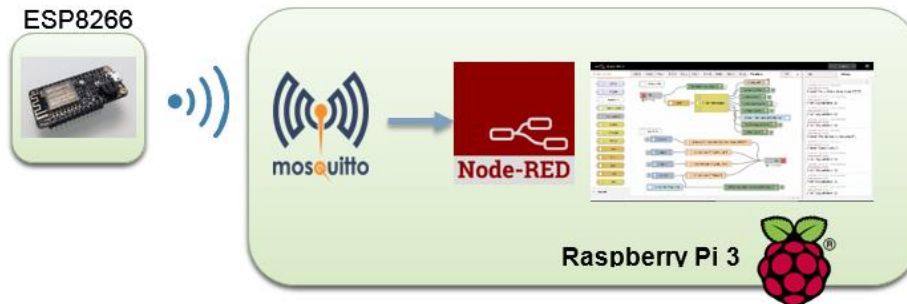


### Activité 3 : Mise en œuvre du MQTT avec un ESP8266

L'objectif de cette activité est de réaliser la mise en œuvre d'une chaîne IOT simplifiée à partir d'un module ESP8266 et d'une carte raspberry :



## Travail préparatoire

Lire le document : [fiche\\_ESP8266.pdf](#)

Ajouter les plugins, le driver et la bibliothèque à l'IDE Arduino en suivant la procédure :

⇒ <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/using-arduino-ide>

Le driver **CP210x\_Windows\_Drivers** (windows 7 – 64bits) est déjà disponible dans les ressources, ainsi que le programme **blink\_esp8266** et le programme **test\_wifi\_esp8266**.

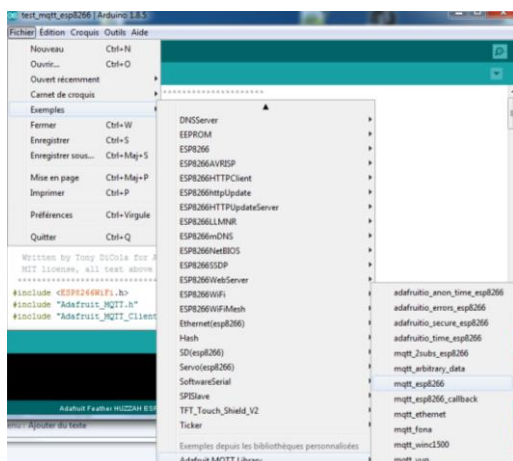
Pour tester le bon fonctionnement du driver et de la carte, ouvrir le programme **blink\_esp8266** et téléverser le dans votre module ESP8266 (choisir la bonne carte et le bon port com). La LED doit clignoter.

Pour tester la connexion wifi, on se connecte à un serveur distant : [wifitest.adafruit.com/testwifi/index.html](http://wifitest.adafruit.com/testwifi/index.html). Pour cela, ouvrir le programme **test\_wifi\_esp8266** (dont il faut modifier les paramètres de connexion wifi en fonction de votre installation) et téléverser le dans votre module ESP8266. Ouvrir le terminal série de l'IDE arduino (vitesse 115200) et vérifier le bon fonctionnement.

## Publier des valeurs par MQTT avec l'ESP8266

Installer la bibliothèque mqtt disponible dans le dossier ressources (la décompresser et la placer dans le dossier MesDocuments/arduino/libraries). Puis relancer votre IDE Arduino.

Vous devez retrouver tous les exemples mqtt fournis par cette bibliothèque :



Programmer votre module ESP8266 avec le code **test\_mqtt\_esp8266** disponible dans le dossier ressources (on modifie les paramètres de connexion wifi et de connexion au broker si besoin).

```
/****** WiFi Access Point *****/

#define WLAN_SSID      "mySSID"
#define WLAN_PASS      "myPW"

/****** Adafruit.io Setup *****/

#define AIO_SERVER      "192.168.1.3" //adresse du broker
#define AIO_SERVERPORT  1883          // use 8883 for SSL
#define AIO_USERNAME    "snir_ab"
#define AIO_KEY         "SNIR@briand"
```

Le programme permet de publier à travers le topic « g133/temperature » sur le broker mosquitto de notre raspberry.

```
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

/****** Publisher *****/
Adafruit_MQTT_Publish temp = Adafruit_MQTT_Publish(mqtt, "g133/temperature");
```

Sur votre raspberry, lancer un subscriber : **mosquitto\_sub -h localhost -t "g133/temperature" -u snir\_ab -P snir@briand**

Vous devez voir apparaître les valeurs publiées par le module ESP8266.

```
pi@BTS_SNIR_AB:~ $ mosquitto_sub -h localhost -t "g133/temperature" -u snir_ab -P SNIR@briand
234
235
236
237
238
239
240
```

Ouvrir un terminale série depuis l'IDE arduino.

```
WiFi connected
IP address:
192.168.1.45
Connecting to MQTT... MQTT Connected!

Sending temperature val 0...OK!

Sending temperature val 1...OK!

Sending temperature val 2...OK!

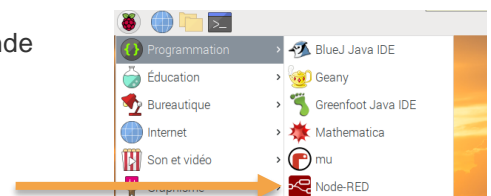
Sending temperature val 3...OK!
```

## Créer une interface web avec Node-Red

Lire l'annexe : Présentation de Node-Red

L'objectif est de réaliser une interface web de visualisation des données MQTT issues du module ESP8266. Pour cela, nous allons utiliser un node MQTT (natif à Node-Red) et son interface graphique (UI).

Lancer Node-Red sur votre raspberry depuis un terminale avec la commande **node-red-start** ou depuis le bureau :



Connectez-vous ensuite par l'intermédiaire de votre navigateur (de votre PC) sur l'interface Node-Red via l'URL de votre raspberry suivie du port 1880.

Sélectionner un nœud d'entrée de type mqtt et le connecter à un nœud dashboard de type gauge :



Si la bibliothèque dashboard n'est pas installée, suivre la procédure suivante :

```
cd ~/.node-red
npm i node-red-dashboard (ne pas prendre en compte les warning)
```

Puis redémarrer votre RPI

Pour plus d'information :

<https://flows.nodered.org/node/node-red-dashboard>

<https://projetsdiy.fr/node-red-dashboard-interface-partie1/>

### Paramétrer le nœud mqtt :

**Edit mqtt in node**

node properties

Server: Add new mqtt-broker... | Topic: Topic | QoS: 2 | Name: Name

**mqtt in > Add new mqtt-broker config node**

Connection tab: Server: localhost | Port: 1883 | Enable secure (SSL/TLS) connection: ☐ | Client ID: Leave blank for auto generated | Keep alive time (s): 60 | Use clean session: ☒ | Use legacy MQTT 3.1 support: ☒

**mqtt in > Add new mqtt-broker config node**

Security tab: Username: snir\_ab | Password: .....

**Edit mqtt in node**

node properties

Server: localhost:1883 | Topic: g133/temperature | QoS: 0 | Name: esp8266

### Paramétrer le nœud gauge :

**Edit gauge node**

node properties

Group: Add new ui\_group... | Size: auto | Type: Gauge | Label: Gauge | Value format: {{value}} | Units: units | Range: min 0 max 10

**gauge > Add new dashboard group config node**

Name: ESP8266 | Tab: Add new ui\_tab... | Width: 6 | Display group name: ☒

**gauge > dashboard group > Edit dashboard tab node**

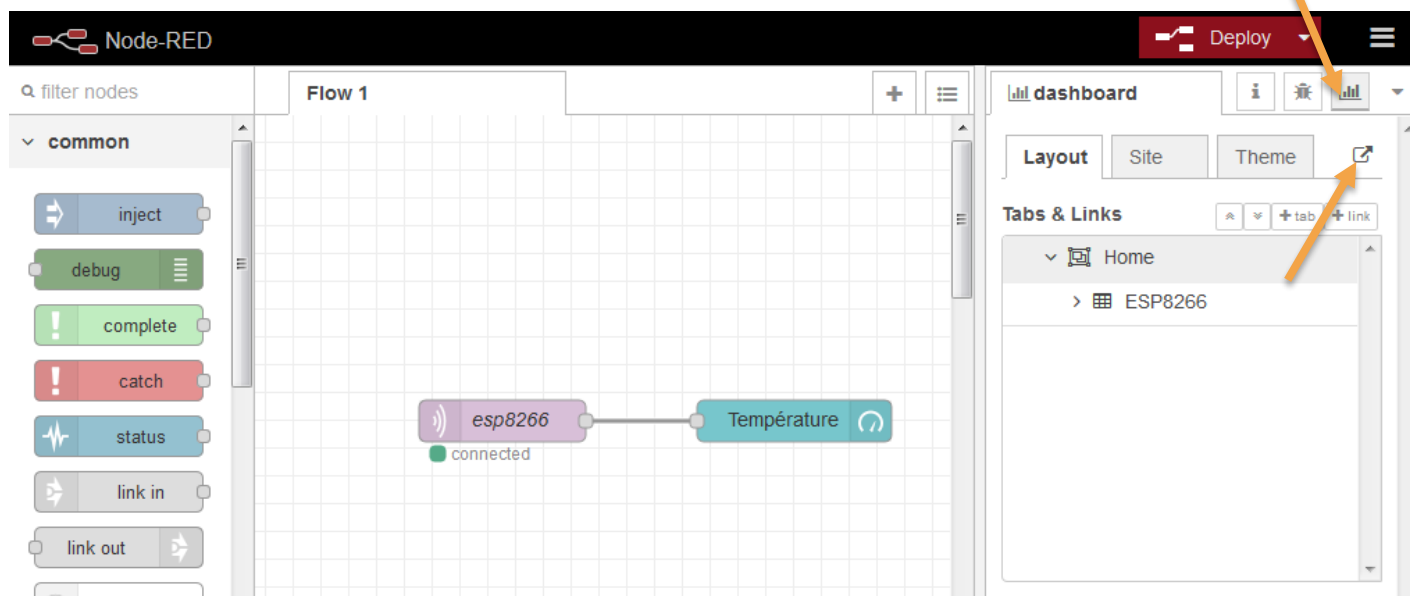
Name: Home | Icon: dashboard

**Edit gauge node**

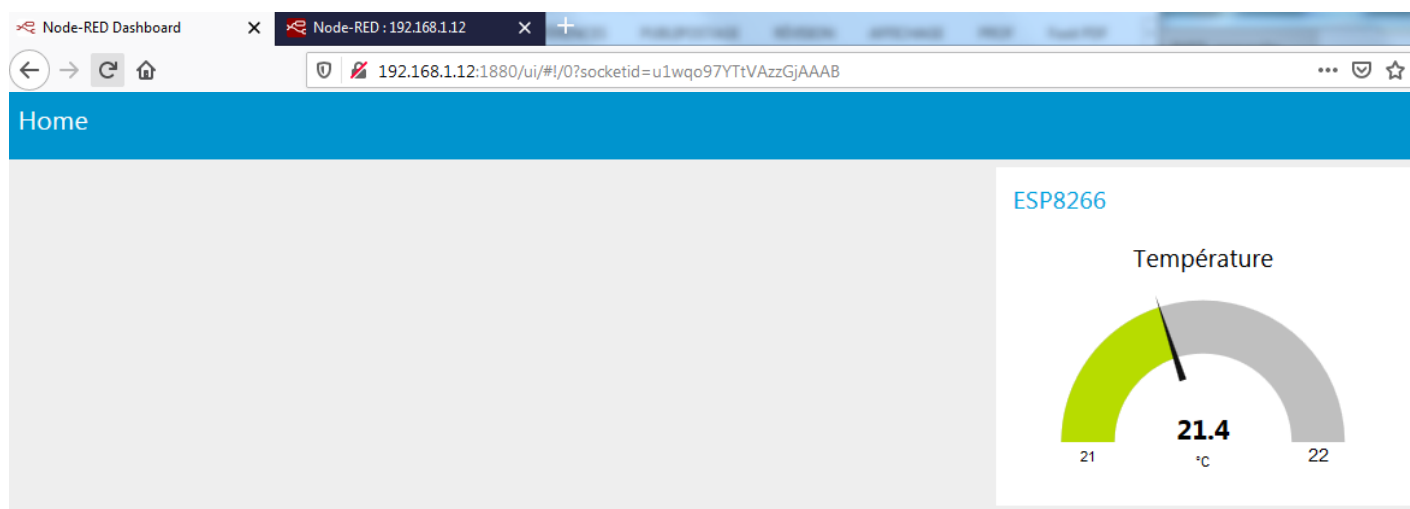
Properties tab

Group: [Home] ESP8266 | Size: auto | Type: Gauge | Label: Température | Value format: {{value}} | Units: °C | Range: min 21 max 22 | Colour gradient: [Green] [Yellow] [Red] | Sectors: 21 optional optional 22 | Name:

Déployer votre flow et visualiser l'interface graphique créée en ajoutant /ui à la fin de l'url ou en cliquant :



Vous devez obtenir le résultat suivant :

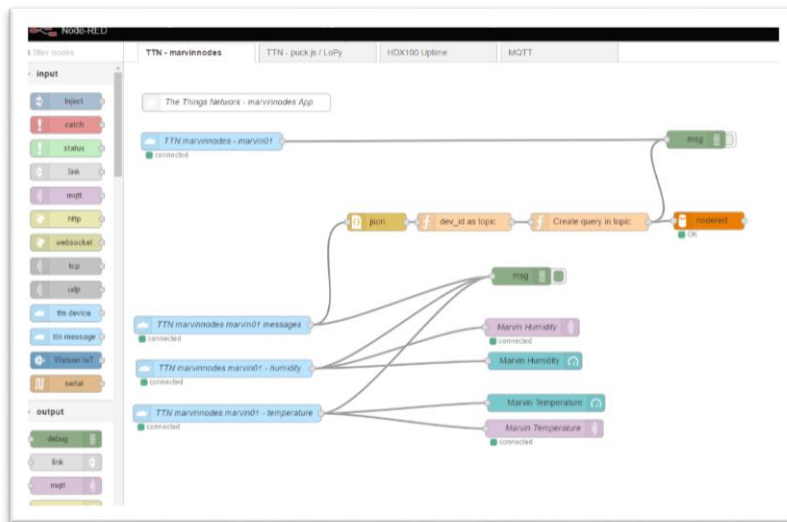


## ANNEXE - Node-Red



NODE-RED est une plateforme, soutenue par IBM, permettant de tirer profit d'un parc d'objets connectés. Elle permet la collecte des données émises et leur transport. Elle a la capacité de créer des flux de données et de traiter l'information reçue, mais aussi de déclencher des actions sur des capteurs actifs. L'objectif premier est la création rapide de nouvelles applications ou services.

Techniquement, il s'agit d'une application reposant sur Node.js et permettant le design des chaînes de traitement dans un environnement Web. Node-Red fonctionne avec une palette de connecteurs et des composants de traitement, que l'on interconnecte entre eux.



Node red est installée par défaut sur la distribution Raspbian pour raspberry. La plateforme possède nativement les blocs mqtt.

### **Pour plus d'information :**

Tutoriel node-red de Marc Silanus : <http://silanus.fr/sin/?p=984>

Tutoriel node-red du site Projets DIY : <https://projetsdiy.fr/node-red-decouverte-sur-raspberry-pi-3-ou-2/>

Présentation de node-red sur le RNR STI eduscol :

<http://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/8054/8054-objets-communicants.pdf>