



# Décoder des signaux radio en temps réel avec **MATLAB** et une carte RTL-SDR

Les avions commerciaux transmettent aux systèmes de contrôle du trafic aérien des messages ADS-B (Automatic Dependent Surveillance – Broadcast) pour communiquer leur position, leur vitesse, leur altitude et autres données de vol. Cet exemple montre comment utiliser MATLAB® et les cartes de radio logicielle (SDR) pour capturer et traiter des données radio en temps réel en utilisant des signaux ADS-B.

Nous utilisons MATLAB, Communications System Toolbox™ et une carte RTL-SDR (basée sur le chipset Realtek RTL2832U) pour recevoir les signaux ADS-B sur un PC, puis détecter et décoder les messages de l'avion.

## Capture des signaux radio à l'aide de la carte RTL-SDR

Nombre des configurations de test pour la capture des signaux sans fil incluent des équipements de test qui coûtent des milliers de dollars. Ces configurations nécessitent souvent plusieurs étapes manuelles, d'abord pour transférer les données depuis l'instrument qui les a capturées vers un PC, puis pour reformater les données binaires en valeurs numériques prêtes à être traitées. A contrario, avec du matériel largement disponible et low-cost ainsi que deux commandes MATLAB, nous accédons aux échantillons IQ directement dans notre espace de travail MATLAB.

Une carte RTL-SDR est un matériel peu coûteux (environ 20 \$) qui peut se brancher sur un port USB. Il est doté d'une fréquence centrale configurable qui varie entre 50 MHz et 2 GHz, et sa bande passante maximale est d'environ 3 MHz. Communications System Toolbox intègre un support package complémentaire pour les cartes RTL-SDR (disponible en [téléchargement](#)), qui inclut des fonctions permettant de configurer la carte RTL-SDR et d'accéder aux données qu'elle reçoit.

La carte RTL-SDR peut capturer les signaux radio ADS-B, qui sont transmis à une fréquence de 1 090 MHz, et utilise la modulation de position d'impulsions pour transmettre les données à 1 Mbit/sec.

Nous configurons la carte RTL-SDR pour capturer les signaux ADS-B à l'aide d'une seule commande MATLAB :

```
RX = comm.SDRRTLReceiver('0','CenterFrequency',1090e6,...  
    'EnableTunerAGC',false,'TunerGain',60,'SampleRate',2.4e6,...  
    'OutputDataType','single','SamplesPerFrame',262144,...  
    'FrequencyCorrection',0);
```

Ensuite, nous effectuons une acquisition de données à partir de la carte RTL-SDR et créons une variable MATLAB, toujours à l'aide d'une seule ligne de code :

```
IQSamples = step(RX);
```

## Détection d'une transmission ADS-B

Après avoir capturé les signaux ADS-B et procédé à l'acquisition des données résultantes à partir de la carte RTL-SDR, nous créons un tracé temporel des données capturées (figure 1).

```
plot(abs(IQSamples));
```

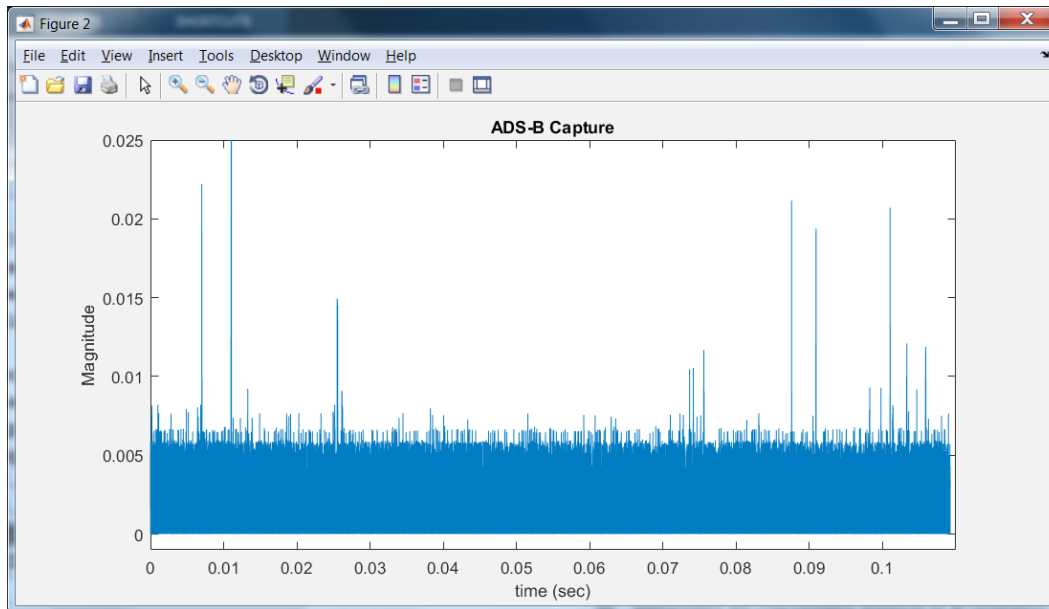


Figure 1. Tracé temporel des signaux radio capturés pendant 0,11 seconde, à l'aide d'une carte RTL-SDR et à une fréquence de 1 090 MHz.

À une fréquence de 1 090 MHz, nous capturons généralement un certain nombre de signaux de courte durée, ainsi qu'une quantité significative de bruit. La plupart des impulsions d'amplitude élevée illustrées en figure 1 correspondent à des formes d'onde existantes utilisées dans le cadre du contrôle du trafic aérien, mais en zoomant pour inspecter les différentes impulsions, nous pouvons constater que le signal généré à environ 0,025 seconde est un message ADS-B décodable (figure 2).

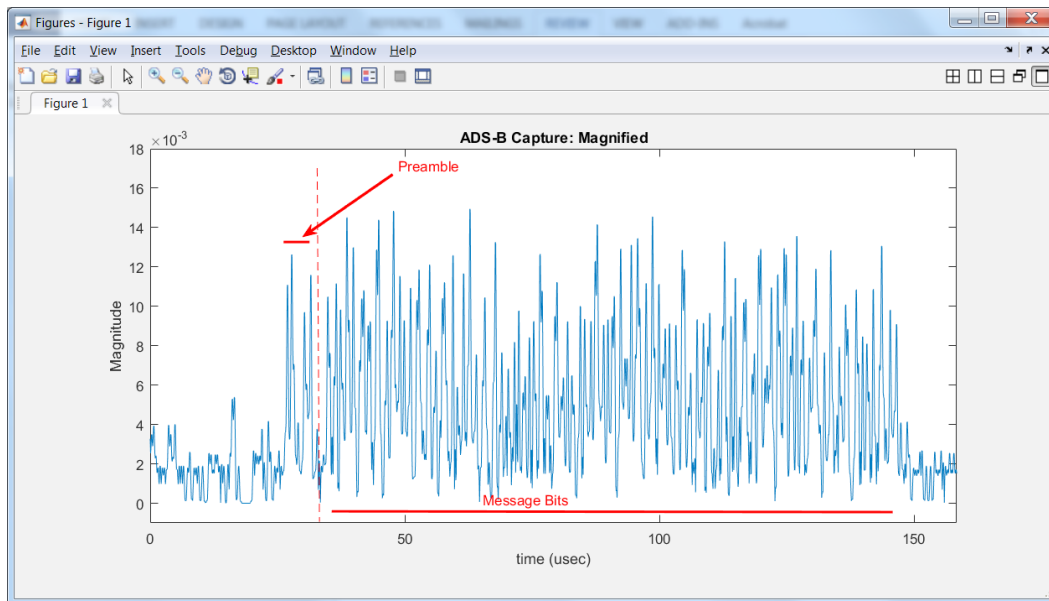


Figure 2. Tracé temporel d'un signal ADS-B.

Le signal ADS-B comprend un préambule de 8 microsecondes, suivi par 112 bits de message qui sont transmis dans les 112 microsecondes qui suivent (figure 2). Pour identifier les transmissions ADS-B potentielles et aligner notre algorithme de détection de bit sur le premier bit du message, nous devons corrélérer le signal capturé avec le modèle de préambule idéal. Pour commencer, nous suréchantillonnons le signal à 12 MHz à l'aide du System object FIRInterpolator de la DSP System Toolbox, puis nous corrélons les données avec le modèle de préambule connu pour déterminer l'alignement de timing correct.

```
% Suréchantillonnage des échantillons IQ par 5 afin d'améliorer
la résolution temporelle
interpFIR=dsp.FIRInterpolator(5);
data = step(interpFIR,IQSamples);

% Calcul de la corrélation avec le préambule ADS-B
(utiliser ce calcul pour identifier le bit de message n°1)
preamble = [ones(1,6) zeros(1,6) ones(1,6) zeros(1,24) ones(1,6)...
            zeros(1,6) ones(1,6)];
correlatedData = xcorr(preamble', flipud(double(abs(data))));
```

Les résultats sont illustrés à la figure 3. Le tracé du haut montre une partie du signal ADS-B suréchantillonné capturé, tandis que le tracé du bas montre le résultat de la corrélation avec le préambule. Dans le tracé du bas, le pic au niveau de l'échantillon 131 indique la plus forte correspondance avec le modèle de préambule dans le signal ADS-B capturé. Étant donné que le modèle que nous utilisons pour la mise en corrélation du préambule a une longueur de 60 échantillons,

et que nous savons que les octets du message ADS-B commencent 36 échantillons après le pic de corrélation, nous pouvons identifier le début des octets de message ADS-B à l'échantillon numéro 167.

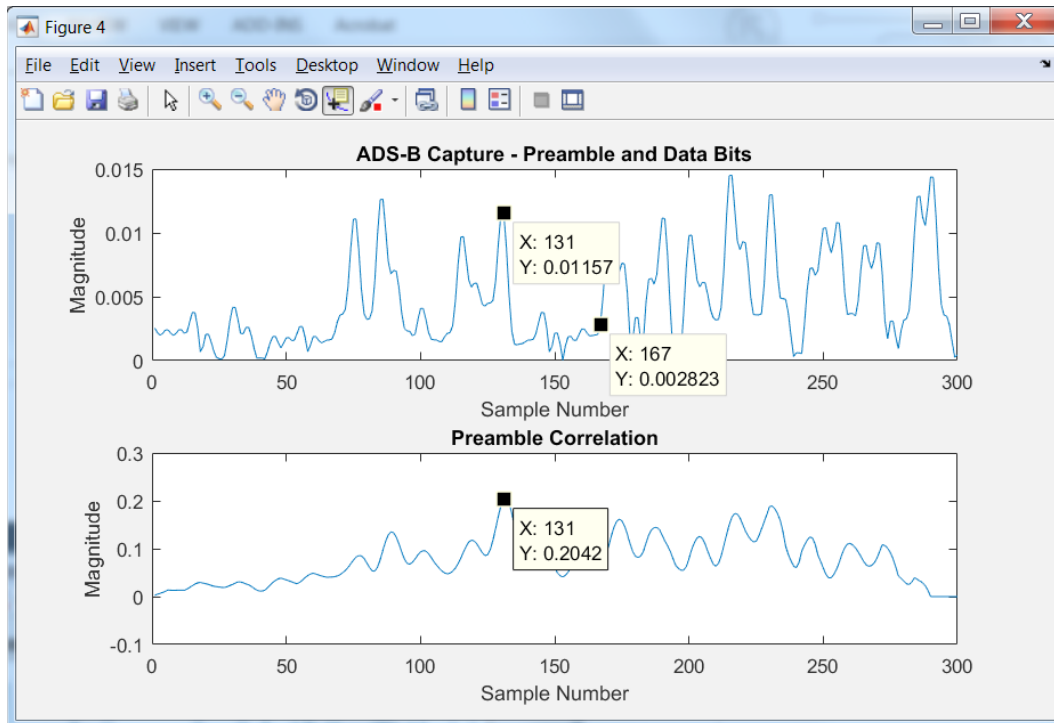


Figure 3. Tracés montrant les signaux ADS-B suréchantillonnés capturés (tracé du haut) et la mise en corrélation avec le préambule ADS-B (tracé du bas).

## Décoder le message ADS-B

Les décisions binaires pour les 112 bits de message sont prises en effectuant une correspondance de modèle des 12 échantillons pour chaque bit de données avec les deux séquences de bits possibles. Les caractères hexadécimaux qui en résultent pour les décisions binaires sont indiqués ci-dessous :

```
rxMsg : 8D A7AD0D 60378784F313A7 3C92F4
```

Les six derniers caractères de ce message (3C92F4) sont une somme de contrôle que nous utilisons pour déterminer si les 88 bits précédents forment un message valide. Nous réalisons le test à l'aide de la fonction polynomiale de la somme de contrôle ADS-B et du System object CRCGenerator de la Communications System Toolbox.

```
g = logical
([1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 1]);
crcADSB = comm.CRCGenerator(g);

y = step(crcADSB,m)';
```

Si le calcul de la somme de contrôle confirme qu'il s'agit d'un message valide, nous extrayons les champs de données et interprétons les valeurs. Dans ce message ADS-B, 24 bits représentent l'identifiant de l'avion, 11 bits représentent l'altitude, 17 bits représentent la latitude et 17 bits représentent la longitude. Nous vérifions ce message décodé en le comparant aux informations de vol sur [flightradar24.com](http://flightradar24.com), et nous nous assurons que nous avons capturé les données correspondant à un avion volant à proximité du siège de MathWorks à Natick, Massachusetts.

L'avion portant l'identifiant A7AD0D se trouve à une altitude de 11 000 pieds

L'avion portant l'identifiant A7AD0D se trouve à une latitude de 41 58 43.0 et à une longitude de 70 50 29.2

Le code ci-dessus constitue la base d'un programme de suivi aérien en direct, qui est fourni à titre d'exemple dans le support package de la carte RTL-SDR. La figure 4 représente une capture de l'affichage du suivi.

### ADSBAirplaneTrackingwithRTLSDR

	Current	Aircraft ID	Flight ID	Latitude(deg)	Longitude(deg)	Altitude(ft)	Speed(knots)	Direction	U/D vel(ft/min)	Time
1		A33B52	N307W...	42.2839	-71.2920	7600	192	SW (237)	1024	16:59:52
2		4BA9D4	THY1	42.2270	-71.3155	39250	447	SW (206)	-960	16:58:33
3	✓	AD3997	AAL147	42.3696	-71.1096	17075	369	W (268)	2240	16:59:53
4		ACB1CB								16:59:05
5		A2AD23								16:57:25
6		A5AC43								16:59:50
7		A3B86C								16:59:53
8		000000								16:58:50
9										
10										
11										
12										
13										
14										
15										

Figure 4. Données de suivi aérien en direct obtenues à l'aide de ADSBAirplaneTrackingwithRTLSDR.m.

## Conclusion

La carte RTL-SDR est un système d'entrée de gamme utile pour expérimenter les fonctions SDR de Communications System Toolbox. Outre les signaux ADS-B, de nombreux autres signaux radio peuvent être capturés et traités grâce à cette association de matériels et de logiciels, notamment les signaux radio AM et FM et les signaux GSM 2G, UMTS 3G et LTE 4G. Un ouvrage en accès libre, intitulé « *Software Defined Radio Using MATLAB and Simulink and the RTL-SDR* » explore ces sujets de façon plus détaillée, avec des exemples pratiques basés sur MATLAB et Simulink®.

Le même processus de travail et la même syntaxe MATLAB que ceux qui sont utilisés dans cet exemple peuvent être appliqués à des cartes SDR plus complexes, notamment Ettus Radio USRP, Avnet PicoZed SDR et les cartes Zynq Radio basées sur FMCOMMS3. Ces options SDR peuvent être utilisées pour des applications nécessitant des bandes passantes plus élevées, des plages de réglage plus larges ou le support du MIMO (Multiple-Input Multiple-Output). Découvrez et téléchargez les supports packages hardware pour Communications System Toolbox :

- *Support Package URSP*
- *Support Package FPGA Radio*
- *Support Package Zynq SDR*

## Produits utilisés

*MATLAB*

*Communications System Toolbox*

*DSP System Toolbox*