

SUPPORT CCF E5	SYSTEME INFOBUS	BTS SN
---------------------------	------------------------	---------------

**TP Mise en œuvre du déplacement GPS virtuel du système INFOBUS.
OUTIL DE PRE RECETTE INTERNE
(Activité préparatoire c.c.f. E5 situation 1 et 2)**

A. Données pédagogiques

Activités / Tâches	A9 mise en œuvre du processus de réception en situation de livraison. T92 installer.	
Compétences	C52 Mettre en œuvre une solution matérielle/logicielle en situation.	
Savoirs associés	S22 documentation exploitée. SF69 exécuter la procédure d'installation. SF70 Exécuter les mesures et tests appropriés.	
Prérequis	Ligne conçue.	
Objectif	Visualiser, valider le déplacement du bus, tester la conformité du fonctionnement au regard de la procédure respectée d'installation (sans la présence du client).	
Objectif(s) associé(s)	Configurer le calculateur pour la réception des trames GPS issu du logiciel "trompeur"U-com.	
Durée	4H	
Conditions	Binôme	
Problématique	Présenter les informations de positionnements reçues par les satellites GPS	
Environnement	Matériel	PC de bureau, ordinateur de Bus SAM
	Documentaire	Document SET Didact : le récepteur GPS du système S.I.V. Documents Ublox : Antaris 4 Modules GPS Datasheet, ANTARIS®4 GPS Modules System Integration Manual (SIM), U-center user guide, ublox5_Protocol_Specifications (GPS.G5-X-07036), GPS Essentials of Satellite Navigation Compendium, TIM-4x Datasheet
	Logiciel	u-com

B. Mise en situation

Lorsque qu'un exploitant d'un réseau d'autobus d'une agglomération met au point un scénario de lignes de bus, c'est-à-dire faire la programmation de la diffusion d'informations sonores et visuelles synchronisées avec les arrêts de la ligne, il doit pouvoir tester ce scénario. Or, le scénario ne peut se déployer que lorsque les coordonnées de géo localisation qui arrivent à l'ordinateur de bord sous forme de trames GPS évoluent. Le test du scénario peut donc se faire soit par un déplacement réel sur le terrain, soit par une simulation du déplacement par injection sur le calculateur de trames GPS convenablement calculées. Cette seconde solution est plus confortable dans la mesure où elle évite un déplacement sur le terrain qui peut être éloigné. Le programme u-com répond à cette problématique et permet la simulation d'un déplacement, basé sur la ligne que l'exploitant vient de créer.

La validation de ce logiciel et l'étude du code source de quelques unes de ses fonctionnalités est l'objet de ce T.P.

C. Procédure d'installation et de configuration du poste expérimental

1. Matériel

Le PC de bureau devra pouvoir disposer de deux ports RS232. On pourra utiliser des adaptateurs USB-RS232 facilement disponibles dans le commerce.

⇒ Relier les deux ports série du PC par le câble RS232.

2. Logiciel

⇒ Installer et/ou ouvrir le logiciel « u-com » sur le PC de bureau.

⇒ Installer et/ou ouvrir le logiciel « u-center » sur le PC de bureau.

3. Validation de la simulation du déplacement

⇒ Logiciel u-center : préciser le nom du port COM d'entrée des trames GPS et ouvrir une fenêtre « Text Console »

⇒ Logiciel u-com

- préciser le numéro du port COM de sortie des trames GPS (de COM1 à COM9)
- choisir le format NMEA et ouvrir le fichier ublox5_NMEA qui se trouve dans le répertoire du logiciel u-center (typiquement C:\Program Files\u-blox\u-center\contents). Vérifier que ce fichier est correctement reconnu par le programme u-com (voir le cadre « Informations »). Cliquer sur le bouton « Envoyer »

⇒ Vérifier que u-com envoie bien des groupes de trames (visualiser les trames envoyées) et vérifier par ailleurs que u-center reçoit bien ces groupes de trames (animation des écrans et visualisation des trames reçues dans la fenêtre Text Console).

D. Trame NMEA : étude de la somme de contrôle

1. Structure de la trame NMEA

⇒ Lire la documentation sur la structure d'une trame NMEA (document u-blox5 Protocol Specification, NMEA Protocol , Protocol Overview)

⇒ Faire une copie d'écran de u-com et choisir une trame NMEA. Montrer que sa structure est bien celle précisée dans le document « u-blox5 Protocol Specification ». Repérer et préciser les champs <Address>, <Value> et <Checksum>.

2. Validation du champ « somme de contrôle »

Définition (Wikipédia) : la somme de contrôle (le terme anglais *checksum* est également employé), parfois appelé « empreinte », est un nombre qu'on ajoute à un message à transmettre pour permettre au récepteur de vérifier que le message reçu est bien celui qui a été envoyé.

⇒ En lisant la documentation sur la structure d'une trame NMEA (document u-blox5 Protocol Specification, NMEA Protocol , Protocol Overview), expliquer comment est calculé le champ « Checksum ».

⇒ Vérifier la valeur du champ Checksum pour la trame choisie précédemment (on pourra utiliser le calculateur de Checksum du site <http://nmeachecksum.eqth.net/>).

E. Trame UBX : étude des sommes de contrôle

⇒ Logiciel u-center : ouvrir une fenêtre « Binary Console »

⇒ Logiciel u-com

- préciser le numéro du port COM de sortie des trames GPS (de COM1 à COM9)
- choisir le format UBX et ouvrir le fichier ublox5_UBX qui se trouve dans le répertoire du logiciel u-center (typiquement C:\Program Files\u-blox\u-center\contents). Vérifier que ce fichier est correctement reconnu par le programme u-com (voir le cadre « Informations »). Cliquer sur le bouton « Envoyer »

⇒ Vérifier que u-com envoie bien des groupes de trames (visualiser les trames envoyées) et vérifier par ailleurs que u-center reçoit bien ces groupes de trames (animation des écrans et visualisation des trames reçues dans la fenêtre Binary Console)

1. Structure de la trame UBX

⇒ Lire la documentation sur la structure d'une trame UBX (document u-blox5 Protocol Specification, UBX Protocol , UBX Packet Structure)

⇒ Faire une copie d'écran de u-com et choisir une trame UBX. Montrer que sa structure est bien celle précisée dans le document « u-blox5 Protocol Specification ». Repérer et donner la valeur des 2 caractères de synchronisation (SYNC CHAR). Repérer et donner la valeur des 2 octets « Class » et « ID ». Repérer et donner la valeur des 2 octets « LENGTH ». Repérer et donner la valeur des 2 octets « CK_A » et «CK_B ».

2. Etude du champ « somme de contrôle »

⇒ En lisant la documentation sur la structure d'une trame UBX (document u-blox5 Protocol Specification, UBX Protocol , UBX Checksum), donner le nom, recopier et expliquer l'algorithme de calcul des octets CK_A et CK_B.

3. Etude d'un code source

Lorsqu'il génère lui même des trames UBX pour une simulation de déplacement pour une ligne donnée (Format CFG), le programme u-com fournit uniquement des trames de type UBX-NAV-SOL. Pour ces trames, il doit calculer les octets CK_A et CK_B.

Le code source C correspondant au calcul des octets CK_A et CK_B est le suivant :

```
unsigned char TrameUbx[60] ;

//Mise à jour des checksum
unsigned char  CKA,CKB;
CKA=0;CKB=0;
for (i=2;i<=57;i++) {CKA+=TrameUbx[i];CKB+=CKA;}
TrameUbx[58]=CKA;TrameUbx[59]=CKB;
```

⇒ Justifier le type et la taille du tableau TrameUbx

⇒ Justifier le type des variables CKA et CKB

⇒ Justifier l'intervalle de variation de l'index de boucle « i »

⇒ Expliquer comment ces lignes de code C permettent de calculer correctement les deux sommes de contrôle de la trame

⇒ En utilisant un environnement de programmation en C, recopier le code ci-dessus avec la définition suivante de la trame UBX ci dessous:

```
unsigned char TrameUbx[60]={ 0xB5, 0x62,0x01, 0x06, 0x34, 0x00, 0x95, 0xDF,
0x08, 0x00, 0x2B, 0xED, 0xFC, 0xFF, 0x7D, 0x06, 0x03, 0x01, 0x95, 0xAF,
0x90, 0x1B, 0x91, 0x4F, 0x93, 0x02, 0xDA, 0x7D, 0xF7, 0x19, 0xB2, 0x02,
0x00, 0x00,
0x00, 0x00, 0x14, 0x00, 0x00, 0x00, 0x32, 0x01, 0x00, 0x08, 0x06, 0xC0,
0x43, 0x40, 0x00, 0x00}
```

⇒ Mettre à jour les checksum en appliquant l'algorithme et vérifier que les deux derniers octets de la trame prennent les valeurs : 0x68, 0x64.

F. Algorithme de calcul de décalage horaire

Lorsqu'il génère lui-même des trames UBX pour une simulation de déplacement pour une ligne donnée (Format CFG), le programme u-com fournit uniquement des trames de type UBX-NAV-SOL. Pour ces trames, il doit fournir le temps UTC. Ce temps UTC est calculé à partir de l'heure courante de l'ordinateur en retranchant le décalage horaire.

Le code source correspondant à ce calcul est donné ci-dessous :

```
// Calcul de l'heure UTC : introduction
//Pour avoir l'heure UTC, on retranche à l'heure système la valeur du
décalage horaire. Ce décalage vaut 1h pour l'heure hiver et 2h pour l'heure
d'été
//Les pays européens observent l'heure d'été à partir du dernier dimanche
du mois de mars jusqu'au dernier dimanche d'octobre.

//On récupère la date courante
Mois=m_codtTimeNow.GetMonth(); //On récupère le mois de 1 à 12
Jour=m_codtTimeNow.GetDay(); //On récupère le jour de 1 à 31
JourSemaine=m_codtTimeNow.GetDayOfWeek();//Jour de la semaine de 1
(dimanche) à 7 (samedi)

//Calcul du décalage horaire correspondant à la date récupérée
if ((Mois < 3) || (Mois > 10)) DecalageHoraire=1; //Commentaire...
if ((Mois > 3) && (Mois < 10)) DecalageHoraire=2; //Commentaire...

if (Mois == 3) //Commentaire...
    {if (Jour < 25) DecalageHoraire=1;
    else {JourLimite=Jour+(8-JourSemaine)%7;
        if (JourLimite > 31)JourLimite-=7;
        if (Jour < JourLimite) DecalageHoraire=1;else DecalageHoraire=2;
        }
    }

if (Mois == 10) //Commentaire...
    {if (Jour < 25) DecalageHoraire=2;
    else {JourLimite=Jour+(8-JourSemaine)%7;
        if (JourLimite > 31)JourLimite-=7;
        if (Jour < JourLimite) DecalageHoraire=2;else DecalageHoraire=1;
        }
    }
```

⇒ Simuler le programme pour les dates suivantes : mercredi 12 janvier, vendredi 30 novembre, mardi 10 mars, vendredi 27 mars, lundi 30 mars, samedi 16 octobre, mardi 26 mars, dimanche 31 octobre.

⇒ Expliquer les conditions de deux premiers « if »

⇒ Rajouter des commentaires aux endroits indiqués « //Commentaires... »

⇒ Quelle est la signification de la variable « JourLimite » et justifier son calcul.

⇒ En utilisant un environnement de programmation en C, implanter le code ci-dessus de calcul du décalage horaire et écrire la ligne qui réalise le calcul du temps UTC à partir du temps système.
