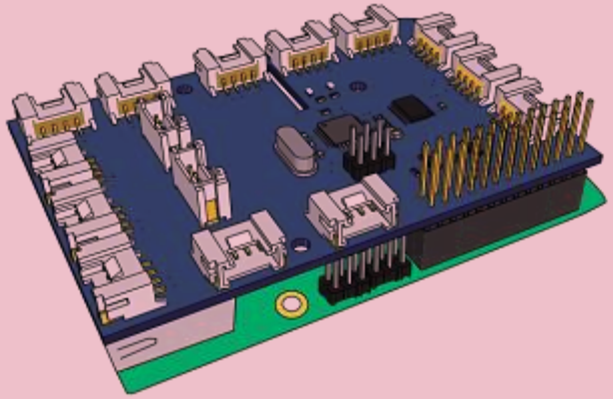
Plateforme Node-RED OCS

Déploiement Node-red sur raspberry pi + grovepi

Node-RED



Mardi 9 Octobre 2018

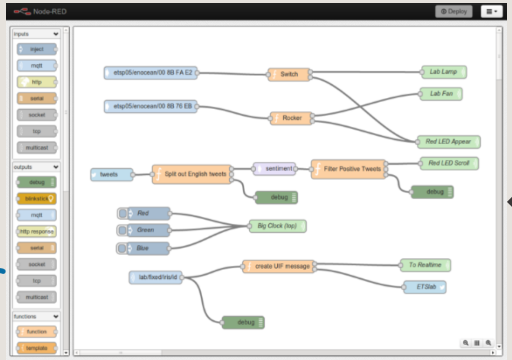
Intelligence Ambiante

OBJETS CONNECTES & SERVICES

Idée générale

Node-RED

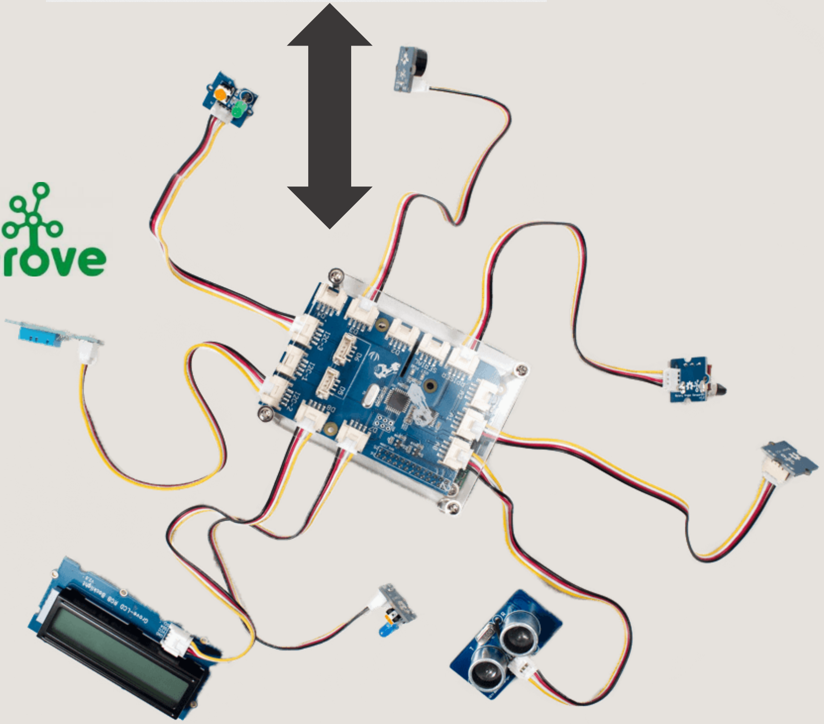
docker



API REST

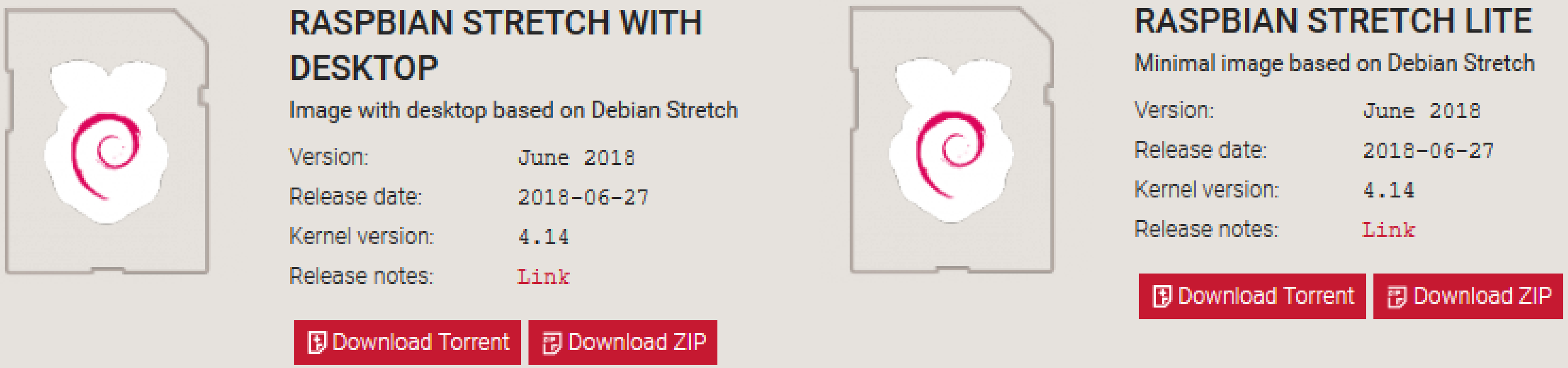
« Intelligence »

+ Grove



Préparation de la carte Raspberry PI

1. **Récupérer l'image RASPBIAN sur https://www.raspberrypi.org/downloads/raspbian/**



RASPBIAN STRETCH WITH DESKTOP

Image with desktop based on Debian Stretch

Version: June 2018
Release date: 2018-06-27
Kernel version: 4.14
Release notes: Link

RASPBIAN STRETCH LITE

Minimal image based on Debian Stretch

Version: June 2018
Release date: 2018-06-27
Kernel version: 4.14
Release notes: Link

2. **Attention! SSH est désactivé par défaut.**

-> Sur la partition BOOT créer le ficher 'ssh' (vide)
-> Reboot

Connection directe avec le PC par cable ethernet

1. **Installer ‘bonjour’**
    1. Télécharger Itunes
    2. Ouvrir l’archive et exécuter Bonjour64.msi

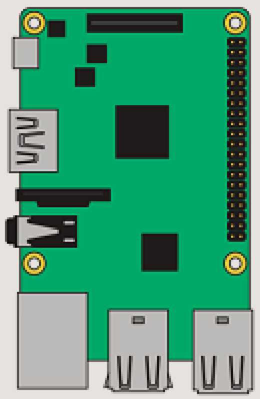
    Disponible également ici: http://trolen.polytech.unice.fr/cours/ocs/td1

2. **Connection avec le raspberrypi**

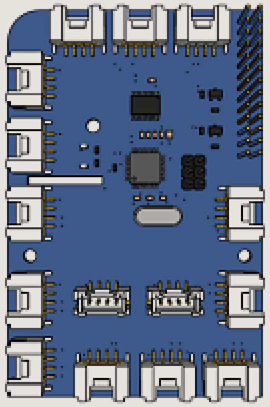
```
$ ssh pi@raspberrypi.local
```

Installation de la suite logicielle grovepi+

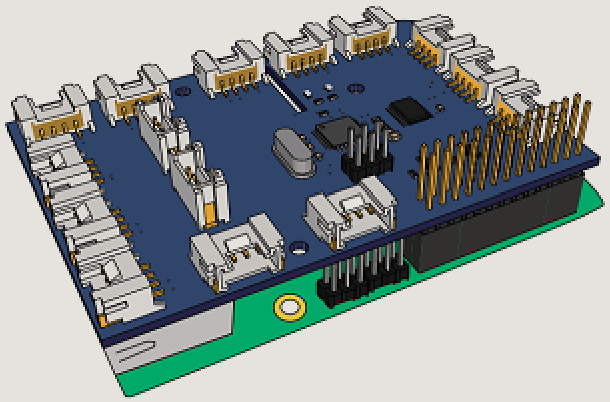
```
$ sudo curl -kL dexterindustries.com/update_grovepi | bash
$ sudo pip install grovepi
$ sudo reboot
```



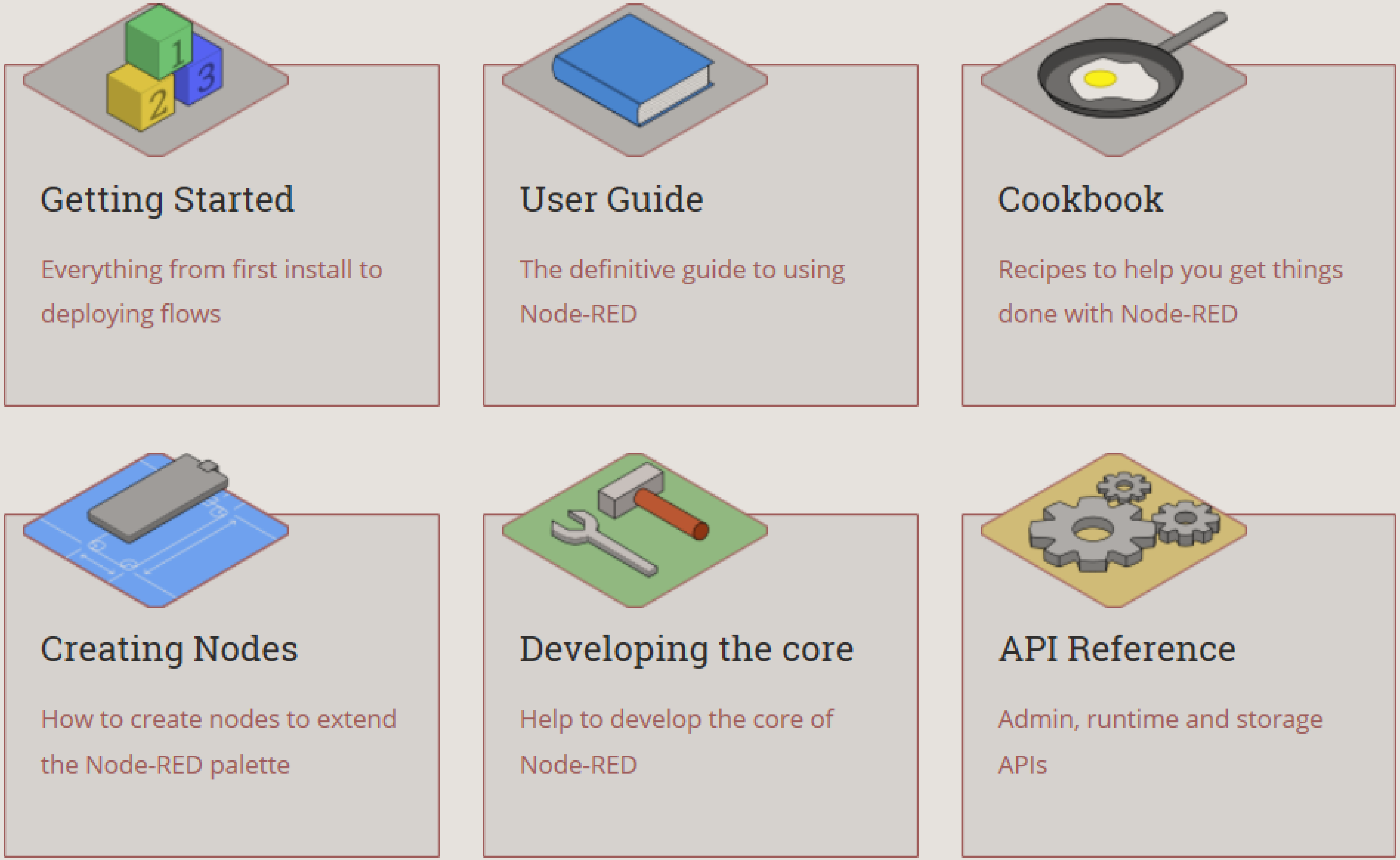
Raspberry Pi



GrovePi+



Documentation node-RED https://nodered.org/docs/



Getting Started

Everything from first install to deploying flows

User Guide

The definitive guide to using Node-RED

Cookbook

Recipes to help you get things done with Node-RED

Creating Nodes

How to create nodes to extend the Node-RED palette

Developing the core

Help to develop the core of Node-RED

API Reference

Admin, runtime and storage APIs

Installation Node-RED

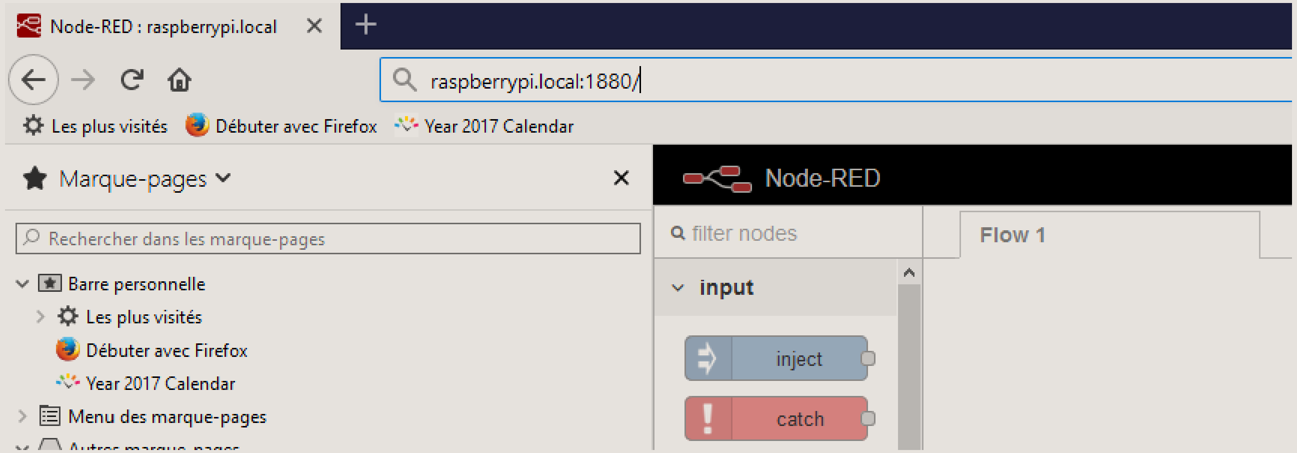
1. Informations sur https://nodered.org/docs/hardware/raspberrypi

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

2. Auto-start on boot

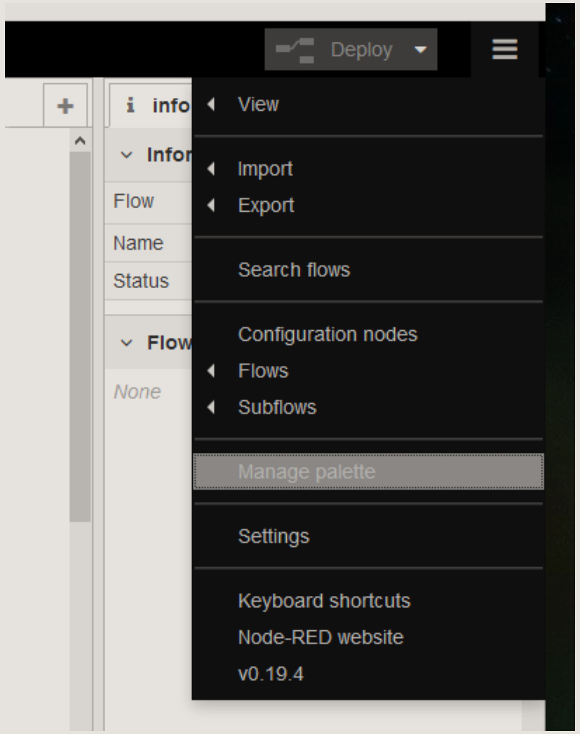
```
$ sudo systemctl enable nodered.service
```

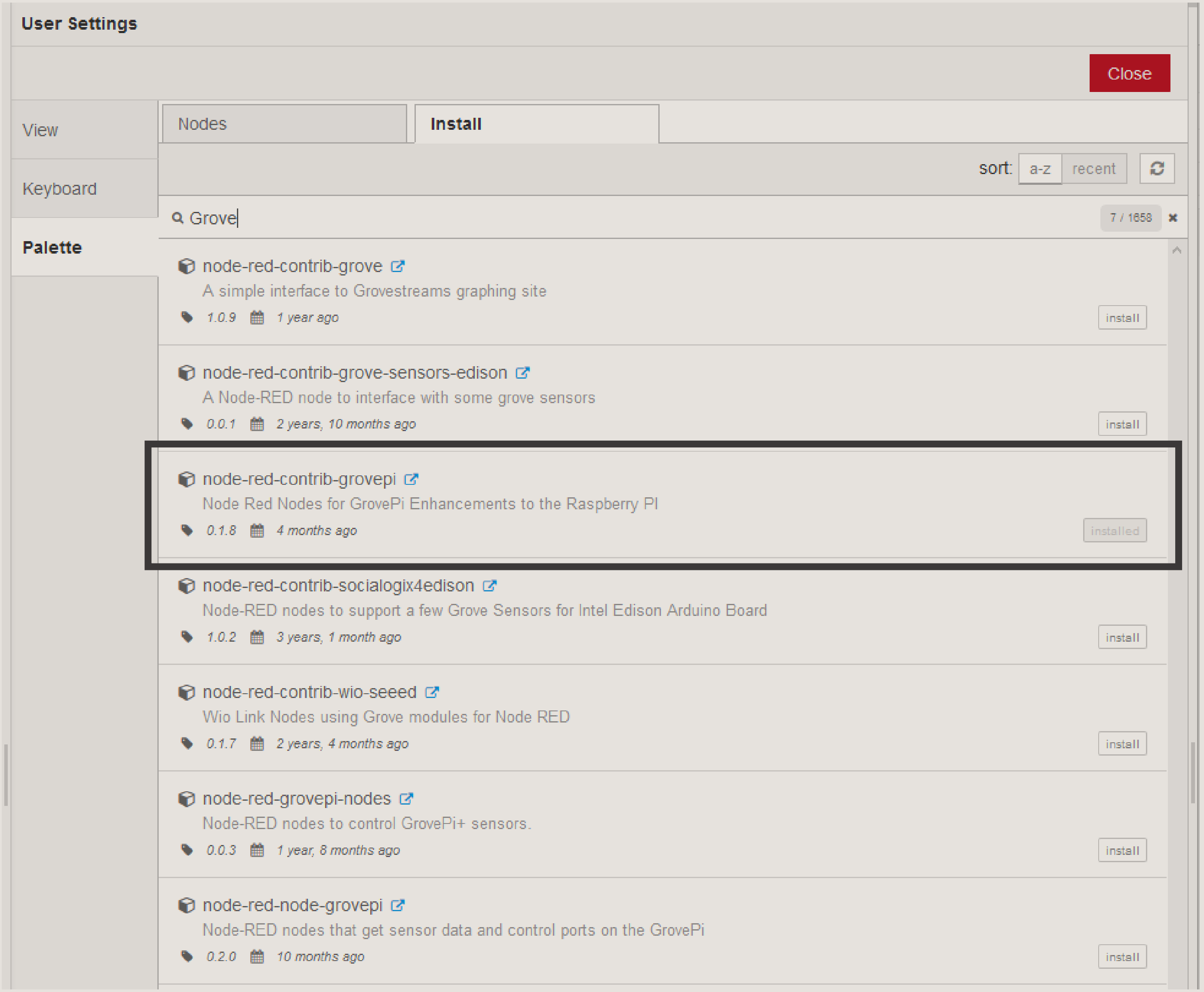
3. Test



raspberrypi.local:1880/

Installation des nodes grovepi+





Grove

node-red-contrib-grove
A simple interface to Grovestreams graphing site
1.0.9 1 year ago install

node-red-contrib-grove-sensors-edison
A Node-RED node to interface with some grove sensors
0.0.1 2 years, 10 months ago install

node-red-contrib-grovepi
Node Red Nodes for GrovePi Enhancements to the Raspberry PI
0.1.8 4 months ago installed

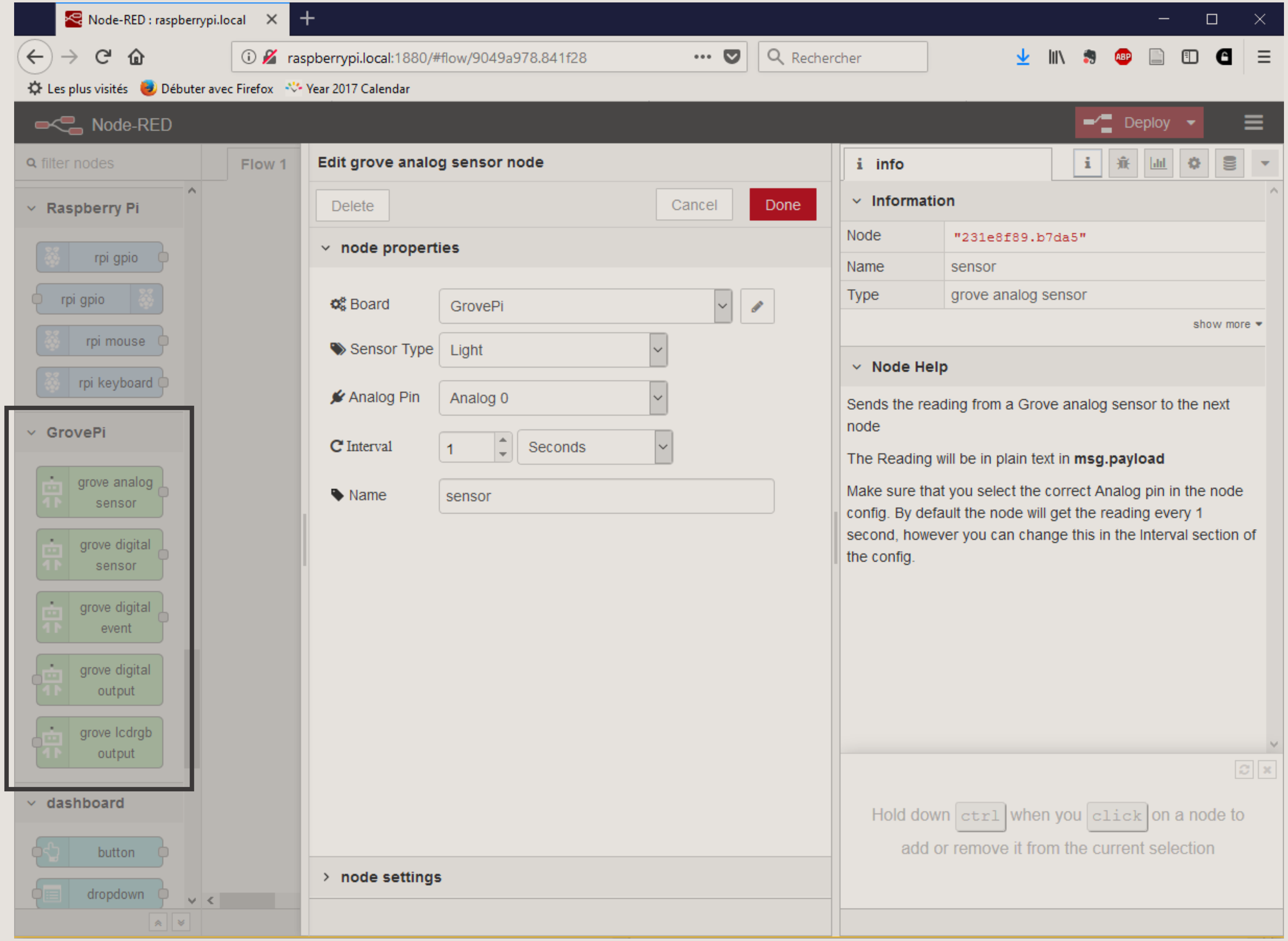
node-red-contrib-socialogix4edison
Node-RED nodes to support a few Grove Sensors for Intel Edison Arduino Board
1.0.2 3 years, 1 month ago install

node-red-contrib-wio-seeed
Wio Link Nodes using Grove modules for Node RED
0.1.7 2 years, 4 months ago install

node-red-grovepi-nodes
Node-RED nodes to control GrovePi+ sensors.
0.0.3 1 year, 8 months ago install

node-red-node-grovepi
Node-RED nodes that get sensor data and control ports on the GrovePi
0.2.0 10 months ago install

Accéder aux i/o du groovepi+ (1/2)



Edit grove analog sensor node

- Board: GrovePi
- Sensor Type: Light
- Analog Pin: Analog 0
- Interval: 1 Seconds
- Name: sensor

Node	"231e8f89.b7da5"
Name	sensor
Type	grove analog sensor

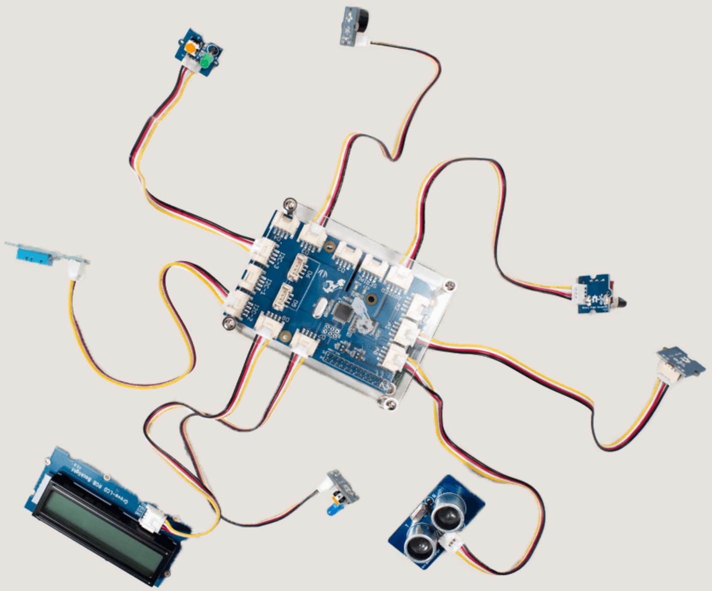
Sends the reading from a Grove analog sensor to the next node

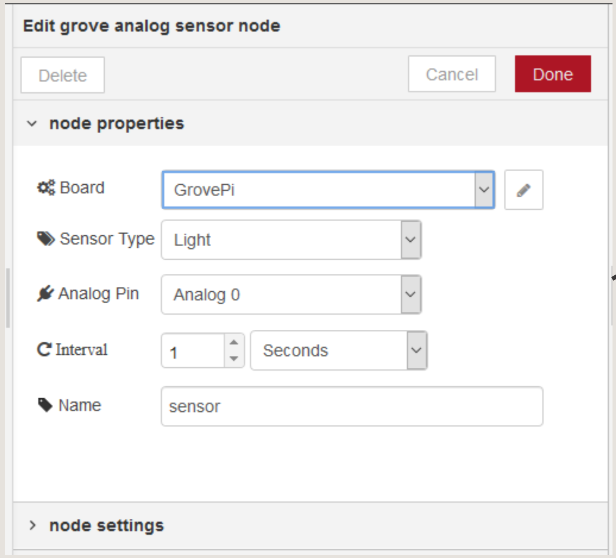
The Reading will be in plain text in **msg.payload**

Make sure that you select the correct Analog pin in the node config. By default the node will get the reading every 1 second, however you can change this in the Interval section of the config.

Hold down ctrl when you click on a node to add or remove it from the current selection

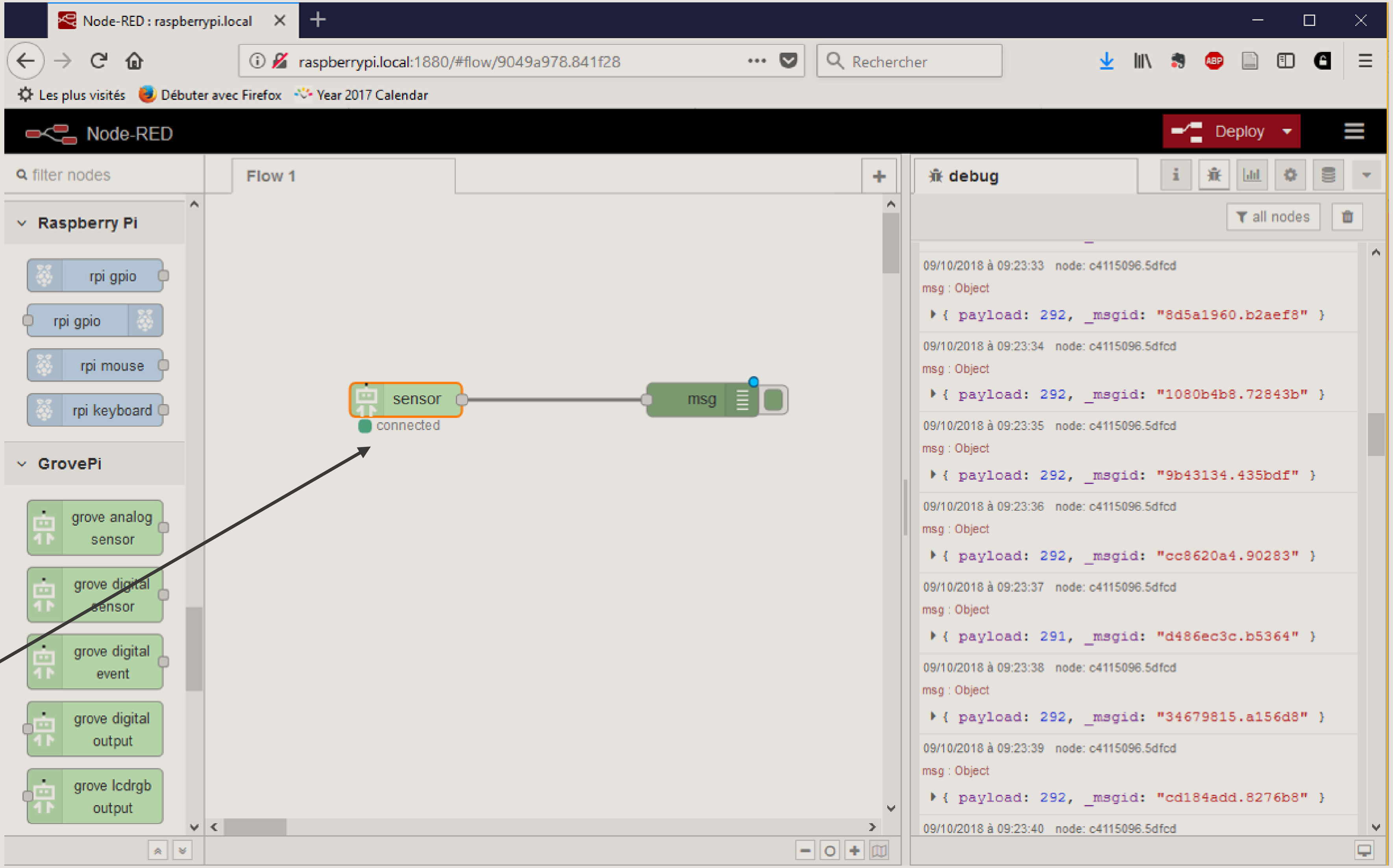
Accéder aux i/o du groovepi+ (2/2)





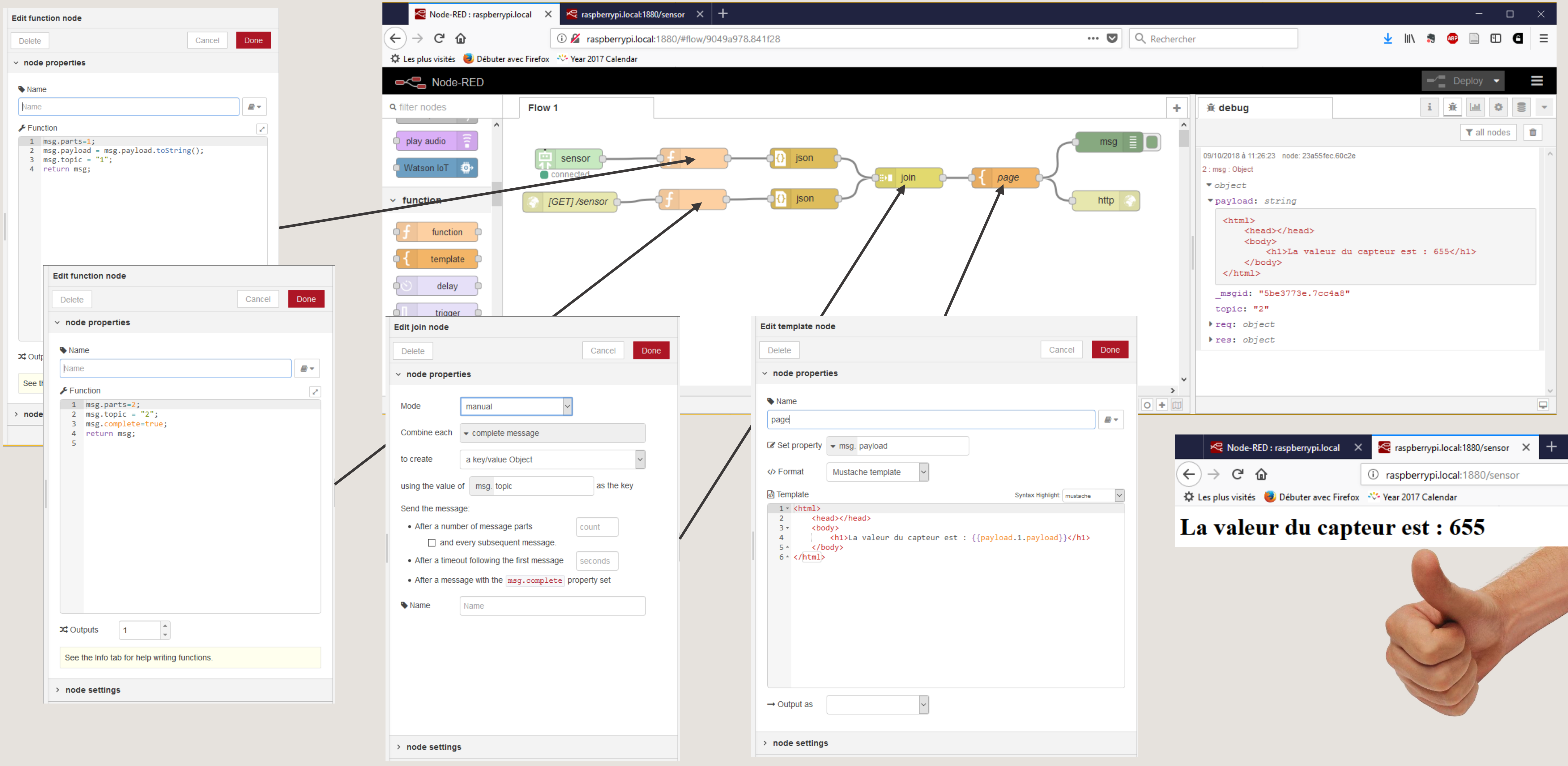
Edit grove analog sensor node

- Board: GrovePi
- Sensor Type: Light
- Analog Pin: Analog 0
- Interval: 1 Seconds
- Name: sensor



```
09/10/2018 à 09:23:33  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "8d5a1960.b2aef8" }
09/10/2018 à 09:23:34  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "1080b4b8.72843b" }
09/10/2018 à 09:23:35  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "9b43134.435bdf" }
09/10/2018 à 09:23:36  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "cc8620a4.90283" }
09/10/2018 à 09:23:37  node: c4115096.5dfcd
msg : Object
▸ { payload: 291, _msgid: "d486ec3c.b5364" }
09/10/2018 à 09:23:38  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "34679815.a156d8" }
09/10/2018 à 09:23:39  node: c4115096.5dfcd
msg : Object
▸ { payload: 292, _msgid: "cd184add.8276b8" }
09/10/2018 à 09:23:40  node: c4115096.5dfcd
```

Construire une API REST (partie 1 : lire les données capteur)



Edit function node

Delete Cancel Done

node properties

Name: Name

Function

```
msg.parts=1;
msg.payload = msg.payload.toString();
msg.topic = "1";
return msg;
```

Edit function node

Delete Cancel Done

node properties

Name: Name

Function

```
msg.parts=2;
msg.topic = "2";
msg.complete=true;
return msg;
```

Outputs: 1

See the Info tab for help writing functions.

node settings

Edit join node

Delete Cancel Done

node properties

Mode: manual

Combine each: complete message

to create: a key/value Object

using the value of msg. topic as the key

Send the message:

- After a number of message parts: count
  - □ and every subsequent message.
- After a timeout following the first message: seconds
- After a message with the `msg.complete` property set

Name: Name

node settings

Edit template node

Delete Cancel Done

node properties

Name: page

Set property: msg. payload

Format: Mustache template

Template — Syntax Highlight: mustache

```
<html>
    <head></head>
    <body>
        <h1>La valeur du capteur est : {{payload.1.payload}}</h1>
    </body>
</html>
```

Output as

node settings

debug

09/10/2018 à 11:26:23 node: 23a55fec.60c2e

2 : msg : Object

```
▾object
  ▾payload: string
      <html>
          <head></head>
          <body>
              <h1>La valeur du capteur est : 655</h1>
          </body>
      </html>
   _msgid: "5be3773e.7cc4a8"
   topic: "2"
  ▸req: object
  ▸res: object
```

raspberrypi.local:1880/sensor

La valeur du capteur est : 655

Construire une API REST (partie 2 : contrôler des actionneurs)

A vous d’essayer!

Développement de nodes

Essayer de suivre le tutorial : https://nodered.org/docs/creating-nodes/

Pour aller plus loin: Docker

Installer docker sur le raspberry pi

```
$ curl -fsSL get.docker.com -o get-docker.sh && sh get-docker.sh
```

Deployer une image node-red + Grovepi+

1. créer le directory /root/mynodered
2.
```
docker run --name nodered --privileged -v /root/mynodered:/root/node-red -p 1880:1880 -d ericbenoit/rpi-nodered-mini
```