

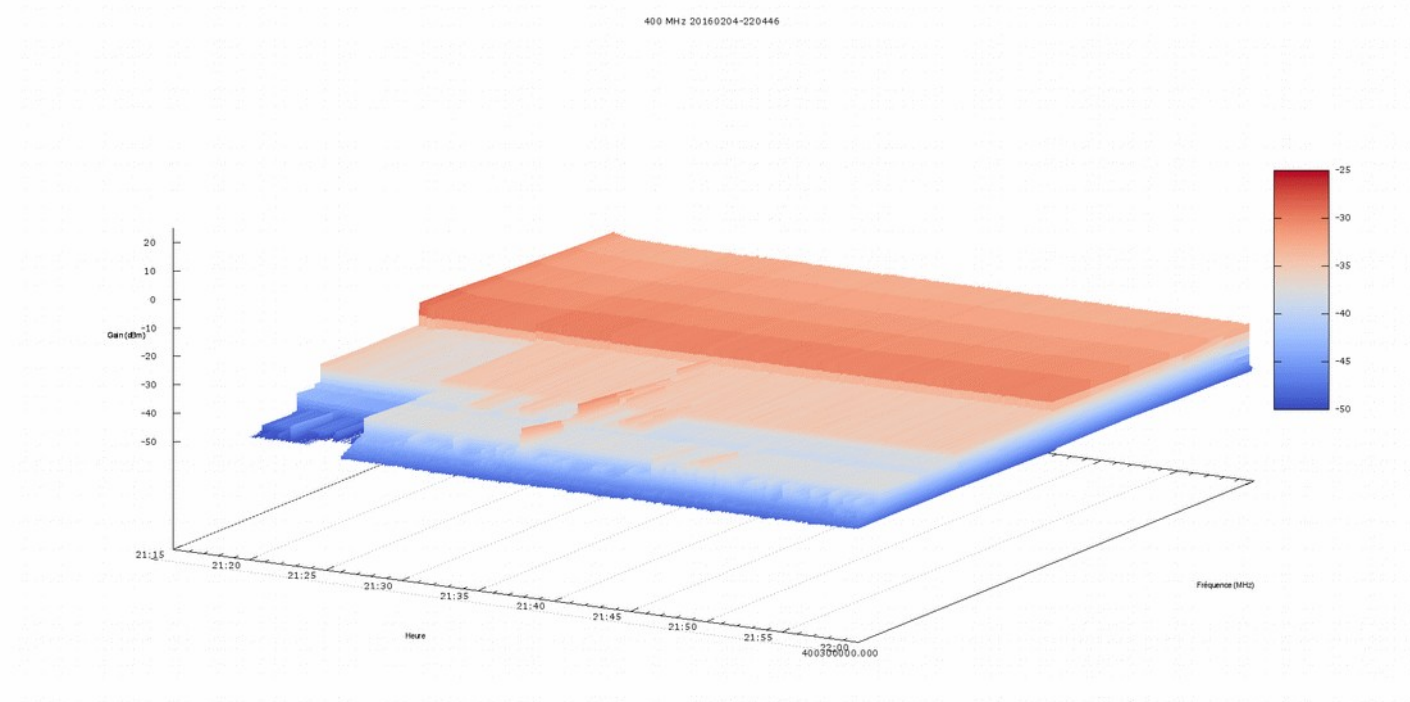
- RS
 - Réception radio sondes meteorologiques 400-406 MHz via RTLSDR
 - Plusieurs décodeurs spécialisés par modèles de sondes attendent un enregistrement sonore déclenché à la suite d'un balayage et d'une détection de puissance
 - Disponible ici :
<https://github.com/rs1729/RS>
- Sources :
https://www.reddit.com/r/RTLSDR/comments/2snqsr/radiosonde_hunting/
<http://radiosonde.eu/>
- Video
<https://www.youtube.com/user/boulderplex>
- Compatible Raspberry Pi 1 modèle B

- RS utilise plusieurs programmes
 - RTLSDR :
 - SDR pour les clés USB
 - Dépôt github ici :
<https://github.com/keenerd/rtl-sdr>
 - rtl_fm
<http://kmkeen.com/rtl-demod-guide/>
 - rtl_power
<http://kmkeen.com/rtl-power/>
 - SOX

- Heatmap (« carte de chaleur »)



- Traitement valeurs et projection avec gnuplot



- Télécharger RS avec git clone depuis :
<https://github.com/rs1729/RS>
soit :
git clone https://github.com/rs1729/RS.git
ou « Download zip »
- Installation « manuelle »

- Compilation des programmes RS
 - Se déplacer dans chaque sous-répertoire du projet
 - Utiliser GCC et ajouter la librairie mathématique :
`gcc nomDuFichierSource.c -lm -o nomDuProgrammeDecodeur`
 - Compiler par exemple `imet1rsb m10x rs41stdin imet1ab rs92gps dfm06 ...`
- Se déplacer dans le répertoire scan, compiler `rs_detect` et `reset_usb`, y copier les programmes décodeurs
 - Vérifier que `sox` et `rtl_fm` et `rtl_power` sont installés et fonctionnels
- Copier `rtlsdr_scan.pl`, vérifier installation de perl et droit d'execution (`chmod +x`) de `rtl_scan.pl`

- Editer la copie de `rtlsdr_scan.pl`
 - Régler la valeur de décalage ppm (programme kal)
 - Kal
 - <https://github.com/steve-m/kalibrate-rtl>
 - `./kal -h`
 - `kal -s GSM850`
 - `kal -c numeroDuCanalPuissanceMaximale`
 - Editer la valeur ppm dans `rtlsdr_scan.pl` ligne 24 variable `my $ppm`

- Contrôler les décodeurs avec le sigidwiki

http://www.sigidwiki.com/wiki/Signal_Identification_Guide

- Rechercher les échantillons des radiosondes RS92 et DFM :

[http://www.sigidwiki.com/wiki/Vaisala_RS92SGP_Weather_Balloon_\(Radiosonde\)](http://www.sigidwiki.com/wiki/Vaisala_RS92SGP_Weather_Balloon_(Radiosonde))

[http://www.sigidwiki.com/wiki/Graw_DFM06_Weather_Balloon_\(Radiosonde\)](http://www.sigidwiki.com/wiki/Graw_DFM06_Weather_Balloon_(Radiosonde))

- Tester les enregistrements :

- Pulseaudio trouver le moniteur de la carte son :

```
pactl list
```

- Tester le détecteur (exemple) :

```
parec -d "alsa_output.pci-0000_00_14.2.analog-stereo.monitor" | sox -t raw -r 48000  
-esigned-integer -b16 -c2 - -t wav - | ./rs_detect -t 8
```

- Tester un décodeur (exemple) et ses options en ligne de commande :

```
parec -d "alsa_output.pci-0000_00_14.2.analog-stereo.monitor" | sox -t raw -r 48000  
-esigned-integer -b16 -c2 - -t wav - 2>/dev/null | ./rs92gps -i 2>/dev/null
```

- Tester un autre décodeur (exemple) :

```
parec -d "alsa_output.pci-0000_00_14.2.analog-stereo.monitor" | sox -t raw -r 48000  
-esigned-integer -b16 -c2 - -t wav - 2>/dev/null | ./rs92gps -i 2>/dev/null
```

- Comparer avec les video disponibles ici :

<https://www.youtube.com/user/boulderplex>

- Démarrer RS en mode « automatique » et journalisation:

```
while true; do ./rtl_sdr_scan.pl | tee >
~/logs/radiosonde/clef/log_$(date '+%Y%m%d_%H%M
%S').txt ; sleep 1; done
```

- Concaténer les journaux de réceptions, les transformer en fichiers GPX avec gpsbabel et les afficher avec Viking, par exemple :

```
cat *.txt | awk -F " " '{print $7,$9}' | sed "s/ /,/" | sed "/^,/d" >
lat_lon.csv ; gpsbabel -i csv -f lat_lon.csv -o gpx -F
lat_lon.gpx
```


RS a été testé sur Raspberry Pi 1 modèle B avec succès !

