



SENSE HAT : LES 6 SENS DE VOTRE RASPBERRY PI

CE QU'IL VOUS FAUT

Raspbian

Où le trouver ? :
www.raspberrypi.org

- Un Raspberry Pi
- Un Sense HAT
- Un compte Google Drive (optionnel)

Difficulté :

Le Sense HAT est un petit module qui va se connecter au port GPIO de votre Raspberry Pi. En plus d'être équipé d'un afficheur, ce dernier embarque un joystick et pas moins de 6 capteurs permettant d'interagir avec le monde extérieur. Un investissement faible (moins de 40 €) pour des milliers de possibilités...



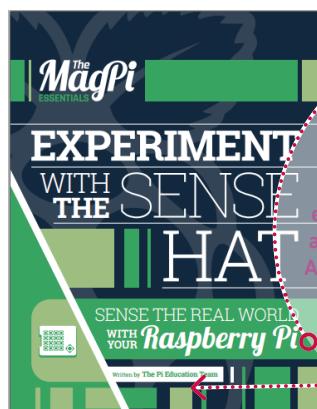
Sense HAT (Hardware Attached on Top)

Il embarque des capteurs laissant imaginer des possibilités infinies qu'il s'agisse de robotique ou de domotique : des capteurs d'humidité, de température, de pression mais aussi un gyromètre, un accéléromètre et un magnétomètre. Le gyromètre calcule les angles d'inclinaison sur 3 axes tandis que l'accéléromètre est en mesure de déterminer l'accélération de -16 à +16 g (1g correspond à la gravité terrestre). Enfin le magnétomètre fait office de boussole. Cette petite machine peut donc se muer en station météo (voir les pages suivantes) mais aussi se repérer dans

l'espace. Bien sûr vous n'êtes pas obligé d'utiliser toutes les fonctions en même temps mais les perspectives sont intéressantes.

UN PRIX ATTRACTIF ET DES MILLIERS DE POSSIBILITÉS

Si on ajoute l'afficheur LED permettant de faire défiler du texte et le mini-joystick, nous sommes en présence d'un des HAT les plus complets pour un prix relativement



Si vous êtes anglophone vous serez peut-être intéressé par le numéro spécial de MagPi dédié au Sense HAT. Gratuit ce dernier est disponible au format PDF mais aussi sur Google Play et App Store. Au programme : comment faire une animation avec les LED, création d'une «eight ball» et d'un simulateur de gravité.
Lien : <https://goo.gl/bPOOPL>

bas : 40 € pour 8 fonctions, c'est 5 € par joujou ! Il ne manque que le GPS ou un module RTC ! Le Sense HAT est tellement prometteur qu'il va même dans l'espace (voir page 42). Bien sûr toutes ces fonctionnalités sont programmables avec Python (ou Scratch). On peut les interpréter, enregistrer les données et les utiliser pour faire un robot ou un appareil de domotique. Seule petite ombre au tableau, la sonde de température, proche de la carte et des LED va forcément indiquer des données plus élevées que la réalité. En été, à l'intérieur et avec les LED allumées, nous avons constaté une différence de +7 °C (ce qui est énorme). Il convient donc d'utiliser cette fonction en ayant ça en tête, en recalibrant la sonde avec la différence que vous aurez constatée. Bien sûr en hiver cet écart sera négligeable. Pour la rédaction, si vous ne deviez acheter qu'un seul HAT, c'est celui-là qu'il vous faut...

→ DES EXEMPLES DE SCRIPTS

Si vous avez la dernière version de Raspbian vous aurez tout ce qu'il faut pour faire fonctionner votre Sense HAT dans `/usr/src/sense-hat/examples/python-sense-hat`. Dans un terminal, changer votre répertoire de travail avec la commande `cd` ou copiez les fichiers dans `/home/pi` si vous préférez laissez le prompt `pi@raspberry:~ $`. Voici les scripts que vous aurez à votre disposition.

`colour_cycle.py` & `rainbow.py` :

Pour afficher toutes les couleurs des LED

`compass.py` :

Va allumer la LED la plus proche du Nord. Attention, il faudra calibrer le magnétomètre pour un résultat convenable :
<https://goo.gl/LPiZoF>

`pygame_joystick.py` :

Permet de tester le joystick

`space_invader.py` :

Affiche un extra-terrestre issu de Space Invaders en utilisant une image PNG de 8x8. Vous voulez faire du «pixel art» vous aussi ? Essayez www.piskelapp.com.

`text_scroll.py` :

Va faire défiler le texte : «One small step for Pi !»

`rotation.py` :

Pour montrer qu'il est possible d'orienter l'afficher dans tous les sens

Rien pour les autres capteurs ? Nous allons y venir... Notez que vous pouvez en rajouter un très pratique ! Lorsque vous arrêtez un programme (`Ctrl + C`), les LED resteront allumées. Pour nettoyer la matrice, ouvrez un éditeur de texte comme LeafPad et tapez :

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()
```

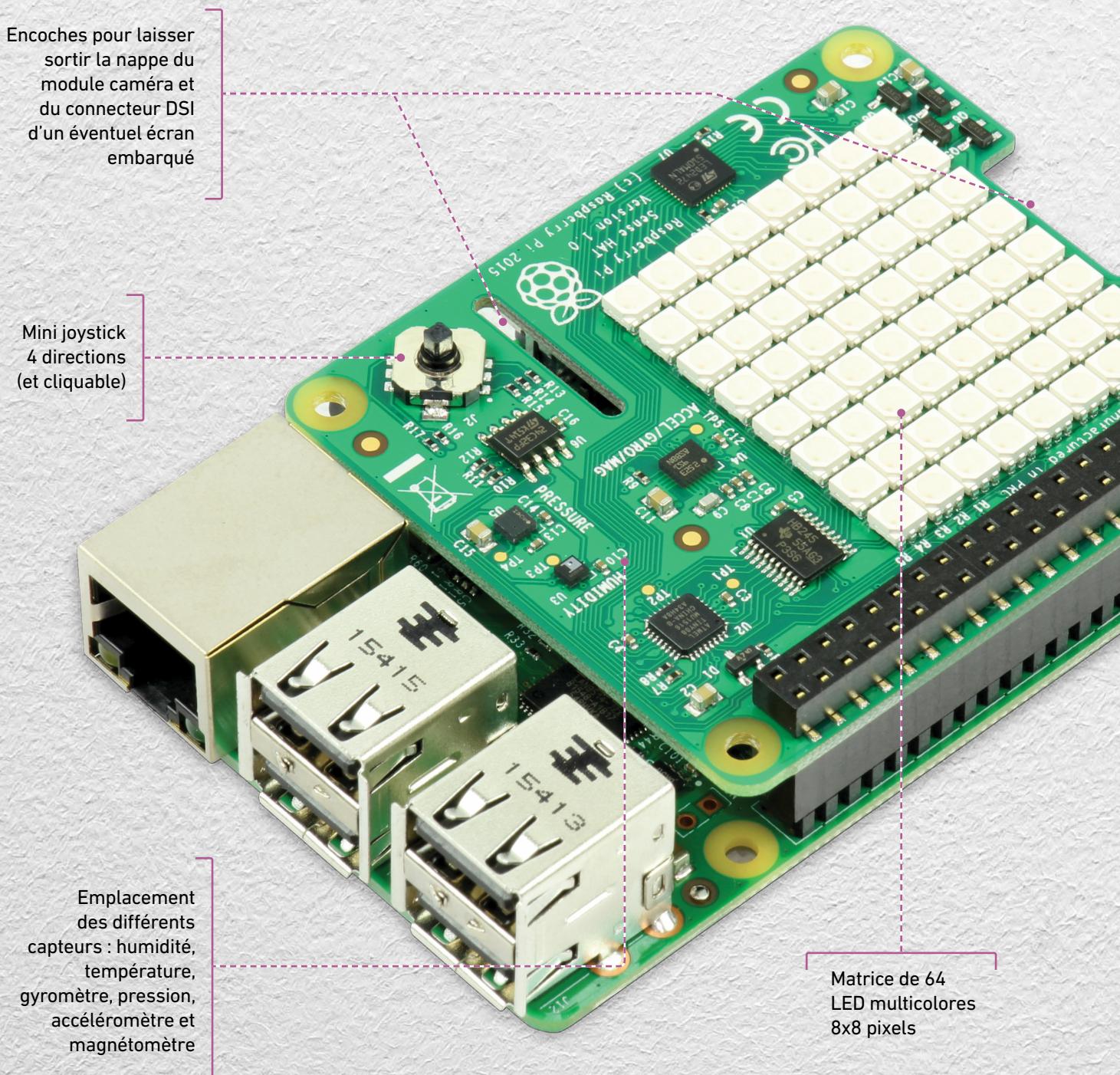
Appelez-le `clear.py` et dans un terminal, lancez :
`python clear.py`

Toutes les LED vont s'éteindre.



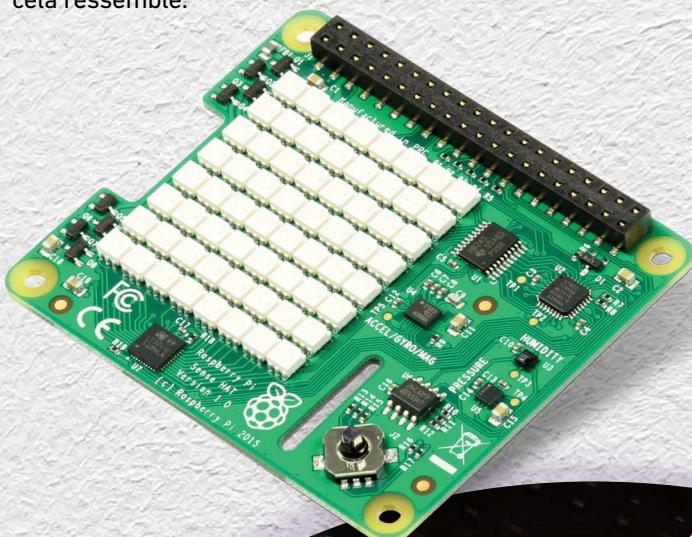


Présentation du

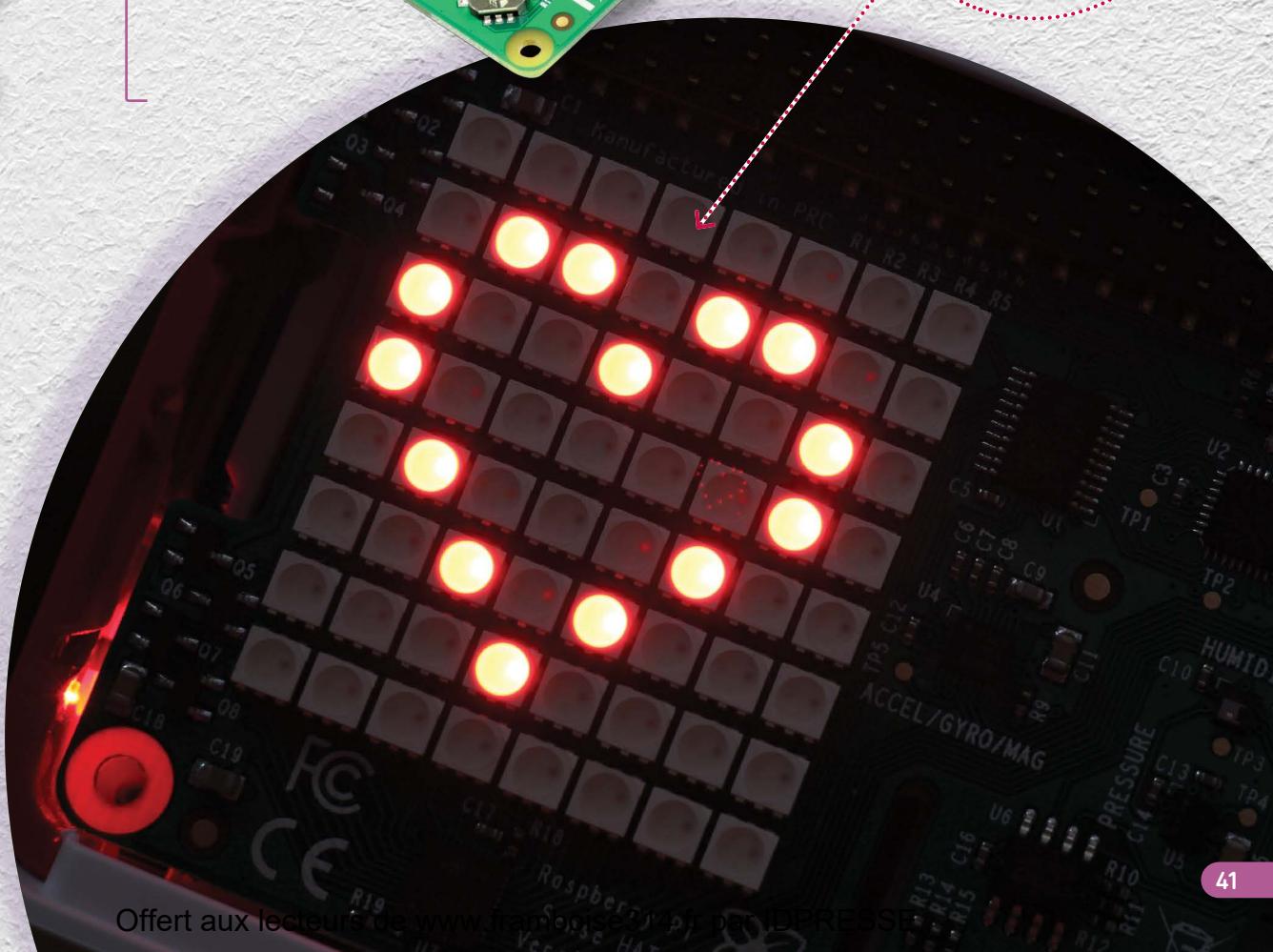
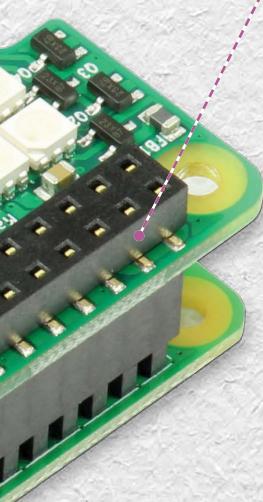


Sense HAT

Le Sense HAT est branché sur les ports GPIO du Raspberry Pi. En le sortant de la boîte, voici à quoi cela ressemble.



À la rédaction
nous aimons
beaucoup le
Sense HAT



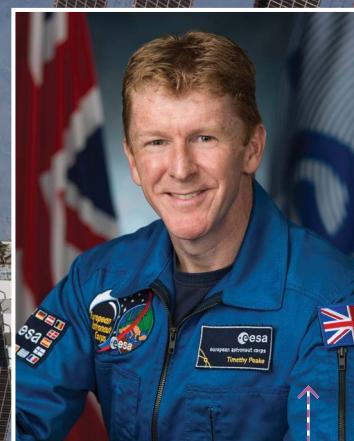


NOS PROJETS COMPLETS



ASTROPI : LE SENSE HAT DANS L'ESPACE !

Initiée par l'agence spatiale britannique, la mission AstroPi avait pour but de faire participer des écoliers à la conception d'expériences scientifiques dans la Station Spatiale Internationale (ISS) en utilisant des Raspberry Pi équipés du Sense Hat. Lancés en décembre 2015, les résultats sont de retour sur terre...



Tim Peake a passé 6 mois dans l'ISS et est revenu sur terre le 18 juin dernier.

C'est le 6 décembre 2015 que sont envoyés dans un cargo Cygnus Ed et Izzy en direction de l'ISS à 400 km au dessus de nos têtes. Les deux compères ne sont pourtant pas astronautes mais deux machines créées pour l'occasion. Les deux joujoux sont de simples Raspberry Pi avec un module Sense HAT, une caméra (standard pour l'un et infrarouge pour l'autre) ainsi qu'un boîtier antichoc imprimable en 3D par Monsieur Tout-le-Monde. Notez que le SenseHAT a lui aussi été spécialement créé pour cette mission. Les deux machines sont parties rejoindre l'astronaute Tim Peake qui les attendait de pied ferme avec des instructions très précises.

UNE EXPÉRIENCE À L'ÉCHELLE DE TOUT UN PAYS

Il faut dire que 6 mois avant le décollage, la mission consistait alors à demander à des écoliers ce qu'ils feraient avec ce type de matériel

à la place des astronautes de l'ISS. Les idées, triées puis validées par des étudiants, ont alors été mises en œuvre par Tim : détection de personnes (ou d'aliens?), affichage du drapeau du pays survolé par l'ISS (en utilisant le magnétomètre), alarme en cas de changement soudain de température, humidité ou pression, jeux de réaction pour contrôler les réflexes des astronautes, détecteur de radiations mortelles (en utilisant la détection de particules dangereuses

grâce à la caméra), etc. Une fois les expériences terminées, un gros fichier CSV plein à craquer de données a été rendu disponible pour les participants et pour tous les curieux <https://goo.gl/Fao1Rn>. Une initiative intéressante mêlant à la fois la Fondation Raspberry, l'ESA (l'agence spatiale européenne), la UK Space Agency, des écoles de toute l'Angleterre et de célèbres Universités.

Lien : <https://astro-pi.org>



Une copie
d'un des deux
AstroPi envoyés
dans l'espace pour
la science



Une mini station météo connectée

Ce petit projet va utiliser les capteurs de température, de pression et d'humidité du Sense HAT pour les publier sur l'afficheur LED ou les envoyer dans votre Google Drive et ainsi garder une trace des données. Suivez le guide...



1 / Ce qu'il vous faut

Commençons par mettre à jour les paquets et installer les éléments qu'il nous faut :

```
sudo apt-get update
sudo apt-get install sense-hat
sudo pip-3.2 install pillow
sudo pip install oauth2client==1.5.2
sudo pip install gspread
sudo pip installPyOpenSSL
```

Si vous avez un message d'erreur concernant ces dépendances, utilisez Google pour savoir comment les mettre à jour avec **--upgrade**.

```
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ sudo pip-3.2 install pillow
Requirement already satisfied (use --upgrade to upgrade): pillow in /usr/lib/pyt
hon3/dist-packages
Cleaning up...
pi@raspberrypi:~ $ pip install --upgrade oauth2client■
```

L'AUTEUR



**RAMIN
SANGESARI**

Ramin Sangesari a 29 ans, vit en à Téhéran en Iran. Diplômé en sciences informatiques et passionné par la programmation, Ramin est le créateur du script MiniWeatherStation que nous avons un peu modifié.

Lien :
[www.hackster.io/
idreams](http://www.hackster.io/idreams)



Le lien entre Google et vous

Google se méfie des applications externes qui viennent interagir avec ses propres services. Il faudra donc demander une autorisation spéciale. Pour l'obtenir, il faudra aller sur ce site :

<https://console.developers.google.com/project>

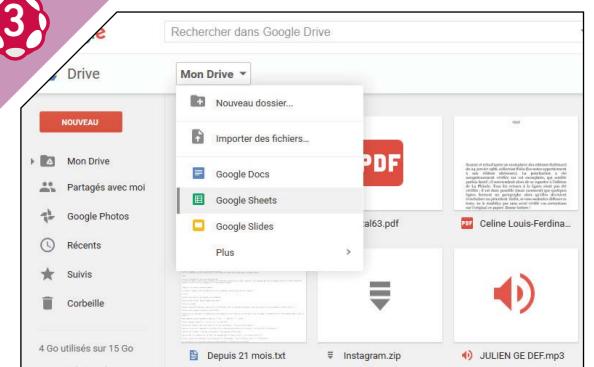
The screenshot shows the 'Identifiants' tab of the Google Cloud Platform 'OAuth 2.0 Client ID' creation interface. It includes fields for 'Clé API', 'ID client OAuth', 'Compte de service', and 'Aidez-moi à choisir'. At the bottom is a 'Créer des identifiants' button.

Cliquez dans **Google APIs** en haut à gauche puis dans **API Google Apps** sélectionnez **Sheets API**. Créez un projet, donnez-lui un nom, faites **Créer** puis **Activer**. Dans **Identifiants** cliquez dans **Créer des identifiants** puis **Clé de compte de services**. Dans **Compte de service** sélectionnez **Compute Engine default service account** et **JSON**. Encore une fois, sélectionnez **Créer**.

The screenshot shows the 'Ouverture de Sense-4fade3c8c435.json' dialog box in Firefox. It asks if you want to open the file with 'Application Windows Wordpad (défaut)' or 'Enregistrer le fichier'. There's also a checkbox for 'Toujours effectuer cette action pour ce type de fichier.' at the bottom.



NOS PROJETS COMPLETS



Le Partage du fichier

Le navigateur va alors automatiquement télécharger un fichier de ce type **Sense-4fade3c8c435.json**.

Ouvrez ce fichier avec un éditeur de texte et cherchez la ligne :

```
client_email»: <375438783019-compute@developer.gserviceaccount.com
```

Dans votre Google Drive, ouvrez un fichier Google Sheets, faites **Partager** et coller

375438783019-compute@developer.gserviceaccount.com dans le champ. Validez.

Pour évitez d'avoir les première 1000 lignes vides, sélectionnez les lignes 2 à 1000 (en commençant la sélection depuis le **2**) puis avec un clic droit faites **Effacer les lignes 2 – 1000**. Placez les fichiers **Sense-4fade3c8c435.json** et **MiniWeatherStation.py**

(prenez notre version modifiée ici : <https://goo.gl/ncZYcs>) dans le même dossier sur votre Raspberry Pi (pour nous, **/home/pi**)

```
Fichier Édition Rechercher Options Aide
#!/usr/bin/python

import json
import sys
import time
import datetime

# libraries
import sys
import urllib
import json
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
from sense_hat import SenseHat

# Oauth JSON File
GDOCS_OAUTH_JSON      = 'Sense2-6e45b1d5494d.json'

# Google Docs spreadsheet name.
GDOCS_SPREADSHEET_NAME = 'Sense'

# How long to wait (in seconds) between measurements.
FREQUENCY_SECONDS       = 30

def login(open_sheet, oauth_key_file, spreadsheet):
```



Personnalisation du code

Ouvrez **MiniWeatherStation.py** avec un éditeur de texte et modifiez les lignes suivant avec le nom du fichier JSON et le nom de votre Google Sheets. Pour nous cela donnera :

Oauth JSON File

```
GDOCS_OAUTH_JSON      = 'Sense-4fade3c8c435.json'
```

Google Docs spreadsheet name.

```
GDOCS_SPREADSHEET_NAME = 'Sense'
```

Et c'est fini ! Il suffit de faire :

```
sudo python2.7 MiniWeatherStation.py
```

L'afficheur LED 8x8 va faire défiler les données. Ces dernières iront aussi prendre place dans les colonnes de votre fichier Google Sheets. Pour stopper le script faites **Ctrl+C**.

```
Fichier Édition Onglets Aide
Cleaning up...
pi@raspberrypi:~ $ sudo python2.7 MiniWeatherStation.py
Logging sensor measurements to Sense every 30 seconds.
Press Ctrl-C to quit.
Temperature (C): 37.5
Humidity: 47.2
Pressure: 0.0

Wrote a row to Sense
Temperature (C): 37.2
Humidity: 45.8
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.4
Humidity: 45.1
Pressure: 1016.6

Wrote a row to Sense
Temperature (C): 37.3
Humidity: 45.2
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.3
Humidity: 45.2
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.5
Humidity: 45.6
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.1
Humidity: 45.8
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.2
Humidity: 45.4
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.4
Humidity: 45.1
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.1
Humidity: 45.5
Pressure: 1016.5

Wrote a row to Sense
Temperature (C): 37.0
Humidity: 46.1
Pressure: 1016.4
```

	A	B	C	D	E
1001	2016-06-20 15:41:37.5		47.2	0.0	
1002	2016-06-20 15:41:37.2		45.8		1016.5
1003	2016-06-20 15:41:37.4		45.1		1016.6
1004	2016-06-20 15:41:37.3		45.2		1016.5
1005	2016-06-20 15:41:37.4		45.2		1016.5
1006	2016-06-20 15:41:37.5		45.6		1016.5
1007	2016-06-20 15:51:37.1		45.8		1016.5
1008	2016-06-20 15:51:37.2		45.4		1016.5
1009	2016-06-20 15:51:37.4		45.1		1016.5
1010	2016-06-20 15:51:37.1		45.5		1016.5
1011	2016-06-20 15:51:37.0		46.1		1016.4
1012					
1013					



MiniWeatherStation : le code complet

Ce code de Ramin Sangesari est sous GNU/GPLv3, vous pouvez donc le modifier et le redistribuer à condition de respecter la licence. Vous en trouverez une copie ici : <http://goo.gl/hpiBlx> et une traduction non officielle ici : <http://goo.gl/jX4kzz>. Notez que nous l'avons nous même modifié pour le faire fonctionner avec Python 2.7 et 3.

```
#!/usr/bin/python

import json
import sys
import time
import datetime

# Librairies nécessaires
import sys
import urllib
import json
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
from sense_hat import SenseHat

# Token du fichier Oauth JSON
GDOCS_OAUTH_JSON      = 'Mini Weather Station-f68343d4a35a.json'

# Nom du fichiers Google Docs
GDOCS_SPREADSHEET_NAME = 'Sense HAT Logs'

# Temps d'attente en secondes entre les mesures
FREQUENCY_SECONDS      = 30

def login_open_sheet(oauth_key_file, spreadsheet):
    """Connect to Google Docs spreadsheet and
    return the first worksheet."""
    try:
        json_key = json.load(open(oauth_key_file))
        credentials = SignedJwtAssertionCredentials(
            json_key['client_email'],
            json_key['private_key'],
            ['https://spreadsheets.google.com/feeds'])
        gc = gspread.authorize(credentials)
        worksheet = gc.open(spreadsheet).sheet1
        return worksheet
    except Exception as ex:
        print ("Unable to login and get
spreadsheet. Check OAuth credentials, spreadsheet
name, and make sure spreadsheet is shared to the
client_email address in the OAuth .json file!")
        print ("Google sheet login failed with
error:"), ex
        sys.exit(1)

sense = SenseHat()
```

```
sense.clear()
print ("Logging sensor measurements to {} every {} seconds.".format(GDOCS_SPREADSHEET_NAME, FREQUENCY_SECONDS))
print ("Press Ctrl-C to quit.")
worksheet = None
while True:
    # Login if necessary.
    if worksheet is None:
        worksheet = login_open_sheet(GDOCS_OAUTH_JSON, GDOCS_SPREADSHEET_NAME)

    # Attempt to get sensor reading.
    temp = sense.get_temperature()
    temp = round(temp, 1)
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)

    # Affichage sur la matrice LED. Vous pouvez
    # supprimer cette partie pour éviter la surchauffe
    # inutile
    sense.clear()
    info = 'Temperature (C): ' + str(temp) +
    'Humidity: ' + str(humidity) + 'Pressure: ' +
    str(pressure)
    sense.show_message(info, text_colour=[255, 0,
0])

    # Print
    print ("Temperature (C): "), temp
    print ("Humidity: "), humidity
    print ("Pressure: "), pressure, "\n"

    # Place les données sur le tableau de Google
    Drive
    try:
        worksheet.append_row((datetime.datetime.now(),
temp,humidity,pressure))
    except:
        # Error appending data, most likely
        # because credentials are stale.
        # Null out the worksheet so a login is
        # performed at the top of the loop.
        print ("Append error, logging in
again")
        worksheet = None
        time.sleep(FREQUENCY_SECONDS)
        continue

    # Attente de 30 secondes avant de poursuivre
    print ("Wrote a row to {}").format(GDOCS_SPREADSHEET_NAME)
    time.sleep(FREQUENCY_SECONDS)
```



RADIO PIRATE SUR LA BANDE FM

CE QU'IL VOUS FAUT

Raspbian

Où le trouver ? :
www.raspberrypi.org

Win32 Disk Imager

Où le trouver ? :
<http://sourceforge.net/projects/win32diskimager>

PiFm

Où le trouver ? :
<http://goo.gl/ADgEC>

PiFmRds

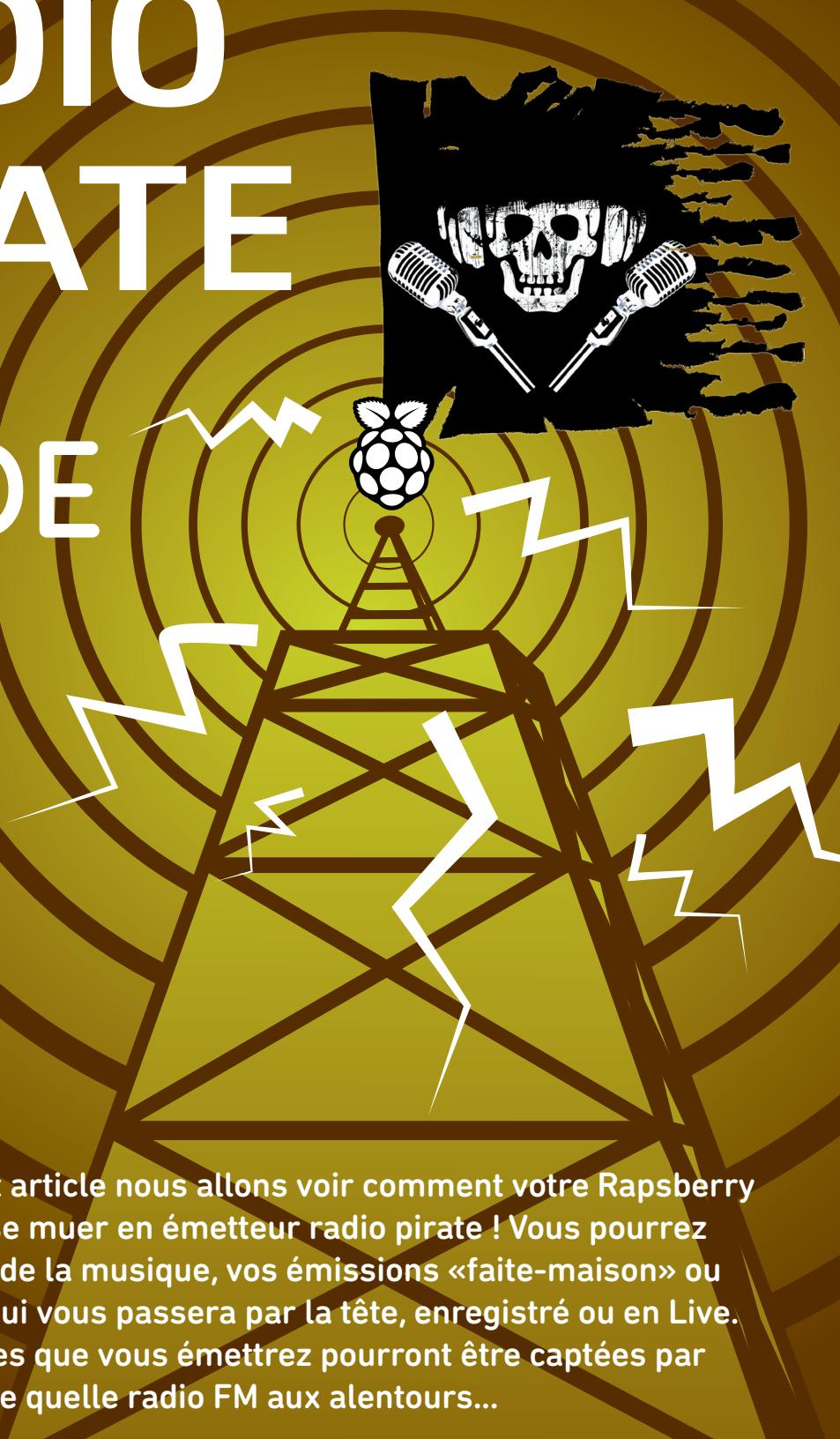
Où le trouver ? :
<https://goo.gl/0iHu1m>

Format Factory

Où le trouver ? :
www.pcftime.com

- Un Raspberry Pi (toutes versions)
- Une carte SD d'au moins 4 Go

Difficulté :



Dans cet article nous allons voir comment votre Raspberry Pi peut se muer en émetteur radio pirate ! Vous pourrez diffuser de la musique, vos émissions «faite-maison» ou tout ce qui vous passera par la tête, enregistré ou en Live. Les ondes que vous émettrez pourront être captées par n'importe quelle radio FM aux alentours...



Dans les années 60 et 70, les radios pirates étaient légion et permettaient de proposer une alternative aux médias jugés trop proches du pouvoir ou de diffuser de la musique interdite. Avec l'essor des radios libres dans les années 80, les radios pirates ont périclité en France. Mais depuis les printemps arabes, ces radios contre-pouvoir refont surface.

GOOD MORNING ENGLAND !

Pourquoi ne pas imaginer diffuser votre propre radio depuis chez vous ? Il vous suffit pour cela d'un Raspberry Pi, de quelques logiciels et d'un fil électrique faisant office d'antenne. Plus ce dernier sera long et plus la portée de votre émission sera importante, mais attention, avec un appareil comme le Raspberry Pi, vous ne pourrez pas dépasser les 100 mètres de portée. Largement de quoi s'amuser un peu, diffuser de la musique à la maison ou dans une résidence d'étudiants. Attention cependant à éviter la case prison (voir encadré). Nous verrons comment ajouter les logiciels à Raspbian puis configurer le Raspberry Pi. Enfin, nous verrons comment personnaliser votre projet en préparant une émission enregistrée. Rappelons enfin que le projet d'origine appelé PiFm fonctionnait très bien avec la première version du Raspberry Pi, mais plus du tout avec les versions 2 et 3. C'est pourquoi nous vous proposons une modification de PiFm : PiFmRds. Cette dernière, censée diffuser un son moins

Même avec une courte portée, vous pouvez imaginer diffuser votre musique ou vos émissions depuis chez vous !

PIRATER LA BANDE FM ? C'EST POSSIBLE AVEC TOUTES LES VERSIONS DE RASPBERRY PI...

pur (mais nous ne l'avons pas constaté), propose par contre la prise en charge du système RDS capable d'afficher le nom de votre station. Pour satisfaire tout le monde, nous avons abordé toutes les versions de la machine dans notre tutoriel...

RADIO PIRATE 76

103.3 MHz sur la bande FM

Centre de Bolbec le 5 juillet 2016

L'émission sera passée 3 fois en boucle de 18 heures à 22 heures avec une interruption de 5 minutes entre chaque session puis nous reprendrons l'antenne en live à partir de 22h30. Vous pourrez participer aux débats en téléphonant au numéro qui vous sera communiqué pendant l'émission.



Au programme :

- Les technologies qui embêtent la NSA (pour le moment)
- Affaire contre le FBI, ce que Apple aimerait bien vous faire croire
- Le chiffrement, un truc de parano ou une bonne habitude ?
- Les "darknets" vus par les médias traditionnels



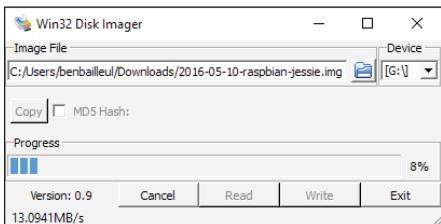
NOS PROJETS COMPLETS

La bande FM à portée de main



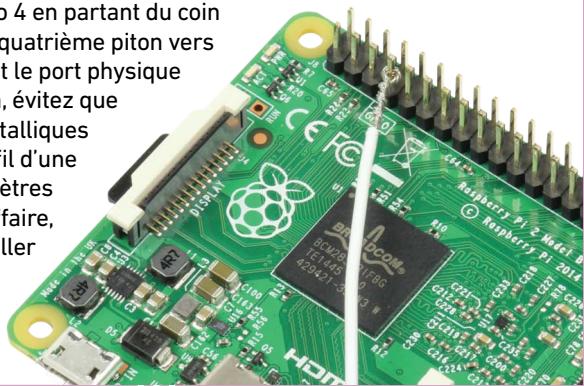
Le logiciel et l'OS

Les logiciels PiFm et PiFmRds qui permettent l'émission des ondes fonctionnent sous Raspbian. Il faudra bien sûr commencer par installer ce système sur votre machine (voir notre article page 24) en vous aidant du logiciel Win32 Disk Imager si vous êtes «Windowsien». Munissez-vous d'une carte d'au moins 4 Go, mais rien ne vous empêche d'utiliser ces logiciels sur le Raspbian que vous utilisez comme PC d'appoint (avec souris/clavier/écran). La communication par SSH est aussi possible (voir page 24).



Les branchements

Introduisez la carte SD dans votre Raspberry Pi et branchez ce dernier sur l'alimentation ainsi que sur un port Ethernet libre de votre routeur/box. Si vous avez un dongle Wi-Fi ou la version 3 du Raspberry, vous pouvez aussi utiliser le réseau sans fil. Branchez le fil électrique sur le port GPIO numéro 4 en partant du coin de la carte. C'est le quatrième piton vers l'intérieur mais c'est le port physique numéro 7. Attention, évitez que d'autres parties métalliques ne se touchent. Un fil d'une vingtaine de centimètres fera amplement l'affaire, mais vous pouvez aller jusqu'à 70 cm pour «booster» le signal.



Installation de PiFm (Raspberry Pi 1)

Nous allons maintenant télécharger, installer et faire fonctionner PiFm depuis le Raspberry 1. Après le prompt, tapez en faisant attention aux capitales :

```
wget www.icrobotics.co.uk/wiki/images/c/c3/Pifm.tar.gz
```

Puis, à la fin du téléchargement, tapez pour décompresser l'archive :

```
tar -xvf Pifm.tar.gz
```

Pour lancer le test, tapez enfin :

```
sudo ./pifm sound.wav 100.1
```

Le 100.1 correspond à la fréquence sur laquelle vous allez émettre. Vous pouvez mettre n'importe quel chiffre entre 87.1 et 108.1. Il ne reste plus qu'à mettre un récepteur radio sur la fréquence que vous avez choisie. Si vous entendez le thème de Star Wars, c'est que vous avez réussi ! C'est en effet le fichier sound.wav de test fourni avec le logiciel. Faites **Ctrl + C** pour arrêter l'émission.

ATTENTION !



Comme vous le savez probablement, l'émission d'ondes radios non autorisée est illégale. Notre projet a un but scientifique et démonstratif, mais vous pouvez quand même tomber sous le coup de la loi. Bien sûr, la portée réduite de notre expérience ne permettra pas d'encombrer les autres communications, mais si vous habitez près d'un commissariat, d'une base militaire, d'un aéroport ou d'un hôpital, nous vous déconseillons d'émettre. Les appareils d'Etats n'utilisent plus ce genre d'ondes, mais vous ne voudriez pas froisser le CSA...

```
Fichier Édition Onglets Aide
pi@raspberrypi: ~ $ wget www.icrobotics.co.uk/wiki/images/c/c3/Pifm.tar.gz
2016-05-17 13:54:26 -> http://www.icrobotics.co.uk/wiki/images/c/c3/Pifm.tar.gz
Résolution de www.icrobotics.co.uk (www.icrobotics.co.uk)... 155.198.3.147
Connexion à www.icrobotics.co.uk (www.icrobotics.co.uk)|155.198.3.147|:80
HTTP request sent, awaiting response... 200 OK
Taille : 5521400 [5,3M] [application/x-gzip]
Sauvegarde en : « Pifm.tar.gz »

Pifm.tar.gz          100%[=====]   5,26M  344KB/s  ds 1
2016-05-17 13:54:40 (384 KB/s)  => « Pifm.tar.gz » sauvegardé [5521400/5521
pi@raspberrypi: ~ $ tar -xvf Pifm.tar.gz
pi@raspberrypi: ~ $ pifm
pi@raspberrypi: ~ $ pifm.c
pi@raspberrypi: ~ $ pifm.py
pi@raspberrypi: ~ $ sudo ./pifm sound.wav 100.1
```



Installation de PiFmRds (Raspberry Pi 2 & 3)

Pour les modèles plus récents de Rapsberry, nous allons utiliser le programme de Christophe Jacquet. Faites :

```
sudo apt-get install libsndfile1-dev
git clone https://github.com/ChristopheJacquet/
PiFmRds.git
```

Allez dans le bon dossier avec :
cd PiFmRds/src

Compilez le programme avec :
make clean
make

Tapez ensuite :

```
sudo ./pi_fm_rds -audio sound.wav
```

Vous devriez entendre un monsieur dire «**FM RDS transmitter using the Raspberry Pi**» en boucle sur la fréquence 107.9.

Toujours dans le dossier PiFmRds/src nous allons pouvoir affiner un peu la commande.

```
sudo ./pi_fm_rds -freq 103.3 -audio sound.wav -pi
FFFF -ps PIRATE -rt 'La Radio Pirate'
```

Dans cet exemple, le son **sound.wav** va être diffusé sur la fréquence **103.3** MHz. Un autoradio RDS affichera **PIRATE** (8 caractères max) puis **La Radio Pirate** (64 caractères max). L'argument -pi ne vous servira pas à grand-chose, mais nous l'avons laissé. Il s'agit du PI code du signal RDS (4 caractères hexadécimaux) qui permet de récupérer un signal si le récepteur s'éloigne un peu trop de la source. C'est très intéressant, mais comme la portée est ici réduite, vous pouvez laisser tomber. Faites **Ctrl + C** pour arrêter l'émission.

```
Fichier Édition Onglets Aide
pi@raspberrypi:~/PiFmRds/src $ sudo ./pi_fm_rds
Using mbox device /dev/vcio.
Allocating physical memory: size = 3403776      mem_ref = 5      bus_addr = fd6bc0
00      virt_addr = 0x7691a000
ppm corr is 0.0000, divider is 1096.4912 (1096 + 2012*2^12) [nominal 1096.4912]
.
PI: 1234, PS: <Varying>
RT: PiFmRds: live FM-RDS transmission from the RaspberryPi*
Starting to transmit on 107.9 MHz*
Termination: cleanly deactivated the DMA engine and killed the carrier.
pi@raspberrypi:~/PiFmRds/src $ sudo ./pi_fm_rds -audio sound.wav
Using mbox device /dev/vcio.
Allocating physical memory: size = 3403776      mem_ref = 5      bus_addr = fd6bc0
00      virt_addr = 0x7690e5000
ppm corr is 0.0000, divider is 1096.4912 (1096 + 2012*2^12) [nominal 1096.4912]
.
Using audio file: sound.wav
Input: 228000 Hz, upsampling factor: 1.00
1 channel, monophonic operation.
Created low-pass FIR filter for audio channels, with cutoff at 12000.0 Hz
RT: "La Radio Pirate"
Starting to transmit on 103.3 MHz.
```



C'est fini !

Avec notre antenne de 20 centimètres depuis l'intérieur d'une maison, nous avons pu émettre jusqu'à une cinquantaine de mètres. Il est possible de booster le signal en mettant votre montage en extérieur et en hauteur. Selon les développeurs, il est possible d'atteindre une portée de 100 mètres. Sachez aussi qu'il est possible de faire fonctionner le Raspberry Pi sur batterie... Dans le tutoriel suivant, nous allons voir comment faire votre propre émission au bon format, que vous utilisiez PiFm ou PiFmRds... Mais avant il nous reste à voir comment faire du «live».



Émission en «live»

Si vous avez un micro USB comme on en trouve sur les jeux karaoké de console de salon, vous pourrez même diffuser votre émission en direct. Branchez le micro sur un des ports USB du Raspberry (cela a fonctionné chez nous avec une webcam équipée d'un micro) et pour PiFmRds, tapez :

```
cd PiFmRds/src
sudo arecord -fS16_LE -r 44100 -Dplughw:1,0 -c 2 -l
sudo ./pi_fm_rds -audio -
```

Avec PiFm faites :

```
sudo arecord -fS16_LE -r 22050 -Dplughw:1,0 -l sudo ./
pifm - 100.1 22050
```

Amusez-vous bien !

```
Fichier Édition Onglets Aide
pi@raspberrypi: ~ cd PiFmRds/src
pi@raspberrypi: ~/PiFmRds/src $ sudo ./pi_fm_rds -freq 103.3 -audio 01.wav -ps PI
RATE -rt 'La Radio Pirate'
Using mbox device /dev/vcio.
Allocating physical memory: size = 3403776      mem_ref = 6      bus_addr = fd37c0
00      virt_addr = 0x7690e000
ppm corr is 0.0000, divider is 1096.4912 (1096 + 2012*2^12) [nominal 1096.4912]
.
Using audio file: 01.wav
Input: 44100 Hz, upsampling factor: 5.17
2 channels, generating stereo multiplex.
Created low-pass FIR filter for audio channels, with cutoff at 12000.0 Hz
PI: 1234, PS: "PIRATE"
RT: "La Radio Pirate"
Starting to transmit on 103.3 MHz.
```



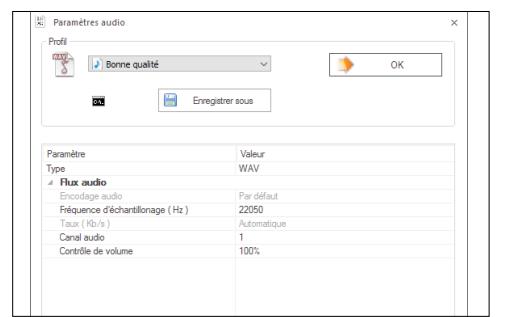
Encodage de votre émission

PAS À PAS



Un format bien précis avec PiFm

Pour pouvoir diffuser vos propres émissions ou chansons avec PiFm, il faudra respecter un format et des spécifications précises. Les fichiers devront être en WAV 16 bit, 22050 Hz et mono. Le nom du fichier ne devra pas comporter d'espace. Commençons par télécharger le logiciel Format Factory et installez-le. Sur la droite dans la catégorie **Audio**, sélectionnez **Tout type vers WAV**.



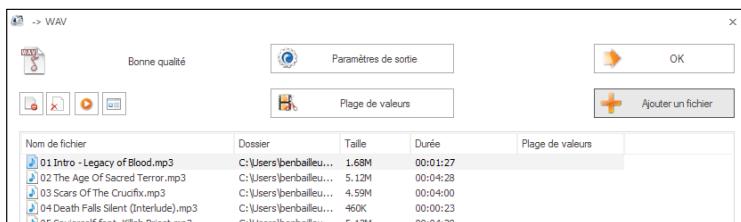
L'encodage

Dans **Paramètres de sortie**, mettez **22050** dans **Fréquence d'échantillonage** et **1** dans **Canal Audio**. Validez et, dans **Ajouter un fichier**, trouvez votre chanson ou l'émission que vous aurez enregistrée avec le **Magnétophone** de Windows par exemple. Choisissez le dossier de destination et faites **OK**. Cliquez sur **Démarrer** en haut de la fenêtre principale et attendez la fin de l'encodage.



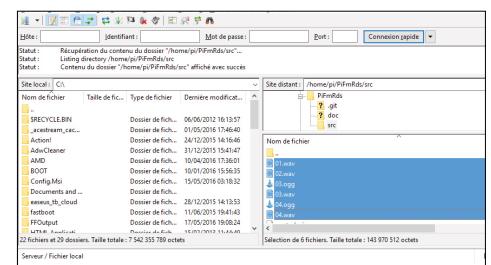
Avec PiFmRds, c'est plus simple

Comme PiFmRds utilise la librairie libsndfile, les fichiers compatibles seront «assaisonnés» correctement à la volée. La MP3 ne fait pas partie du lot, il faudra tabler sur du WAV ou de l'OGG. Si votre fichier est stéréo, PiFmRds reproduira la stéréo. Là aussi le logiciel Format Factory vous sera précieux pour convertir vos fichiers (**44100** d'échantillonage et **Canal Audio 2** pour la stéréo). Sous Linux, vous avez SoundConverter et iTunes sous Mac bien sûr...



Placez votre fichier

Pour placer votre fichier son sur le Raspberry Pi, vous pouvez le faire avec une clé USB (chemin **/media/pi**) ou via FileZilla. Dans **Fichier**, choisissez **Gestionnaire de Sites**, puis **Nouveau Site** et sur la droite entrez l'adresse IP du Raspberry, le port **22**, protocole **SFTP** et dans **Type d'authentification**, mettez **Normale**. Tapez **pi** comme identifiant et **raspberry** comme mot de passe. En faisant **Connexion**, vous devriez avoir accès aux entrailles de la bête. Déplacez votre fichier **test.wav** dans le répertoire principal qui s'appelle **pi** (pour PiFmRds, le répertoire est toujours **pi/PiFmRds/src**). Il ne vous reste qu'à lancer la diffusion. Voilà ! Vous émettez votre propre émission ! Votre petite amie écoute telle station FM sur 102.3 ? Amenez-vous près de chez elle avec votre message romantique enregistré et diffusez-le. Quand on n'a pas de guitare...

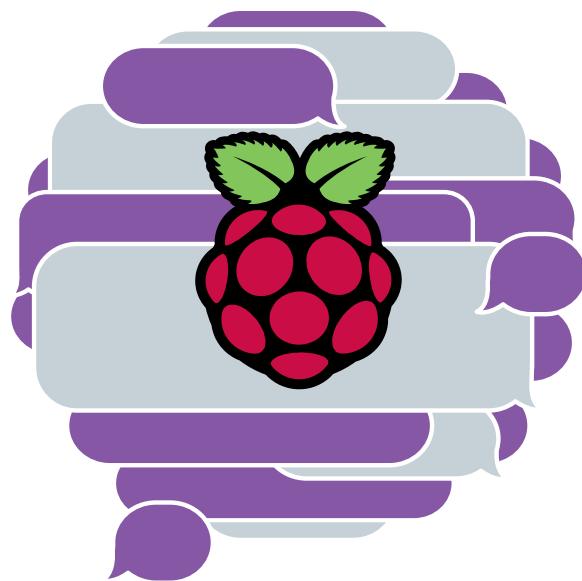




NOS PROJETS COMPLETS

RASPISMS : PROGRAMMEZ ET AUTOMATISEZ vos SMS

Avez-vous déjà eu besoin d'envoyer un SMS alors que vous n'aviez pas de réseau téléphonique ? Probablement, à coup sûr même si vous habitez en «zone blanche», ces zones du territoire français confrontées au paradoxe de posséder internet, mais toujours pas de téléphone portable ! Aujourd'hui, fin de la galère, nous allons vous expliquer comment envoyer des SMS avec le Raspberry Pi, une clef 3G et la plate-forme RaspiSMS !



CE QU'IL VOUS FAUT

RaspiSMS

Où le trouver ? :
<http://raspisms.raspbian-france.fr/download>

Win32 Disk Imager

Où le trouver ? :
<http://goo.gl/eZNTZA>

- Un Raspberry Pi (toutes versions)
- Une clef 3G/GSM
- Un abonnement à un opérateur téléphonique

Difficulté :

Bien plus qu'un simple système d'envoi et de réception de messages, RaspiSMS est une plate-forme complète qui pourra répondre à de très nombreux besoins. Ainsi, le logiciel permet bien entendu d'envoyer et de recevoir des SMS, mais il permet également de gérer des contacts et des groupes de contacts, d'envoyer des SMS à des heures programmées, de contrôler un serveur par SMS, de faire suivre des SMS par e-mail et bien d'autres choses ! Si RaspiSMS peut

être utilisé comme un simple outil de conversation pour les utilisateurs aux besoins les plus simples, les développeurs ou les entreprises peuvent s'en servir pour des actions beaucoup plus poussées. Ainsi, grâce à sa gestion des «SMS STOP» et sa possibilité de programmer des SMS à heure fixe, les petites entreprises peuvent utiliser RaspiSMS pour garder un contact avec leurs clients et les informer de nouveautés. Du côté des développeurs, une API leur permet de programmer des SMS sur la plate-forme depuis leur code. Ils peuvent



ainsi envoyer des messages à leurs utilisateurs, pour valider une inscription par exemple. De leur côté, les administrateurs systèmes peuvent se servir de RaspiSMS en le couplant à des outils de monitoring afin d'être avertis sur leur téléphone en cas de panne de leur serveur. Serveur qu'ils pourront éventuellement prendre en main par SMS s'ils ne peuvent accéder à internet ! D'autres usages peuvent encore être inventés, la plupart des outils étant facilement interconnectables avec RaspiSMS !

UN OUTIL AUX POSSIBLITÉS MULTIPLES ADAPTÉ AUX DEVELOPPEURS ET AUX ENTREPRISES !

Pour les personnes qui désirent installer RaspiSMS et qui disposent d'une carte SD vierge ou dont ils ne souhaitent pas conserver le contenu actuel, Raspbian France propose une image au format IMG de Raspbian avec RaspiSMS préinstallé. Sans doute la méthode la plus simple, l'installation de RaspiSMS sur une carte SD dédiée se fait en quelques minutes ! L'avantage de cette méthode réside dans la possibilité d'avoir un système Raspbian complet sur lequel vous pourrez installer tous les autres logiciels que vous souhaitez ! Mais si vous avez déjà un Raspbian, vous pouvez aussi utiliser un paquet au

format DEB (voir notre encadré). Cette méthode permet de ne pas changer tout le système (pratique si vous n'avez pas plusieurs cartes micro SD) tout en ne nécessitant que le téléchargement d'un fichier de 1 Mo (contre 2,62 Go pour le pack complet).

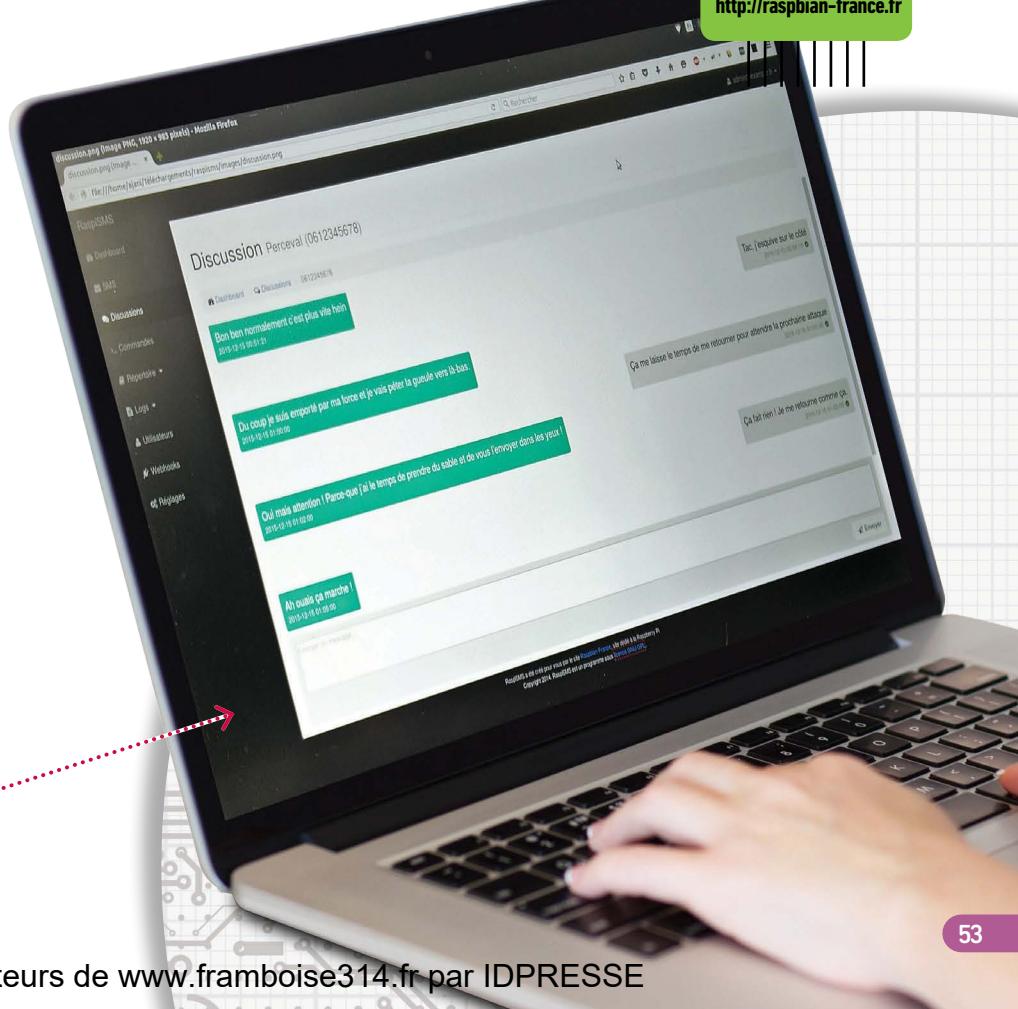
LES PETITES ENTREPRISES PEUVENT UTILISER RASPISMS POUR GARDER CONTACT AVEC LEURS CLIENTS SANS PAYER POUR UN SERVEUR DÉDIÉ.

LES AUTEURS



À l'origine de RaspiSMS, on retrouve le blog Raspbian France qui propose des tutoriels en français autour du Raspberry Pi. C'est parce qu'il a été bloqué de nombreuses fois par l'impossibilité d'envoyer des SMS dans des zones disposant d'Internet, mais pas du réseau GSM qu'un des auteurs décide d'utiliser son Raspberry Pi pour en faire une plate-forme d'envoi de SMS par internet !

Lien :
<http://raspbian-france.fr>



Confectionnez-vous
un serveur SMS
à la maison que
vous pourrez gérer
de n'importe quel
navigateur...



NOS PROJETS COMPLETS

Installation et utilisation de RaspiSMS



Télécharger l'image de RaspiSMS



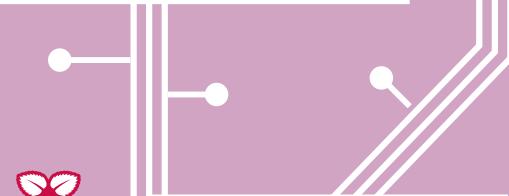
Commençons par télécharger l'image de RaspiSMS en suivant notre lien dans **Les versions>Package .img**. Une fois l'archive en votre possession, vous devrez la décompresser deux fois pour obtenir le fichier .img. Si vous êtes sous Windows, utilisez le logiciel 7-zip. Pour les Linuxiens, utilisez simplement la commande suivante : **tar zxvf /chemin/du/fichier.tar.gz**. Maintenant que nous avons l'image décompressée, nous allons la copier sur notre carte SD.

→ INSTALLER RASPISMS SUR UN SYSTÈME DÉJÀ EXISTANT

Si vous possédez déjà un système Raspbian qui tourne et que vous ne souhaitez pas perdre tout ce qui est dessus, il est possible d'installer RaspiSMS en utilisant un paquet .deb. Avantages de cette méthode, vous pouvez rajouter RaspiSMS à un système déjà existant sans perdre vos données. Par ailleurs, vous pourrez installer RaspiSMS sur une autre machine qu'un Raspberry Pi, tant que celle-ci utilise un système Linux basé sur Debian (Ubuntu par exemple). Inconvénient, vous devrez mettre un tout petit peu plus les mains dans le cambouis. Sur la page de téléchargement, procurez-vous le fichier **raspisms-2.0.deb**, allez dans le dossier où vous l'avez téléchargé avec la commande cd et entrez les commandes suivantes :

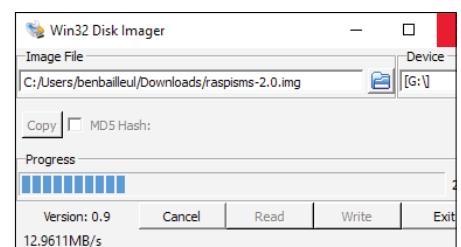
```
sudo apt-get update
sudo dpkg -i ./raspisms-2.0.deb (il faut être dans le même dossier que le fichier raspisms-2.0.deb pour lancer la commande).
sudo apt-get -f install
```

Au cours de l'opération, le système vous remontera probablement des erreurs. Ne vous en faites pas, la dernière commande ci-dessus est justement là pour les régler. Au bout de quelques secondes/minutes, le système vous demandera si vous souhaitez configurer le logiciel automatiquement. Assurez-vous que vous avez bien branché votre clé 3G à la Raspberry Pi et répondez que vous souhaitez configurer le logiciel automatiquement. Il ne vous reste plus qu'à suivre les indications du logiciel étape par étape.



Copier l'image sur la carte SD (Windows)

Pour mettre le contenu de ce dernier dans la carte, il faudra utiliser le logiciel Win32 Disk Imager. Téléchargez le logiciel et munissez-vous d'une carte SD d'au moins 8 Go. Lancez le programme, sélectionnez **raspisms-2.0.img** avec l'icône en forme de dossier puis spécifiez l'emplacement de la carte SD dans la colonne **Device**. Ne vous trompez pas sous peine d'effacer le contenu d'un de vos disques durs ! Faites **Write** et attendez la fin du processus.





Copier l'image sur la carte SD (Linux)

Démontez la carte si elle a été montée automatiquement, trouvez le chemin de votre carte SD, probablement quelque chose comme **/dev/mmcblk0**. Il est possible que vous ayez des partitions sur la carte avec des noms comme **/dev/mmcblk0p1**. Ignorez-les.

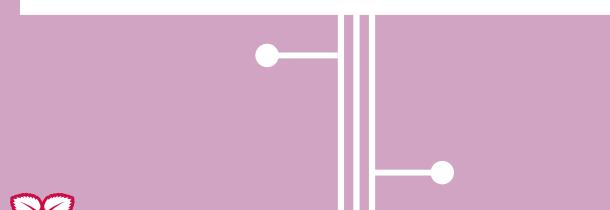
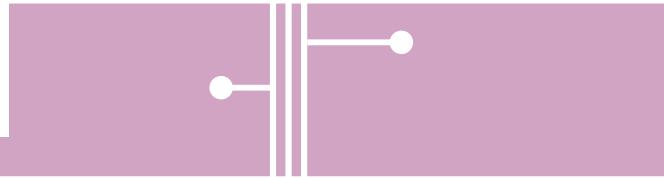
Une fois le chemin de votre carte identifié, lancez simplement la commande suivante :

```
sudo dd if=/chemin/du/fichier.img of=/chemin/de/votre/carte
```

Attendez la fin de la commande (celle-ci est un peu longue et n'affiche aucun signe d'avancement. Ne vous en faites pas, c'est parfaitement normal).

```
ubuntu@ubuntu:/media/40ECF465ECF4569E
File Edit View Search Terminal Help
255 heads, 63 sectors/track, 964 cylinders
Units = cylinders of 10605 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Device Boot Start End Blocks Id System
/dev/sdb1 * 1 965 7747583 b W95 FA
T32
Partition 1 has different physical/logical beginnings (non-Linear):
phys=(1023, 254, 63) logical=(0, 0, 3)
Partition 1 has different physical/logical endings:
phys=(1023, 254, 63) logical=(964, 135, 3)
ubuntu@ubuntu:/media/40ECF465ECF4569E$ sudo dd if=/dev/zero of=/dev/sda5 bs=8M
```



Configurer le code PIN

Maintenant que nous avons copié RaspiSMS sur la carte SD, mettez celle-ci dans votre Raspberry Pi et branchez-le. Nous vous recommandons d'utiliser une clef 3G Huawei E169 qui fonctionne parfaitement avec la Raspberry Pi ! Pour des tests, vous pouvez même vous contenter d'utiliser votre carte SIM actuelle.

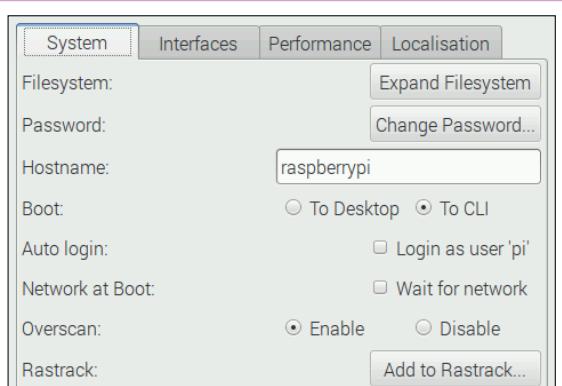
Pour information, les identifiants par défaut sont les suivants :
RaspiSMS > admin@example.fr et admin
Raspberry Pi > pi et raspberry
MySQL > root et raspberry

Dans un premier temps, nous allons modifier un fichier de configuration pour adapter le code PIN à votre carte SIM.Modifier le fichier **/etc/gammu-smsdrc**, avec la commande **sudo nano /etc/gammu-smsdrc** par exemple. Vous pouvez aussi taper **startx** pour accéder à l'interface graphique et changer ces paramètres avec LeafPad. Ajoutez la ligne **pin = XXXX** où vous mettrez votre code PIN à la place des **XXXX**. N'oubliez pas de sauvegarder le fichier.



Étendre la partition

Maintenant, nous allons étendre le système à toute la carte. Pour cela, lancez la commande **sudo raspi-config**. Choisissez **Expand Filesystem** et validez. Profitez-en pour changer la langue du clavier ou les mots de passe. Faites **Finish** et dites que vous voulez redémarrer. De même il est possible de faire ces réglages depuis l'interface graphique si vous n'êtes pas à l'aise avec les lignes de commande. Profitez-en aussi pour activer le WiFi si vous disposez d'un dongle ou d'un Raspberry Pi 3 (voir notre article sur Raspbian page XX). Voilà, l'installation est terminée !





NOS PROJETS COMPLETS



Petit tour du propriétaire

Maintenant que nous avons installé RaspiSMS, nous allons faire un petit tour des fonctionnalités disponibles et des différents écrans de l'application.

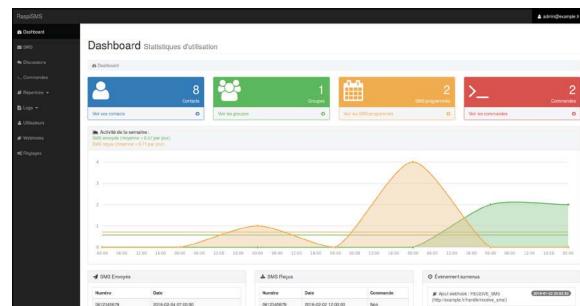
Pour vous connectez à RaspiSMS, rien de plus simple. Il vous suffit d'ouvrir votre navigateur internet et de vous connecter à l'adresse **http://ip_du_raspberry_pi/RaspiSMS** (respectez les capitales). Par exemple, chez nous cela correspond à l'adresse **http://192.168.1.38/RaspiSMS**. Vous pourrez trouver l'adresse de votre Raspberry Pi dans la page de configuration de votre box.

Postes connectés				
#	Adresse MAC	Nom d'hôte	Adresse IP	Port
1	00:04:30:28:4a:14	NETGEM-28a14	192.168.1.53	LAN 1
2	e8:99:c4:b7:29:31	android-ef670c3bc6c27af7	192.168.1.63	WiFi
3	00:1b:67:0c:a7:d5	-	192.168.1.52	Femtocell
4	7c:4f:b5:73:6c:ca	BAILLEUL	192.168.1.94	WiFi
5	00:0b:b4:10:a8:bc	STORAGE-10AABC	192.168.1.76	LAN 3
6	00:22:3f:ff:9b:b2	PC-de-Olsana	192.168.1.77	WiFi
7	b8:27:ab:95:46:f3	raspberrypi	192.168.1.105	WiFi



Connexion et accueil

Une fois que vous êtes connecté, vous arrivez sur la page **Dashboard**. Le menu à gauche vous permet d'accéder aux différentes parties de l'application et en haut à droite, votre nom qui vous permet de vous déconnecter ou d'accéder à votre profil. Cette page vous montre quelques statistiques sur votre utilisation de RaspiSMS ainsi qu'un graphique des SMS envoyés et reçus au cours de la semaine. En dessous du graphique, trois tableaux vous affichent un résumé des derniers SMS envoyés et reçus ainsi qu'une liste des derniers événements enregistrés par RaspiSMS.



SMS programmés

Pour accéder à la programmation d'un SMS pour une heure donnée, cliquez sur **SMS** dans le menu. Vous pouvez cocher les lignes que vous souhaitez sélectionner et les modifier ou les supprimer en cliquant sur **Action pour la sélection** en bas à droite. Vous pouvez également ajouter un nouveau SMS en cliquant sur le bouton **Ajouter un SMS programmé** en bas à gauche. Vous retrouverez ces comportements sur la partie **Commandes**, **Contacts**, **Groupes** et **Utilisateurs**, ainsi qu'un comportement proche pour toute la gestion de la partie **Logs**.



Discussions

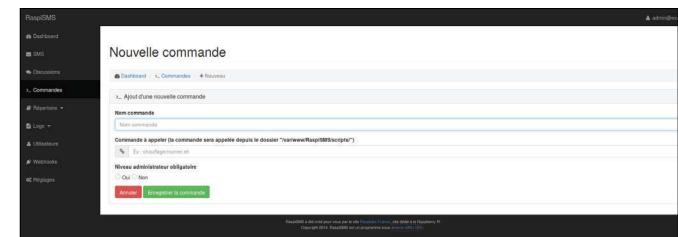
En ajoutant un nouveau SMS programmé, vous serez renvoyé vers la page d'ajout. Vous pourrez définir le texte de votre SMS, la date d'envoi (une date passée déclenchera l'envoi immédiat du SMS), un ou plusieurs numéros au(x)quel(s) envoyer le SMS (ces numéros seront automatiquement passés au format international). Vous pourrez également renseigner des contacts et des groupes auxquels envoyer le SMS. Si la fonction est activée, vous pourrez choisir d'envoyer le SMS au format «flash» qui s'affiche directement sur l'écran du téléphone, un peu comme un popup. Deuxième fenêtre, celle de **Discussions**. Elle vous permettra d'échanger de façon fluide avec un interlocuteur de votre choix.



Les commandes, pour les utilisateurs avancés

Plutôt à destination des administrateurs systèmes ou des personnes voulant faire de la domotique, la fenêtre des **Commandes** vous permet de définir des scripts sur le serveur que vous pourrez appeler par SMS. Pour cela, ajouter une commande comme nous l'avons fait pour les SMS programmés. Dans un premier temps, renseignez le nom de la commande. C'est lui qui sera utilisé dans le SMS de déclenchement. Dans un second temps, renseignez le chemin relatif du script à appeler, depuis le dossier `/var/www/RaspiSMS/scripts/`. Notez que le script devra être exécutable ! Une fois la commande enregistrée, vous pouvez la déclencher en envoyant un SMS au format suivant au numéro de RaspiSMS :

**[<nom commande>:<si besoin paramètres supplémentaires passés à la commande>][[login:<email user>]
[password:<password user>]**



Nouvel utilisateur

Copyright © 2014 Raspbian France. Tous droits réservés à la Framboise314.fr.

Les autres entrées

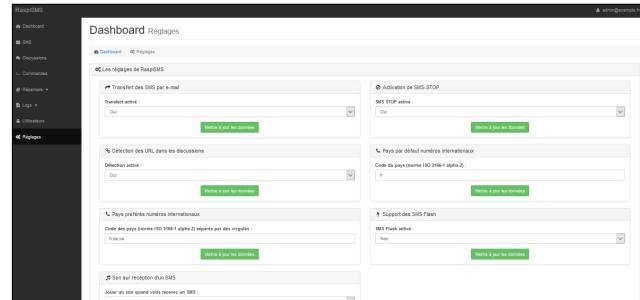
En plus des fonctionnalités directement liées aux SMS, RaspiSMS vous permet de gérer des contacts et des groupes de contacts dans la partie **Répertoire**. Dans **Logs** vous pourrez gérer la fonctionnalité SMS STOP, dans **Événements** vous trouverez l'historique de vos actions tandis que dans **Utilisateurs** vous pourrez gérer plusieurs comptes. Lors de la création d'un utilisateur, vous devez définir son adresse e-mail. Le mot de passe n'est pas obligatoire. Si vous n'en définissez pas, un mot de passe sera généré aléatoirement et envoyé à l'adresse e-mail de l'utilisateur.



Les réglages

Au fur et à mesure qu'il a gagné en fonctionnalités, RaspiSMS a également eu besoin de plus de finesse dans ses réglages. Une fenêtre dédiée est donc présente sous le nom **Réglages**.

Ici vous pouvez donc régler certains détails : transfert des SMS par e-mail, activation des SMS STOP, détection des URL dans les discussions, pays par défaut pour les numéros (le pays doit être représenté par son code sur deux lettres à la norme ISO 3166-1 alpha-2), pays préférés (possibilité de mettre plusieurs pays en les séparant par une virgule), SMS flash et signal sonore en cas de réception.



→ PARTICIPER AU DÉVELOPPEMENT DE RASPISMS

Si vous êtes développeur et que vous souhaitez modifier RaspiSMS ou participer à son amélioration, n'hésitez pas à allez faire un tour sur le GitHub du projet, <https://github.com/RaspbianFrance/RaspiSMS>.

Si vous avez des questions ou besoin d'aide, n'hésitez pas à aller faire un tour sur la documentation de RaspiSMS, <http://raspisms.raspbian-france.fr/documentation> et sur <http://raspbian-france.fr> pour poser vos questions ! Ce projet est sous GNU/GPLv3, vous pouvez donc le modifier et le redistribuer à condition de respecter la licence. Vous en trouverez une copie ici : <http://goo.gl/rIS4jU> et une traduction non officielle ici : <http://goo.gl/SISnM3>.