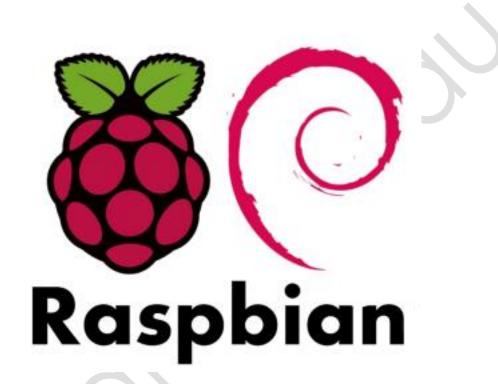


Florian MOREAU

BTS SIO 1



03/09/2018 PAGE **1** SUR **16**



Florian MOREAU

BTS SIO 1

Avez-vous déjà pensé de contrôler une Freebox Player par la voix ? Ce tutoriel va montrer comment, avec un Raspberry Pi 3 (un mini-ordinateur) avec NodeJS (plateforme orienté applications réseau en JavaScript).

Ce tutoriel est tiré des explications du site : https://aymkdn.github.io/assistant-plugins/

Som	maire	
1 Pi	rérequis	3
1.1	Raspberry Pi 3	3
1.2	Compte et code secret IFTTT	
1.3	Compte et code secret PushBullet	4
2 In	stallation de NodeJS	5
2.1	Mise à jour vérification et nettoyage	5
2.2	Installation de NodeJS	5
3 Le	es plugins	6
3.1	Création de leurs emplacements de base	6
3.2	Ajout des plugins souhaités	7
3.3	Configuration	7
3.4	Premier lancement	9
4 C	réation d'un ordre	9
	nnexe (commande)	

03/09/2018 PAGE **2** SUR **16**



Florian MOREAU

BTS SIO 1

1 Prérequis

1.1 Raspberry Pi 3

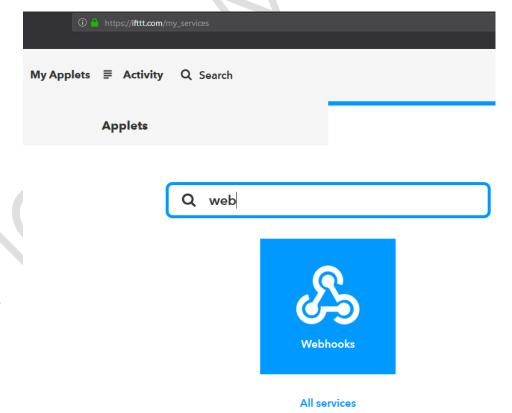
Il faut posséder un Raspberry Pi 3 pour ce tutoriel avec un système d'exploitation nommé Raspbian. C'est un dérivé de Debian spécialement pour Raspberry.

1.2 Compte et code secret IFTTT

Iftt est un service web gratuit qui permet aux utilisateurs de créer des enchaînements d'instructions simple.

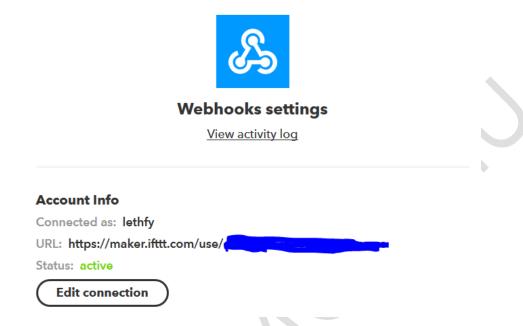
Rendez-vous sur https://ifttt.com afin de créer un compte à l'aide de votre compte Google.

Une fois le compte créé, il faut aller chercher le mot de passe WebHook afin de pouvoir faire communiqué la Google Home avec la Freebox Player. Il se situe dans la catégorie Services puis WebHooks et enfin aller dans les paramètres (Settings).



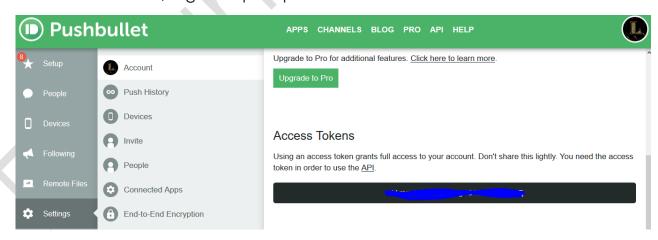
03/09/2018 PAGE **3** SUR **16**

Le code se situe dans le lien affiché à l'écran. Sur la capture d'écran, le code est caché. Celui-ci servira plus tard.



1.3 Compte et code secret PushBullet

Connecter vous à PushBullet avec le même compte Google que IFTTT. Puis aller dans les paramètres (Settings) et cliquer sur « Create Access Token » afin de générer un code secret, à garder pour plus tard.



03/09/2018 PAGE **4** SUR **16**



Florian MOREAU

BTS SIO 1

2 Installation de NodeJS

2.1 Mise à jour vérification et nettoyage

Tout d'abord, mettre à jour le système Raspbian du Raspberry avec les commandes suivantes :

sudo apt-get update

sudo apt-upgrade

Il faut ensuite vérifier la version de NodeJS en tapant ceci dans la console :

node -v

Si la version est trop ancienne, ou que l'on n'est pas sûr que c'est la dernière version, ou qu'il y a un soucis, la suppression est possibl avec cette commande :

sudo apt-get remove nodejs

2.2 Installation de NodeJS

Il faut vérifier quelle version de ARM possède le système avec cette commande : cat /proc/cpuinfo

Il faut utiliser le site *http://nodejs.org/dist/* pour installer la version souhaitée. Cidessous un exemple avec la dernière version disponible -v9.3.0- en version *linux-armv61*:

Il faut se placer sur la racine utilisateur (ATTENTION : ne pas installer en super utilisateur nommé Root).

cd ~

On télécharge le fichier souhaité avec cette commande : (wget + lien du fichier).

wget http://nodejs.org/dist/v9.3.0/node-v9.3.0-linux-armv6l.tar.gz

03/09/2018 PAGE **5** SUR **16**



Florian MOREAU

BTS SIO 1

Il faut ensuite le décompresser comme ceci : (tar xvzf + lien du fichier).

tar xvzf node-v9.3.0-linux-armv6l.tar.gz

Ensuite, pour pouvoir utiliser les commandes **node** et **npm** (ce sont les commandes de NodeJS) de partout on va modifier le fichier ~/.bashrc:

nano ~/.bashrc

À la fin du fichier rajouter la ligne :

export PATH=\$PATH:/home/pi/node-v9.3.0-linux-armv6l/bin

Puis enregistrer avec CTRL X, suivi par O (pour Oui) ou Y (pour Yes) selon la langue.

Enfin taper la commande ci-dessous pour prendre en compte la modification du fichier :

source ~/.bashrc

Maintenant, taper **node -v** devrait retourner v9.3.0. (ou votre nom de version choisis).

3 Les plugins

3.1 Création de leurs emplacements de base

Télécharger trois fichiers qu'il faut placer dans le répertoire ~/assistant-plugins/:

mkdir ~/assistant-plugins

cd ~/assistant-plugins

wget https://raw.githubusercontent.com/Aymkdn/assistant-plugins/master/docs/install/package.json wget https://raw.githubusercontent.com/Aymkdn/assistant-plugins/master/docs/install/index.js wget https://raw.githubusercontent.com/Aymkdn/assistant-plugins/master/docs/install/postinstall.js

03/09/2018 PAGE 6 SUR 16



Florian MOREAU

BTS SIO 1

ATTENTION: toutes les commandes suivantes sont à faire dans le dossier ~/assistant-plugins/ créé juste au-dessus.

3.2 Ajout des plugins souhaités

lci, il y a besoin de trois plugins : Freebox (pour avoir les commandes de la Freebox Player), notifier (pour que Pushbullet envoi la commande souhaitée) et iftt (pour que ifttt fassent le lien entre parole et commande). Voici les commandes :

npm install assistant-freebox --save

npm install assistant-notifier@latest --save

npm install assistant-ifttt@latest --save

Une fois tous les plugins souhaités installé, tapé la commande suivante :

npm install --loglevel error && npm run-script postinstall

3.3 Configuration

Se rendre dans le fichier de configuration dénommé configuration.json.

Nano configuration.json

Il faut désormais coller le code de WebHooks trouvé au début du tutoriel dans iftt après key entre les doubles quottes.

```
"plugins": {
   "iffH": {
      "key":"dph-Wyhz1Zxlw89WZchMAV"
   }
}
```

03/09/2018 PAGE **7** SUR **16**

L'adresse IP du Google Home se trouve sur l'application Google Home de votre téléphone et la collé au niveau d'host toujours entre doubles quottes.

```
"plugins": {
    "notifier": {
        "host":"192.168.0.13"
    }
}
```

Si vous avez plusieurs Google Home, il est possible de les configurer et cibler. Pour ce fait, le fichier de configuration doit ressembler à ça :

```
"plugins": {
    "notifier": {
        "host": {
            "salon":"192.168.0.13",
            "cuisine":"192.168.0.14",
            "chambre d'alexandre":"192.168.0.10",
            "chambre des parents":"192.168.0.11"
        }
    }
```

Enfin, au niveau de **pushbullet_token**, mettre entre doubles quottes le code secret trouvé sur **Pushbullet** au début du tutoriel.

03/09/2018 PAGE **8** SUR **16**



Florian MOREAU

BTS SIO 1

3.4 Premier lancement

Il suffit de taper la commande **node index.js** pour exécuter le programme. Il faudra surement aller appuyer sur la flèche de droite sur la Freebox afin de valider cette connexion. Votre Raspberry est prêt à recevoir des ordres de la Google Home. Il faut maintenant les créer sur ifttt.

4 Création d'un ordre

Se rendre sur le site https://ifttt.com, dans Applets puis New Applets.

Cliquer sur This:

if this then that

Trouver et sélectionner **Google Assistant**:

Choose a service

Step 1 of 6

Q google



03/09/2018 PAGE **9** SUR **16**



Florian MOREAU

BTS SIO 1

Sur l'onglet suivant, les trois premiers champs servent à écrire une phrase que l'on dira après le fameux « **OK Google** ». Cela sert à avoir trois versions de l'ordre. **Exemple : « éteins la Freebox », « arrête la Freebox », « Freebox Off ».**

Ensuite, il y a 4 solutions disponible:



Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

Say a phrase with a

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Set Nest thermostat to 68." **Use the # symbol to specify where you'll say the number ingredient

Say a phrase with a text ingredient

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Post a tweet saying 'New high score." **Use the \$ symbol to specify where you'll say the text ingredient

Say a phrase with both a number and a text ingredient

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Block time for 'exercise' at 6 PM."
**Use the # symbol to specify where you'll say the number ingredient and \$ where you'll say the text ingredient

« Say simple phrase » c'est un ordre qui ne contiendra pas de variable.

(Exemple : Eteindre la freebox).

« Say a phrase with a number » c'est un ordre contenant un numéro (Exemple : Baisse le son de 4). Il faut remplacer le numéro par un # dans l'ordre.

(Exemple: Baisse le son de #)

« Say a phrase with a text ingredient » c'est un ordre contenant une variable de type chaîne de caractère (Exemple: Ouvre mon dossier photo 2017) il faut remplacer la variable que l'on souhaite (Nom d'un dossier ou de chaîne) par un \$. (Exemple: « mets la chaîne \$ », « Ouvre mon dossier \$ »)

La dernière, il faut que Google Home répondre.

Exemple: « D'accord, je zappe sur \$ », « Ok, j'éteins la Freebox ».

Veuillez ensuite sélectionner la langue dans laquelle doit parler et comprendre la Google Home. Puis cliquer sur **« Create Trigger ».**

03/09/2018 PAGE **10** SUR **16**



Florian MOREAU

BTS SIO 1

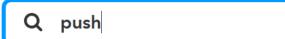
Sélectionner ensuite le bouton *That*.

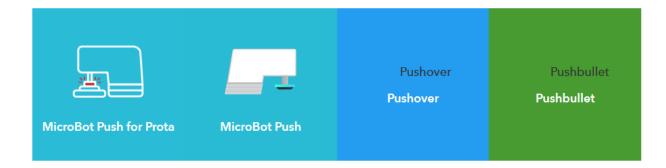


Puis sélectionner le service PushBullet.

Choose action service

Step 3 of 6



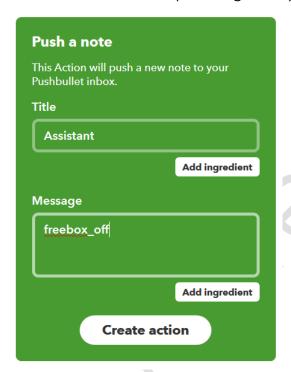


Prendre l'option **Push a note** :



03/09/2018 PAGE **11** SUR **16**

Ici, mettre dans *Title « Assistant »* puis taper la commande de votre choix dans le champ Message puis sélectionner « *Create action* ». Les commandes sont situées dans l'index de ce tutoriel. Puis faites un test (on obligatoire) et cliquer sur Finish.



Et voilà, votre voix peut maintenant contrôler votre Freebox Player!

03/09/2018 PAGE **12** SUR **16**



Florian MOREAU

BTS SIO 1

5 Annexe (commande)

red: envoie la commande red (touche rouge de la télécommande)

yellow: envoie la commande yellow (touche jaune de la télécommande)

blue : envoie la commande blue (touche bleue de la télécommande)

green: envoie la commande green (touche verte de la télécommande)

up : envoie la commande up (flèche haut)

down: envoie la commande down (flèche bas)

left: envoie la commande left (flèche gauche)

right: envoie la commande right (flèche droite)

OK: envoie la commande OK

mute: envoie la commande mute (sourdine)

play: envoie la commande play

fwd: envoie la commande fwd (avance rapide)

bwd: envoie la commande bwd (retour rapide)

waitXXXX: enclenche un timer de XXXX millisecondes

on: envoie la séquence power suivi d'un timer de 7 secondes (wait7000)

off: envoie la commande power

tv: envoie la séquence home, wait2000, red, ok, wait4000

unmute: envoie mute

home: envoie la séquence home, wait2000, red

back: envoie la commande red

pause: envoie la commande play

videos: envoie la séquence home, wait2000, red, right, ok

direct: envoie la séquence green, ok

03/09/2018 PAGE **13** SUR **16**



Florian MOREAU

BTS SIO 1

enregistrements: envoie la séquence home, wait2000, red, up, ok

soundDown: envoie la commande vol_dec

soundUp: envoie la commande vol_inc

programUp: envoie la commande prgm_inc

programDown : envoie la commande prgm_dec

folder XYZ: permet de chercher le répertoire XYZ dans search_path (qui est défini dans la configuration), puis de s'y rendre

Zappe sur ABC ou zappe sur la 123 : permet de zapper sur la chaine ABC ou sur la chaine dont le numéro est 123 (exemple : freebox_zappe sur la 1 ou freebox_zappe sur TF1)

On peut aussi utiliser *X pour effectuer X fois la même action

Exemple: freebox_soundUp*5 équivaut à :

freebox_soundUp,soundUp,soundUp,soundUp

6 Perte de connexion

Parfois, il arrive que la box redémarre ou que le réseau internet rencontre des soucis. Pour palier à ce problème, il va falloir mettre en place deux scripts en shell qui permettent de contrôler la connexion entre le Raspberry et la box ainsi que de redémarrer les services de notre serveur quand celui-ci retrouvera la connexion.

6.1 Les scripts

Le premier script, sert à **contrôler régulièrement** (ici toute les minutes) l'état de la connexion entre le Raspberry et la box. Il suffit de **créer un nouveau dossier dans Home** nommé script, et d'y **créer un nouveau fichier en .sh** (ici, pingTest.sh).

Voici les commandes :

Cd /home

Mkdir script

Nano script/pingTest.sh

03/09/2018 PAGE **14** SUR **16**

Une fois dans le fichier, recopier le code suivant (en adaptant les adresses IP si nécessaire)

```
#!/bin/bash

sleep 1
ping 192.168.0.1 -c5 -q

while [ $? != 1 ]

do
ping 192.168.0.1 -c5 -q
echo "Connexion Success"
sleep 60
done
echo "Connexion Lost "
./pingDead.sh
```

Le second script, ici nommé pingDead.sh, doit être créé dans le même dossier que le premier. Il testera la connexion quand elle sera perdue afin de redémarrer le service de notre serveur nodeJS que si le Raspberry lui-même trouve le réseau. Pour ce faire, taper la commande :

Nano script/pingDead.sh

Puis, recopier le code suivant :

```
#!/bin/bash

sleep 1
ping 192.18.0.1 -c5 -q

while [ $? == 1 ]

do
ping 192.168.0.1 -c5 -q
echo "Connexion Lost"
done
echo "Connexion Success "
pm2 restart index
sleep 10
./pingTest.sh
```

03/09/2018 PAGE **15** SUR **16**



Florian MOREAU

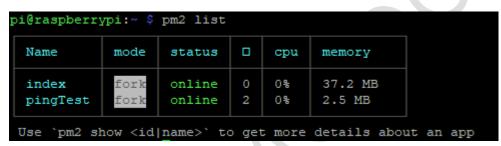
BTS SIO 1

6.2 Démarrage automatique

Au démarrage du Raspberry, il n'y a pour l'instant que NodeJS, grâce au service PM2, qui s'exécute. Il faut utiliser ce même service pour démarrer nos scripts. Il suffit de démarrer que le premier script, étant donné que le second n'est exécuté que si le premier rencontre une erreur dans le test de connexion.

Pour ce faire, il suffit de taper la commande **pm2 start script/pingTest.sh** (à adapter en fonction du nom du premier script).

Ensuite, **vérifier que les processus** de **NodeJS** (ici index) et du **script** (ici PingTest) sont bien lancé comme ci-dessous grâce à la commande **pm2 list**.



Pour finir, il suffit de d'entrer la commande **pm2 save** puis redémarrer la machine avec **sudo reboot**.

Avec la commande **pm2 monit**, il y aura maintenant **deux processus**, un qui donnera l'état de **nodeJS** (index), l'autre donnera **l'état du script** (pingTest).

```
0] index
           Mem:
                xx index > [assistant-freebox] Url => http://hdl.freeb
[ 1] pingTest
                 xx ox.fr/pub/remote control?code=27407982&key=power
                 xx index > [assistant-freebox] Commande « off »
                 xx exécutée
                 xx index > [assistant-freebox] Commande « on »
                 xx pingTest > PING 192.168.0.1 (192.168.0.1) 56(84)
                 xx bytes of data.
                  x pingTest > --- 192.168.0.1 ping statistics ---
                 xx pingTest > 5 packets transmitted, 5 received, 0%
                  x packet loss, time 4147ms
                 xx pingTest > rtt min/avg/max/mdev = 0.538/0.575/0.664
                 xx /0.054 ms
                 xx pingTest > Connexion Success
```

Le Freebox Player est désormais contrôlable par la box, avec un résolution de problème causé par une perte de connexion internet.

03/09/2018 PAGE **16** SUR **16**