

# **Tâche 6 – Couche MAC ADS-B**

## **Implémentation de la couche MAC**

### **Durée 3h**

#### **Projet TS229 – Année 2019/2020**

Guillaume Ferré, Romain Tajan et Baptiste Laporte-Fauret

### **Pré-requis**

La **tâche 3** est nécessaire pour réaliser entièrement cette tâche. Peut être commencée après lecture du sujet.

### **Objectifs**

Le but de cette partie est de développer une fonction Matlab qui convertit un message binaire en registre afin de pouvoir en extraire les données sur les appareils concernés.

### **Structure des trames ADS-B**

Les signaux émis par les appareils pour l'ADS-B ont une durée de 120  $\mu s$ . Ils sont constitués des parties suivantes :

- le préambule (identique à celui de la section précédente),
- le format de la voie descendante,
- la capacité,
- l'adresse OACI (Organisation de l'Aviation Civile Internationale) de l'appareil,
- les données ADS-B,
- les bits de contrôle de parité.

La durée de chacune des parties ainsi que leur position dans une trame ADS-B sont représentées en Figure 2. Nous détaillons maintenant la fonction des différentes parties de la trame.

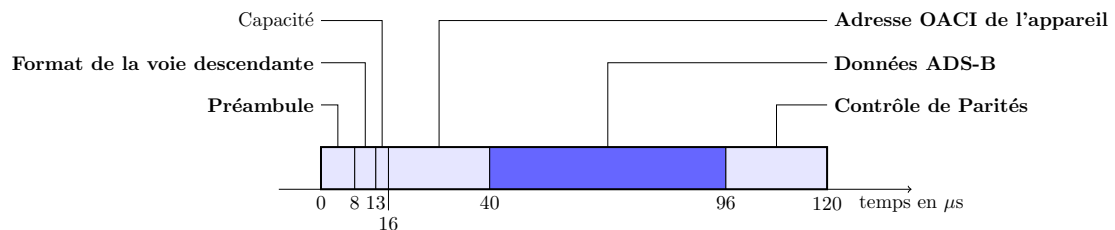


FIGURE 1 – Format d'une trame ADS-B

### Le préambule

Il sert pour la synchronisation temporelle et fréquentielle des signaux. Le préambule est identique quelque soit les types de signaux ADS-B transmis, il dure  $8\mu s$  et est identique à celui utilisé dans la section précédente.

### Le format de la voie descendante

Le format de la voie descendante (parfois noté DF pour Downlink Format) est codé sur 5 bits et indique le type de la trame envoyée. Les valeurs particulières de ce format sont les suivantes

- 11 : message d’acquittement
- **17 : message de type ADS-B**
- 18 : message de type TIS-B (Traffic information service – broadcast)
- 19 : message de type ADS-B militaire (crypté)

Dans ce projet, nous nous intéresserons seulement aux trames avec **DF = 17**.

### La capacité

La capacité est notée CA (pour CAPacity). Elle est codée sur 3 bits et représente un sous-type de trame envoyée. Nous ne nous préoccupons pas de sa signification.

### L’adresse OACI de l’appareil

L’OACI est l’Organisation de l’Aviation Civile Internationale, elle définit entre autre une immatriculation pour les aéroports, aérodromes et appareils volants. Dans les trames ADS-B, l’adresse OACI de l’appareil (parfois notée AA pour Aircraft Address) est codée sur 24 bits. Chaque appareil possède une adresse unique tout au long de sa vie, quelque soit sa compagnie ou son plan de vol.

Dans ce projet, il est primordial de pouvoir décoder correctement cette adresse car elle servira à connaître de quel appareil proviennent les données décodées. Cette adresse sert aussi pour trouver un avion dans les bases de données des tours de contrôles, ou des radars virtuel (tels que <http://planefinder.net> ou <http://www.flightradar24.com>). On pourra donc se servir des radars virtuels afin d’avoir plus d’informations sur les appareils ayant transmis les signaux reçus.

### Les données ADS-B

Dans le cadre de données envoyée pour ADS-B (DF = 17), les messages sont composés de 56 bits et durent  $56\mu s$ . Les messages pouvant être transmis sont répertoriés dans des registres. Le contenu de ces messages dépend du type de registre. On trouve entre autres des registres pour les informations de position au sol, de **position en vol**, de **signes d’identification** et de vitesse. Ceux qui vont nous intéresser sont ceux écrits en gras, les derniers étant des bonus.

Dans tous les cas, les registres sont constitués de 56 bits. Les 5 premiers bits du message constituent le code du format des données (FTC-Format Type Code en anglais). Ces bits indiquent si le message correspond à un message de position au sol, un message de position en vol ou un message d’identification. Les tables de correspondances entre les 5 bits et le type de message sont données dans les Tables 1 à 3 en Annexe A.

La structure des trames de position en vol est donnée dans la Table 1 ci-dessous. Pour la suite du projet, nous ne considérerons ni les 2 bits de *surveillance* ni le bit de *type d’antenne* ni le bit *indicateur de temps UTC*. L’encodage de l’altitude ainsi que ceux des latitude et longitude sont donnés en Annexe.

La structure des trames d’identification est donnée Table 2. L’identification de l’appareil est composée de 8 caractères, chacun étant encodé sur 6-bits. La table de correspondance entre les 6 bits et les caractères est donnée en Table 5 dans l’Annexe D.

TABLE 1 – Composition du message de position en vol

Index binaire	Champs	
1	MSB	Format Type Code
⋮		
5	LSB	
6	MSB	Surveillance Status
7	LSB	
8	Indicateur de type d'antenne	
9	MSB	Altitude
⋮		
20	LSB	
21	Indicateur de temps UTC	
22	Indicateur de format CPR	
23	MSB	Latitude encodée avec CPR
⋮		
39	LSB	
40	MSB	Longitude encodée avec CPR
⋮		
56	LSB	

## Sous-tâches

**Sous-tâche 1** *Quelles valeurs de FTC correspondent à des trames de position en vol ? Même question pour les messages d'identifications.*

Afin de rendre synthétique votre code MATLAB, un registre sera représenté par une structure possédant les champs `format` (contenant DF), `adresse` (contenant AA), `type` (contenant le Format Type Code-FTC), `nom` (contenant le nom de l'appareil), `altitude` (contenant l'altitude), `timeFlag` (contenant l'indicateur de temps UTC), `cprFlag` (contenant l'indicateur CPR), `latitude` (contenant la latitude) et `longitude` (contenant la longitude). La façon la plus simple d'initialiser cette structure est de de la forme suivante

```
registre = struct('adresse',[],'format',[],'type',[],'nom',[], ...
'altitude',[],'timeFlag',[],'cprFlag',[], ...
'latitude',[],'longitude',[], 'trajectoire', []);
```

Pour initialiser un des champs de cette structure vous utiliserez la syntaxe suivante

```
registre.nom = 'AF1234';
registre.altitude = 34000;
```

Tous les messages envoyés pour l'ADS-B ne comportent pas systématiquement toutes les informations. En effet, les informations transmises dépendent du type de registre en train d'être envoyé et donc de la partie "message" d'une trame ADS-B. La structure de cette partie "message" est donnée en Annexe pour des messages de position en vol, position au sol et d'identification de l'appareil.

**Sous-tâche 2** - *Matlab - Écrire la fonction MATLAB `bit2registre` qui prend en argument un vecteur de 112 bits et un registre à mettre à jour, qui extrait les informations du vecteur binaire et qui renvoie le registre mis à jour seulement si le CRC ne détecte pas d'erreur.*

TABLE 2 – Composition du message d'identification

Index binaire	Champs		
1	MSB	Format Type Code	
⋮			
5	LSB		
6	MSB	Catégorie de l'appareil	
⋮			
8	LSB		
9	MSB	Caractère 1 de l'identifiant	
⋮			
14	LSB		
15	MSB	Caractère 2 de l'identifiant	
⋮			
20	LSB		
21	MSB	Caractère 3 de l'identifiant	
⋮			
26	LSB		
27	MSB	Caractère 4 de l'identifiant	
⋮			
32	LSB		
33	MSB	Caractère 5 de l'identifiant	
⋮			
38	LSB		
39	MSB	Caractère 6 de l'identifiant	
⋮			
44	LSB		
45	MSB	Caractère 7 de l'identifiant	
⋮			
50	LSB		
51	MSB	Caractère 8 de l'identifiant	
⋮			
56	LSB		

## Vérification

**Sous-tâche 3 - Matlab** - Afin de tester votre code, utilisez le fichier `adsb_msgs.mat`. Ce fichier contient les 112 bits suivant le préambule de plusieurs trames issues d'un même appareil à la fréquence d'échantillonnage  $F_e = 4$  MHz (ces trames sont rangées en colonnes). Extrayez toutes les informations de ces trames et affichez la trajectoire de l'avion considéré. Vérifier que la trajectoire obtenue est comparable à celle représentée en Fig. 2.

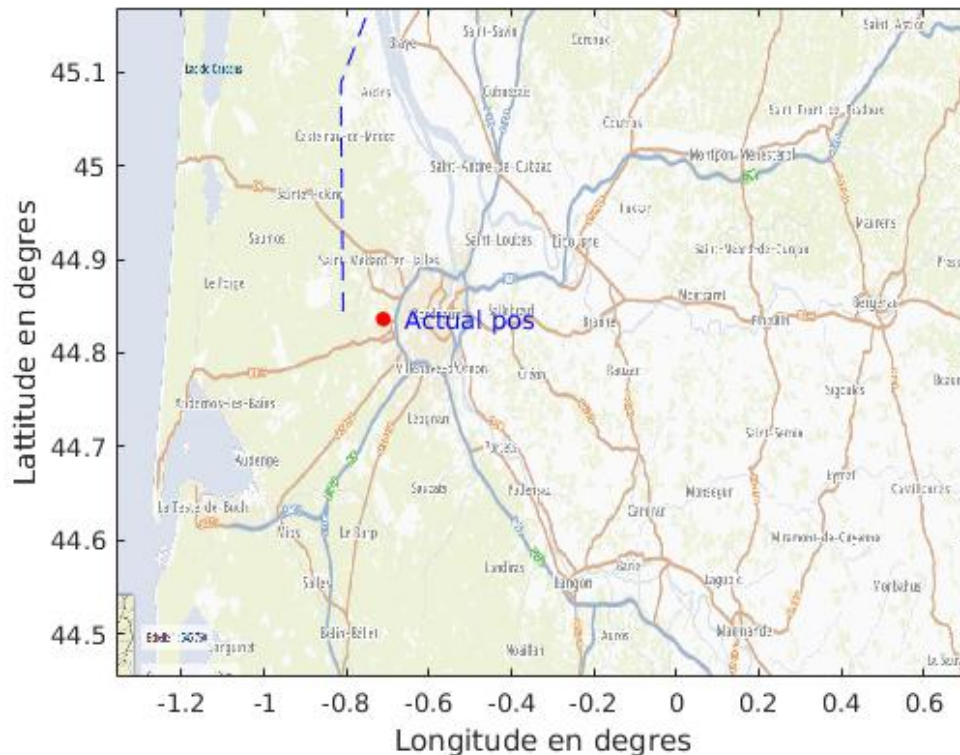


FIGURE 2 – Trajectoire obtenue à partir d'une trame

## Validation

Faites valider votre travail par votre encadrant afin de passer à la tâche suivante.