maker

Fabriquer son récepteur ADS-B avec un Raspberry Pi

Allongé dans l'herbe par ce beau jour d'été, je contemple les traînées laissées sur l'azur du ciel par les avions qui me survolent... Où peuvent-ils bien aller ? Bon sang, il doit bien y avoir un moyen de le savoir ! Et c'est parti : après quelques recherches sur Internet, je farfouille dans la boîte à bricoles j'en en sors une clé TNT avec son antenne et un Raspberry Pi. Prêt pour recevoir les infos ADS-B en direct...



François MOCQ Créateur du blog www.framboise314.fr Auteur du livre Raspberry Pi 2 - Editions ENI

Objectif de cet article

Le but de cet article est de fabriquer un récepteur ADS-B avec un dongle TNT et un Raspberry Pi. Avec ce récepteur vous pourrez bien entendu connaître la position mais aussi d'autres paramètres de vol des avions qui circulent dans votre zone. Les données pourront s'afficher en mode console sur votre écran, ou sur une page Web affichée sur un autre poste ou un smartphone. Mais l'histoire ne s'arrête pas là, puisque si vous vous inscrivez auprès de Flightradar24 et que vous leur envoyez les données recueillies, vous bénéficierez du téléchargement gratuit de l'appli – elle ne coûte que 3,59€ - pour votre smartphone ainsi que d'un compte premium dont nous verrons les avantages plus loin.

ADS-B

Chaque avion de ligne embarque aujourd'hui un transpondeur ADS-B (Automatic Dependent Surveillance-Broadcast). La position exacte de l'appareil est déterminée grâce à un récepteur GPS. A ces données de position (lattitude, longitude et altitude) sont ajoutés le numéro du vol, l'indicatif de l'avion, les informations de vitesse et de vitesse ascensionnelle, le cap...

Ces données sont transmises sous forme de « burst » binaire (112 bits) non crypté sur 1090MHz. Dans ce mode S, les informations sont envoyées régulièrement en fonction de la phase de vol, par exemple toutes les dix secondes en vol normal et toutes les secondes en phase d'approche. Ces émissions peuvent être reçues par des stations au sol ou par les autres avions. Chaque avion peut ainsi connaître avec précision la position des appareils qui l'entourent et utiliser ces informations pour prévenir les collisions.

L'émission de ces données est omnidirectionnelle et peut être recue facilement au sol par un système simple, muni d'une antenne et d'un récepteur. Cette installation permet de suivre les avions en vol sans nécessiter un radar beaucoup plus coûteux Fig.1.

Récepteur TNT USB

Des récepteurs TNT = Télévision Numérique terrestre (appelés en anglais DVB-T = Digital Video Broadcasting - Terrestrial) sont disponibles sur les sites de vente en ligne (photo 1) pour des prix allant de quelques euros à une vingtaine d'euros. Je disposais de plusieurs exemplaires suite à des essais de réception TV finalement pas très heureux... Mais mes dongles TNT allaient enfin pouvoir reprendre vie !

Pour arriver à construire des récepteurs vendus à un prix aussi bas, les fabricants ont recours à des composants spécialisés et fortement intégrés. C'est ainsi qu'une fois ouvert (photo 2), un dongle TNT ne laisse apparaître que trois circuits intégrés, accompagnés de quelques composants annexes Fig.2.

Ces récepteurs sont vendus pour recevoir les bandes radio FM (87,5 à



108 MHZ), TV TNT en UHF (470 à 790 MHz). Enfin... ça c'est la version officielle. Parce que dans la réalité, les composants mis en oeuvre permettent de couvrir une gamme de fréquences beaucoup plus large que celle qui est annoncée. Et c'est bien ce qui nous intéresse puisque la fréquence ADS-B à recevoir se situe au delà de 1 GHz. Avec le tuner R820T utilisé sur ce dongle, la réception des fréquences comprises entre 24 MHZ et 1850 MHz devient possible, ouvrant la voie à de nombreuses applications SDR (Software Defined Radio = Radio logicielle). Le tuner R820T se charge de la réception des signaux qu'il transmet au décodeur RTL2832U, chargé de les numériser et de les transmettre sous forme d'octets via le port USB jusqu'à l'ordinateur (PC ou dans notre cas, Raspberry Pi !). Celui-ci se charge alors du traitement des échantillons en démodulant les signaux et/ou en calculant le spectre des signaux.reçus. Avec cette méthode, l'ordinateur est capable de traiter un spectre (un ensemble de signaux) s'étendant sur plusieurs mégaherz.

Réception des signaux ADS-B avec le Raspberry Pi

Les composants du récepteur TNT

Pour recevoir les signaux ADS-B avec le Raspberry Pi et un dongle DVB-T USB, l'antenne livrée avec le dongle suffit. Il ne faudra pas attendre des résultats extraordinaires mais les premiers essais que j'ai réalisés avec l'antenne posée sur un appui de fenêtre au premier étage ont permis la réception d'avions situés à près de 150 Km. Il faut dire que le ciel est particulièrement dégagé vers le sud. Il n'y a ni bâtiment proche, ni obstacle naturel (colline par exemple) qui pourrait empêcher la réception des signaux. A cette fréquence, même si on constate des réflexions et de la dispersion, la meilleure réception a lieu quand les deux antennes



liste du matériel

Pour le matériel :

- 1 Raspberry Pi 2
- 1 boîtier pour le Raspberry Pi 2
- 1 alimentation 5V / 2A avec connecteur micro USB
- 1 carte SD 16Go Samsung
- 1 clé DVB-T modèle SPC-0155
- 1 antenne (livrée avec la clé USB)

sont à vue. Enfin, quand l'avion est à 100 Km, on ne voit pas grandchose, mais les ondes, elles, passent bien. Le décodeur RTL2832U transforme les signaux reçus en données de la forme I/Q, I représentant l'amplitude instantanée du signal et Q sa phase. Au risque de gacher vos vacances d'été, je vous rappellerai juste que la porteuse modulée peut s'écrire sous la forme I cos (f.t) + Q sin (f.t). Ça pique un peu non ?

Choisir le système :

Il y a deux possibilités pour préparer la carte microSD. La première est une installation directe de Raspbian. Elle nécessite la mise en oeuvre de Win32DiskImager pour la création de la carte et le transfert d'une image comportant deux partitions, une partition FAT et une partition ext4. Vous choisirez cette option si vous êtes à l'aise avec les outils utilisés. Pour les débutants NOOBS est le bon choix. Il ne nécessite pas de connaissance particulière et s'installe par simple copie des fichiers sur la carte microSD.

Préparer la Carte SD :

Choisissez une carte micro SD de bonne qualité. J'utilise pour ma part des Samsung EVO 16 Go qu'on trouve pour moins de 10^o. Evitez les cartes microSD "pas chères" trouvées sur certains site d'enchères en ligne. De nombreux utilisateurs de ces cartes à la capacité inférieure à la valeur annoncée ont connu des déboires lors de l'installation de NOOBS, celui-ci indiquant que la capacité est insuffisante pour l'installation. Le



message d'erreur "Available Space : 0" vous alertera sur ce problème. SI vous récupérez une carte SD précédemment utilisée pour équiper votre récepteur ADS-B, la Fondation recommande de formater la carte avec SD Formatter 4.0 (https://www.sdcard.org/downloads/formatter_4/) avant d'installer NOOBS Fig.3.

Rendez vous sur la page de téléchargement de la Fondation (https://www.raspberrypi.org/downloads/). NOOBS existe en deux versions. La version "Offline and network install" représente 742 Mo de téléchargement. Elle embarque Raspbian et permet d'installer le système sur un Raspberry Pi non relié à Internet, ou sur un modèle A+ qui ne dispose pas de prise Ethernet. La version "Network install only" ne représente que 21,8 Mo de téléchargement, mais elle ne peut être utilisée que sur un Raspberry Pi connecté à Internet car cette image ne contient aucun système et doit télécharger les systèmes à installer. Choisissez la version qui correspond le mieux à vos besoins. Une fois le fichier .zip récupéré, décompressez l'archive dans un dossier. Ouvrez le dossier et copiez les fichiers qui s'y trouvent sur votre carte microSD. Votre système est prêt à démarrer !

Démarrer le Raspberry Pi

Connectez le clavier, la souris et l'écran HDMI. Branchez l'alimentation, le Raspberry Pi démarre et affiche son écran coloré. NOOBS se lance. Vous aboutissez sur la fenêtre de choix du système d'exploitation à installer (photo 3). Cochez la case figurant au début de la ligne de Raspbian, choisissez Français pour la langue du système et du clavier (en bas de l'écran), puis cliquez sur l'icône **Install** en haut à gauche. Servez-vous une bière bien fraîche et installez vous confortablement... Il y en a pour un moment **Fig.4**.

Quand le processus d'installation est achevé, Raspbian démarre et vous atterrissez sur l'écran de configuration de Raspbian, **raspi-config**. La première ligne est sélectionnée. Elle permet à Raspbian d'occuper toute la place disponible lorsque vous l'installez seul sur une carte. Avec NOOBS, cette option n'est pas disponible. Déplacez vous sur <**Finish>** avec la touche de tabulation, puis validez ce choix avec la touche Entrée. Vous arrivez sur la console. Pour ce premier démarrage vous n'avez pas besoin de vous logger. Par la suite vous utiliserez le login **pi** et le mot de passe **raspberry**. Si vous souhaitez relancer l'utilitaire de configuration de Raspbian, exécutez la commande **sudo raspi-config**. Commencez par mettre votre système à jour:

pi@raspberrypi ~ \$ sudo apt-get update pi@raspberrypi ~ \$ sudo apt-get upgrade

Connecter le Dongle TNT



Brancher votre dongle USB TNT sur un des ports USB du raspberry Pi. Pour vérifier que votre dongle est bien détecté, tapez la commande **dmesg**. Parmi les lignes qui apparaissent, repérez les lignes suivantes qui indiquent que le récepteur TNT a bien été détecté, puisque les circuits intégrés RTL2832U et R820T sont mentionnés. Nous allons pouvoir continuer l'installation.

[44.837951] usb 1-1.2: new high-speed USB device number 5 using dwc_otg

- [45.040538] usb 1-1.2: dvb_usb_v2: found a 'Realtek RTL2832U reference design' in warm state
- [45.103415] DVB: registering new adapter (Realtek RTL2832U reference design)
- [45.147362] r820t 0-001a: Rafael Micro r820t successfully identified
- [45.179781] IR RC5(x/sz) protocol handler initialized
- [45.222421] usbcore: registered new interface driver dvb_usb_rtl28xxu

En fait Raspbian a installé le pilote permettant de regarder la télévision avec le dongle DVB-T (TNT). Pour qu'il ne vienne pas perturber la réception des signaux ADS-B, nous allons le placer en liste noire (blacklist) : Créez un fichier **rtlsdr.conf** dans le répertoire /etc/modprobe.d

pi@raspberrypi ~ \$ cd /etc/modprobe.d pi@raspberrypi /etc/modprobe.d \$ sudo nano rtlsdr.conf

Une fois que nano est ouvert ajouter la ligne suivante dans le fichier :

blacklist dvb_usb_rtl28xxu

Ensuite sauvegardez le fichier **rtlsdr.conf** avec **CTRL X**. Ainsi blacklisté, le module ne sera plus chargé par le système. Vérifiez si le pilote est chargé avec la commande **lsmod** :

pi@raspberrypi ~\$ lsmod | grep dvb_usb_rtl28xxu

 dvb_usb_rtl28xxu
 14195 0

 rtl2830
 7227 1 dvb_usb_rtl28xxu

 rtl2832
 8911 2 dvb_usb_rtl28xxu

 dvb_usb_v2
 13368 1 dvb_usb_rtl28xxu

 rc_core
 16932 14 ir_sharp_decoder,ir_xmp_decoder,lirc_dev,ir_lirc_codec,dvb_

 usb_rtl28xxu,ir_rc5_decoder,ir_nec_decoder,ir_sony_decoder,ir_mce_kbd_decoder,ir_jrc_decoder,dvb_usb_v2,ir_rc6_decoder,ir_sony_decoder

Le pilote est effectivement chargé puisque nous avons connecté le dongle TNT. Il faut le supprimer :

pi@raspberrypi ~ \$ sudo modprobe -r dvb_usb_rtl28xxu

Installation du pilote rtl-sdr

Les dongles comportant un décodeur de type RTL2832U sont gérés par le logiciel **rtl-sdr**. Ce logiciel développé par un groupe de passionnés permet de récupérer directement les données du signal I/Q, ce qui transforme votre dongle USB TNT en récepteur radio à large bande au moyen d'un nouveau pilote. En clair, cela signifie qu'un dongle USB TNT peut remplacer pour une vingtaine d'euros un scanner qui coûtait plusieurs centaines d'euros il y a quelques années... La vitesse maximale d'échantillonnage est aux alentours de 2,4 Me/s (méga échantillons par seconde). C'est à partir de ces données que le signal est reconstitué. Pour installer le logiciel, nous avons besoin des outils de construction et de la librairie **libusb** pour que les applications puissent accéder facilement aux périphériques USB.

pi@raspberrypi ~ \$ sudo apt-get -fym install git cmake build-essential libusb-1.0-0-dev

Maintenant nous pouvons récupérer rtl-sdr sur github et le compiler sur notre Raspberry Pi.

pi@raspberrypi ~ \$ cd git pi@raspberrypi ~/git \$ git clone git://git.osmocom.org/rtl-sdr.git Cloning into 'rtl-sdr' ... remote: Counting objects: 1607, done. remote: Compressing objects: 100% (701/701), done. remote: Total 1607 (delta 1174), reused 1213 (delta 898) Receiving objects: 100% (1607/1607), 344.85 KiB | 10 KiB/s, done. Resolving deltas: 100% (1174/1174), done. pi@raspberrypi ~/git \$ cd rtl-sdr pi@raspberrypi ~/git/rtl-sdr \$ mkdir build pi@raspberrypi ~/git/rtl-sdr \$ cd build pi@raspberrypi ~/git/rtl-sdr/build \$ cmake .. -DINSTALL_UDEV_RULES=ON -- The C compiler identification is GNU 4.6.3 -- Check for working C compiler: /usr/bin/gcc -- Check for working C compiler: /usr/bin/gcc – works .../... pi@raspberrypi ~/qit/rtl-sdr/build \$ sudo make install Scanning dependencies of target convenience_static

pi@raspberrypi ~ \$ mkdir git

[5%] Building C object src/CMakeFiles/convenience_static.dir/convenience/convenience.co Linking C static library libconvenience_static.a

pi@raspberrypi ~/git/rtl-sdr/build \$ **sudo ldconfig** pi@raspberrypi ~/git/rtl-sdr/build \$ **sudo cp ../rtl-sdr.rules /etc/udev/rules.d**/

Le pilote rtl-sdr est installé et nous pouvons passer à l'installation de dump1090.

Installation du logiciel Dump1090

Le logiciel **dump1090** est un logiciel de décodage des signaux ADS-B mode S spécialement destiné à être utilisé avec les périphériques USB gérés par **rtl-dsr**. Il est capable de décoder des signaux faiblement reçus sur 1090 MHz. Il supporte le protocole TCP30003 qui permet de retransmettre les messages à FlightRadar24, par exemple. Il met à disposition de l'utilisateur un serveur HTTP qui affiche les avions détectés sur une carte Google Map mais peut afficher les données en mode console. Dump1090 existe en version Windows et heureusement pour nous, également en version Raspberry Pi !

Pour installer le logiciel dump1090, nous allons le récupérer sur github et le compiler :

pi@raspberrypi ~ \$ cd ~/git

pi@raspberrypi ~/git \$ git clone git://github.com/MalcolmRobb/dump1090.git Cloning into 'dump1090'... remote: Counting objects: 1401, done. remote: Total 1401 (delta 0), reused 0 (delta 0), pack-reused 1401 Receiving objects: 100% (1401/1401), 4.13 MiB | 329 KiB/s, done. Resolving deltas: 100% (837/837), done. pi@raspberrypi ~/git \$ cd dump1090 pi@raspberrypi ~/git/dump1090 \$ make gcc -02 -g -Wall -W `pkg-config --cflags librtlsdr` -c dump1090.c

.../...

.../...

Tout est prêt pour lancer le logiciel dump1090 et récevoir les avions qui se trouvent dans les parages :

pi@raspberrypi ~/git/dump1090 \$ **./dump1090 --quiet --net --enable-agc &** [1] 2325 pi@raspberrypi ~/git/dump1090 \$ Found 1 device(s):



0: Realtek, RTL2838UHIDIR, SN: 00000001 (currently selected) Found Rafael Micro R820T tuner Max available gain is: 49.60 Setting gain to: 49.60 Exact sample rate is: 2000000.052982 Hz Gain reported by device: 49.60

Connectez-vous depuis un navigateur sur l'adresse de votre Raspberry Pi au port 8080, vous obtenez une carte indiquant les avions en vol près de chez vous (Fig.4). Leur position est actualisée à chaque réception de trame ADS-B et si vous cliquez sur un avion, vous affichez les données le concernant. Vous pourrez trouver plus d'informations le concernant en cliquant sur les liens vers FlightRadar24, FlightStats ou FlightAware. Fig.5.

Personnaliser dump1090

Si vous souhaitez adapter dump1090 à vos besoins, le fichier *config.js* situé dans le répertoire *public_html* de dump1090 vous permet de régler successivement:

- l'utilisation des valeurs métriques ou de miles nautiques,
- le centre de la carte (latitude longitude et zoom)
- la couleur des marqueurs (MarkerColor)
- l'emplacement du repère (SiteSHow, SiteLat et SiteLon)
- la présence de cercles indiquant la distance des avions

Le fichier config.js ci-dessous contient mes propres réglages.

// -----

//

// -- Valeurs de sortie -----// Affichage métrique : true pour afficher en km Metric = true; // true ou false

// -- Configuration de la carte ------// Latitude et Longitude en format décimal // Indique le centre de la carte qui sera affichée // J'ai choisi de centrer sur mon antenne ADS-B CONST_CENTERLAT = 46.799107; CONST_CENTERLON = 4.418899; // Niveau de zoom google map de 0 à 16, 0 est le plus éloigné CONST_ZOOMLVL = 5;

// -- Paramètres des marqueurs ------// Couleur par défaut des marqueurs MarkerColor = "rgb(127, 127, 127)"; SelectedColor = "rgb(225, 225, 225)"; StaleColor = "rgb(190, 190, 190)";

// -- Dessin des cercles donnant la distance des avions ------// Les cercles ne sont affichés que si SiteShow est sur true SiteCircles = true; // true ou false // En miles nautiques ou en km (selon la valeur de 'Metric') // J'ai choisi d'afficher des cercles à 50km, 100km, 150km et 200km SiteCirclesDistances = new Array(50,100,150,200);

Protéger l'installation

A défaut de trouver une ancienne boîte de glace à la framboise qui aurait harmonieusement complété l'ensemble, j'ai "investi" dans une boîte étanche achetée un peu plus de 2€ dans une grande surface. Les trous percés pour laisser passer les câbles ont été colmatés avec du silicone utilisé en plomberie. La fixation des différents éléments a été faite avec un adhésif double face prévu pour une utilisation à l'extérieur (**Fig.4**).

Envoyer des données à Flightradar24

Pourquoi envoyer les données à FlightRadar24 (ou à un autre service de radar virtuel ?) : pour partager les informations recueillies. Elles seront affichées sur le site de FlightRadar24. Vous bénéficierez aussi de l'option premium qui vous permet d'accéder à des données supplémentaires **Fig.6.** Sur la capture d'écran de smartphone (photo 6) vous voyez de gauche à droite : la carte avec les avions en vol, un des avions est sélectionné et sa trajectoire est affichée. Le type de l'avion, sa destination et un certain nombre de paramètres. Par exemple ici, un Airbus A319 de EasyJet va se poser à Bâle et il est en phase de descente (-1024 fpm = - 312 m/minute). Le paysage tel que peut le voir un passager de l'avion (s'il n'y a pas de nuages...).

Pour fournir vos données à FlightRadar24, vous devez envoyer un flux de données compatible TCP30003. Lancez cette fois le logiciel dump1090 avec les paramètres suivants :

./dump1090 --quiet --net --net-sbs-port 30003 --enable-agc &

Il faut maintenant installer le logiciel de FlightRadar24 pour transférer les données :





pi@raspberrypi ~ \$ **mkdir fr24**

pi@raspberrypi ~ \$ cd fr24

pi@raspberrypi ~/fr24 \$ wget https://dl.dropboxusercontent.com/u/66906/fr24feed_arm-rpi_242.tgz --2015-06-03 20:24:39-- https://dl.dropboxusercontent.com/u/66906/fr24feed_arm-rpi_242.tgz Résolution de dl.dropboxusercontent.com (dl.dropboxusercontent.com)... 174.129.251.235, 184.73. 223.97, 174.129.255.54, ...

Connexion vers dl.dropboxusercontent.com

(dl.dropboxusercontent.com)|174.129.251.235|:443...connecté.

requête HTTP transmise, en attente de la réponse...200 OK

Longueur: 276631 (270K) [application/x-gtar]

Sauvegarde en : «fr24feed arm-rpi 242.tgz»

100%[======>] 276 631 441K/s ds 0,6s 2015-06-03 20:24:44 (441 KB/s) - «fr24feed_arm-rpi_242.tgz» sauvegardé [276631/276631] pi@raspberrypi ~/fr24 \$ tar zxvf fr24feed_arm-rpi_242.tgz fr24feed_arm-rpi_242

Le transfert de données vers FlightRadar24 nécessite une clé qui vous sera délivrée après votre inscription :

pi@raspberrypi ~/fr24 \$./fr24feed_arm-rpi_242 -signup

Remplissez le questionnaire. Préparez vous à fournir les coordonnées décimales de l'endroit où vous vous trouvez. Une fois que vous aurez reçu la clé (la mienne est remplacée par des x ;)), vous pouvez partager vos données :

[i]FR24Feed v242 - built on Aug 15 2014/08:32:38 [i]Downloading configuration...OK [i]Parsing configuration...OK [c]Interval: 5s [c]Latitude: 46.8000 [c]Longitude: 4.4200 [c]GND: YES [c]NonADSB: YES [i][stats]Stats thread started [n]defined 1 server [i]Source defined as Basestation [n]LFLH1@83.140.21.66:8099/UDP [n]connecting [b]connecting to 127.0.0.1:30003 [b]connected [b]working [n]connected [n]switching to UDP [n]working [i]Data feed time difference OK abs(20:32:39.080 - 20:32:39.042)=0 [i]sent 1 planes in 1 packets [i]sent 1 planes in 1 packets [i]sent 1 planes in 1 packets [n]pinging the server [i]sent 1 planes in 1 packets

Après une phase d'initialisation votre système est opérationnel et commence à remonter des informations vers FlightRadar24 comme l'indiquent les dernières lignes. Vous bénéficiez de l'option premium à partir de cet instant et elle sera maintenue quelques jours si vous cessez d'envoyer des données au serveur.

Si vous souhaitez que votre Raspberry Pi redémarre automatiquement sur dump1090 après sa mise sous tension, il suffira d'écrire un script de lancement de dump1090 et de fr24 et de le faire exécuter lors du démarrage de Raspbian.

Améliorer le système

L'antenne et son câble

L'antenne est un élément clé de ce genre de récepteur. Le petit morceau d'antenne livré avec le dongle TNT permet certes de recevoir quelques avions. Un proverbe radioamateur dit :"Si tu achètes un récepteur à 1000€, achète une antenne à 1000€ !". En effet il ne sert à rien de déployer des trésors de technologie si votre récepteur ne reçoit rien. Le remplacement de cette antenne par une "vraie" antenne 1090 MHz augmentera considérablement le signal reçu et donc la distance à laquelle les avions seront détectés. Ceci explique la présence d'une prise de type "N" sur le câble d'antenne de mon installation (à gauche sur la

photo 4). Ce modèle de prise est utilisé sur les antennes professionnelles car il procure des pertes relativement faibles. J'ai opté pour une antenne de type FR24 équipée d'une prise N, le boîtier étant monté sur le mat d'antenne. Ceci élimine pratiquement le problème des pertes dans le câble de liaison entre l'antenne et le récepteur. Si vous choisissez de relier l'antenne au dongle TNT par un câble coaxial, ne lésinez pas sur sa qualité (et son prix) au risque d'être déçu par les résultats ! **Fig.7**.



Antenne FR24 (© FLightRadar24)

Le dégagement

Un autre paramètre important dans la réception des signaux aux fréquences de l'ADS-B est le dégagement de l'antenne. La portée étant principalement à vue, l'antenne doit être bien dégagée. Si vous êtes encaissé entre des immeubles ou situé au fond d'une vallée, la portée sera fortement réduite. Dans certains cas, le montage de l'antenne sur un mât de plusieurs mètres assure un dégagement suffisant pour améliorer la réception. Il faudra juger de la nécessité de surélever l'antenne en fonction votre situation.

Préamplifier le signal

Dans l'arsenal des moyens disponibles pour améliorer la réception, il y a encore la possibilité d'intercaler un préamplificateur de signal entre l'antenne et le récepteur. Ce préamplificateur devra être situé au plus près de l'antenne et être muni d'un filtre pour éliminer les signaux non souhaités susceptibles de provoquer des interférences et de dégrader la réception. C'est une amélioration que je compte tester prochainement.

Conclusion

Au travers de cette description, vous avez pu vous apercevoir que le Raspberry Pi est extrêmement souple et qu'il peut s'adapter à de nombreux projets. Dans le domaine de la réception des signaux ADS-B c'est un auxiliaire précieux qui permet de traiter les signaux reçus 2' heures sur 24, sans monopoliser un PC. Avec un matériel minimum (un Raspberry Pi et son alimentation, une carte SD, une clé USB TNT et son antenne) il est déjà possible d'obtenir des résultats intéressants. J'espère avoir donné envie à certains d'entre vous de se lancer à la chasse aux signaux émis par les avions qui nous survolent en permanence.