envoyés les uns après les autres
- 1 caractère = 1 valeur dans  $[0 ; 255]$  = 1 octet = 8 bits
- 1 caractère est précédé d'un
  - Bit de départ (STARTBIT)
  - 1 ou 2 bits de fin (STOPBITS)
- Donc 1 caractère transmis donne lieu à 10 bits reçus



Source : [https://perso-etis.ensea.fr/andry/cours/L3/liaison\\_serie.pdf](https://perso-etis.ensea.fr/andry/cours/L3/liaison_serie.pdf)

# Bits ou Bytes ?

- Bit = 0 ou 1
- Byte = 8 bits = Octet en français

# BAUD RATE - Vitesse de transmission

- 1 octet est donc précédé d'un bit et suivi d'un bit, au minimum
- En combien de temps se transmet la lettre B ?
  - B se traduit en octet par 0x42
  - 0x42 s'écrit en binaire 01000010
  - Un bit de début
  - Supposons qu'il y a un seul bit de stop, ça fait 10 bits
  - Si on transmet à 9600 bauds (bits/s),  $10/9600$  fait environ

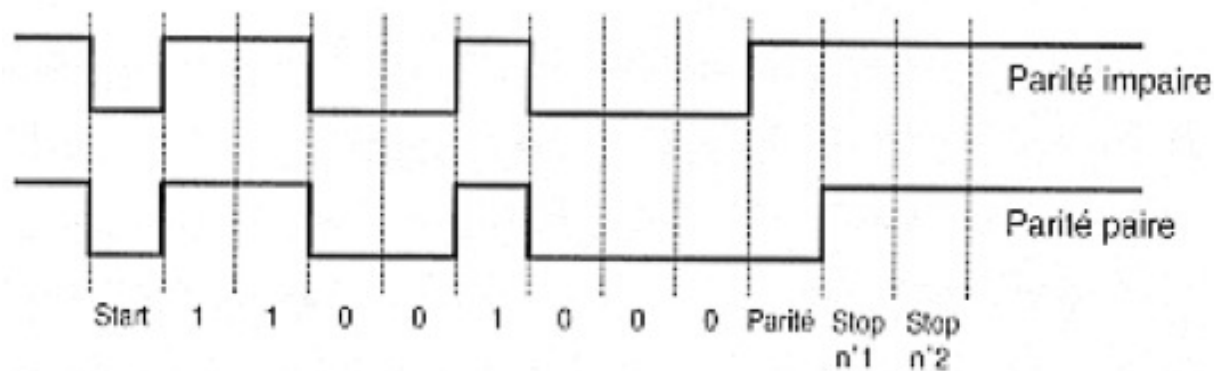
1 ms pour transmettre « B »

# PARITYBIT - Le bit de parité

- Pour vérifier l'intégrité de la transmission

On compte le nombre de bits à 1 de l'octet sans les bits de début et de fin

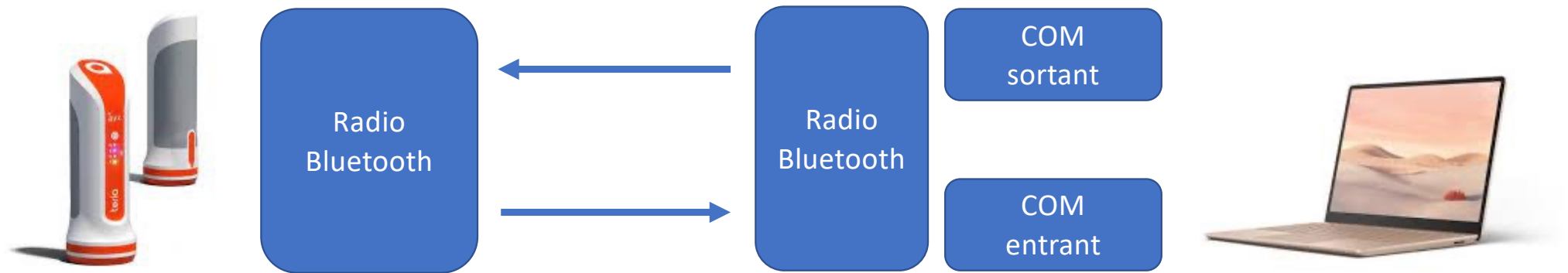
Si c'est pair, alors  $\text{PARITYBIT} = 1$ , si c'est impair  $\text{PARITYBIT} = 0$



# Appairage Bluetooth

Le **COM sortant** de votre PC est celui qui a établi la connexion avec le GNSS

En réponse, le GNSS a établi une connexion avec votre PC sur votre **COM entrant**



# Ecoute du port COM sortant

- 1) Allez dans votre interface de réglages Bluetooth et lancez la connexion au GNSS
- 2) Le COM sortant est celui qui reçoit les données. Il faut l'identifier dans vos **paramètres Bluetooth avancés**.
- 3) Installez et lancez le client Teria Ntrip puis:
  - a - Paramétrez la connexion
  - b - Lancez la connexion
  - c - Allez dans details et maximisez la fenêtre
  - d – Cliquez sur « Data COM PORT » pour afficher les données reçues  
elles apparaissent tout à droite de la fenêtre



# Zoom sur les trames GGA et GST

## 4) Expliquer la constitution de la trame GGA

- Informations
- Unités
- Référentiels

## 5) Expliquer la constitution de la trame GST (idem)

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

Nous allons étudier le fonctionnement de l'application WinForm **Serial Port Listener** RS Log.

Le dossier de projet Visual Studio est disponible sur les fichiers Teams de notre équipe.



**Télécharger le dossier de projet dans votre workspace Visual Studio et lancez le en double cliquant sur le fichier SerialPortListener.csproj**

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

## I – Test de l'application

- 1) Au préalable, il faut s'assurer que l'application Ntrip n'écoute plus le port COM sortant. En effet, un port COM ne peut être écouté que par une seule application à la fois.
- 2) Démarrer le projet SerialPortListener
- 3) Démonstration du fonctionnement (professeur)
- 4) Observer les trames reçues:
  - 1) De quels types sont-elles ?
  - 2) Quelles informations contiennent-elles ?

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

## II – Analyse de l'application et de son code

- 1) Expliciter l'architecture de l'application dans un diagramme UML
- 2) Comment peut-on scruter le fonctionnement de l'application instruction par instruction ?
- 3) Définir la ligne de code pour le départ de l'inspection
- 4) Lancer l'inspection et scruter les valeurs des variables au fil de l'exécution

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

## II – Analyse de l'application et de son code

### 1) Paramétrage des ports séries et gestion de leurs évènements

#### 1) Le Serial Port Manager

- 1) Quelle librairie windows utilise-t-il ?
- 2) Quels sont ses attributs et leurs rôles ?
- 3) Quels sont les rôles des diverses méthodes ?
- 4) A quoi correspondent les EventHandler ?
- 5) Quelle autre classe est définie dans SerialPortManager.cs ?

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

## II – Analyse de l'application et de son code

### 1) Connexion

1) Identifier le port qui reçoit les données du GNSS (le sortant)

1) Pour ce faire, direction les paramètres bluetooth

2) Puis paramètres bluetooth avancés, onglet Port COM

2) Allumer le GNSS et le placer dehors debout

3) Paramétrer et lancer la connexion sur l'interface de l'appli

1) COM sortant, 9600 bauds, 8 databits, Parity None, Stop bits One

2) Cliquer sur Start Listening

4) Normalement, les trames GGA et GST défilent

# Etude d'une application WinForm pour lire les trames reçues sur le port COM sortant

## II – Analyse de l'application et de son code

### 1) Connexion

### 2) Récupération des trames

- 1) Quelle méthode est lancée automatiquement lorsqu'une nouvelle trame est transmise par le GNSS ?
- 2) Quel est le lien entre cette méthode et la réception d'une trame sur le port COM ?
- 3) Quel objet transporte la trame en bit ?
- 4) Quelle ligne de code convertit la trame en bits en chaîne de caractères ?
- 5) Dans quel composant la chaîne de caractère est-elle affichée ?



# Création d'objets de géopositionnement

- Pour chaque trame reçue, nous allons exploiter la conversion d'ASCII à Texte, et créer un Bean, à savoir un objet C# contenant toutes les informations transmises.
- Etant donné que le GNSS nous envoie deux types de trames (GGA et GST), nous allons définir deux types de Bean, donc créer deux classes:
  - La classe GGA
  - La classe GST

# Création d'objets de géopositionnement

## I – Conception de Bean

### 1) Objets GGA

#### 1) Quelles sont les types de données pour chaque infos de la trame ?

```
$GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,0000*0E
```

```
$GPGGA      : Type de trame  
064036.289 9: Trame envoyée à 06 h 40 min 36 s 289 (heure UTC)  
4836.5375,N : Latitude 48,608958° Nord = 48° 36' 32.25" Nord  
00740.9373,E: Longitude 7,682288° Est = 7° 40' 56.238" Est  
1          : Type de positionnement (le 1 est un positionnement GPS)  
04         : Nombre de satellites utilisés pour calculer les coordonnées  
3.2        : Précision horizontale ou HDOP (Horizontal dilution of precision)  
200.2,M    : Altitude 200,2, en mètres  
,,,,,0000  : D'autres informations peuvent être inscrites dans ces champs  
*          : séparateur de checksum  
0E         : Somme de contrôle de parité, un simple XOR sur les caractères entre $ et *
```



Nom de l'attribut
_date
_north
_east
_typeLocation
_nbSat
_altitude
_nextdata



Type
DateTime
double
double
int
int
double
string

#### 2) Créer la classe GGA.cs avec le concepteur UML de Visual Studio

- 1) Déclarer les attributs
- 2) Surcharger le constructeur (à la création de l'objet, les attributs doivent être remplis)
- 3) Générer les propriétés

# Création d'objets de géopositionnement

## I – Conception de Bean

### 1) Objets GGA

### 2) Objets GST

#### 1) Quelles sont les types de données pour chaque infos de la trame ?

Field	Description
1	UTC time of GGA fix
2	RMS value of the standard deviation of the range inputs to the navigation process (range inputs include pseudoranges and DGPS corrections)
3	Standard deviation of semi-major axis of error ellipse, in meters
4	Standard deviation of semi-minor axis of error ellipse, in meters
5	Orientation of semi-major axis of error ellipse, in degrees from true north
6	Standard deviation of latitude error, in meters
7	Standard deviation of longitude error, in meters
8	Standard deviation of altitude error, in meters

#### 2) Créer la classe GST.cs avec le concepteur UML de Visual Studio

- 1) Déclarer les attributs
- 2) Surcharger le constructeur (il prendra un string en paramètre)
- 3) Générer les propriétés

Nom de l'attribut	Type
_date	DateTime
_RMS	double
_SDmajor	double
_SDminor	int
_SDmajEllipseError	int
_SDlatError	double
_SDlongError	double
_SDaltError	double

# Création d'objets de positionnement

I – A chaque nouvelle trame, on crée un objet

- 1) Reconnaître la trame
- 2) Créer l'objet en lui passant la trame
- 3) Dans le constructeur de l'objet:
  - 1) Découper la trame
  - 2) Convertir les données
  - 3) Initialiser les attributs

# Création d'objets de positionnement

I – Stocker les objets dans un ArrayList dans le mainform

- 1) ArrayList GGA
- 2) ArrayList GST

Et maintenant, tout est possible !