

2020 - 2021

# COAP

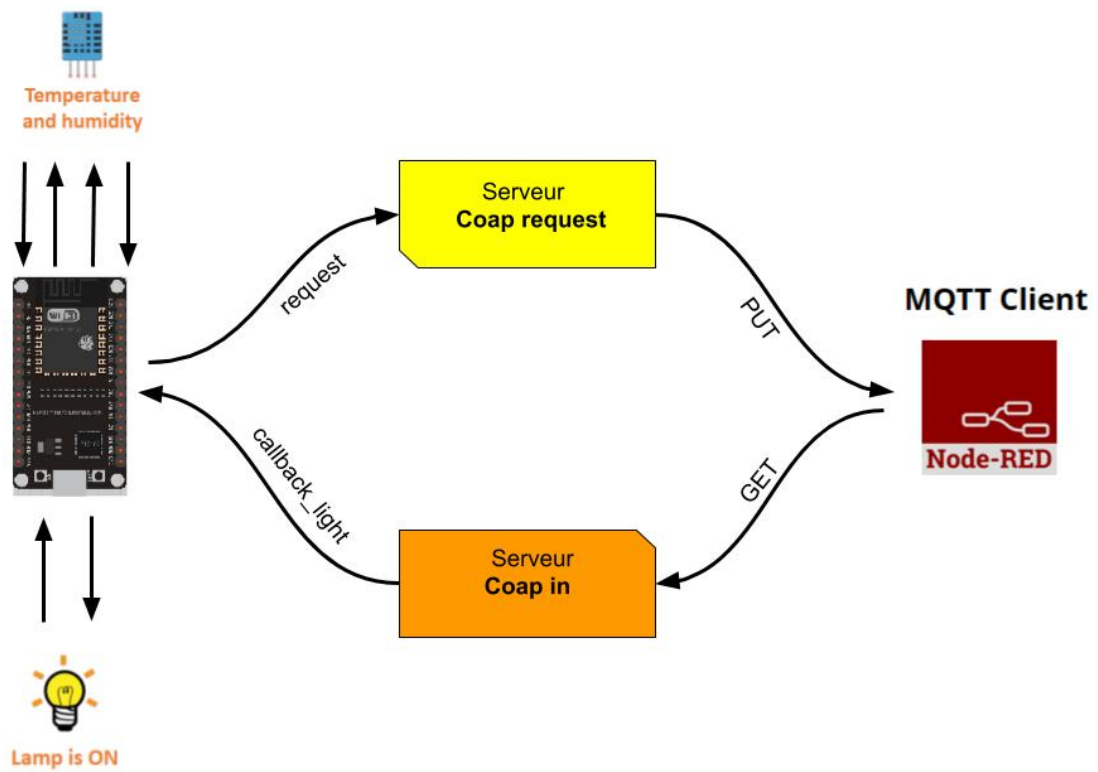
Implémentation du COAP pour une utilisation  
uplink/downlink



## Table des matières

Schéma synopsis.....	2
Branchement des différents modules du projet .....	3
Installation de NODE-RED.....	4
Récupération de l'adresse IP de l'ESP32 .....	6
Configuration de NODE-RED après l'installation afin d'établir une communication COAP .....	7
Préparation du Arduino IDE .....	15

## Schéma synopsis

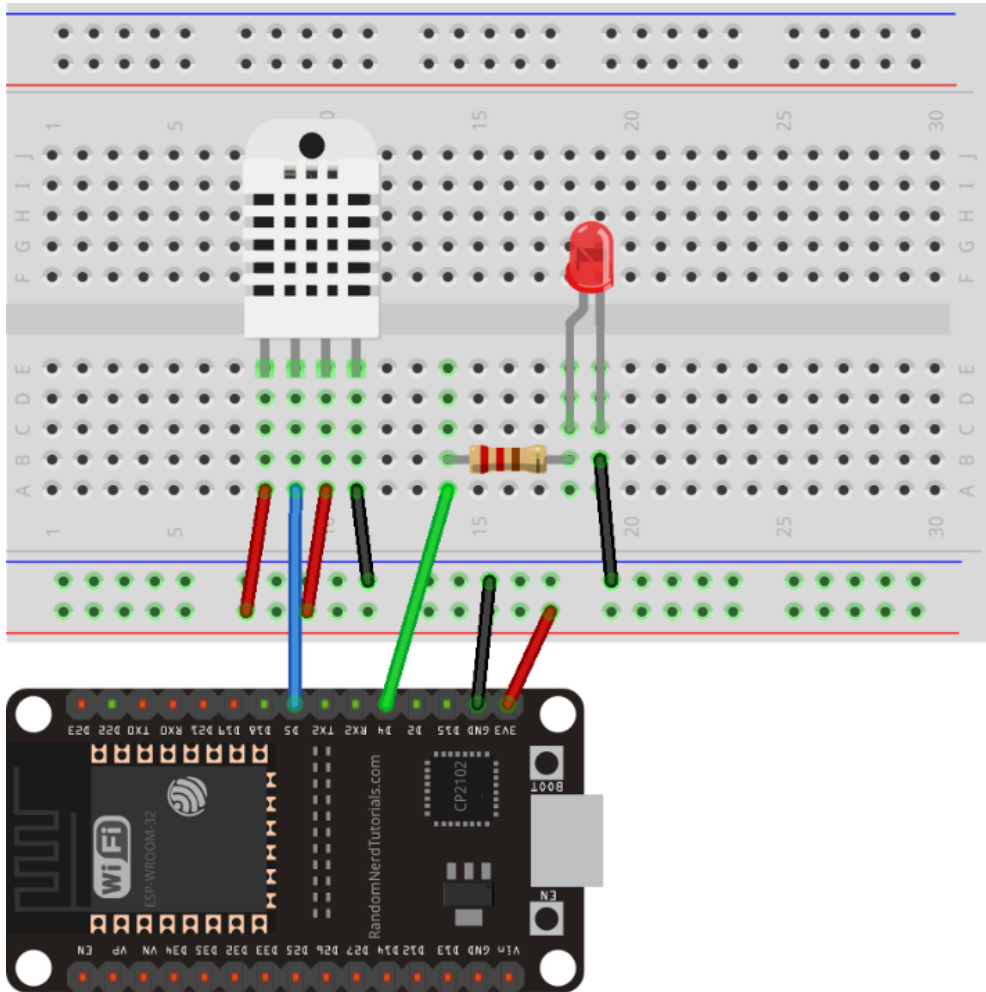


## Liste du matériel

Voici la liste du matériel nécessaire pour réaliser le projet :

- ESP32
- LED
- DHT22

## Branchement des différents modules du projet

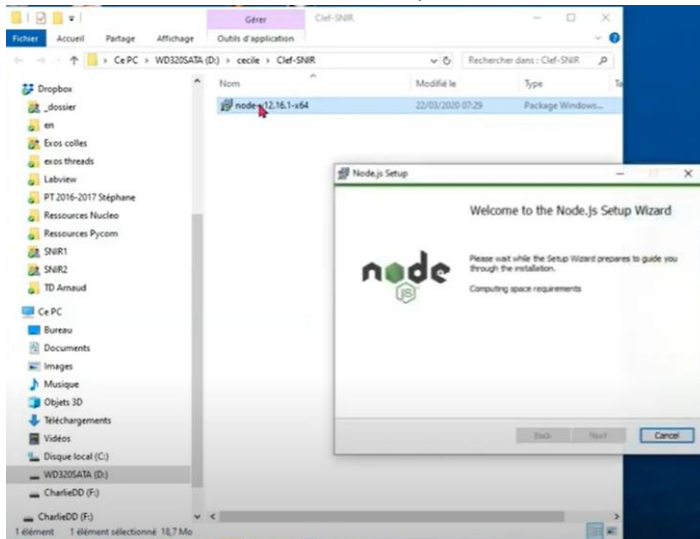


fritzing

# Installation de NODE-RED

## 1) Installation de Node.js

- Cliquer sur ce lien <https://nodejs.org/en/> et installer la version LTS (Long-Term support)
- Exécuter l'installateur (le fichier qui vient d'être téléchargé)



- Suivre les différentes étapes en maintenant une configuration par défaut
  - Vérifier l'installation :
    - Ouvrir un terminal
    - Taper `node -v` et vérifier la version installée
    - Faire de même pour vérifier l'installation du gestionnaire de paquet npm
- Taper `npm -v`

```
Invite de commandes
Microsoft Windows [version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\cecile>node -v
v12.16.1

C:\Users\cecile>npm -v
6.13.1

C:\Users\cecile>
```

## 2) Installation de Node-Red via npm (gestionnaire de paquet)

- Taper dans le terminal la commande suivante :

*npm install -g --unsafe-perm node-red*

```
Invite de commandes
Microsoft Windows [version 10.0.18362.720]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\cecile>node -v
v12.16.1

C:\Users\cecile>npm -v
6.13.1

C:\Users\cecile>npm install -g --unsafe-perm node-red
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
C:\Users\cecile\AppData\Roaming\npm\node-red -> C:\Users\cecile\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\cecile\AppData\Roaming\npm\node-red -> C:\Users\cecile\AppData\Roaming\npm\node_modules\node-red\red.js
+ node-red@1.0.4
updated 1 package in 16.85s

C:\Users\cecile>
```

- Exécuter Node-Red en tapant la commande suivante :

*node-red*

```
C:\Users\cecile>node-red
22 Mar 11:34:12 - [info]

Welcome to Node-RED
=====

22 Mar 11:34:12 - [info] Node-RED version: v1.0.4
22 Mar 11:34:12 - [info] Node.js version: v12.16.1
22 Mar 11:34:12 - [info] Windows_NT 10.0.18362 x64 LE
22 Mar 11:34:12 - [info] Loading palette nodes
22 Mar 11:34:13 - [info] Settings file : \Users\cecile\.node-red\settings.js
22 Mar 11:34:13 - [info] Context store : 'default' [module=memory]
22 Mar 11:34:13 - [info] User directory : \Users\cecile\.node-red
22 Mar 11:34:13 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Mar 11:34:13 - [info] Flows file : \Users\cecile\.node-red\flows_DESKTOP-SQG3D6I.json
22 Mar 11:34:13 - [info] Creating new flow file
22 Mar 11:34:13 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

22 Mar 11:34:13 - [info] Starting flows
22 Mar 11:34:13 - [info] Started flows
22 Mar 11:34:13 - [info] Server now running at http://127.0.0.1:1880/
=
```

- Accéder à l'interface graphique à l'aide d'un navigateur en accédant à l'adresse indiquée à la dernière ligne.

Comme vous pouvez le voir notre serveur tourne en local sur le port 1880.

On peut y accéder à l'adresse <http://127.0.0.1:1880/>

## Récupération de l'adresse IP de l'ESP32

Nous programmerons l'ESP32 en utilisant l'IDE Arduino. Pour télécharger du code sur votre ESP32 à l'aide de l'IDE Arduino, [vous devez installer l'ESP32](#) . Vous devrez également installer deux bibliothèques supplémentaires pour que tout soit prêt pour votre ESP32.

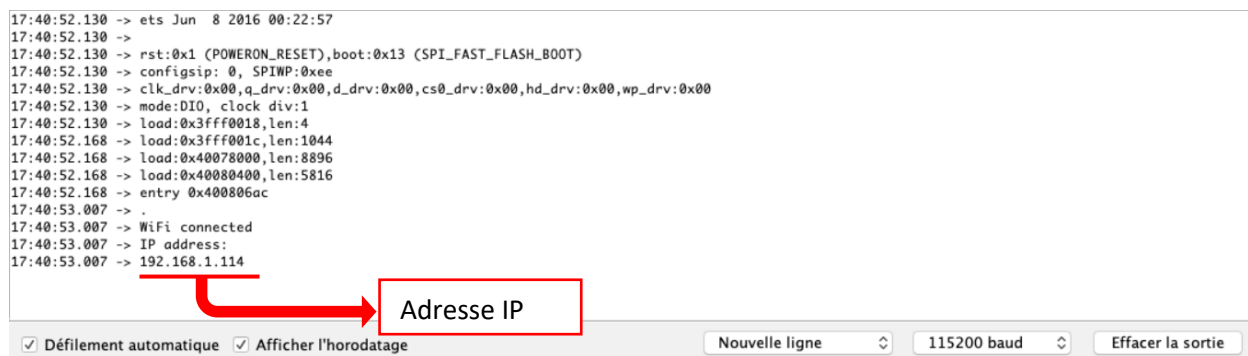
Ensuite, télécharger le [fichier permettant de retrouver l'adresse IP](#).

Au moment de téléverser le code vérifiez que vous ciblez la bonne carte sur Arduino.

Récupérer l'adresse IP sur le moniteur série d'arduino :

- Cliquer sur **Outils** → **Moniteur série**
- Paramétrer la vitesse de communication du moniteur série à 115200

```
17:40:52.130 -> ets Jun 8 2016 00:22:57
17:40:52.130 ->
17:40:52.130 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
17:40:52.130 -> configsip: 0, SPIWP:0xee
17:40:52.130 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
17:40:52.130 -> mode:DIO, clock div:1
17:40:52.130 -> load:0x3fff0018,len:4
17:40:52.168 -> load:0x3fff001c,len:1044
17:40:52.168 -> load:0x40078000,len:8896
17:40:52.168 -> load:0x40080400,len:5816
17:40:52.168 -> entry 0x400806ac
17:40:53.007 -> .
17:40:53.007 -> WiFi connected
17:40:53.007 -> IP address:
17:40:53.007 -> 192.168.1.114
```



Adresse IP

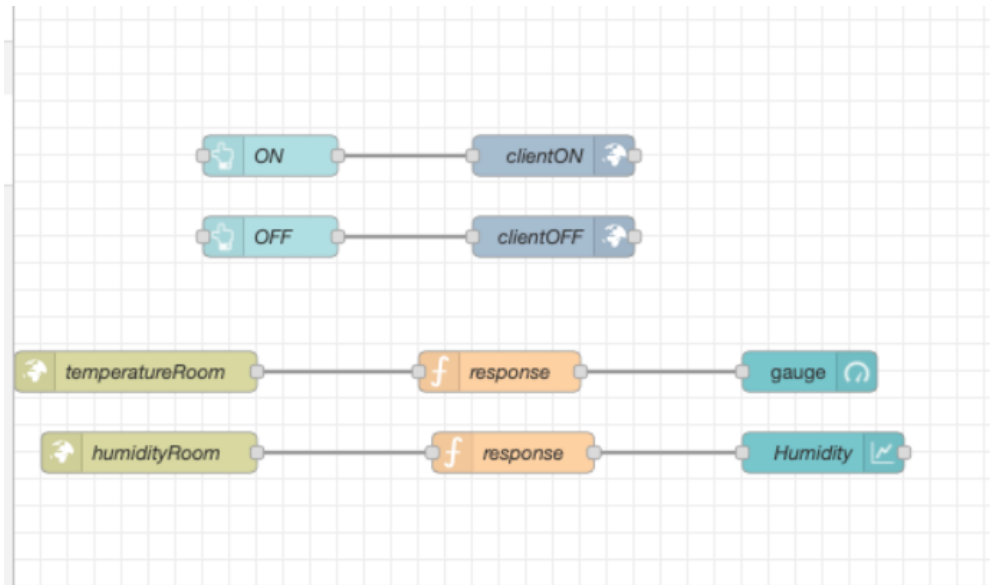
☒ Défilement automatique ☒ Afficher l'horodatage

Nouvelle ligne 115200 baud Effacer la sortie

## Configuration de NODE-RED après l'installation afin d'établir une communication COAP

Dans cette partie nous allons configurer et établir une communication COAP à partir de node-red.

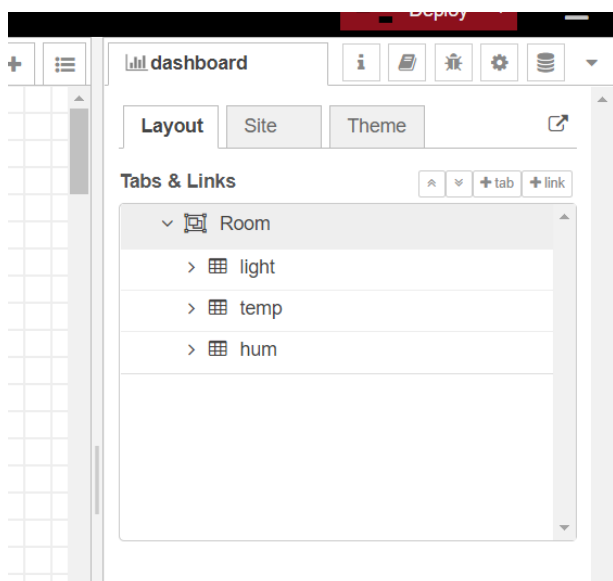
Voici le résultat final attendu :



### 1. Création des Topics

- Sur node-red local on sélectionne **dashboard** puis **Layout** pour créer une tab en cliquant sur **+tab**, la tab constituée de groupe de capteur

En positionnant la souris sur la tab créée des options vont apparaître, cliquez sur **+group** et vous pouvez ajouter vos périphériques externes (actionneurs, capteurs...).

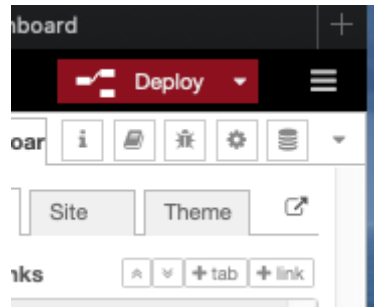




Nommer vos tabs avec les mêmes noms que sur l'image ci-dessus afin de pouvoir utiliser notre projet.

## 2. Créer les Nodes sur node-red

- Télécharger la palette **node-red-contrib-coap**

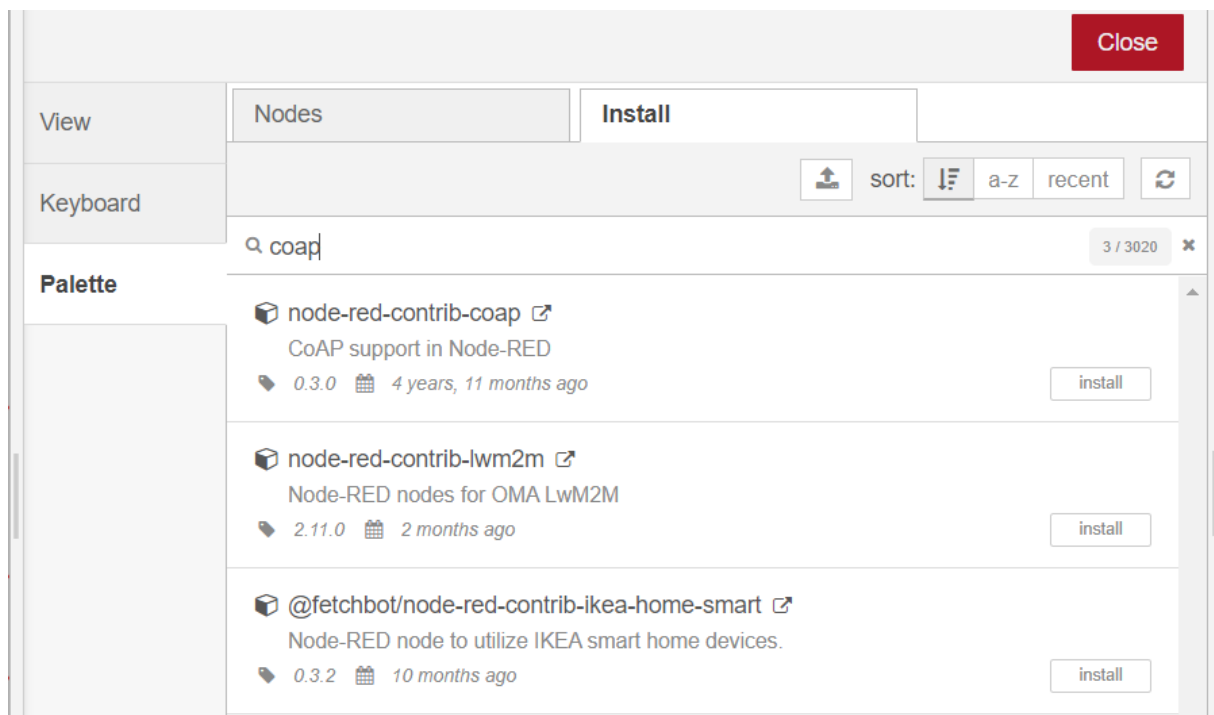


- Cliquez sur le **bouton à droite de « Deploy »**

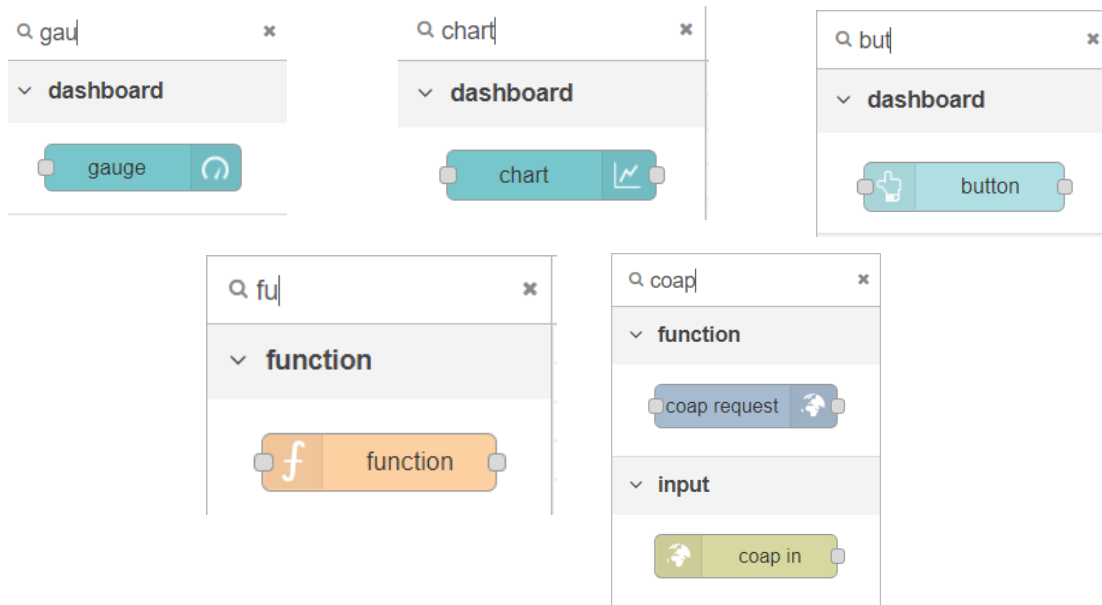


→ **manage palette** → **install** → **node-red-contrib-coap**

- Installer le paquet en cliquant sur **install**



- Maintenant vous pouvez importer les nodes à l'aide de la barre de recherche en haut à gauche sur Node-red et les placer comme sur le résultat final attendu présenté au début de cette partie.



- Configuration des nodes qui devront faire appel aux tab et group créer précédemment.
  - Cliquez sur l'un des deux « **button** » et configurez-le comme ci-dessous :

The image shows the 'Edit button node' configuration window. It includes a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' tab is active, showing various settings:
 

- Group:** '[Room] light' (with a dropdown arrow and edit icon)
- Size:** 'auto' (with a dropdown arrow)
- Icon:** 'optional icon' (with a text input field)
- Label:** 'ON' (with a text input field)
- Tooltip:** 'optional tooltip' (with a text input field)
- Colour:** 'optional text/icon color' (with a text input field)
- Background:** 'optional background color' (with a text input field)
- When clicked, send:** (checked)
  - Payload:** 'a\_z' (with a dropdown arrow and text input field)
  - Topic:** 'ON' (with a text input field)
- If msg arrives on input, emulate a button click:** (unchecked checkbox)
- Name:** 'ON' (with a text input field)

 At the bottom, there is an 'Enabled' checkbox which is currently checked.

- Cliquez sur le deuxième « **button** » et configurez-le comme ci-dessous :

**Edit button node**

Delete Cancel Done

**Properties**

Group [Room] light

Size auto

Icon optional icon

Label OFF

Tooltip optional tooltip

Colour optional text/icon color

Background optional background color

When clicked, send:

Payload a\_z

Topic OFF

→ If msg arrives on input, emulate a button click: ☐

Name OFF

Enabled

- Cliquez sur l'un des « **COAP request** » et configurez-le comme ci-dessous :

**Edit coap request node**

Delete Cancel Done

**Properties**

URL coap://192.168.1.114/light/ON

Method GET

Observe? ☒

Raw buffer? ☒

Content format text/plain

Name clientON

Tip: Leave url or method blank if you want to set them via msg properties.

Enabled

Au niveau de l'URL, veuillez indiquer l'adresse IP trouvé précédemment pour l'ESP32.

- Cliquez sur l'autre « **COAP request** » et configurez-le comme ci-dessous en mettant l'adresse IP de votre ESP32 :

**Edit coap request node**

Delete Cancel Done

**Properties**

URL: coap://192.168.1.114/light/OFF

Method: POST

☐ Observe?

☐ Raw buffer?

Content format: text/plain

Name: clientOFF

Tip: Leave url or method blank if you want to set them via msg properties.

Enabled

- Cliquez sur l'un des « **COAP in** » et configurez-le comme ci-dessous :

**Edit coap in node**

Delete Cancel Done

**Properties**

Server: IoT CoAP Server

Method: PUT

url: /temp

Name: temperatureRoom

Enabled

- Cliquez sur l'autre « **COAP in** » et configurez-le comme ci-dessous :

**Edit coap in node**

Delete Cancel Done

**Properties**

Server IoT CoAP Server

Method PUT

url /hum

Name humidityRoom

- Cliquez sur l'un « **response** » et configurez-le comme ci-dessous :

**Edit function node**

Delete Cancel Done

**Properties**

Name response

Setup Function Close

```

1 msg.payload = parseFloat(msg.req.payload);
2 return msg;

```

Outputs 1

Enabled

- Faites de même pour l'autre « **response** »

- Cliquez sur le **Gauge** et configurez-le comme ci-dessous :

The 'Edit gauge node' window shows the following configuration:

- Group:** [Room] temp
- Size:** auto
- Type:** Gauge
- Label:** gauge
- Value format:** {{value}}
- Units:** units
- Range:** min -30, max 30
- Colour gradient:** Green, Yellow, Red
- Sectors:** -30, optional, optional, 30
- Name:** (empty)

Buttons: Delete, Cancel, Done. A 'Properties' tab is active.

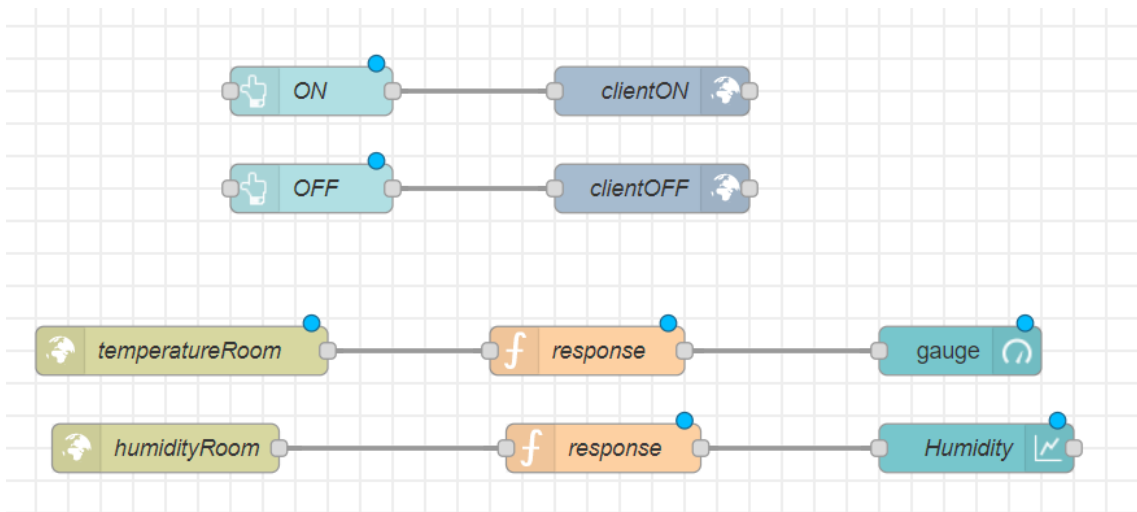
- Cliquez sur le **chart** et configurez-le comme ci-dessous :

The 'Edit chart node' window shows the following configuration:

- Group:** [Room] hum
- Size:** auto
- Label:** Humidity
- Type:** Line chart (with 'enlarge points' checkbox)
- X-axis:** last 1 hours OR 1000 points
- X-axis Label:** HH:mm:ss (with 'as UTC' checkbox)
- Y-axis:** min, max
- Legend:** None (with 'Interpolate' dropdown set to linear)
- Series Colours:** Blue, Light Blue, Orange, Green, Light Green, Red, Pink, Purple, Light Purple
- Blank label:** display this text before valid data arrives
- Name:** Humidity

Buttons: Delete, Cancel, Done. A 'Properties' tab is active.


- Voici le résultat attendu en prenant soin de connecter les nodes :



Une fois terminé, votre node-red sera prêt

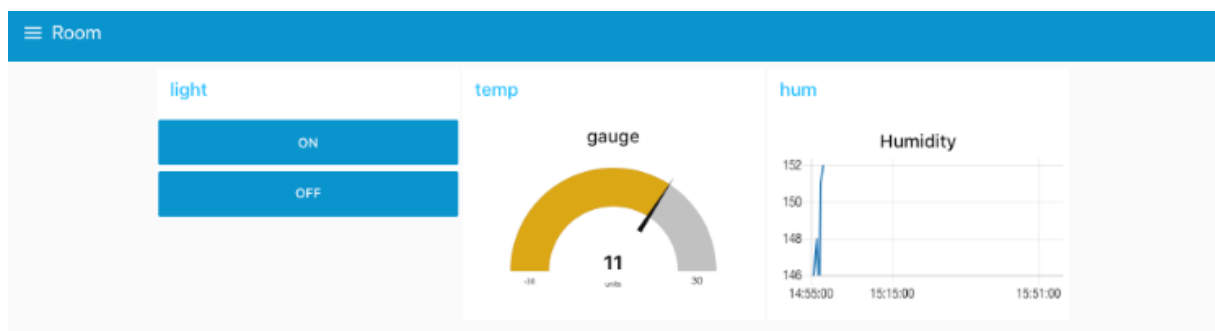
- Cliquer sur « Deploy » afin de déployer l'application



- Visualiser l'aperçu du Dashboard en cliquant sur ce bouton dans le menu **dashboard** : 



Voici l'aperçu attendu de notre dashboard :



## Préparation du Arduino IDE

Veuillez vérifier que vous avez bien installé la librairie de l'ESP32 lors de [la récupération son adresse IP](#) qui était nécessaire lors des précédentes étapes.

La bibliothèque de capteurs DHT permet d'utiliser facilement n'importe quel capteur DHT pour lire la température et l'humidité avec vos cartes ESP32 ou Arduino.

- [Installer la librairie DHT](#)
- Dézipper le fichier et modifier son nom en DHT
- Déplacer le au dossier des librairies arduino

Vous pouvez télécharger le code source pour l'utilisation du COAP, LED et DHT11 avec un ESP32 sur le lien drive suivant [sur le lien drive suivant](#).

Au moment de téléverser le code vérifiez que vous ciblez la bonne carte sur Arduino.

Sur le drive vous trouverez aussi un [fichier JSON](#) que vous pouvez importer directement sur node-red et avoir une configuration identique à celle décrite ci-dessus dans la partie configuration de node-red.