

UTILISATION DES "TAGS" POUR SYNCHRONISER UN FLUX DE DONNÉES; APPLICATION AU DÉCODAGE DES SIGNAUX ADS-B.

Thomas Lavarenne¹

¹ Lycée Blaise Cendrars SEVRAN (93)

Nous proposons ici l'utilisation d'une fonctionnalité de GNU Radio pour synchroniser un flux de données par comparaison avec une suite de valeurs précisées par l'utilisateur. Le logiciel ajoute un "Tag" au flux de données permettant ainsi sa synchronisation sur la fenêtre de visualisation ainsi que la possibilité de déclencher l'enregistrement d'un fichier sur le disque dur, facilitant ainsi le traitement ultérieur de décodage. Nous verrons que cette méthode s'applique particulièrement bien pour l'étude des signaux ADS-B.

Mots-clés: Tags, ADS-B.

1 Introduction

Les signaux ADS-B ont été largement étudiés et de nombreuses ressources utilisant GNU Radio sont disponibles sur le net sur le sujet. L'intérêt de cette démarche est de présenter l'utilisation des blocs "Correlate Access Code - Tag", "Burst Tagger" et "Tagged File Sink" permettant d'obtenir au final un fichier contenant uniquement les données qui nous intéressent, ce qui facilitera grandement le traitement informatique de décodage.

2 Configuration expérimentale

L'ADS-B se réceptionne sur la fréquence de 1090 MHz à l'aide de la clé DVB-T RTL2832u. Les données sont envoyées à un débit de 1 Mbit.s^{-1} . Une fréquence d'échantillonnage (Samp_rate) de 2 MHz permettra d'obtenir exactement deux échantillons par bit, permettant ainsi de décoder les informations (encodage Manchester).

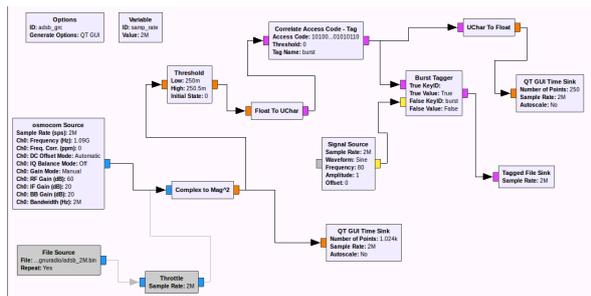


Figure 1: Graphe GNU Radio Companion pour l'acquisition et le traitement des trames ADS-B.

Le bloc "Threshold" définit une valeur seuil afin d'obtenir un flux de données uniquement de 1 ou de 0. Ce flux est converti en "unsigned char" afin de pouvoir être traité par le bloc "Correlate Access Code - Tag". C'est ce bloc qui réalise la corrélation entre le flux de données et le code indiqué par l'utilisateur. Si la suite de valeur indiquée est re-

connue, le bloc attache un "Tag", ici "burst", à la fin de la suite de valeur trouvée. Après reconversion en type "float", l'affichage du signal s'effectue sur le bloc "QT GUI Time Sink" en choisissant le type de synchronisation sur "Tag" et indiquant la valeur "burst" précédemment indiquée. Le code est choisi d'après la documentation de la trame [1] (premières valeurs attendues).

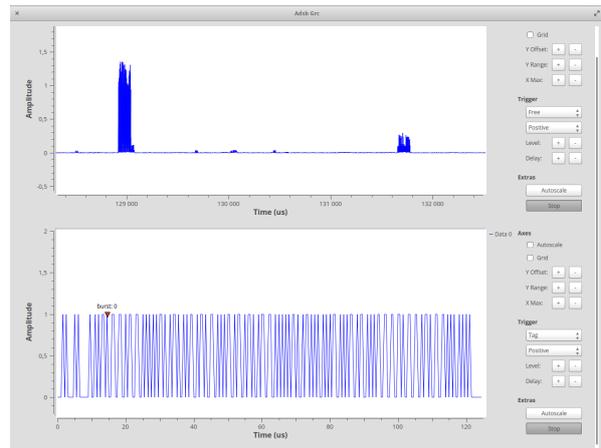


Figure 2: En-haut: les trames ADS-B sans aucune synchronisation (sortie du bloc "Complex To Mag²"). En-bas: trame synchronisée faisant apparaître le Tag "burst".

Le bloc "Burst Tagger" associé au bloc "Tagged File Sink" permet l'enregistrement des trames dans des fichiers séparés, la première valeur est celle suivant immédiatement le tag.

Le traitement s'effectue ensuite avec un script python qui extrait les données du fichier binaire et les traite en utilisant les informations données dans la documentation [1].

```

#Chemin des dossiers home et adsb
Chemin_home='/home/lavarenne'
Chemin_dossier_adsb='/home/lavarenne/gnuradio/adsb'

while(True):
    time.sleep(2)

    #déplacer les fichiers trames dans le dossier adsb
    os.system('mv {0}/file* {1}/'.format(Chemin_home,Chemin_dossier_adsb))

    #lecture du nom des fichiers et insertion dans la liste fichier
    fichier=[]
    for files in os.listdir(Chemin_dossier_adsb):
        fichier.append(files)

    # boucle sur le nbr de fichiers dans le dossier adsb
    for k in range(0,len(fichier)):

        # Ouvrir le fichier binaire et afficher la trame
        with open('{0}/{1}'.format(Chemin_dossier_adsb,fichier[k]), "rb") as binary_file:
            data = binary_file.read()
            byte = map(ord,data[:10*2])
            trame=[]
            trame_dec=""
            try:
                for i in range(0,10):
                    trame.append(byte[2*i])
                    trame_dec+=str(trame[i])
            except:
                pass

            print ''
            print '.....'
            print ''

            print "la trame binaire est:"
            msg="100001"+trame_dec
            print msg

```

Figure 3: Partie du script python permettant l'affichage de la trame complète.

Une méthode simple pour obtenir les valeurs binaires est de ne garder qu'un échantillon sur deux. En effet, le signal enregistré contient deux échantillons par bit: 01 pour la valeur 0 ou 10 pour la valeur 1.

3 Résultats

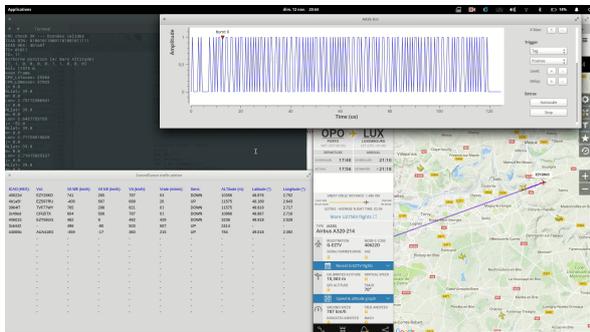


Figure 4: Décodage des trames et comparaison temps réel sur le site www.flightradar.com.

Le script peut être amélioré pour afficher un récapitulatif des informations décodées en temps réel sur une petite interface graphique.

La comparaison avec les informations données par le site www.flightradar.com permet de valider le bon fonctionnement de notre décodage.

4 Conclusion

Une simple clé DVB-T équipée du chipset RTL2832u, associée à la puissance de GNU Radio permet de réceptionner de manière très efficace les trames ADS-B. Le traitement du flux de données par GNU Radio lui-même permet de réduire au minimum la partie décodage, les trames binaires sont quasiment enregistrées directement dans un fichier en sortie du logiciel (il faut simplement utiliser la règle d'encodage considérée). L'avantage de cette démarche est de pouvoir être adaptée lors d'autres études, comme le décodage des trames ACARS (Aircraft Communication Addressing and Reporting System) ou RDS (Radio Data System).

References

- [1] The 1090MHz Riddle. An open-access book about decoding Mode-S and ADS-B data <http://mode-s.org/decode/>