

# L'ADSB

Présentation de l'ADSB sur wikipedia

[https://fr.wikipedia.org/wiki/Automatic\\_dependent\\_surveillance-broadcast](https://fr.wikipedia.org/wiki/Automatic_dependent_surveillance-broadcast)

## But :

permettre la localisation du trafic aérien sur un fond de carte

## Scénario :

lors d'une catastrophe naturelle ou d'un événement, ce système permettrait d'informer le centre de crise du trafic aérien autour de la zone. Il permettrait notamment de synchroniser les phases logistiques.

## Matériel :

- [nano ordinateur \(raspberry pi, orange pi,...\)](#)
- [clef SDR \(type nooelec avec TCXO\)](#)
- [Filtre 88-108 MHz](#)
- [antenne 1090 MHz](#)

## Budget masses (Poids des éléments) :

- [nano ordinateur \(raspberry pi, orange pi,...\)](#) : 29g (pour un raspberry pi 3a+); 44g (raspberry pi2, 9g (raspberry pi zero)
- [clef SDR \(type nooelec avec TCXO\)](#) : 30g
- [Filtre 88-108 MHz](#) : 21g
- [antenne 1090 MHz](#) : 7g
- [\[F4EED\]](#) [je verrais mieux des dipôles ADSB](#)

## Logiciel :

- [Raspbian](#) pour l'OS du Raspberry Pi

- Obligatoire

- [dump 1090](#)

- ou tous autre fork

- Commentaire: [technologie déjà maîtrisée](#)
- **Testé**

## Procédure installation :

1. Installer Raspbian stretch : <https://www.raspberrypi.org/downloads/raspbian/>
2. Installer dump1090 sur le raspberry; en mode terminal lancer les commandes suivantes :
  - `sudo apt-get update`
  - `sudo apt-get install -y git build-essential debhelper rtl-sdr`
  - `sudo apt-get install -y librtlsdr-dev libusb-1.0-0-dev pkg-config`
  - `sudo apt-get install -y fakeroot curl cron lighttpd`
  - `sudo mkdir ~/build-dump-mut`
  - `cd ~/build-dump-mut`
  - `sudo git clone https://github.com/mutability/dump1090.git`
  - `cd ~/build-dump-mut/dump1090`
  - `sudo dpkg-buildpackage -b (soyez patient)`
  - `cd ~/build-dump-mut`
  - `sudo dpkg -i dump1090-mutability_1.15~dev_*.deb`
  - `sudo lighty-enable-mod dump1090`
  - `sudo /etc/init.d/lighttpd force-reload`
  - `sudo dpkg-reconfigure dump1090-mutability`
  - pour la plus part des utilisations accpeter les valeurs par défaut en appuyant sur la touche entrée.
  - pour les valeurs suivantes entrez les valeurs comme indiqu ce dessous:
  - (a) RTL-SDR dongle to use: 0
  - (b) Votre latitude de réception (au format décimal): xx.xxxx
  - © Votre longitude de réception (au format décimal): yy.yyyy
  - (d) Interface address to bind to (blank for all interfaces): remove default 127.0.0.1 and leave blank.

## Correction d'un bug sur raspbian strech :

- `sudo wget -O /etc/udev/rules.d/rtl-sdr.rules`  
`"https://raw.githubusercontent.com/osmocom/rtl-sdr/master/rtl-sdr.rules" * sudo reboot`

**Avantage du ballon :** \* la montée en altitude du récepteur permet de voir les avions qui volent loin à faible altitude (apparition dès 175 pieds – environ 55m – constatés sur Roissy CDG) [Retour aux tests radio envisagés](#) ===== Radiodiffusion type bande FM + RDS ===== \* But : permettre une émission sur la bande de radiodiffusion, 88-108 MHz, incluant la partie RDS et pouvoir ainsi diffuser de l'information à la population sur un simple récepteur FM \* Scénario : lors d'une catastrophe naturelle ou événement, diffuser les informations de première nécessité à la population mais aussi diffuser l'alerte. \* Matériel : Raspberry PI zero \* Poids des modules : \* Raspberry PI zero W : 9g \* Système d'exploitation : \* Raspbian \* Solution retenue et testé : \* **PiFmRds sur raspberry pi zéro** : point fort l'utilisation des transmission d'alerte \* **Testé** - Est aussi utilisable : \* rpitx de F5OEO [Retour aux tests radio envisagés](#) ===== APRS ===== \* But : permettre le positionnement des radioamateurs et des sources de données (météo,...), présent dans la zone de portée radio du ballon, sur un fond de carte \* Scénario : lors d'une catastrophe naturelle ou événement, géolocaliser les convois et permettre la transmission de données vers le centre de crise. \* Matériel : A Définir \* Modem APRS ? \* nano ordinateur (raspberry pi, orange pi,...) \* Lime SDR mini? \* Module type DR818 \* Poids des modules : \* Modem APRS ? : \* nano ordinateur (raspberry pi, orange pi,...) : \* Lime SDR mini? : 38g avec boîtier \* Module type DR818 : ? \* logiciel : A Définir \* Raspbian \* direwolf \* **Testé** \* **Ce qui a été testé c'est uniquement la réception APRS (igate) avec une clef SDR rajouté sur le projet relai numérique** [Retour aux tests radio envisagés](#) ===== Relais radioamateur analogique, transpondeur, transpondeur linéaire ===== \* But : permettre la mise en place de liaison radioamateur de grande portée en analogique \* Scénario : lors d'une catastrophe naturelle ou événement, assurer les communications

radio entre les équipes déployées. \* Matériel : \* Poids des modules : \* **Non Testé** Retour aux tests radio envisagés ===== Relais multimode (DSTAR/DMR/C4FM/...) ===== \* But : permettre la mise en place de liaison radioamateur de grande portée en DSTAR/DMR/C4FM/... \* Scénario : lors d'une catastrophe naturelle ou événement, assurer la résilience des communications entre deux sites séparés d'une grande distance \* Matériel : \* nano ordinateur (raspberry pi, orange pi, ...) \* Raspberry Pi DUPLEX hotSPOT MMDVM | DMR D-Star C4FM POCSAG \* Logiciel : Pi-Star \* Commentaire: technologie déjà maîtrisée Avantage d'utiliser Pi-Star : - Image raspbian préconfigurée, rien n'est à installer en plus. - Relativement simple de configuration. - Un système de protection en écriture pour la carte SD est déjà en place (ce qui permet de ne pas tuer la carte SD si redémarrage à chaud. Idée à la noix : \* faire tourner la partie ADSB (dump1090) sur le même raspberry pi \* **Testé** Remarque de Vivien F4FWH suis à l'exercice caraïbe wave 2019 : \* Ne pas surcharger le raspberry pi, sur l'exercice celui-ci est monté à 50-60 °C \* **Testé uniquement en DMR** Retour aux tests radio envisagés ===== TNT ===== \* But : permettre l'émission d'une vidéo TNT diffusant des informations sur la crise en cours \* Scénario : idem FM/RDS \* Matériel : \* Poids des modules : \* **Non testé** Retour aux tests radio envisagés ===== LORA ===== \* But : permettre la mise en place du protocole LORA pour permettre de connexion des objets connectés au sol en eux. \* Scénario : lors ou avant d'une catastrophe naturelle ou événement, permettre la création d'un réseau de capteurs dont les données seraient partagées et mises à profit de systèmes d'alerte par exemple (sismographes, marégraphe, etc. \* Matériel : \* Poids des modules : \* **Non testé** Retour aux tests radio envisagés ===== AIS ? ===== \* But : permettre la localisation du trafic maritime sur un fond de carte \* Scénario : idem ADSB \* Matériel : \* Poids des modules : \* **Non testé** Retour aux tests radio envisagés ===== Liaison wifi entre la station au sol et le ballon ===== \* But : Permettre la mise en place d'une liaison wifi entre le ballon et la station au sol. Pour pouvoir faire de l'upload de fichier sur le matériel embarqué, mais aussi de prendre la main sur les nano ordinateurs embarqués. \* Scénario : lors d'une catastrophe naturelle ou événement, servir de relais pour la téléphonie, site web d'information etc. \* Matériel : \* voir si un des nano PC déjà embarqué pourrait faire office de point d'accès wifi \* la station sol pourrait être du type Nano station d'ubiquity de manière à avoir une antenne avec un peu de gain pointé en direction du ballon ? \* Idée d'ajout d'un connecteur pour l'antenne externe du raspberry pi: [https://web.archive.org/web/20181019182655im\\_/https://www.dorkbotpdx.org/blog/wramsdell/external\\_antenna\\_modifications\\_for\\_the\\_raspberry\\_pi\\_3](https://web.archive.org/web/20181019182655im_/https://www.dorkbotpdx.org/blog/wramsdell/external_antenna_modifications_for_the_raspberry_pi_3) \* Poids des modules : \* Page dédiée: <https://projet-eonef.frama.wiki/raspiWifi> \* Sur le paramétrage du point d'accès, privilégier la bande 5GHz. Si vous activez un canal en 2.4GHz, pensez bien à désactiver la prise en charge 802.11b (débits 1, 2, 5.5 et 11Mbps). Un peu de lecture détaillée sur le sujet 802.11b is poison, IEEE 802.11mc. \* **Testé** Retour aux tests radio envisagés ===== Hors projet ballon ===== ===== Réception image météo satellite ===== \* But : permettre la réception d'image météo \* Scénario : permettre de voir la météo à venir après rupture des canaux de réception d'info météo \* Matériel : \* Antenne : type QFH : <http://bbayle.com/satellites/3eme.html> \* partie réception automatisée : \* Poids des modules : \* **Testé** Retour aux tests radio envisagés ===== Liaison Phonie (HF) ===== \* But : permettre des liaisons longue distance \* Scénario : \* Matériel : \* Antenne : G5RV [http://f5ad.free.fr/Liens\\_coupes\\_ANT/F/F6DDR%20G5RV.htm](http://f5ad.free.fr/Liens_coupes_ANT/F/F6DDR%20G5RV.htm) \* transceiver : n'importe quel poste HF (pour les tests du week end nous avons un FT897 \* Poids des modules : \* Contact établie : Guadeloupe \* **Testé** Retour aux tests radio envisagés ===== Liaison Numérique (HF) ===== \* But : permettre des liaisons longue distance \* Scénario : \* Matériel : \* Antenne : G5RV [http://f5ad.free.fr/Liens\\_coupes\\_ANT/F/F6DDR%20G5RV.htm](http://f5ad.free.fr/Liens_coupes_ANT/F/F6DDR%20G5RV.htm) \* transceiver : n'importe quel poste HF (pour les tests du week end nous avons un FT897 \* Logiciel : WSJT-X configuré en mode FT8 \* Poids des modules : \* Contact établie : Etats Unis, Guyane, ... \* **Testé** Retour aux tests radio envisagés ===== Installation des divers logiciels ===== ===== Installation de raspbian ===== \* But : Permettre l'installation du système d'exploitation Raspbian sur les nano ordinateurs embarqués de type raspberry pi. \* Installation de Raspbian pour le Raspberry Pi sur carte micro SD avec Etcher (Windows, Linux, Mac) Retour vers la partie installation des logiciels

From:

<https://projet-eonef-2.frama.wiki/> - **Projet EONEF 2**

Permanent link:

<https://projet-eonef-2.frama.wiki/adsb?rev=1561911682>

Last update: **2019/06/30 18:21**

