

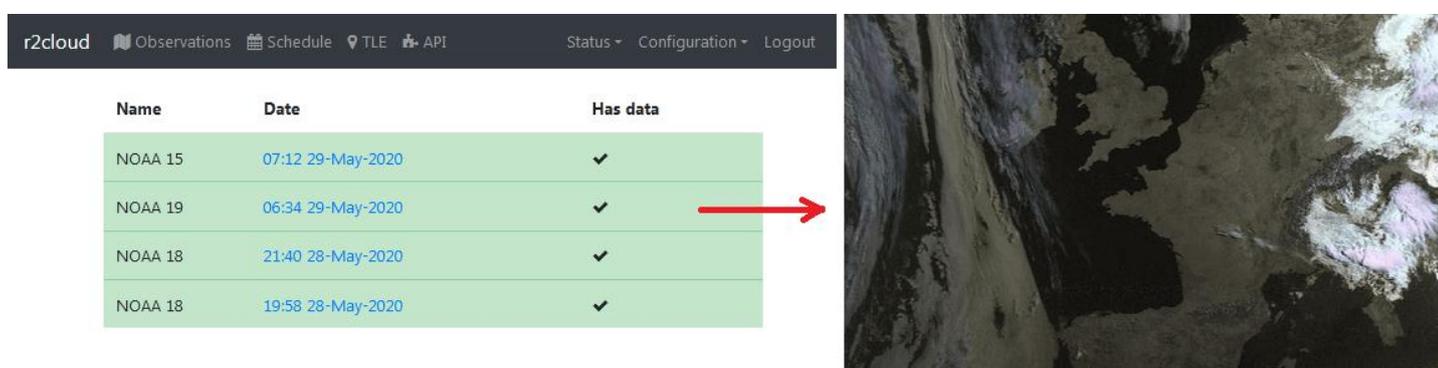
7.1 Présentation de r2cloud.

Cela fait longtemps que les satellites NOAA (National Oceanic and Atmospheric Administration) sont en orbite. Ils émettent en continu des images météo sur 137 MHz. Actuellement au nombre de trois (NOAA15,18 et 19), ils ont été rejoints par un satellite russe Meteor-M2 qui diffuse des images en couleur.

L'avantage d'utiliser un Raspberry Pi pour recevoir les images météo est évident : on n'a aucun regret de le laisser allumé 24 h / 24. Son logiciel met à jour automatiquement les paramètres orbitaux des satellites (TLE : « Two-Line Elements ») et gère la réception et le décodage des images.

De plus **r2cloud** décode également les télémesures des CubeSat. Dès qu'un nouveau CubeSat est en service, la mise à jour est automatique.

Concernant les aériens, l'utilisation d'une antenne bi-bande suffit pour démarrer. Mais si l'on cherche à obtenir la meilleure réception possible, une antenne QFH quadrifilaire 137 MHz est facile à construire.

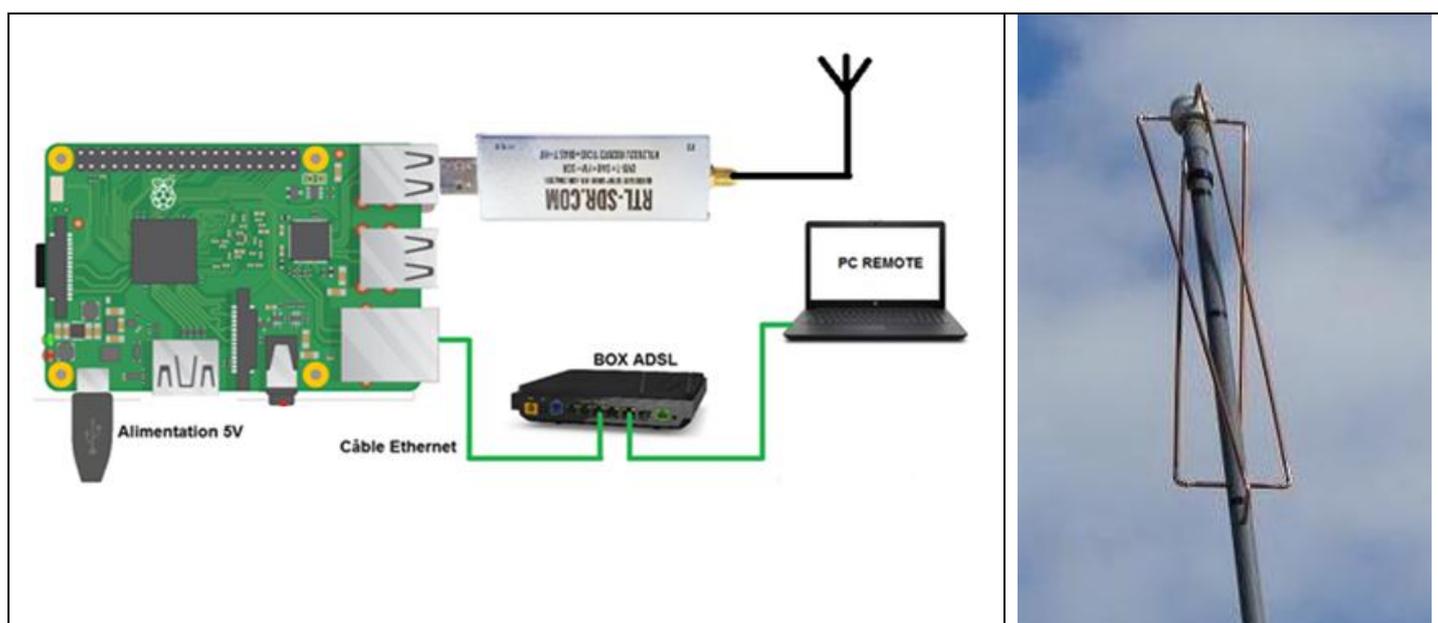


The screenshot shows the r2cloud web interface. At the top, there are navigation links: Observations, Schedule, TLE, API, Status, Configuration, and Logout. Below this is a table with the following data:

Name	Date	Has data
NOAA 15	07:12 29-May-2020	✓
NOAA 19	06:34 29-May-2020	✓
NOAA 18	21:40 28-May-2020	✓
NOAA 18	19:58 28-May-2020	✓

A red arrow points from the second row (NOAA 19) to a satellite image on the right, which shows a view of Earth from space, including clouds and landmasses.

Côté matériel, un Raspberry Pi et une clé RTL-SDR suffisent. Pas besoin de moniteur ni de clavier connecté au Raspberry Pi, même pour la mise en service.



7.2 Installation du logiciel r2cloud

<https://github.com/dernasherbrezon/r2cloud>

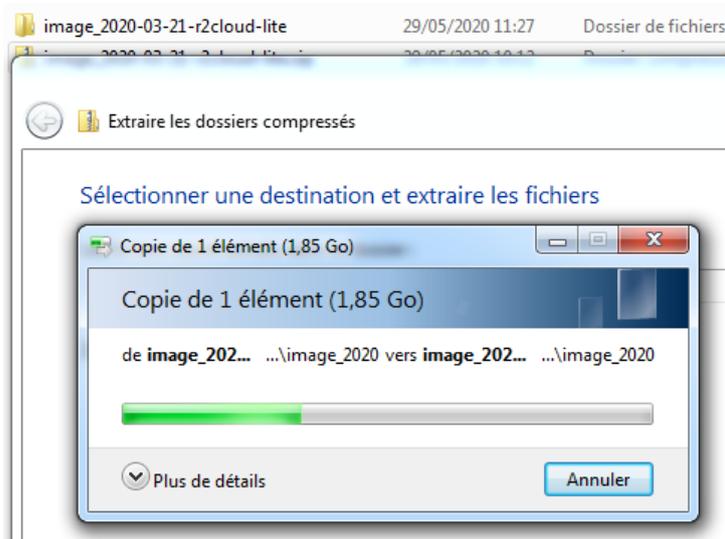
A partir du lien ci-dessus, téléchargez le fichier **image_2020-03-21-r2cloud-lite.zip** et privilégiez le mode d'installation facile.

Installation

1. Install r2cloud

- From the image. This is simplest way and require brand new SD card
 - Download the **latest** official image
 - Insert SD card into the card reader and flash it. You could use [Etcher](#) to do this
 - Insert SD card into the card reader and create file `r2cloud.txt` in the root directory. This file should contain any random string. This string is a login token. This token will be used during initial setup.

Extrayez le fichier compressé :

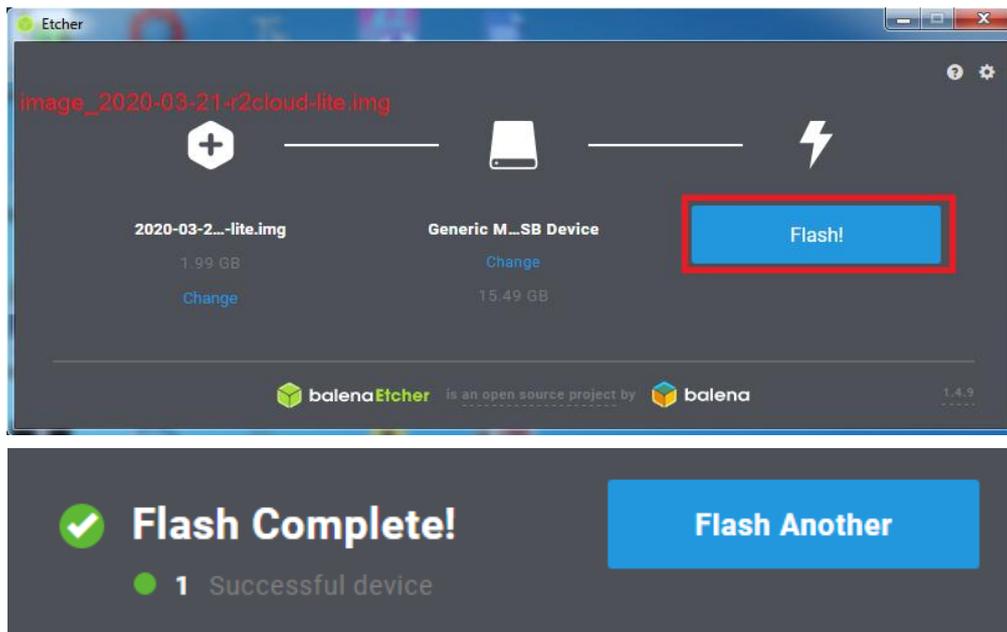


Téléchargez et installez l'utilitaire **etcher** pour copier l'image sur une carte mSD :

<https://www.balena.io/etcher/>



Sélectionnez le fichier `image_2020-03-21-r2cloud-lite.img`, le lecteur de la carte mSD, puis cliquez sur Flash.



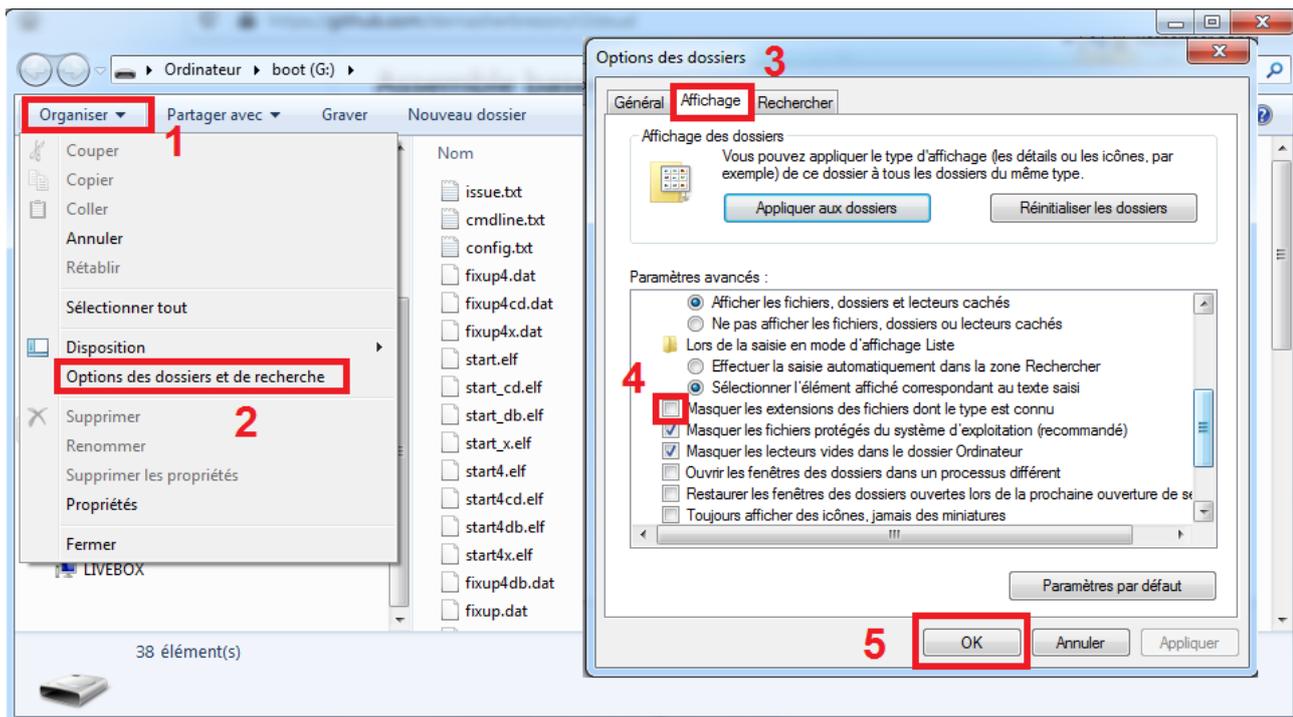
Une fois la copie terminée, débranchez et rebranchez le lecteur de carte mSD.

Sous Windows, la fenêtre ci-dessous apparaît.

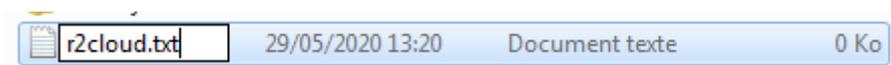
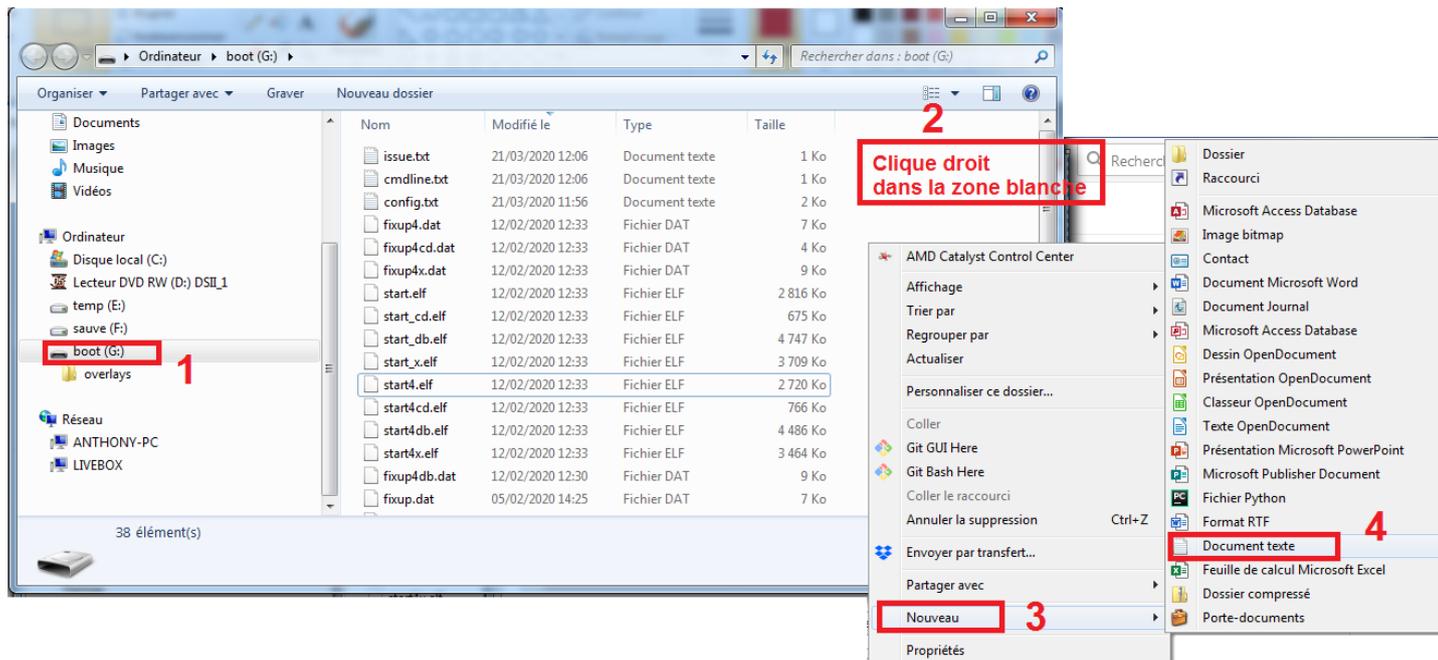
Consultez avec l’explorateur de fichiers la partition nommée boot.



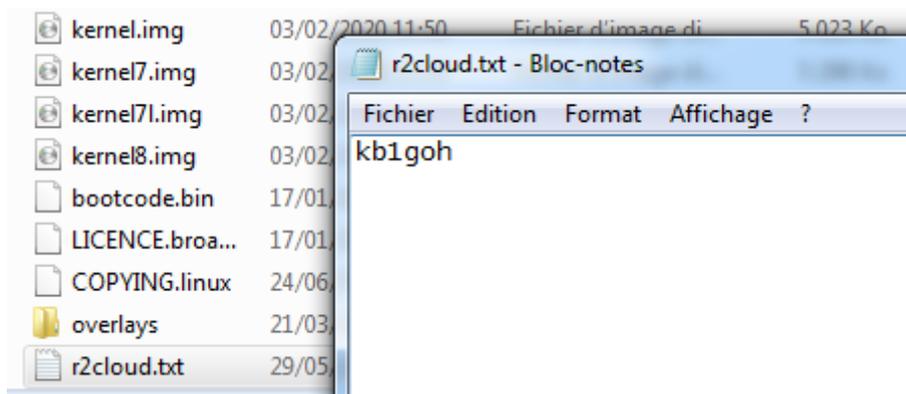
Dans l’explorateur, veillez à ce que l’on puisse voir les extensions de fichiers en suivant les étapes 1 à 5 (case 4 décochée).



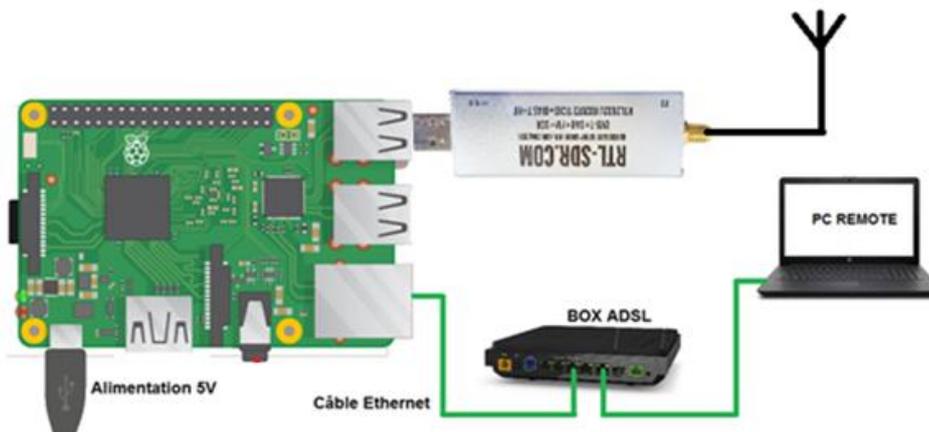
Créerz un nouveau fichier nommé **r2cloud.txt** :



Ouvrez le fichier **r2cloud.txt** et indiquez votre login. Pour faire simple, utilisez votre indicatif. N'oubliez pas d'enregistrer le fichier.



Insérez la carte mSD dans le Raspebry Pi, puis mettez sous tension l'ensemble avec la clé RTL-SDR branchée :

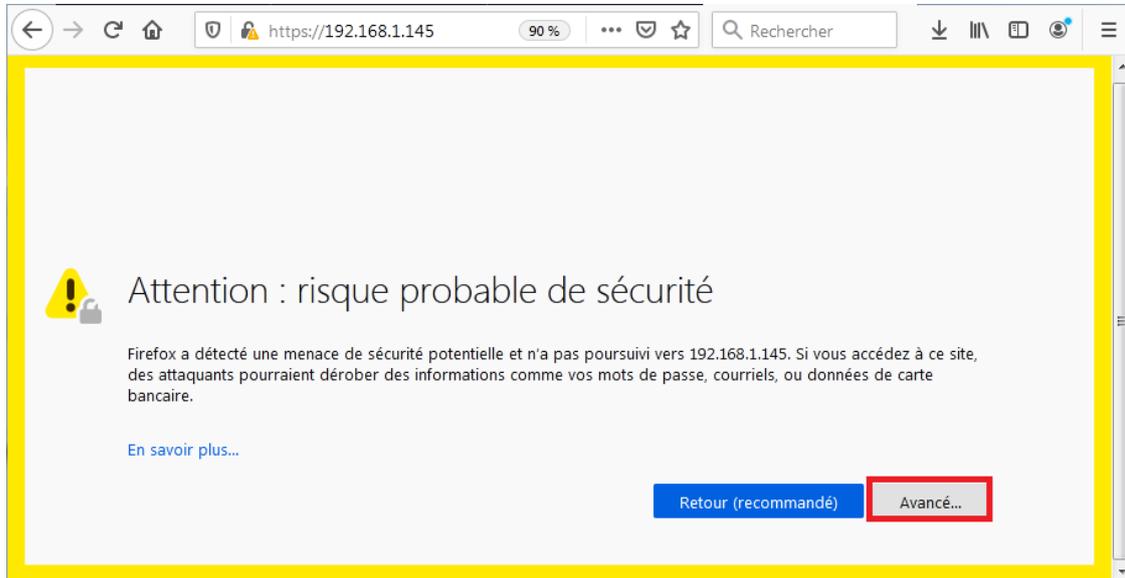


7.3 Prise en main du logiciel r2cloud

Utilisez le nom de domaine déjà préconfiguré dans le Raspberry PI <https://raspberrypi.local>, ou utilisez l'adresse IP du Raspberry PI connecté sur le réseau local.

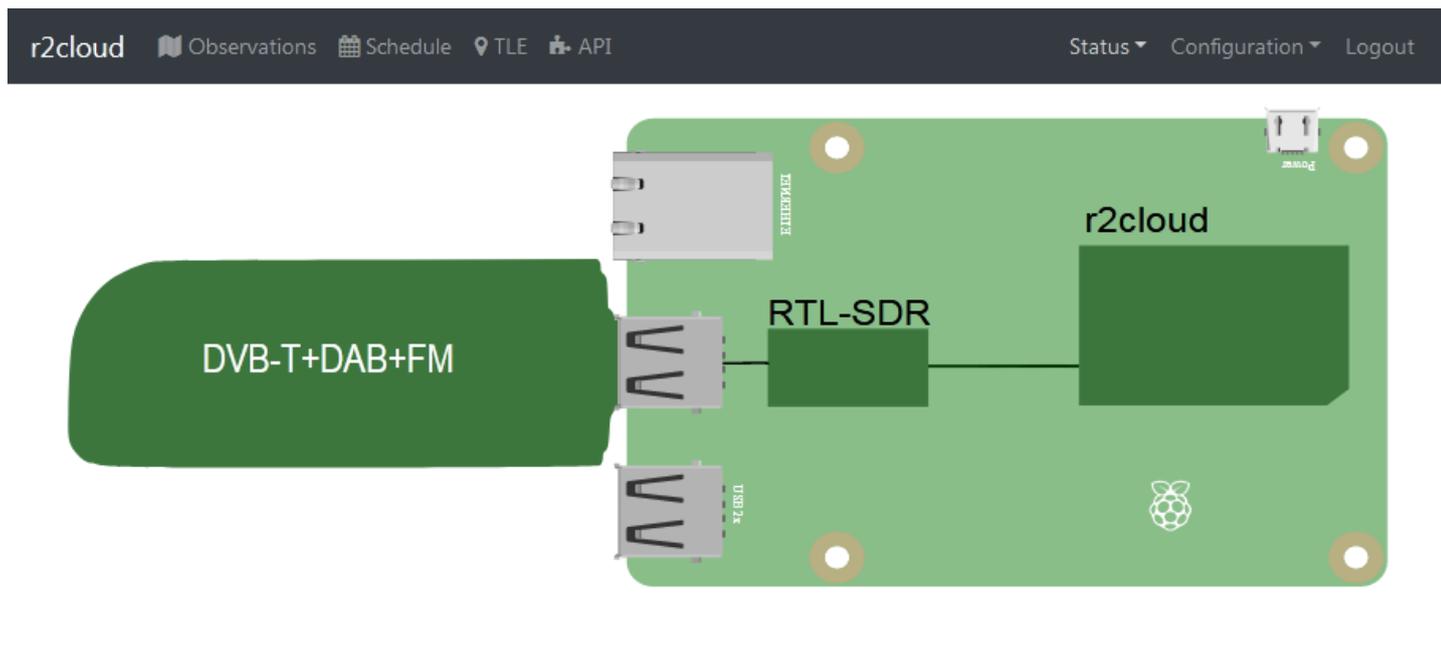
Rappel : Pour connaître l'adresse IP du Raspberry Pi, reprenez le tutoriel Partie 1.

Lors de la connexion, acceptez l'exception de sécurité :



<p>Saisissez le login choisi précédemment dans le fichier texte. Saisissez également un email et un mot de passe.</p>	<p>Saisissez vos coordonnées longitude et latitude. (Cordonnées en degrés décimaux)</p>	<p>Acceptez les mises à jour automatiques.</p>
<p>Setup</p> <p>This is a first time login. Please setup account:</p> <ol style="list-style-type: none"> 1. Insert your SD card into the card reader 2. Open SD card contents and create the file r2cloud.txt there 3. Put some keyword into this file on a single line 4. Save the file and open this page once again 5. Enter the keyword, new email and password <p>kb1goh</p> <p>kb1goh@yahoo.com</p> <p>.....</p> <p>Setup</p>	<p>Setup coordinates</p> <p>Base station coordinates are essential for scheduling satellite observations</p> <p>47.890242</p> <p>0.276770</p> <p>Automatically detect using current browser.</p> <p>Detect</p> <p>Next</p>	<p>Automatic updates</p> <p>Automatic updates will keep your base station up-to-date. We constantly deliver new functionality and bug fixes</p> <p><input checked="" type="checkbox"/> Enable</p> <p>Back Finish</p>

La page principale doit apparaître :

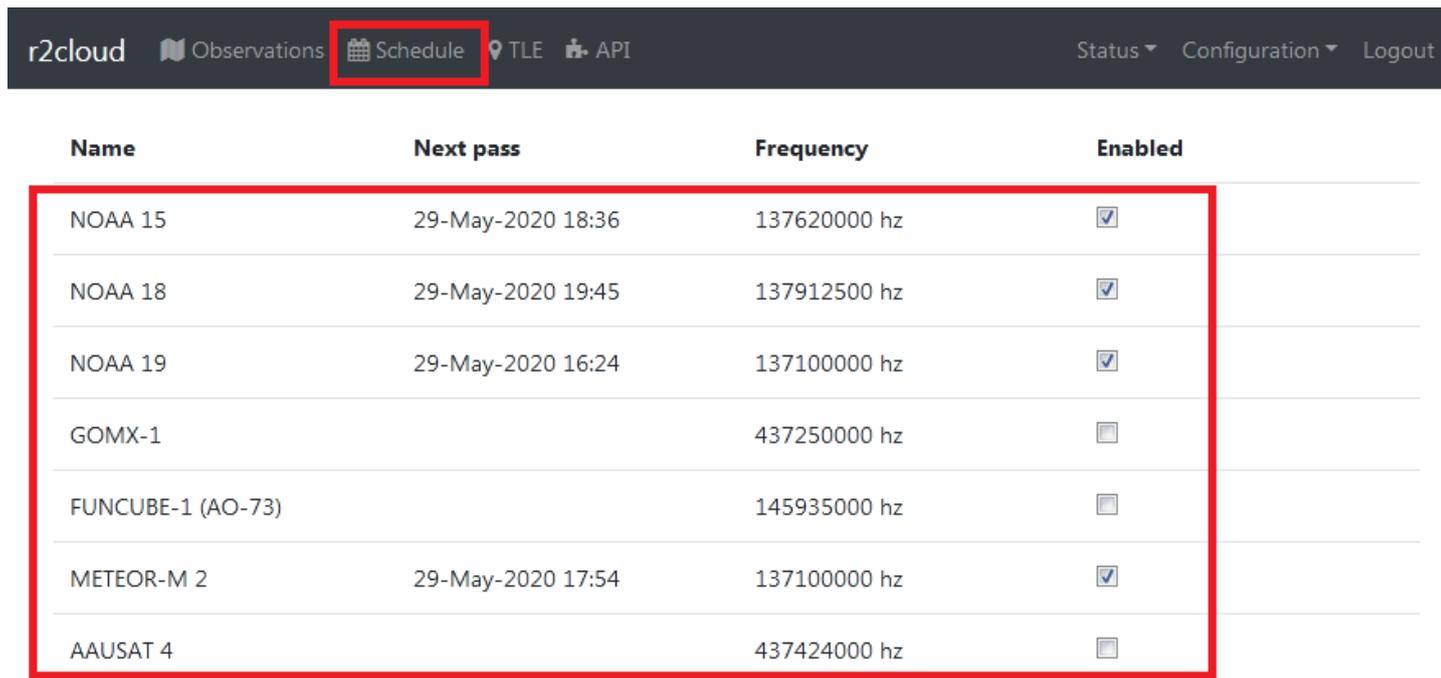


Dans le menu **Configuration - General**, vérifiez l'exactitude des coordonnées géographiques du site de réception.

Dans le menu TLE, la mise à jour n'est pas encore faite. Il faudra attendre plusieurs heures :

Le TLE n'est pas à jour	Le TLE est à jour (en vert)

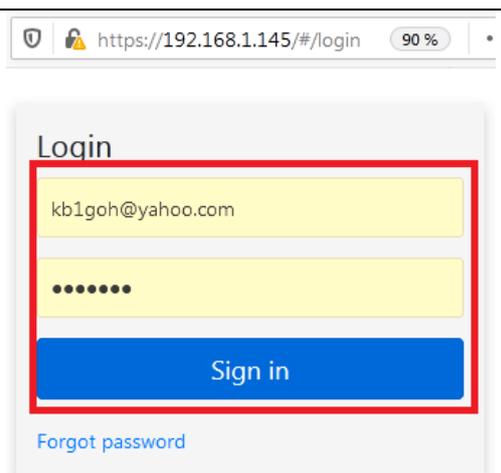
Dans le menu Schedule, sélectionnez les satellites à recevoir :



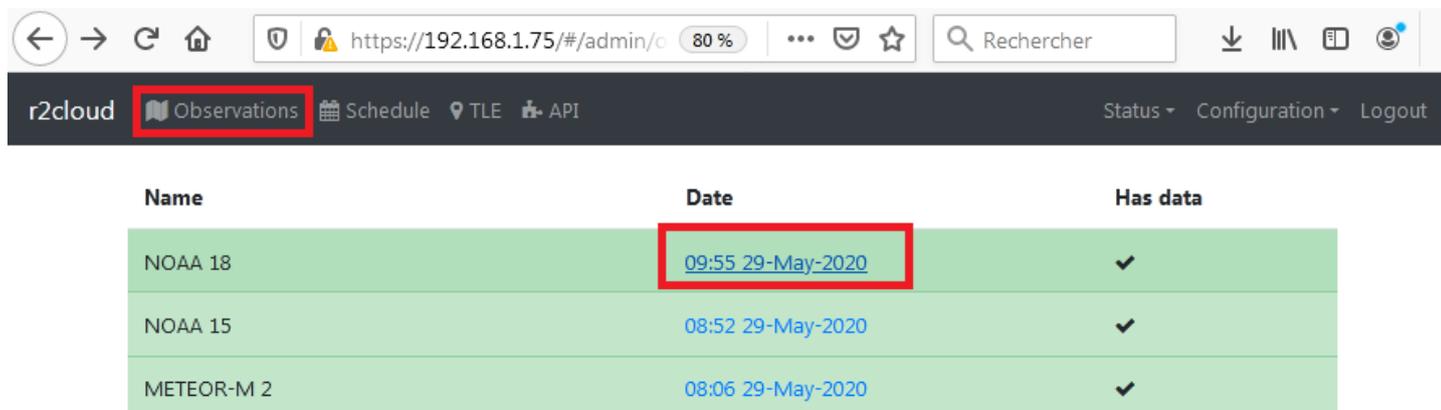
Name	Next pass	Frequency	Enabled
NOAA 15	29-May-2020 18:36	137620000 hz	<input checked="" type="checkbox"/>
NOAA 18	29-May-2020 19:45	137912500 hz	<input checked="" type="checkbox"/>
NOAA 19	29-May-2020 16:24	137100000 hz	<input checked="" type="checkbox"/>
GOMX-1		437250000 hz	<input type="checkbox"/>
FUNCUBE-1 (AO-73)		145935000 hz	<input type="checkbox"/>
METEOR-M 2	29-May-2020 17:54	137100000 hz	<input checked="" type="checkbox"/>
AAUSAT 4		437424000 hz	<input type="checkbox"/>

Laissez sous tension le Raspberry Pi et attendez 24 à 48 heures.

Pour consulter les images, il faudra systématiquement se logger.

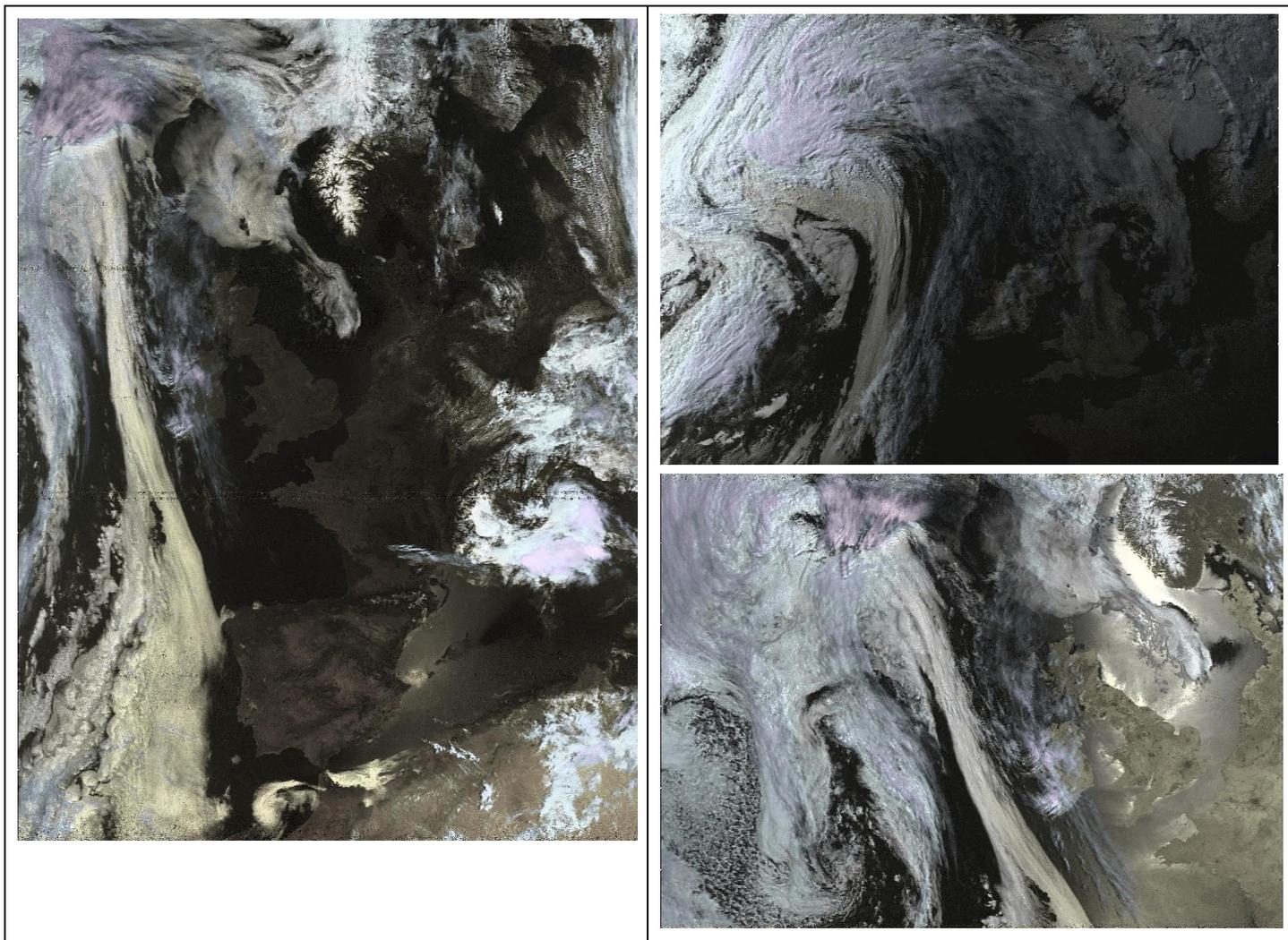


Dans le menu **Observations**, cliquez sur la date du satellite concerné (si des données ont été reçues, la ligne apparaît en couleur verte).

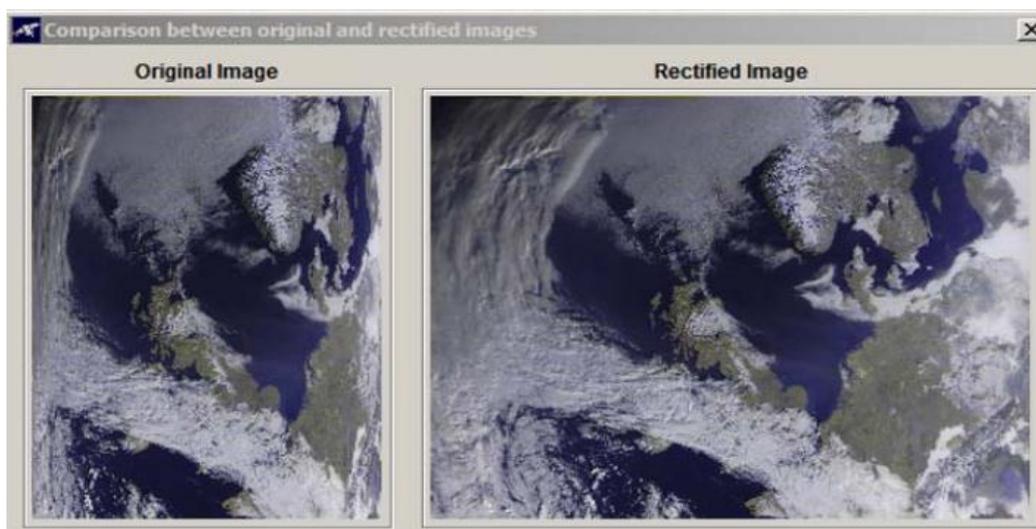


Name	Date	Has data
NOAA 18	09:55 29-May-2020	<input checked="" type="checkbox"/>
NOAA 15	08:52 29-May-2020	<input checked="" type="checkbox"/>
METEOR-M 2	08:06 29-May-2020	<input checked="" type="checkbox"/>

L'image reçue apparaît ensuite.



Les images couleurs du satellite **meteor** sont déformées sur les côtés. C'est normal, mais le programme [SmoothMeteor](#) permet de redresser ces images.



7.4 Dépannage

Si au bout de 48 heures aucune image n'est reçue, il faudra vérifier l'antenne et la connection entre la clé RTL-SDR et le Raspberry Pi.

La connexion en ssh avec l'utilitaire Putty et le Raspberry PI n'est pas possible. Il faudra relier un moniteur et un clavier au Raspberry Pi.

Raspberrypi login : pi

Password : raspberry mais comme le clavier est en qwerty, tapez : rqsberry

Tapez ensuite :

```
rtl_test
```

```
pi@raspberrypi:~ $ rtl_test
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
 49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
^CSignal caught, exiting!
User cancel, exiting...
Samples per million lost (minimum): 0
pi@raspberrypi:~ $ ^C
```

Pour accéder au tiret du bas en qwerty, il faudra faire **shift +)**

7.5 Présentation du logiciel radiosonde auto rx

Comme r2cloud pour les satellites, le logiciel **radiosonde auto rx** permet de recevoir et d'afficher sur une carte la position des radiosondes météo régulièrement envoyées.

https://github.com/projecthorus/radiosonde_auto_rx/wiki

Actuellement, les modèles de radiosondes prises en charge par le logiciel sont les suivantes :

- Vaisala RS92 (experimental support for the RS92-NGP).
- Vaisala RS41.
- Graw DFM06/DFM09/DFM17/PS-15.
- Meteomodem M10.
- Internet iMet-4 (and 'narrowband' iMet-1 sondes).
- Lockheed Martin LMS6, 400 MHz and 1680 MHz variants (including the new 'LMS-X' type).
- Meisei iMS-100.

7.6 Installation du logiciel radiosonde auto rx

Il n'y a pas de fichier image prêt à l'emploi. Pour installer le logiciel **radiosonde auto rx**, il faudra au préalable préparer une carte mSD avec « Raspbian Lite » et installer les bibliothèques RTL-SDR.

Résumé de l'installation préliminaire :

- HAM Radio avec le Raspberry Pi Partie 4 : installation complète, sauf la sauvegarde de la carte mSD qui peut être réalisée ultérieurement) ;
- HAM Radio avec le Raspberry Pi Partie 5.6 : installation de la clé RTL-SDR, comme le résume les commandes suivantes :

Mise à jour des dépôts et des logiciels (il est conseillé d'utiliser l'utilitaire **PuTTY**) :

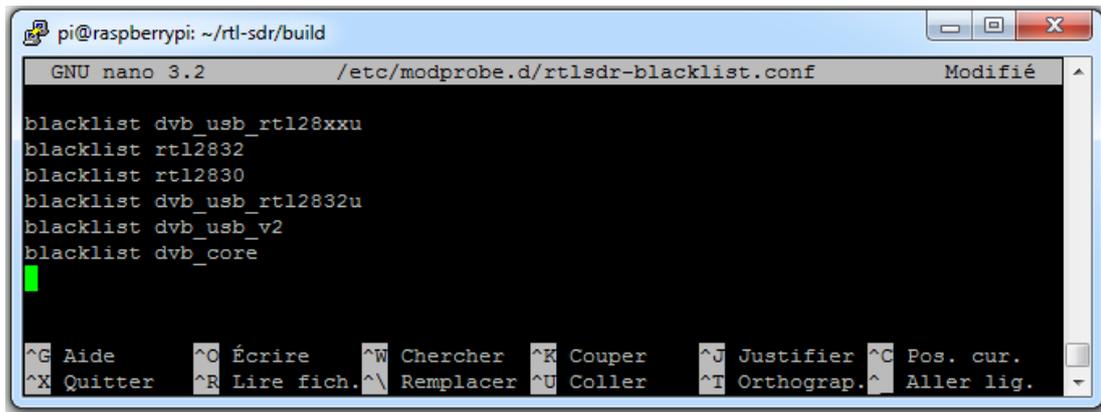
```
sudo apt update
sudo apt upgrade
```

Installation de l'utilitaire **git** :

```
sudo apt install git
```

Installation des dépendances pour **radiosonde auto rx** :

```
sudo apt-get install python-numpy python-setuptools python-crcmod python-requests python-dateutil python-pip sox git build-essential cmake usbutils libusb-1.0-0-dev rng-tools libsamplerate-dev
```

```

pi@raspberrypi: ~/rtl-sdr/build
GNU nano 3.2 /etc/modprobe.d/rtlsdr-blacklist.conf Modifié
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
blacklist dvb_usb_rtl2832u
blacklist dvb_usb_v2
blacklist dvb_core
  
```

Enregistrez le fichier à l'aide des touches **Ctrl + o**, puis la touche entrée pour valider l'enregistrement.

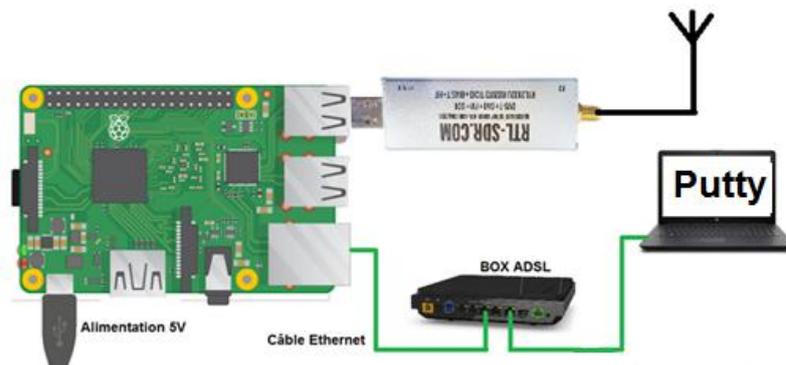
Sortez de l'éditeur à l'aide des touches **Ctrl + x**.



Redémarrez le Raspberry Pi : `pi@raspberrypi:~/rtl-sdr/build $ sudo reboot`

`sudo reboot`

Vérification : Connectez la clé RTL-SDR sur un port USB libre du Raspberry Pi.



Vérifiez la présence de la clé RTL-SDR.

Avec LXTerminal ou PuTTY, tapez la ligne de commande suivante :

`lsusb`

Localisez la clé RTL-SDR : **RTL238 DVB-T**

Cas du Raspberry Pi 3 :

```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 004: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Cas du Raspberry Pi 4 :

```
pi@raspberrypi:~ $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Avec LXTerminal ou PuTTY, tapez la ligne de commande suivante (attention : **tiret du bas**) :

```
rtl_test
```

La clé RTL-SDR devrait être reconnue.

Si ce n'est pas le cas, débranchez et rebranchez la clé, et recommencez le test.

```
pi@raspberrypi:~ $ rtl_test
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
^CSignal caught, exiting! ctrl+c
User cancel, exiting...
Samples per million lost (minimum): 0
pi@raspberrypi:~ $ ^C
```

Installation du logiciel **radiosonde Auto RX** :

```
git clone https://github.com/projecthorus/radiosonde_auto_rx.git
```

```
ls
```

```
pi@f4goh:~ $ git clone https://github.com/projecthorus/radiosonde_auto_rx.git
Clonage dans 'radiosonde_auto_rx'...
remote: Enumerating objects: 37, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 5202 (delta 17), reused 18 (delta 10), pack-reused 5165
Réception d'objets: 100% (5202/5202), 12.45 MiB | 1008.00 KiB/s, fait.
Résolution des deltas: 100% (3551/3551), fait.
pi@f4goh:~ $ ls
radiosonde_auto_rx  rtl-sdr
pi@f4goh:~ $
```

N'oubliez pas la touche tabulation TAB. Le réflexe pour taper les deux lignes de commandes sans faire de copier-coller dans PuTTY doit être :

```
cd ra TAB a TAB ./b TAB
```



```
cd radiosonde_auto_rx/auto_rx
./build.sh
```

```
pi@f4goh:~ $ cd radiosonde_auto_rx/auto_rx/
pi@f4goh:~/radiosonde_auto_rx/auto_rx $ ./build.sh
Building dft_detect
Building RS92/RS41/DFM/LMS6/iMS Demodulators
Building LMS6-1680 Demodulator.
Building iMet Demodulator.
Building fsk-demod utils from codec2
Copying files into auto_rx directory.
Done!
pi@f4goh:~/radiosonde_auto_rx/auto_rx $
```

Copiez le fichier de configuration :

```
cp station.cfg.example station.cfg
```

Editez le fichier de configuration `station.cfg` avec l'éditeur nano :

```
pi@f4goh:~/radiosonde_auto_rx/auto_rx $ nano station.cfg
```

https://github.com/projecthorus/radiosonde_auto_rx/wiki/Configuration-Settings

Ci-dessous, la configuration minimale :

Modification	Avant	Après
<pre>##### # RADIOSONDE SEARCH SETTINGS # #####</pre> <p>Fréquences de réception min et max</p>	<pre># the sondes are min_freq = 400.05 max_freq = 403.0</pre>	<pre># the sondes are min_freq = 400.05 max_freq = 406.0</pre>
<pre>##### # STATION LOCATION # #####</pre> <p>Définir la position géographique de la station d'écoute. Coordonnées en degrés décimaux.</p>	<pre># Used by the Habitat [location] station_lat = 0.0 station_lon = 0.0 station_alt = 0.0</pre>	<pre># Used by the Habitat UpL [location] station_lat = 47.890242 station_lon = 0.276770 station_alt = 80.0</pre>
<pre>##### # HABITAT UPLOAD SETTINGS # #####</pre> <p>Identification sur le site tracker.habhub.org</p>	<pre>habitat_enabled = False # Uploader callsign, as shown above. uploader_callsign = CHANGEME AUTO RX # Upload listener position to Habitat upload_listener_position = True # Uploader Antenna Description. # If upload_listener_position is ena uploader_antenna = 1/4 wave monopole</pre>	<pre>habitat_enabled = True # Uploader callsign, as shown a uploader_callsign = F4GOH # Upload listener position to H upload_listener_position = True # Uploader Antenna Description. # If upload_listener_position i uploader_antenna = diamond x200</pre>

7.7 Prise en main du logiciel radiosonde auto rx

Exécutez le programme avec Python (veillez à exécuter le programme depuis le répertoire `radiosonde_auto_rx/auto_rx`)

```
python auto_rx.py
```

```
pi@f4goh:~/radiosonde_auto_rx/auto_rx $ python auto_rx.py
2020-05-30 09:43:36,434 INFO:Reading configuration file...
2020-05-30 09:43:39,303 INFO:Config - Tested SDR #0 OK
2020-05-30 09:43:39,309 INFO:Started Flask server on http://0.0.0.0:5000
* Serving Flask app "autorx.web" (lazy loading)
* Environment: production
2020-05-30 09:43:39,313 INFO:Telemetry Logger - Started Telemetry Logger Thread.
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
2020-05-30 09:43:39,315 INFO:OziMux - Started OziMux / Payload Summary Exporter
2020-05-30 09:43:39,755 INFO:Version - Local Version: 1.3.1 Current Master Version: 1.3.1
2020-05-30 09:43:39,756 INFO:SDR #0 has been allocated to Scanner.
2020-05-30 09:43:42,608 INFO:Scanner #0 - Starting Scanner Thread
2020-05-30 09:43:42,613 INFO:Scanner #0 - Running frequency scan.
```

Une fois le programme **radiosonde auto rx** exécuté, ouvrez avec un navigateur la page Web avec l'adresse IP du Raspberry PI sur le port 5000.

<http://192.168.1.145:5000/>

ou avec **votre** nom de domaine <http://f4goh:5000/>

Current Task: SDR #0: Scanning

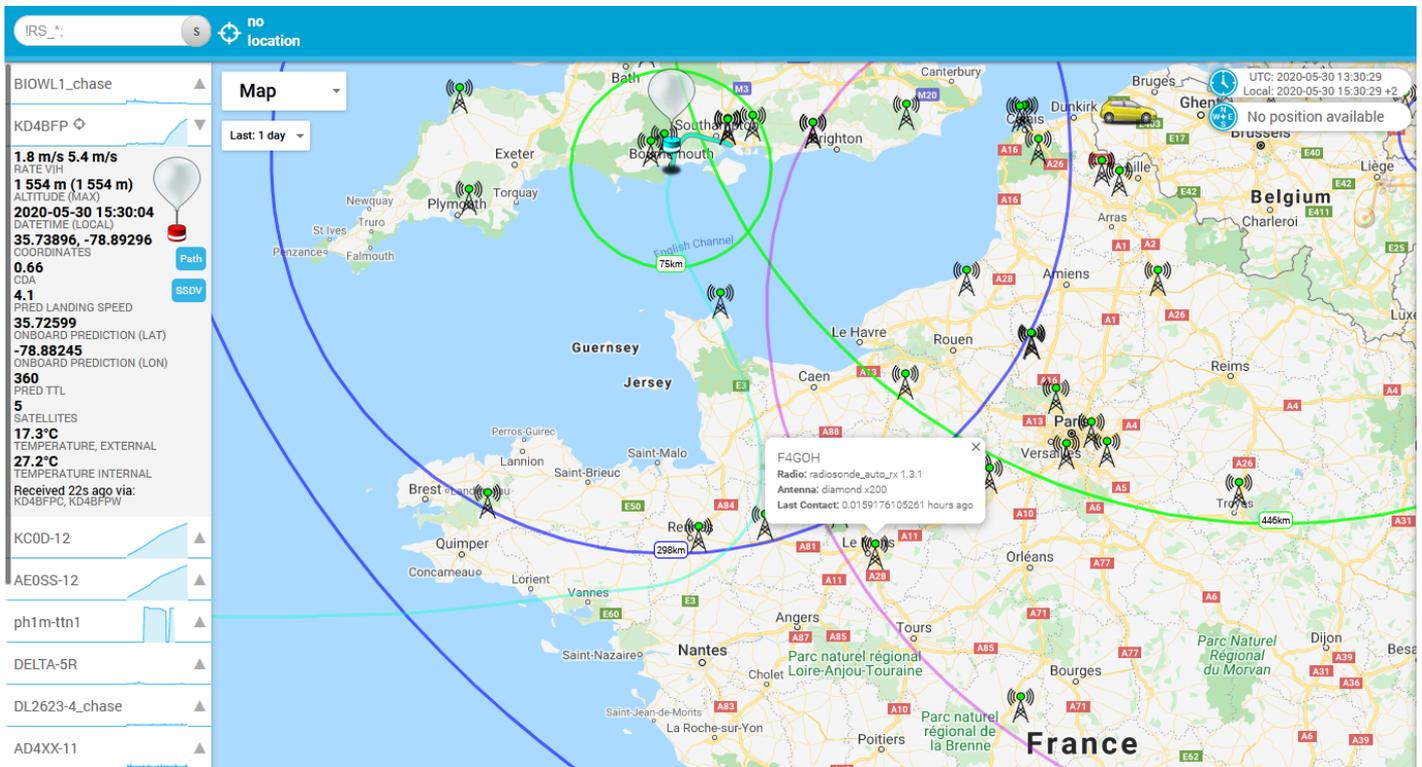
SDR	Age	Type	Freq (MHz)	ID	Time	Frame	Latitude	Longitude	Alt (m)	Vel (kph)	Asc (m/s)	Temp (°C)	RH (%)	Az (°)	El (°)	Range (km)	Ot
None																	

Auto-Follow Latest Sonde: Hide Map:

Une carte devrait apparaître avec, s'il y en a, la position des ballons.

Avec le navigateur, consultez la page Web suivante :

<http://tracker.habhub.org/>



Votre station devrait être présente parmi la liste. Ce site indique également la présence de ballons sondes et la possibilité pour votre station de les décoder.

Pour quitter le programme, appuyez sur les touches Ctrl+c :

```
403.05 402.98 402.92]
^C2020-05-30 10:07:11,924 INFO:Web - Flask Server Shutdown. ctrl+c
2020-05-30 10:07:11,925 INFO:Starting shutdown of all threads.
2020-05-30 10:07:11,926 INFO:Scanner #0 - Waiting for current scan to finish...
2020-05-30 10:07:23,134 INFO:Scanner #0 - Scanner Thread Closed.
2020-05-30 10:07:23,402 INFO:Telemetry Logger - Stopped Telemetry Logger Thread.
pi@f4goh:~/radiosonde_auto_rx/auto_rx $
```

Démarrer automatiquement « radiosonde auto rx » au démarrage du Raspberry PI

(Depuis le répertoire `radiosonde_auto_rx/auto_rx`)

```
cd ~/radiosonde_auto_rx/auto_rx/
```

Copiez le fichier de démarrage :

```
sudo cp auto_rx.service /etc/systemd/system/
```

Redémarrez le Raspberry PI :

```
sudo reboot
```

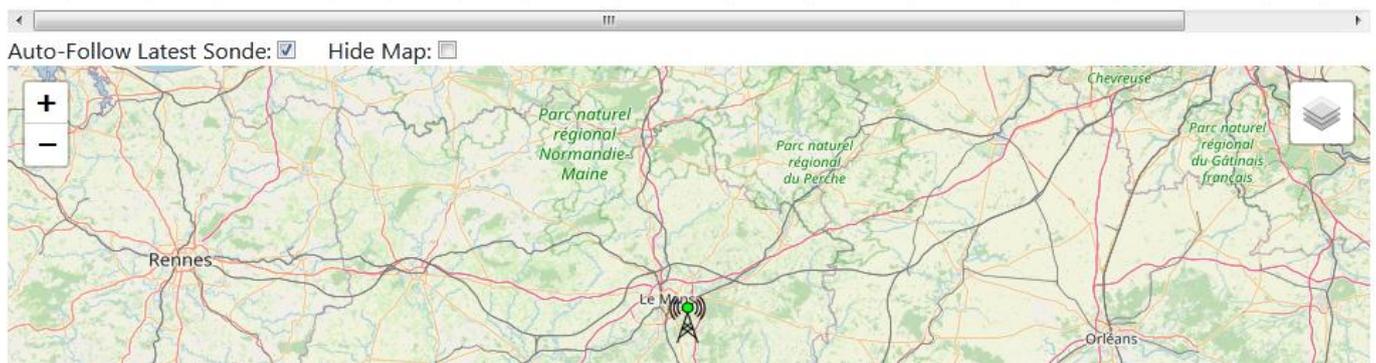
Après une minute, vérifiez le bon chargement de la page HTML dans le navigateur :

<http://192.168.1.145:5000/> ou avec **votre** nom de domaine <http://f4goh:5000/>

Radiosonde Auto-RX Status Version: 1.3.1

Current Task: SDR #0: Scanning

SDR	Age	Type	Freq (MHz)	ID	Time	Frame	Latitude	Longitude	Alt (m)	Vel (kph)	Asc (m/s)	Temp (°C)	RH (%)	Az (°)	El (°)	Ran
None																



Pour supprimer « **radiosonde auto rx** » au démarrage du Raspberry PI, effectuez la commande suivante, puis redémarrez :

```
sudo rm /etc/systemd/system/auto_rx.service
```

A suivre :

Suivi du vol des avions ADS-B : <https://flightaware.com/adsb/piaware/build>