

## Raspberry, Node-red, Mqtt et Freeboard

!! STATUT: BROUILLON !!

Je suis parti d'un kit Raspberry Pi 0. La carte seule est à 5€50 chez Kubii. Le starter Kit se trouve pour moins de 30 euros.



Pour la partie réseau soit vous achetez un adaptateur usb/RJ45, soit vous montez un dongle Wifi.

Le démarrage et le paramétrage de départ est assez classique. Je suis parti sur une image Raspbian en Stretch.

Pour un dongle Wifi il faut renseigner le fichier de conf wifi

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=FR

network={
ssid="VOTRE RESEAU"
psk="VOTRE MOT DE PASSE"
proto=RSN
key_mgmt=WPA-PSK
pairwise=CCMP TKIP
group=CCMP TKIP
}
```

Ensuite on met à jour le tout:

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
apt-get autoremove
apt-get clean
```

Il faut ensuite installer homeassistant.

```
sudo python3 -m pip install homeassistant
```

Si vous partez comme moi pour une solution GrovePi il faut:

### Installation de GrovePi

1. se placer dans /home/pi
2. récupérer le dépôt depuis dexter

```
git clone https://github.com/DexterInd/GrovePi.git
```

Se rendre dans le dossier /home/pi/Grovepi/Script et rendre le script install.sh executable

```
sudo chmod +x ./install.sh
```

Puis lancer ce script:

```
sudo ./install.sh
```

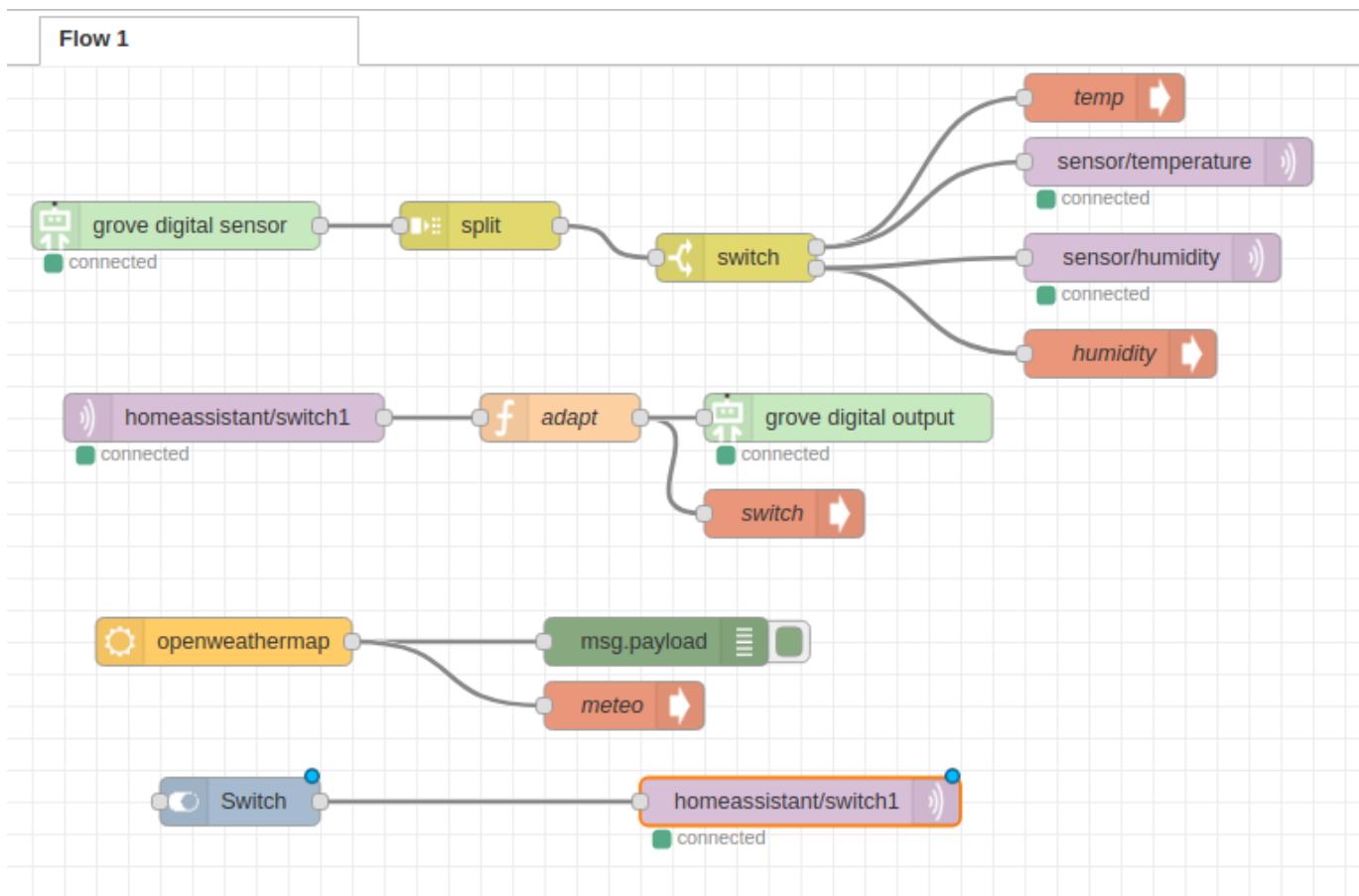
A la fin vous vous retrouvez avec Arduino installé et le Raspberry prêt à prendre la main sur le GrovePi+.

Vous pouvez donc soit programmer directement avec Arduino et téléverser avec le programmeur Raspberry GPIO soit installer le firmware depuis

```
cd /home/pi/Grovepi/Firmware  
./firmware_update.sh
```

Répondre "y" et appuyer sur une touche.

## Installation des modules node-red



Il faut maintenant lancer une fois node-red

```
node - red
```

Puis se placer dans le répertoire node-red de pi:

```
cd /home/pi/.node-red
```

Installer ensuite node-red-contrib-grovepi

```
npm install node-red-contrib-grovepi
```

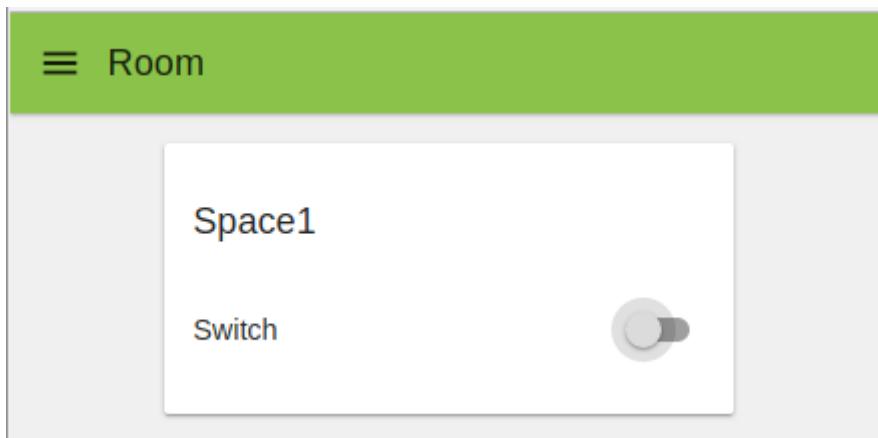
Nous allons en profiter pour installer également Freeboard qui permet la création de tableau de bord pour nos objets connectés. Node-red et freeboard fonctionnent ensemble à merveille.



```
cd ~/.node-red
git clone https://github.com/Freeboard/freeboard.git
npm install freeboard
git clone https://github.com/urbiworx/node-red-contrib-freeboard.git
npm install node-red-contrib-freeboard
```

Pour pouvoir créer des interfaces sur node-red pour nos objets connectés, installons node-red-contrib-ui

```
npm install node-red-contrib-ui
```



Pour récupérer les données météo installez node-red-node-openweathermap

```
npm install node-red-node-openweathermap
```

Il faudra se créer un compte sur <https://openweathermap.org/appid> et récupérer la clé.

### Installation de mqtt

Pour pouvoir autoriser les Websockets pour Mqtt (utile pour récupérer les trames mqtt sous Freeboard) il faut suivre scrupuleusement le tutoriel suivant  
<http://ouilogique.com/mosquitto-mqtt-raspberry/>

### Ajout de plugin pour freeboard

Paho Mqtt:

```
cd /home/pi
git clone https://github.com/alsm/freeboard-mqtt
```

Créer un dossier mqtt dans

```
cd ~/.node-red/node-red-contrib-freeboard/node_modules/freeboard/plugins/
mkdir mqtt
cd mqtt
```

Copier les fichiers js dans mqtt

```
cp /home/pi/freeboard-mqtt/mqtts31.js ~/.node-red/node-red-contrib-
freeboard/node_modules/freeboard/plugins/
cp /home/pi/freeboard-mqtt/paho.mqtt.plugin.js ~/.node-red/node-red-contrib-
freeboard/node_modules/freeboard/plugins/
```

Editer le fichier paho.mqtt.plugin.js et vérifier la section suivante:

```
"external_scripts" : [
    "plugins/mqtt/mqtts31.js"
],
```

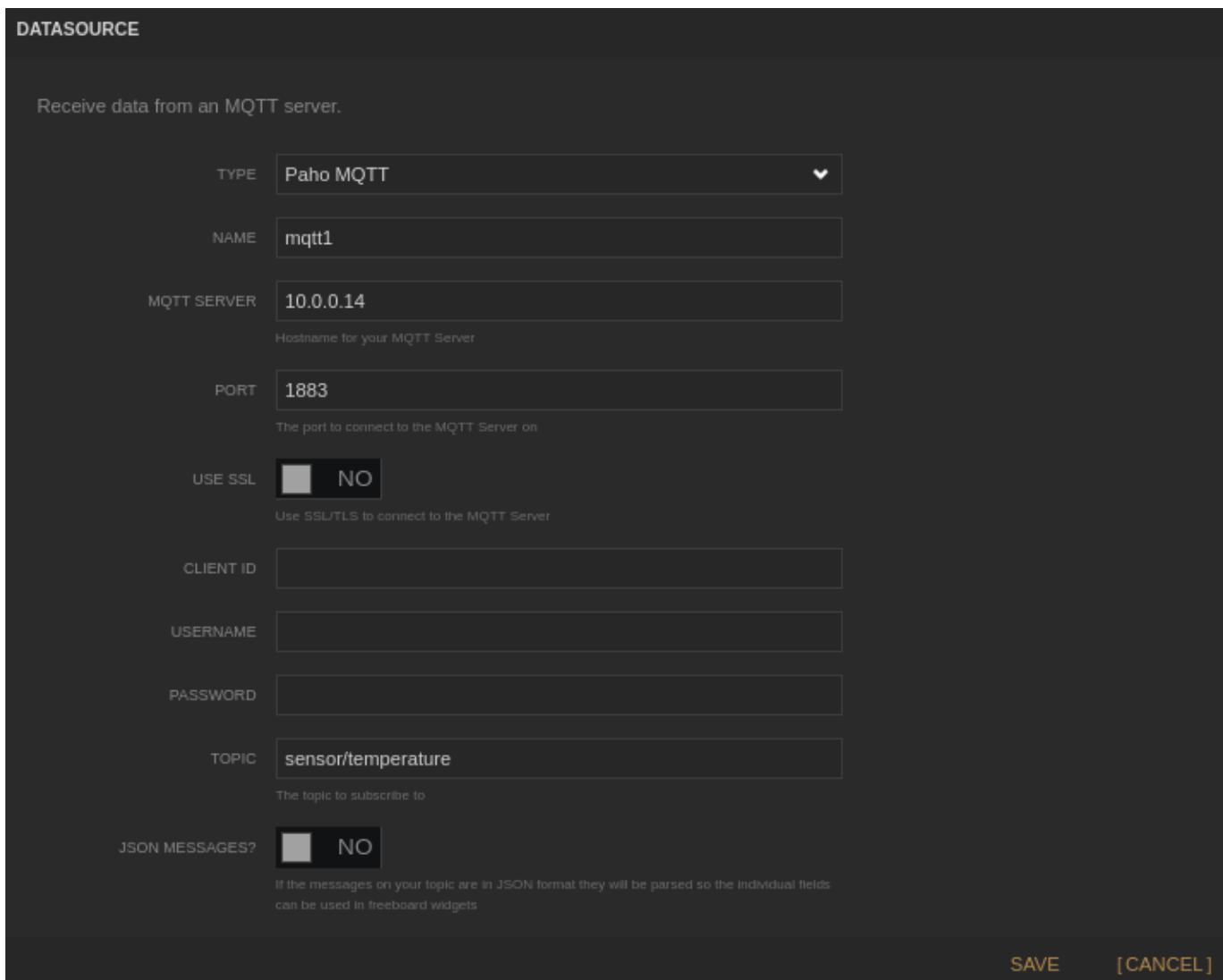
Editer le fichier index.html

```
nano ~/node-red/node-red-contrib-freeboard/node_modules/freeboard/index.html
```

Et près de head.js ajouter le plugin paho.mqtt.plugin.js

```
head.js("js/freeboard.js",
    "plugins/mqtt/paho.mqtt.plugin.js",
```

Sous Freeboard on peut ajouter maintenant un datasource de type “paho mqtt”:



Bon on peut transmettre des données à Freeboard mais comment actionner des équipements à partir de Freeboard ? Il existe un plugin nommé switch.js qui permet de faire le travail demandé. Souci il faudra construire des scripts php pour communiquer avec mqtt, pas de soucis c'est assez simple.

```
cd /home/pi
git clone https://github.com/onlinux/freeboard-plugins
cd freeboard-plugins
cp *.js ~/node-red/node-red-contrib-freeboard/node_modules/freeboard/plugins/
```

Last update:

2020/09/26 7\_raspberry\_node-red\_et\_freeboard https://mistert.freeboxos.fr/dokuwiki/doku.php?id=7\_raspberry\_node-red\_et\_freeboard  
15:15

```
cp *.png ~/.node-red/node-red-contrib-freeboard/node_modules/freeboard/img/
```

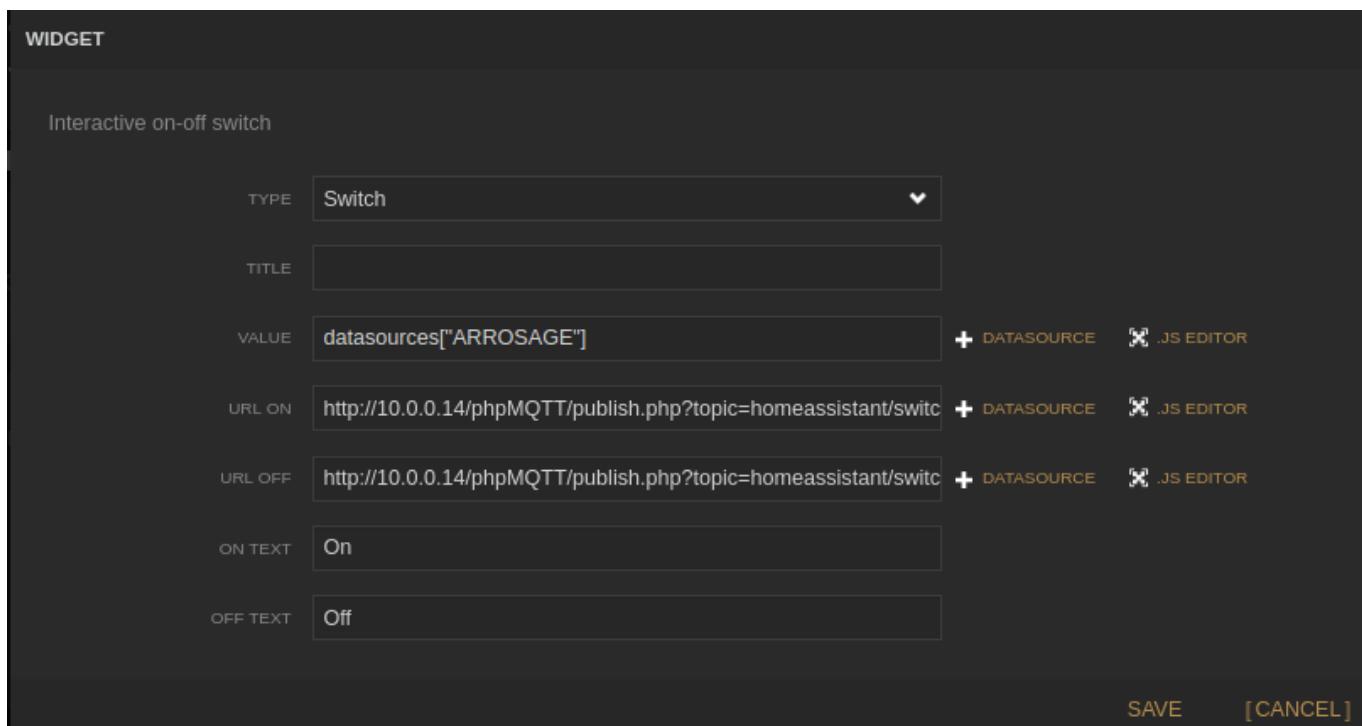
On modifie le fichier index.html en ajoutant les plugins dans la section head.js

```
nano ~/.node-red/node-red-contrib-freeboard/node_modules/freeboard/index.html
head.js("js/freeboard.js",
    "plugins/mqtt/paho.mqtt.plugin.js",
    'plugins/switch.js', <-- AJOUTER CECI
```

Se rendre sur freeboard:

```
http://10.0.0.14:1880/freeboard/
```

On peut ajouter un switch ...



Vous noterez la présence des deux scripts qui vont gérer les clics:

```
http://10.0.0.14/phpMQTT/publish.php?topic=homeassistant/switch1&msg=ON1
http://10.0.0.14/phpMQTT/publish.php?topic=homeassistant/switch1&msg=OFF1
```

Il faut maintenant installer un serveur Web sur le raspberry avec Apache et Php

```
sudo apt install apache2
sudo apt install php php-mbstring
```

Dans /var/www/html récupérer le git phpMQTT

```
cd /var/www/html
```

```

git clone https://github.com/bluerhinos/phpMQTT
cd phpMQTT
sudo nano publish.php

<?php
header('Access-Control-Allow-Origin: *');

if ($_GET) {
    extract($_GET);
    // $topic
    // $msg
}

require("./phpMQTT.php");

$server = "localhost";      // change if necessary
$port = 9001;                // change if necessary
$username = "";              // set your username
$password = "";              // set your password
$client_id = "phpMQTT-publisher"; // make sure this is unique for connecting
to sever - you could use
uniqid()

$mqtt = new phpMQTT($server, $port, $client_id);

if ($mqtt->connect(true, NULL, $username, $password)) {
    $mqtt->publish("$topic", "$msg", 0);
    $mqtt->close();
    echo("Success!\n");
} else {
    echo "Time out!\n";
}

```

On fixe les bons droits apache

```

sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/

```

La boucle est bouclée, node-red peut maintenant contrôler l'ensemble.

## Avantages

1. Ce n'est pas cher (la wio link est entre 15 et 20 euro et dispose de peu d'E/S)
2. On a la main sur l'ensemble du système sans dépendre de services externes (protection des données, exit cayenne, freeboard.io, etc ...)
3. C'est évolutif et libre
4. Pas de programmation lourde en python ou en C arduino mais possibilité d'y recourir au besoin
5. Possibilité de proposer une SDCARD ad hoc prête à l'emploi

## Inconvénients

Last update:

2020/09/26 7\_raspberry\_node-red\_et\_freeboard https://mistert.freeboxos.fr/dokuwiki/doku.php?id=7\_raspberry\_node-red\_et\_freeboard  
15:15

1. Faut un peu bosser :)
2. Demande une maîtrise minimale d'html et nodejs et debian

From:

<https://mistert.freeboxos.fr/dokuwiki/> - **Wiki de Sébastien TACK**



Permanent link:

[https://mistert.freeboxos.fr/dokuwiki/doku.php?id=7\\_raspberry\\_node-red\\_et\\_freeboard](https://mistert.freeboxos.fr/dokuwiki/doku.php?id=7_raspberry_node-red_et_freeboard)

Last update: **2020/09/26 15:15**