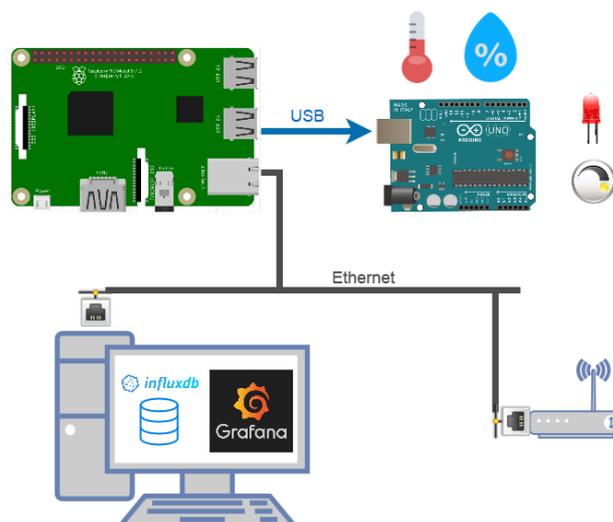


TP : Développement d'un objet connecté wifi/Ethernet



1 Introduction

Les objets connectés ou l'Internet des Objets (IoT : Internet of Things) sont des termes très utilisés de nos jours même si bien des personnes ont encore du mal les définir.

Au sens large, les objets connectés sont des objets communicants. Ils sont à la fois des émetteurs, des capteurs d'informations qui émettent, mais également qui peuvent interagir avec d'autres machines (M2M) comme des serveurs ou d'autres objets connectés. Les données générées par ces objets sont capitalisées dans des centres de données (data centers) et de nouveaux usages de cette donnée apparaissent autour de la santé, du sport, des produits financiers par exemple.

Les objets connectés connaissent une croissance exponentielle. Des estimations indiquent qu'ils seraient près de 15 milliards en circulation dans le monde actuel. Ils pourraient être plus de 50 milliards d'ici 2020.

Pour connecter les objets, les technologies actuelles peuvent être utilisées (RFID, Wi-Fi, GSM, Bluetooth, Z-Wave, ZigBee, ...) mais de nouveaux réseaux se développent au niveau mondial. Ils doivent permettre d'envoyer (mais aussi recevoir dans certains cas) des très petits messages sur des longues portées sans passer par des systèmes coûteux de réseaux mobiles et en consommant peu d'énergie.

Au cours de ce TP, vous allez mettre en œuvre un objet connecté directement au réseau local. Il s'agira d'une régulation de température construite autour d'une carte Arduino. L'IHM de supervision et de contrôle sera réalisé au moyen de Node Red sur un ordinateur embarqué Raspberry Pi et l'historisation des données sera assurée par un serveur de bases de données InfluxDB installé sur un ordinateur PC de la salle. On accèdera à l'historique des mesures un dashboard construit à partir de Grafana.

2 Evaluation

L'évaluation portera sur :

- Un compte rendu détaillé explicitant les différentes opérations menées durant le TP.
Pensez à prendre des copies d'écran pour illustrer ces opérations.
- L'avancement dans la réalisation de l'objet connecté et déploiement de la solution.

3 Arduino

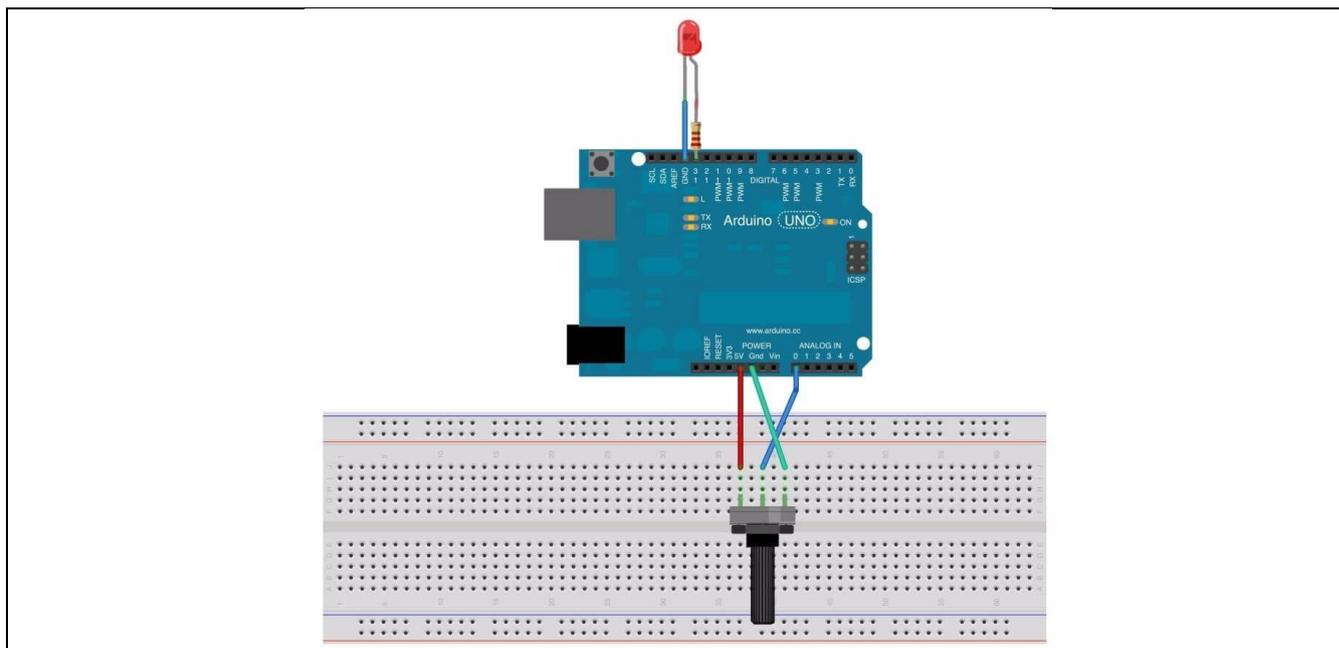
3.1 Installation de l'environnement de développement

Sur votre PC, bootez sur Ubuntu et téléchargez l'environnement de développement Arduino. Décompactez-le dans le dossier **Documents** et exécutez-le :

```
cd arduino-1.8.8
./arduino
```

3.2 Le montage

Le but est de piloter une LED en TOR (Tout ou Rien) et de lire l'état d'un potentiomètre.



Pour réaliser ce montage vous disposez du matériel suivant :

- Une carte Arduino Leonardo
- Un shield de connectique Grove
- Un module potentiomètre rotatif Grove
- Un module LED Grove
- Divers câbles de connexion.

Réalisez le montage et faites valider par l'enseignant.

3.3 Le programme

A partir des programmes d'exemple (*File -> Examples -> 1-Basics*) **Blink** et **AnalogReadSerial**, réalisez un programme qui permet d'allumer la LED lorsque le potentiomètre dépasse la moitié de sa course.

Faites une démonstration du fonctionnement à l'enseignant.

3.4 Mesure de la température et de l'humidité relative

Vous disposez d'un module grove capteur de température DHT11. Installez ci-nécessaire la librairie DHT Sensor Library v1.2.3 :

Sketch -> Include Library -> Manage Libraries -> Filter your search

Ouvrez le programme d'exemple DHTtester. Analysez le programme et connecté le capteur en conséquence. Définissez le type de capteur que vous utilisé (DHT11).

A partir des programmes précédents, réalisez un programme qui permet d'acquérir une consigne de température via le potentiomètre et allumera la LED seulement si la température mesurer est inférieure à la consigne. La température, l'humidité relative, la valeur de la consigne et l'état de la LED sont à retourner vers le terminal via la liaison Serial toute les 2 secondes selon le protocole suivant :

`<Temperature>;<Humidité>;<Consigne>;<EtatLED>\n`

Faites une démonstration du fonctionnement à l'enseignant.

4 IHM de supervision et contrôle

4.1 Node Red



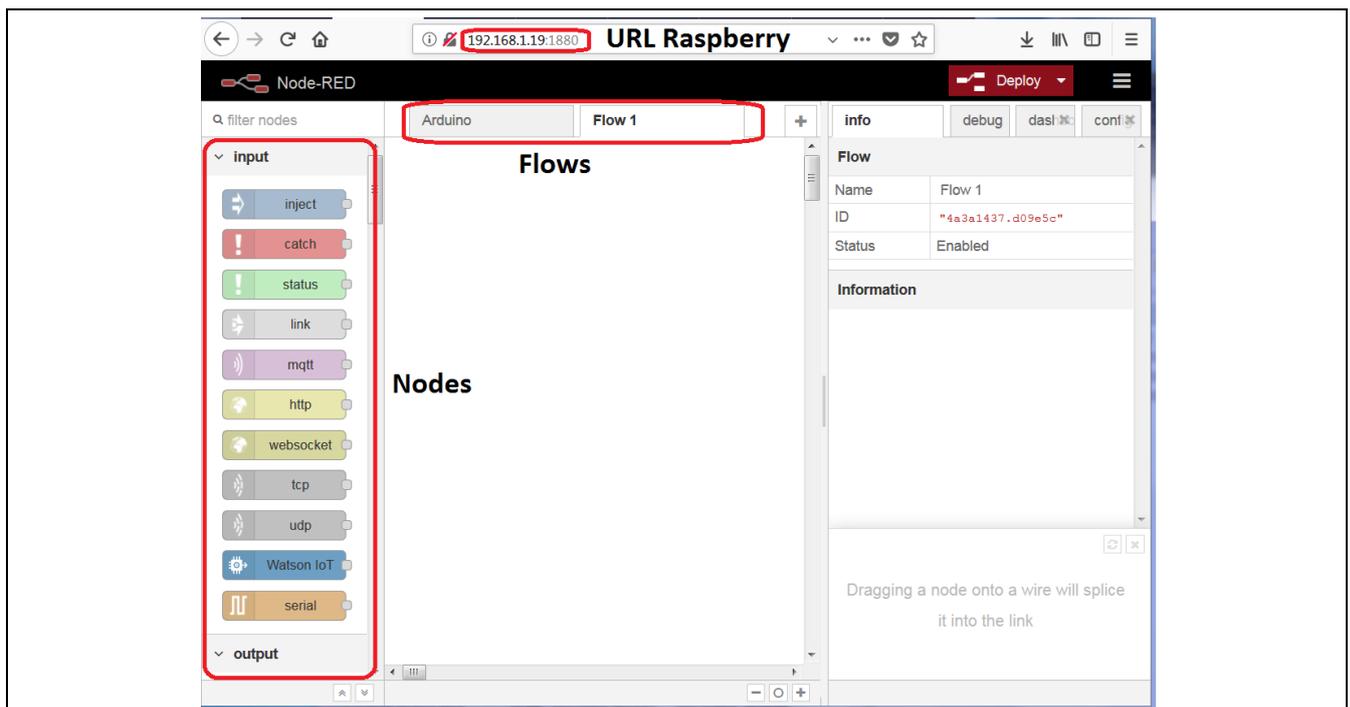
Node-RED est un outil puissant pour construire des applications de l'Internet des Objets (IoT) en mettant l'accent sur la simplification de la programmation qui se fait grâce à des blocs de code prédéfinis, appelés «nodes» pour effectuer des tâches. Il utilise une approche de programmation visuelle qui permet aux développeurs de connecter les blocs de code ensemble. Les nœuds connectés, généralement une combinaison de nœuds d'entrée, de nœuds de traitement et de nœuds de sortie, lorsqu'ils sont câblés ensemble, constituent un «flow».

Sur un Raspberry Pi qui tourne sur Raspbian, Node Red est déjà installé. Pour le lancer, il suffit de cliquer sur le menu **Programmation** puis **Node Red**.

Node Red peut également être lancé ou stopper à partir d'un terminal :

```
$ node-red-start
$ node-red-stop
```

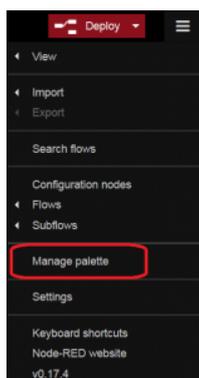
L'accès à l'interface de programmation se fait à partir d'un navigateur. Il est préférable d'utiliser le navigateur d'un ordinateur connecté sur le même réseau que le Raspberry.



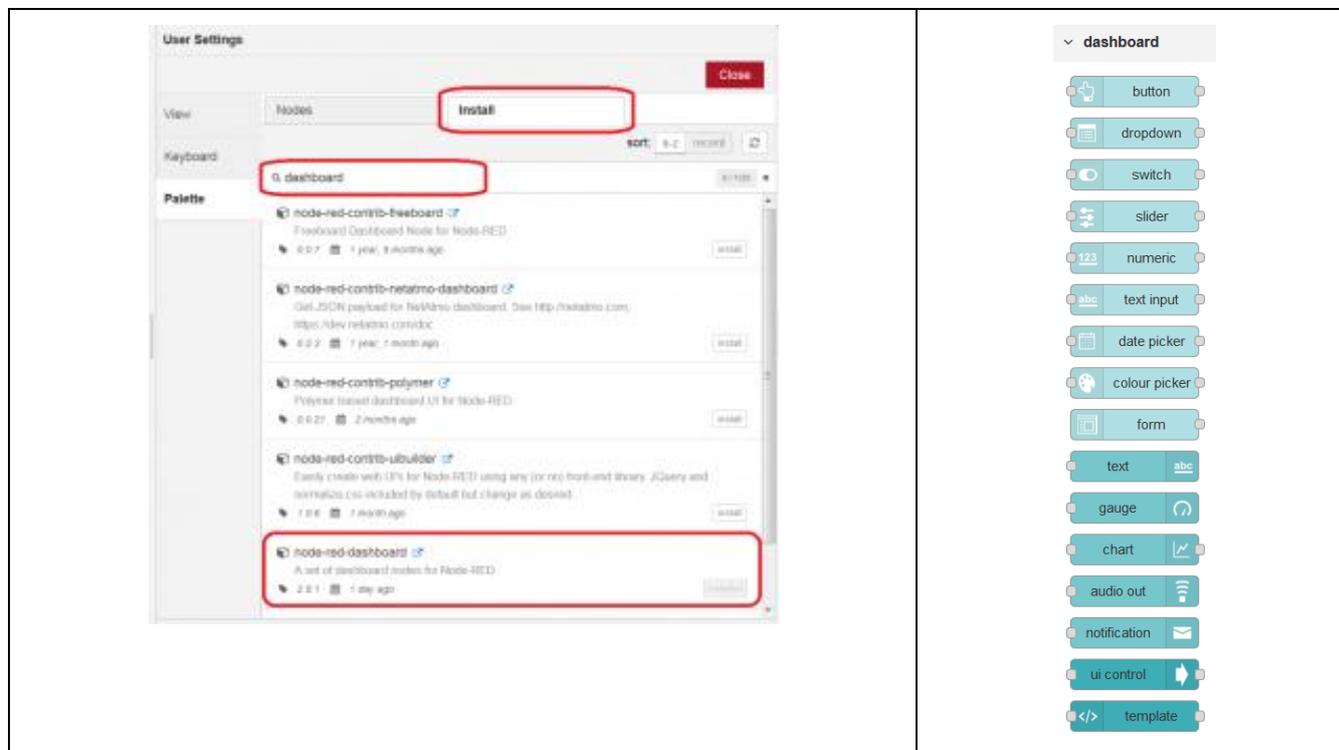
La palette de Nodes d'origine est déjà bien fournie mais il en manque au moins une essentielle qui va permettre de réaliser l'interface web. Elle se nomme **Dashboard**. Il vous faudra peut être installer au préalable le gestionnaire de Nodes. L'environnement d'exécution est construit sur **Node.js**, le gestionnaire de Nodes utilisé par Node Red est donc naturellement npm :

```
$ node-red-stop
$ sudo apt-get update
$ sudo apt-get install npm
$ cd /home/pi/.node-red
$ npm rebuild
$ node-red-start
```

On dispose désormais du gestionnaire de nodes dans le menu :



Cliquez sur **Manage Palette** puis recherchez la palette de nodes à installer. Une nouvelle palette est ensuite disponible :



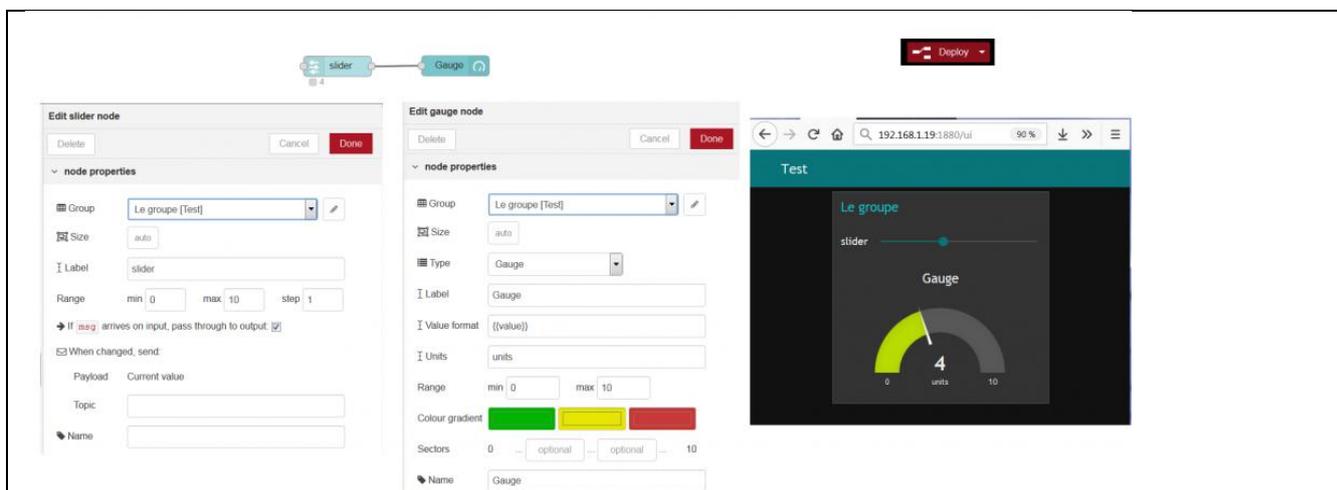
Faites constater à l'enseignant.

4.2 Premier flow

Commençons par réaliser un flow très simple composé d'un node **Slider** et d'un node **Gauge**. Pour modifier la configuration par défaut des nodes, il suffit de double-cliquer dessus.

Pour que ces composants soient bien disposés sur la page web, il faut les mettre dans un **Tab**, puis dans un **Groupe**, dans la boîte de sélection du groupe, cliquez sur **Add new ui_group**, puis créer un groupe (ici : "Le groupe") et faite de même pour le Tab (ici : "Test").

Cliquez ensuite sur le bouton **Done**, puis déployer le Flow. On accède au site web généré par l'url : http://ip_raspberry:1880/ui

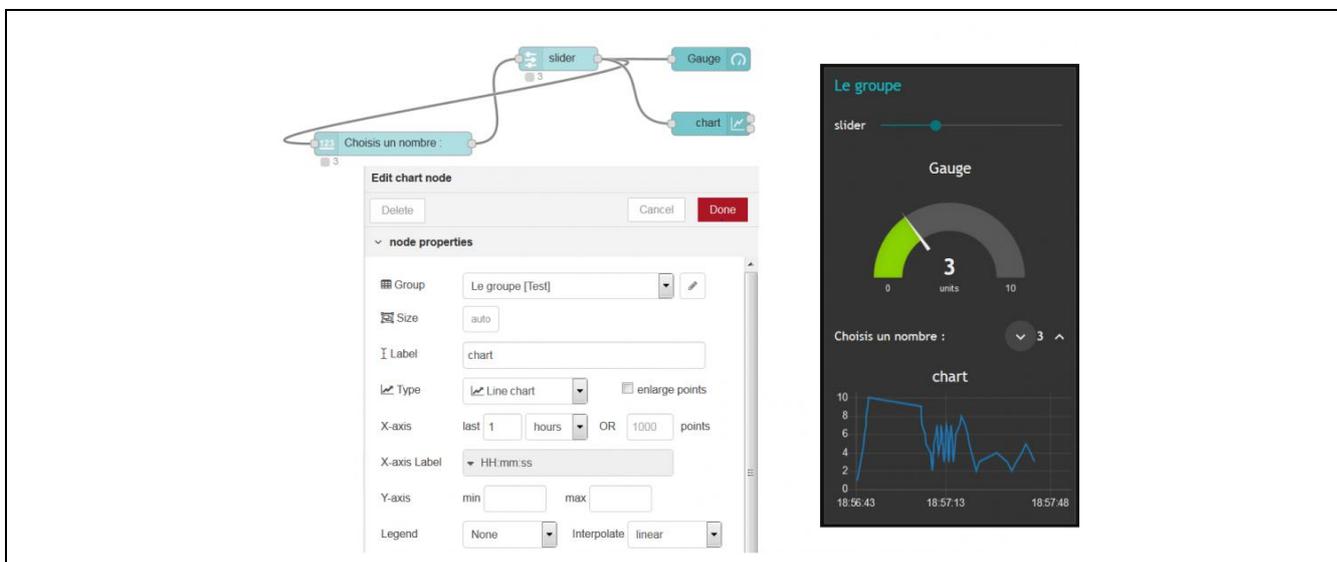


Actionnez le **Slider** et constatez l'action sur la jauge (**Gauge**).

Modifiez le flow en ajoutant un **node Numeric**, déployez et faites constater le résultat :



Pour conserver un historique des valeurs affichés, on peut utiliser un node **Chart** :



Si la propriété **Y-axis** est laissée vide, l'échelle est automatique. L'affichage de la courbe est réactualisé à chaque nouvelle valeur.

Nous allons maintenant rajouter une information textuelle en bas du groupe. Cette information contiendra par exemple la valeur du **slider**, la date et l'heure de modification. L'ajout d'un node **Text** ne suffit pas car il permet d'ajouter un label et le contenu d'un message. Or si on se contente de connecter la sortie du node **slider** à l'entrée du node **Text**, on pourra écrire :

```
Label="Valeur = "
Value Format="{ {msg.payload} }"
```

Il nous faut intercaler un node pour insérer la date dans le message à afficher. Nous allons utiliser un node **Function**, ce node permet d'effectuer n'importe quel traitement sur le flux entrant pour fabriquer un ou plusieurs flux sortants. Le code est du **JavaScript** :

```
var now      = new Date();
var year     = now.getFullYear();
var month    = now.getMonth()+1;
var day      = now.getDate();
var hour     = now.getHours();
var minute   = now.getMinutes();
var second   = now.getSeconds();
msg.payload  =  day+"/"+"
               month+"/"+"
               year+" à "+"
               hour+": "+"
               second+" => "+"
               msg.payload;

return msg;
```

A noter : Les messages transmis dans les flux d'entrée et de sortie des nodes sont des objets. C'est l'attribut **payload**, soit "**charge utile**" en français qui contient l'information à transmettre. Pour observer cette charge utile de message vous pouvez utiliser un node **Debug**.

The screenshot displays the Node-RED web interface. On the left, a flow named 'Flow 1' is visible, starting with a 'Choisis un nombre' slider node. This node is connected to four other nodes: 'Gauge', 'chart', 'msg payload', and 'date'. The 'date' node is connected to a 'Valeur le' text node. The 'debug' console on the right shows the message objects being processed, including the payload value and the formatted date-time string. On the far right, a mobile app preview shows a gauge displaying the value '6' and a chart showing a line graph with a data point at the end. A red box highlights the text 'Valeur le 18/2/2018 à 13:9 => 6' at the bottom of the app preview.

Faites constater à l'enseignant.

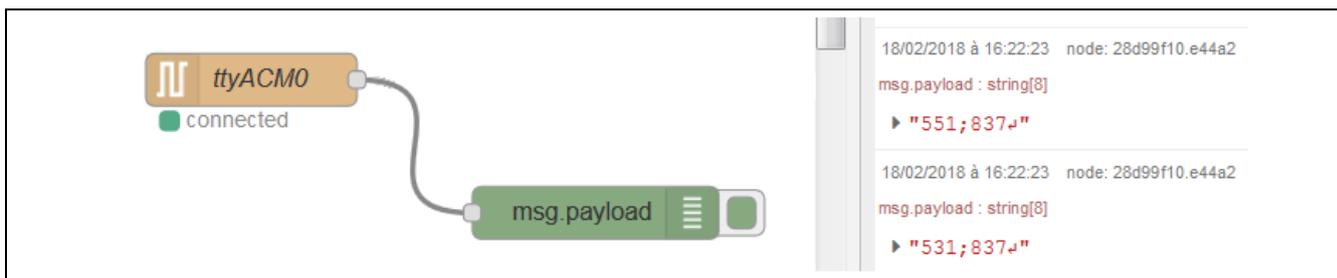
4.3 Interaction avec Arduino

Attention : Avant de commencer un nouveau Flow, pensez à supprimer ou désactiver (double-clic sur l'onglet du flow puis **Disabled**) sous peine de servir deux flows, soit deux programmes en même temps, ce qui risque d'entraîner de curieux effets.

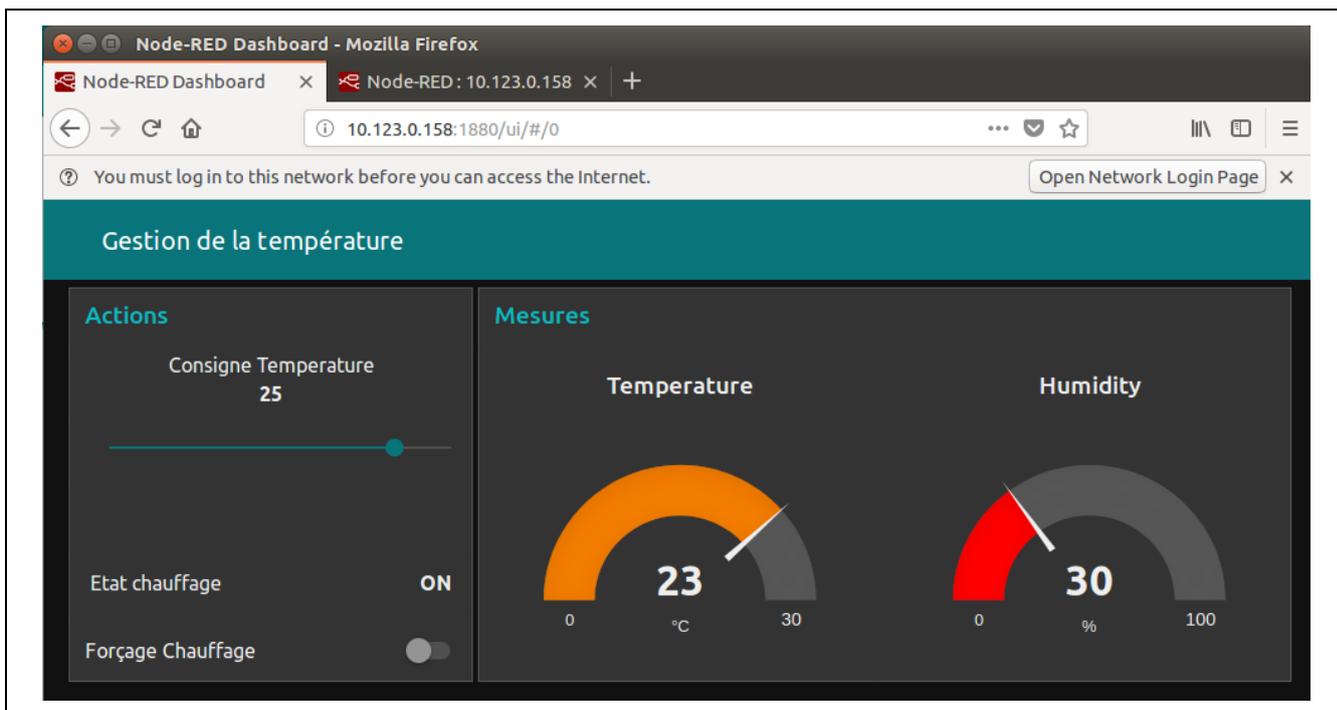
Les informations issues de la carte Arduino parviennent à la Raspberry via le port série généralement nommé **/dev/ttyACM0**. Nous allons donc placer sur notre flow un node **Serial** et le configurer pour une connexion à ce port avec comme caractéristiques :

- débit : 9600 bauds
- Stop : 1
- parité : aucune
- contrôle de flux : aucun
- caractère de séparation : "\n"

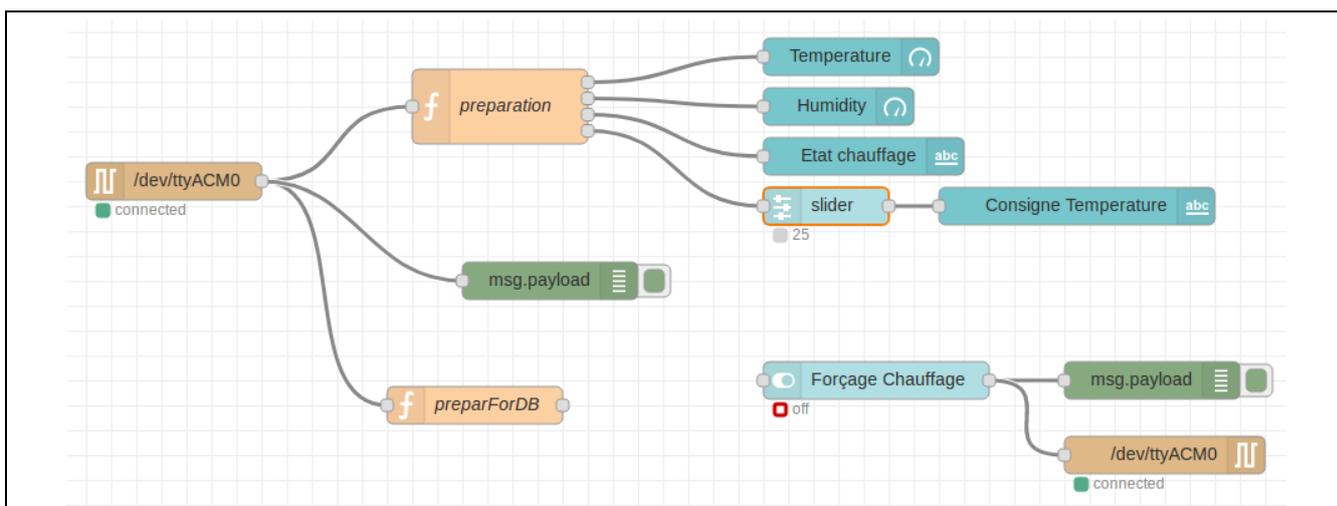
Un node **Debug** permettra d'observer la bonne acquisition de la trame envoyée par Arduino.



Réalisons maintenant un flow complet pour exploiter le potentiel de notre montage au travers de cette IHM :

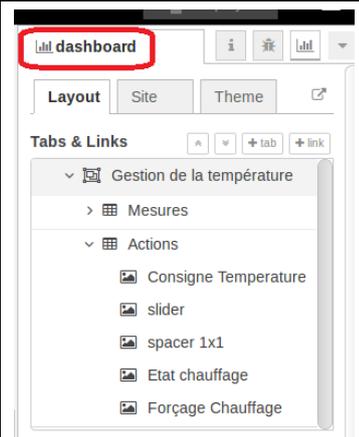


Commençons par la saisie :



4.3.1 Préparation de la page web

Group	Node name	Node type	Size	Fonction
Mesures	Temperature	Gauge	6x5	Affiche la température (°C) Bleu(0-18°C)Orange(18-25°C)Rouge(25-30°C)
	Humidity	Gauge	6x5	Affiche l'humidité relative (%) Rouge(0-40%)Vert(40-60%)Bleu(60-100%)
Actions	Consigne Temperature	Text	auto	Affiche la valeur de la consigne de température
	(Slider)	Slider	auto	Range : 0-30 step 1
	(Spacer)	Spacer	1x1	Séparation (Dashboard Layout)
	Etat chauffage	Text	auto	Indique ON ou OFF
	Forçage Chauffage	Switch	auto	Force le chauffage à ON ou OFF



4.3.2 Code des fonctions

preparation ()

```

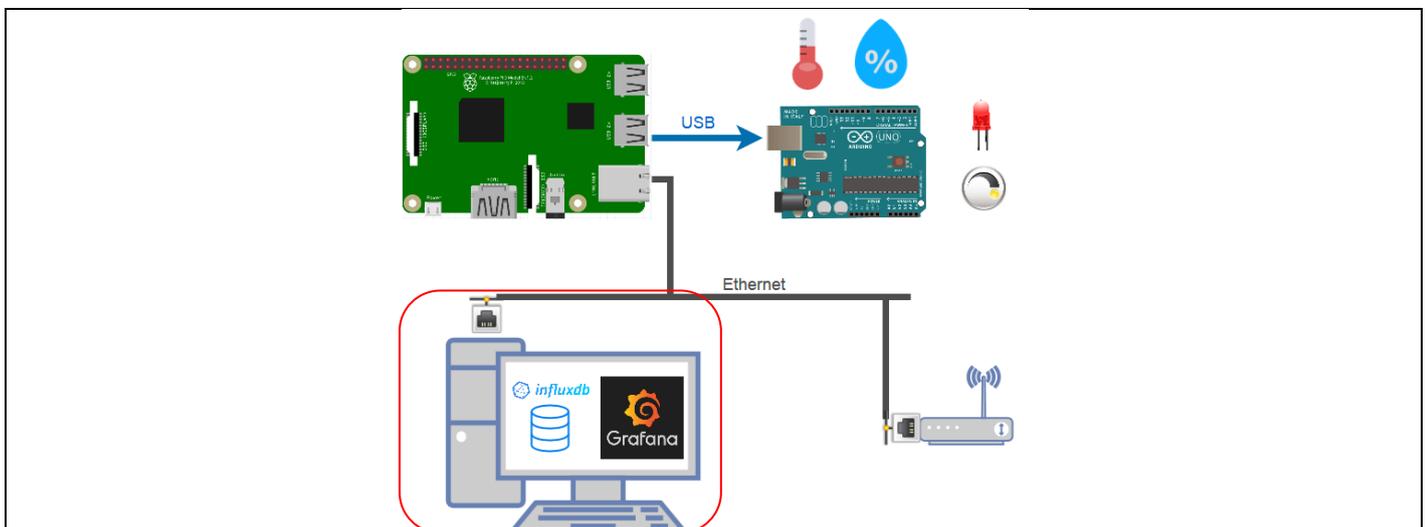
var o;
var temp={};
var hum={};
var stateLed={};
var cons={};
o = msg.payload.split(";");
hum.payload=o[0];
temp.payload=o[1];
cons.payload=o[3];
if(o[2]=="1") stateLed.payload="ON";
else stateLed.payload="OFF";

return [hum,temp,stateLed,cons];
    
```

Faites constater à l'enseignant.

5 Historisation des données

Pour historiser les données, nous allons utiliser un gestionnaire de bases de données orienté séries chronologiques, ici **InfluxDB**. Pour accéder facilement aux données, vous allez créer un dashboard à l'aide de l'outil **Grafana** sur votre PC Ubuntu :



5.1 Installation de InfluxDB

```
$ sudo apt-get update
$ curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
$ source /etc/lsb-release
$ echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list

$ sudo apt-get update && sudo apt-get install influxdb
```

5.2 Démarrage du service

```
$ sudo service influxdb start
$ influx
...
InfluxDB shell 1.7.3
Enter an InfluxQL query
>
```

5.3 Créer et utiliser la base

```
> CREATE DATABASE chauffage
> SHOW DATABASES
...
chauffage
> USE chauffage
Using database chauffage
```

5.4 Insérer des données dans une table

```
> INSERT historical,mesure=temperature value=24
> INSERT historical,mesure=humidity value=45
> SELECT * FROM historical
name: historical
time                mesure          value
-----
1549290274188669098 temperature     24
1549290274189929451 humidity        45
>
```

5.5 Automatiser l'insertion des données

Nous allons modifier le flow Node-Red pour insérer automatiquement les données. Pour cela, il nous faut ajouter les nodes de gestion de InfluxDB :

5.5.1 Suppression des données dans la table

```
> DROP SERIES FROM historical
> SELECT * FROM historical
>
```

5.5.2 Installation des nodes de gestion de InfluxDB

A partir du gestionnaire de palettes de Node-Red, installez le node *node-red-contrib-influxdb*



5.5.3 Insertion des données

Modifiez le flow Node-Red pour assurer une insertion des données dans la base InfluxDB :

Le serveur InfluxDB est installé sur votre PC et écoute les requêtes sur le port 8086.

Measurement indique le nom de la table dans laquelle les données seront introduites.

La fonction *preparForDB()* fourni un objet json contenant les noms des champs et les valeurs à enregistrer :

preparForDB()

```
var o;
var temp;
var hum;
var cons;
var stateLed;
o = msg.payload.split(";");
hum = parseInt(o[1]);
temp = parseInt(o[0]);
cons = parseInt(o[3]);
stateLed = o[2];

msg.payload = [{
  "humidity" : hum,
  "temperature" : temp,
  "order" : cons,
  "stateLed" : stateLed
}];
return msg;
```

N'oubliez pas de déployer le flow !

Faites constater l'insertion des données à l'enseignant.

```
> SELECT * FROM historical
name: historical
time                humidity order stateLed temperature
----                -
1549295413210398929 37      20    0      23
1549295415476852641 37      20    0      23
1549295417752839537 37      20    0      23
1549295420027007197 37      20    0      23
1549295422299668872 37      20    0      23
1549295424576904176 37      20    0      23
1549295426851769621 37      20    0      23
1549295429126275721 37      20    0      23
1549295431398993075 37      20    0      23
1549295433674050496 37      20    0      23
```

5.6 Installation de grafana

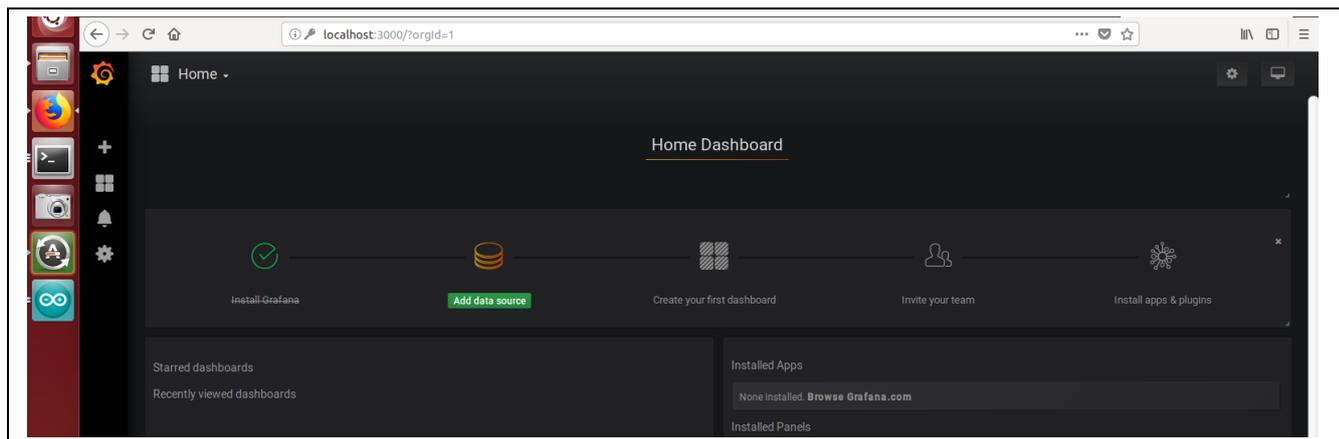
```
$ echo "deb https://packagecloud.io/grafana/stable/debian/ wheezy main" | sudo tee
/etc/apt/sources.list.d/grafana.list
$ curl https://packagecloud.io/gpg.key | sudo apt-key add -
...
$ sudo apt-get update && sudo apt-get install grafana
```

5.7 Démarrer le service grafana

```
$ sudo service grafana-server start
```

Grafana diffuse et écoute sur le port 3000, vous accéder à l'interface de configuration au moyen d'un navigateur avec l'URL <http://localhost:3000>

Par défaut, Username = **admin** et password = **admin**.



Add data source -> InfluxDB :

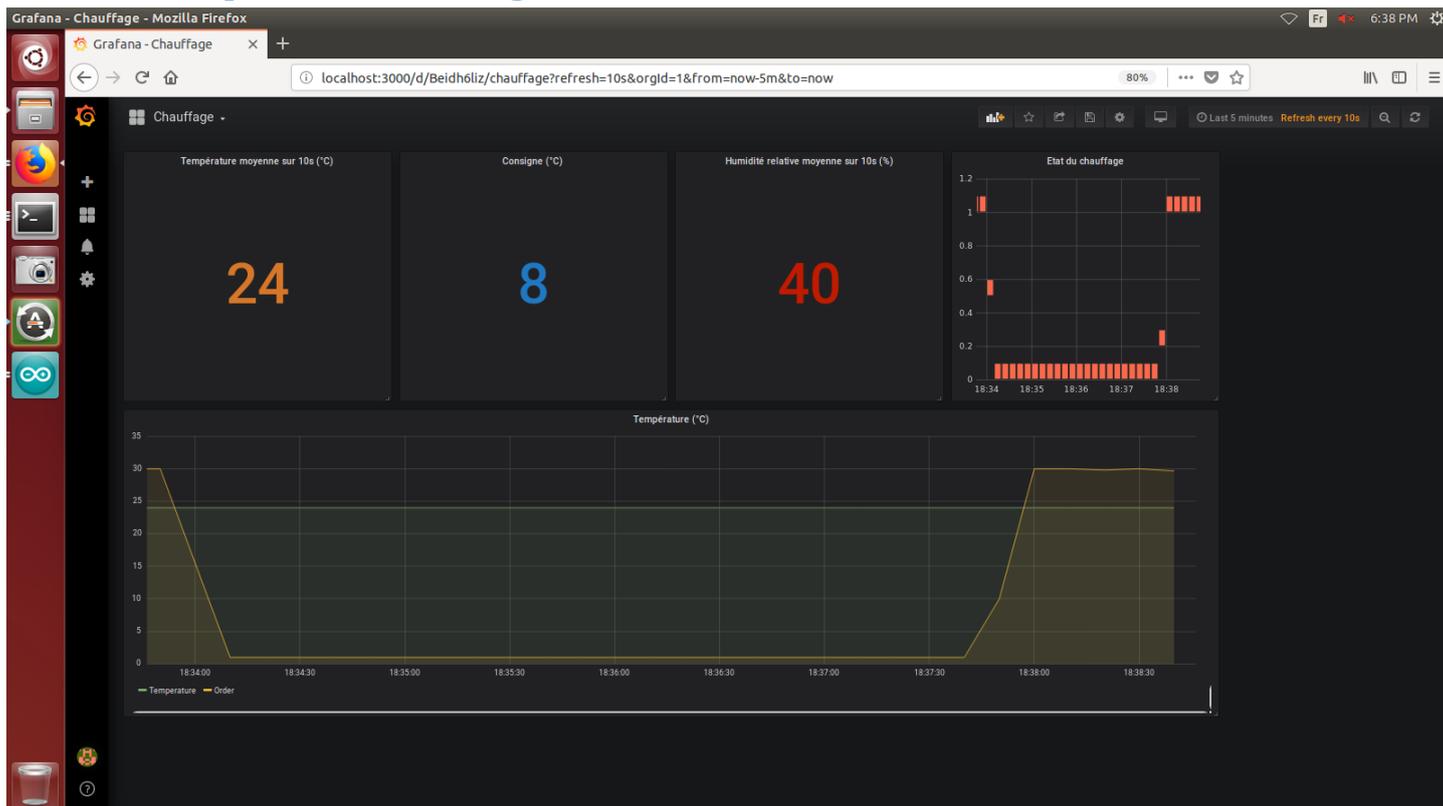
- **Name** : Chauffage
- **URL** : <http://localhost:8086>
- **Database** : chauffage

5.8 Dashboard historique

- Cliquez sur **+** puis **Dashboard** dans le menu de gauche.
- Dans l'onglet **Add**, cliquez sur **Graph**
- Cliquez sur le titre du graph puis **Edit**
- Dans l'onglet **Metrics** on formule la requête
 - FROM default **historical**
 - SELECT field(**temperature**) mean()
 - GROUP BY time(**10s**) fill(null)
 - FORMAT AS Times series
 - ALIAS BY **Temperature**
- Dans l'onglet **General** indiquez le titre du graph : **Température (°C)**
- Sur la partie droite en haut de l'écran, indiquez la durée de la fenêtre d'affichage et la période de rafraîchissement : **Last 30 minutes** et **Refresh every 10s**
- Sauvegardez (icône disquette) retournez sur le dashboard (flèche blanche sur fond bleu en haut à droite)

Concevez votre dashboard et faites constater par l'enseignant.

5.9 Exemple de dashboard grafana



6 Allez plus loin...

S'il vous reste du temps et que vous possédez un compte google et/ou Twitter, vous pouvez essayer de mettre en œuvre une notification automatique à partir de Node Red....

Cependant, votre compte google doit être configuré pour accorder l'accès aux applications moins sécurisées <https://myaccount.google.com/lesssecureapps>

<input type="checkbox"/>	☆	moi	Alert ! - La led à changer d'état : OFF
<input type="checkbox"/>	☆	moi	Alert ! - La led à changer d'état : ON

7 Références web

7.1 Raspberry Pi

- <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- <http://www.raspberry-projects.com/pi/category/programming-in-c>
- <http://www.adafruit.com/pdfs/raspberrypi2modelb.pdf>
- <https://learn.adafruit.com/downloads/pdf/introducing-the-raspberry-pi-2-model-b.pdf>
- https://en.wikipedia.org/wiki/ARM_Cortex-A7

7.2 Node Red

- <http://noderedguide.com/>
- <https://nodered.org/>

7.3 InfluxDB

- <https://www.influxdata.com/>
- <https://docs.influxdata.com/influxdb/v1.7/introduction/installation/>
- https://docs.influxdata.com/influxdb/v1.7/administration/authentication_and_authorization
- <https://docs.influxdata.com/influxdb/v1.7/introduction/getting-started/>

7.4 Grafana

- <https://grafana.com/>
- <http://docs.grafana.org/installation/debian/>
- http://docs.grafana.org/guides/getting_started/
- <http://docs.grafana.org/features/datasources/influxdb/>

7.5 Divers

- https://fr.wikipedia.org/wiki/Internet_des_objets
- <https://www.usine-digitale.fr/objets-connectes/>
- https://fr.wikipedia.org/wiki/Web_des_objets
- <https://www.objetconnecte.com/guide-objets-connectes/quest-ce-quun-objet-connecte/>
- <http://silanus.fr/>
- <https://github.com/msilanus>