

Travaux pratique du cours : “Découverte du monde connecté”

G Leclercq

Version 1.3.2

2020

Table des matières

1 Objectifs de ces TP.....	6
2 TP broker mosquitto.....	6
2.1 Configuration.....	6
2.1.1 Le broker.....	7
2.1.2 les clients.....	7
2.2 Résultats attendus.....	8
2.3 Démonstrations.....	8
2.4 Résultats obtenus.....	10
3 TP Node red.....	11
3.1 Configuration.....	11
3.1.1 Connexion au site Node-Red public : FRED.....	12
3.1.2 Quelques liens utiles pour découvrir Node-Red.....	12
3.2 TP1 construire son premier flux.....	13
3.2.1 Résultats attendus.....	13
3.2.2 Démonstration.....	13
3.2.2.1 Découvrir l'Interface Homme Machine (IHM) de node-red	13
3.2.2.2 Découvrir comment créer un flux.....	13
3.2.2.3 Découvrir comment tester un flux (flow).....	14
3.2.2.4 Découvrir comment enregistrer ou télécharger un flux.....	14
3.2.3 Résultats obtenus.....	22
3.3 TP2 météo.....	23
3.3.1 Résultats attendus.....	23
3.3.2 Démonstration.....	23
3.3.2.1 Importation du flux météo.....	23
3.3.3 Résultats obtenus.....	23
3.4 TP3 slider-gauge.....	25
3.4.1 Résultats attendus.....	25
3.4.2 Démonstration.....	25
3.4.2.1 Importation du flux slider-gauge.....	25
3.4.3 Résultats obtenus.....	25
3.5 TP4 http-hello.....	27
3.5.1 Résultats attendus.....	27
3.5.2 Démonstration.....	27
3.5.3 Résultats obtenus.....	32
3.6 TP5 ecoute-MQTT.....	32
3.6.1 Résultats attendus.....	32
3.6.2 Démonstration.....	33
3.6.3 Résultats obtenus.....	34
3.7 TP6 publish-MQTT.....	34
3.7.1 Résultats attendus.....	34

3.7.2	Démonstration.....	35
3.7.3	Résultats obtenus.....	38
3.8	TP7 temp&humidite-interieur.....	38
3.8.1	Résultats attendus.....	38
3.8.2	Démonstration.....	39
3.8.3	Résultats obtenus.....	41
3.9	TP8 bonjour.....	41
3.9.1	Résultats attendus.....	41
3.9.2	Démonstration.....	41
3.9.3	Résultats obtenus.....	42
4	TP Node red avec passerelle entre brokers MQTT public et privé.....	42
4.1	Configuration.....	42
4.2	TP1 Allumer Led Verte en local.....	44
4.2.1	Résultats attendus.....	44
4.2.2	Démonstration.....	44
4.2.3	Résultats obtenus.....	47
4.3	TP2 Allumer Led Verte depuis l'extérieur.....	47
4.3.1	Résultats attendus.....	47
4.3.2	Démonstration.....	47
4.3.3	Résultats obtenus.....	48
5	TP Sonnette connectée.....	48
5.1	Objectifs de ce TP.....	48
5.2	Configuration, scénario et résultats attendus.....	51
5.2.1	Étape 1 : Je suis à l'extérieur et quelqu'un sonne à ma porte d'entrée qui est verrouillée.....	51
5.2.2	Étape 2 : J'utilise la camera de mon domicile pour reconnaître qui a sonné.....	54
5.2.3	Étape 3 : je déverrouille à distance la porte d'entrée de mon domicile.....	54
5.2.4	Résultats obtenus.....	55
5.2.5	Application de ce TP à d'autres usages.....	55
5.2.6	Annexe : câblage de test.....	56
5.2.7	Annexe : Logiciel installé dans l'ESP32.....	56
5.2.7.1	Boot.py :.....	56
5.2.7.2	main.py :.....	58

Index des illustrations

Illustration 1: plateforme pour les TP broker mosquitto.....	7
Illustration 2: configuration MyMQTT sur chacun des smartphones : intérieur et extérieur.....	8
Illustration 3: publication depuis le pc linux situé en intérieur au topic "cig/temperature".....	9
Illustration 4: publication depuis un smartphone intérieur ou extérieur au topic "cig/temperature".....	9
Illustration 5: souscription (sub) sur le pc linux au topic "cig/temperature".....	10
Illustration 6: souscription (sub) sur le pc pi@raspberry aux topics de type "cig/#".....	10
Illustration 7: plateforme pour les TP Node red.....	11
Illustration 8: création d'un flux dans node-red.....	14
Illustration 9: importation d'un flux Node-Red.....	15
Illustration 10: suite et fin d'importation d'un flux Node-Red.....	22
Illustration 11: Flux météo.....	24
Illustration 12: affichage météo dans la page https://fred.sensetecnic.com/	24
Illustration 13: slider-gauge afficher dans la page: https://fred.sensetecnic.com/	26
Illustration 14: slider-gauge affiché dans le dashboard en page https://cig_91.fred.sensetecnic.com/api/ui/#/0	26
Illustration 15: flux hello-world.....	30
Illustration 16: http-hello, configuration du node page.....	31
Illustration 17: résultat hello-world.....	32
Illustration 18: dashboard ecoute-MQTT.....	33
Illustration 19: ecoute-MQTT.....	34
Illustration 20: publish-MQTT.....	35
Illustration 21: dashboard publish-MQTT.....	36
Illustration 22: publish-MQTT-écran 'debug'.....	38
Illustration 23: flux temp&humidite-interieur.....	39
Illustration 24: dashboard 1 temp&humidite-interieur.....	40
Illustration 25: dashboard 2 temp&humidite-interieur.....	40
Illustration 26: flux bonjour.....	41
Illustration 27: plateforme pour les TP Node red avec passerelle entre broker MQTT mosquitto.....	42
Illustration 28: rôle de la passerelle MQTT.....	43
Illustration 29: flux raspberrry.....	46
Illustration 30: Commande Led Verte depuis la zone privée.....	46
Illustration 31: flux cde raspberrry sur le Node-Red public : FRED.....	47
Illustration 32: Commande Led Verte depuis l'extérieur.....	48
Illustration 33: Quelqu'un sonne à mon domicile !.....	49
Illustration 34: étape 3: Je déverrouille la porte.....	50
Illustration 35: Quelqu'un sonne à ma porte de mon domicile !.....	51
Illustration 36: Application Node Red sur smart phone.....	52
Illustration 37: Flux node-red installé dans l'application NodeRed Mobile.....	53
Illustration 38: je deverrouille ma porte à distance.....	54
Illustration 39: câblage de test.....	56

1 Objectifs de ces TP

Les Travaux Pratiques développés ci-après consistent à découvrir comment mettre en place des services dans le monde connecté .

2 TP broker mosquito

2.1 Configuration

La plateforme utilisée est présentée ci-dessous.

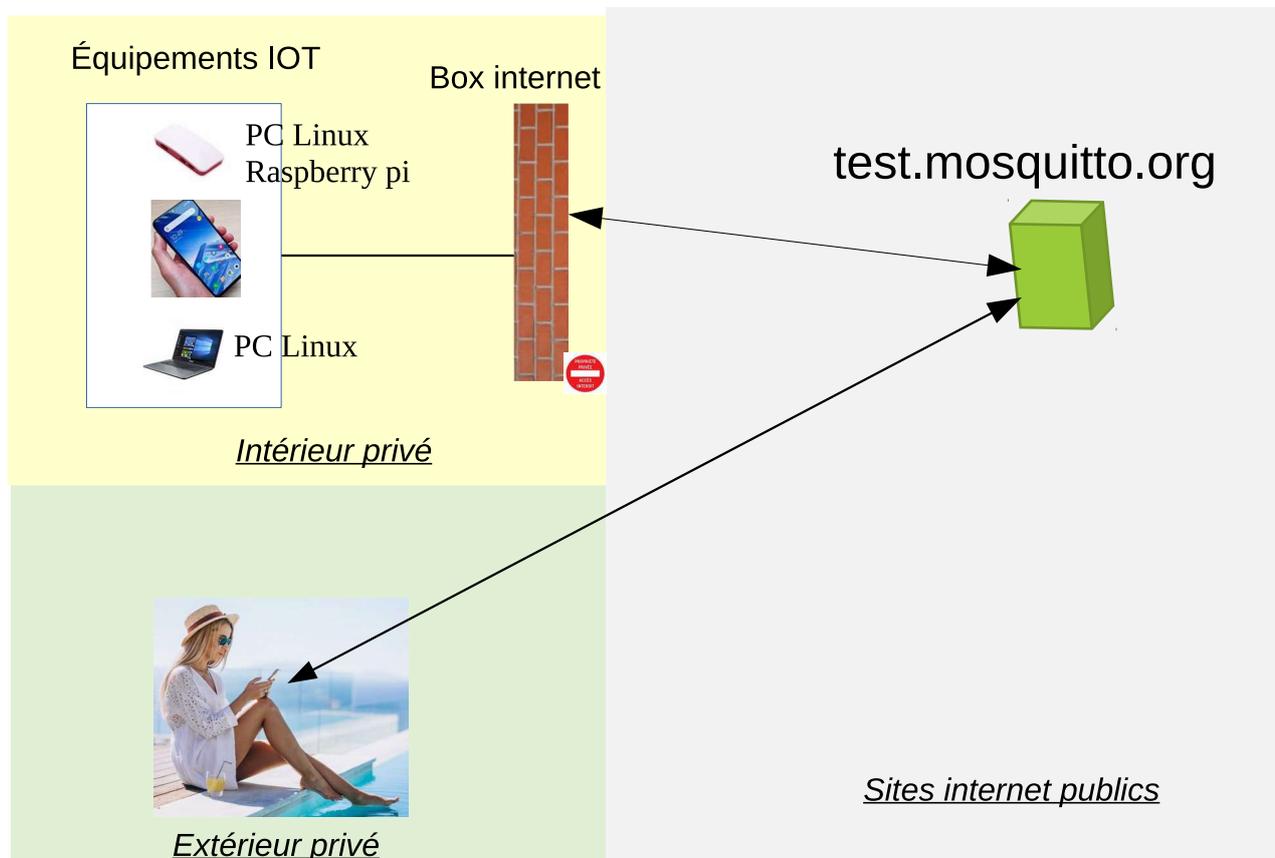


Illustration 1: plateforme pour les TP broker mosquitto

2.1.1 Le broker

Le broker utilisé est le site internet : test.mosquitto.org

2.1.2 les clients

Les clients du broker connecté en intérieur d'un local en wifi à une box internet :

- les terminaux hébergés dans le PC Linux
- les terminaux hébergés dans le PC Linux raspberry pi
- l'application MyMQTT hébergée dans un smart phone android

Le client du broker connecté en extérieur sur le réseau 4G constitué de :

- l'application MyMQTT hébergée dans un smart phone android

2.2 Résultats attendus

Rappel rapide de MQTT: [ici](#)

Les résultats attendus consistent à vérifier que chaque client peut publier (pub) ses données sur des 'topics' et que les autres clients qui souscrivent (sub) à ces topics, reçoivent bien ces données.

2.3 Démonstrations

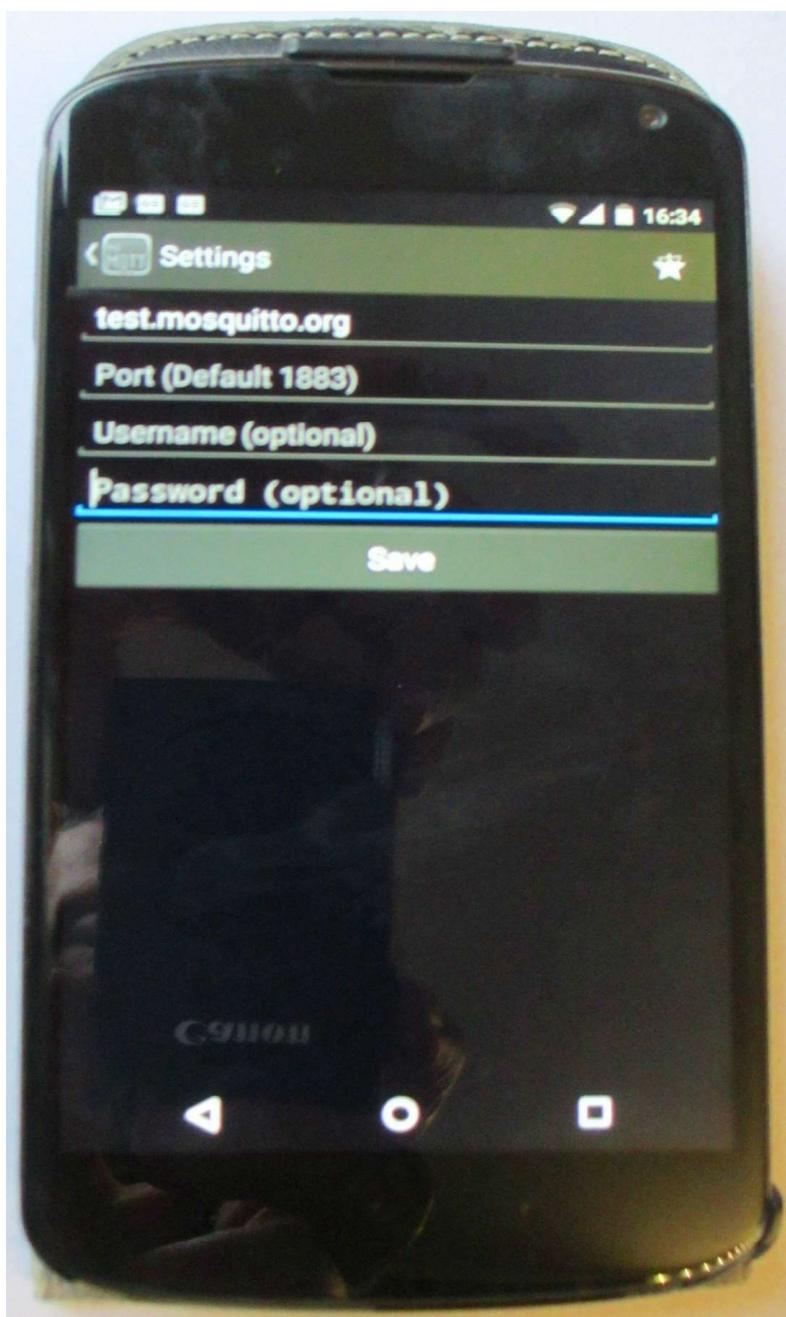
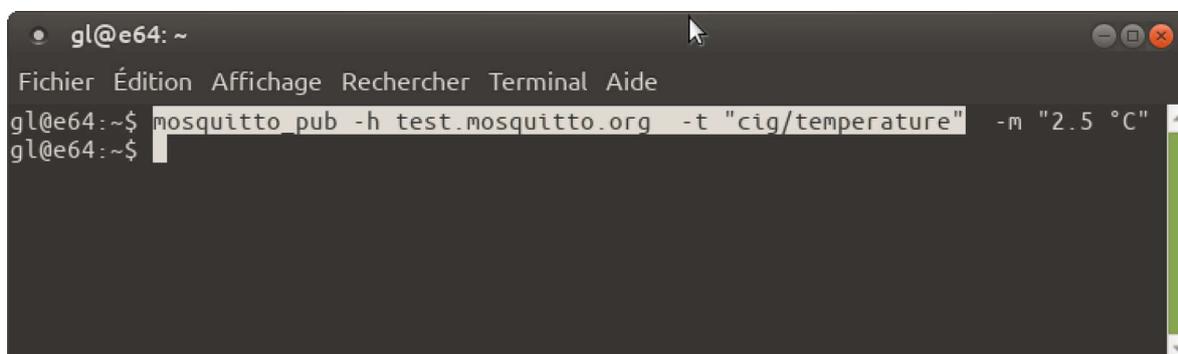


Illustration 2: configuration MyMQTT sur chacun des smartphones : intérieur et extérieur

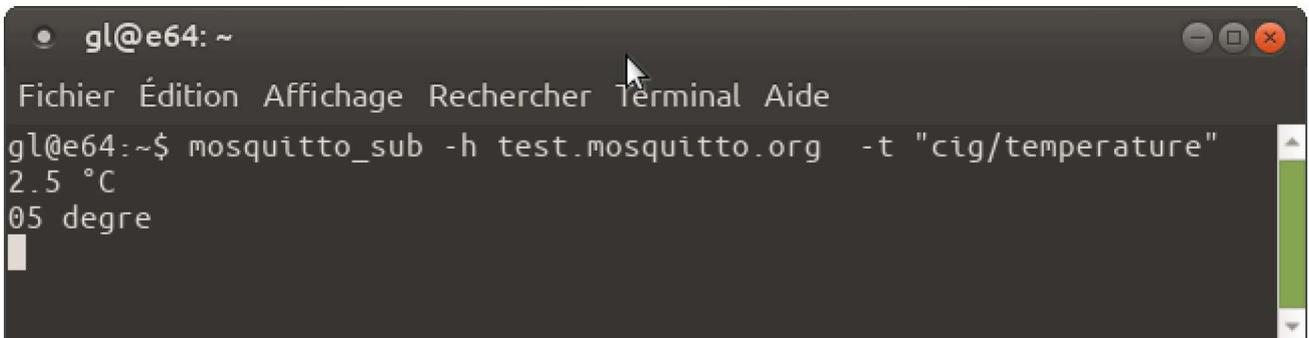


```
gl@e64: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
gl@e64:~$ mosquitto_pub -h test.mosquitto.org -t "cig/temperature" -m "2.5 °C"  
gl@e64:~$
```

Illustration 3: publication depuis le pc linux situé en intérieur au topic "cig/temperature"

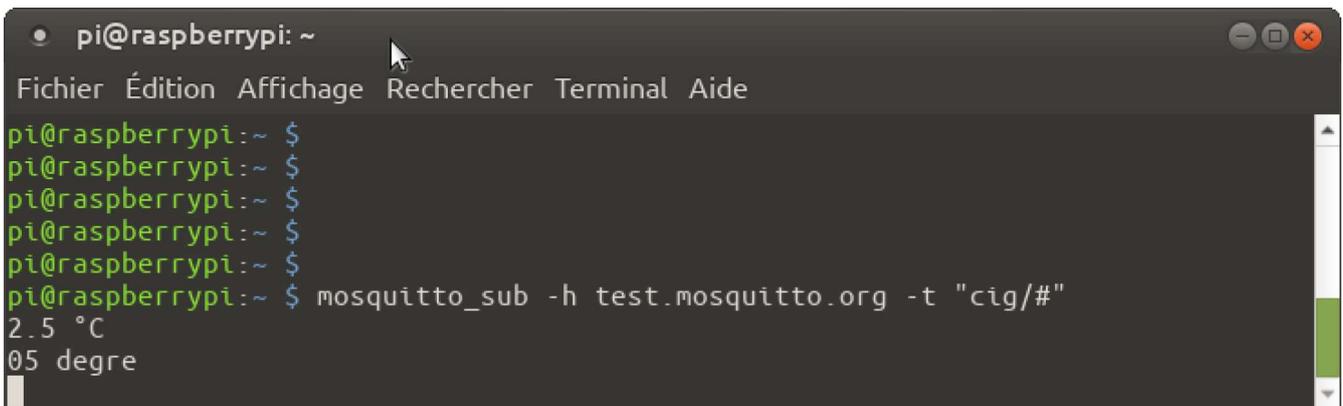


Illustration 4: publication depuis un smartphone intérieur ou extérieur au topic "cig/temperature"



```
gl@e64: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
gl@e64:~$ mosquitto_sub -h test.mosquitto.org -t "cig/temperature"  
2.5 °C  
05 degre
```

Illustration 5: souscription (sub) sur le pc linux au topic "cig/temperature"



```
pi@raspberrypi: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $ mosquitto_sub -h test.mosquitto.org -t "cig/#"  
2.5 °C  
05 degre
```

Illustration 6: souscription (sub) sur le pc pi@raspberry aux topics de type "cig/#"

2.4 Résultats obtenus

Les résultats obtenus sont conformes aux résultats attendus.

3 TP Node red

3.1 Configuration

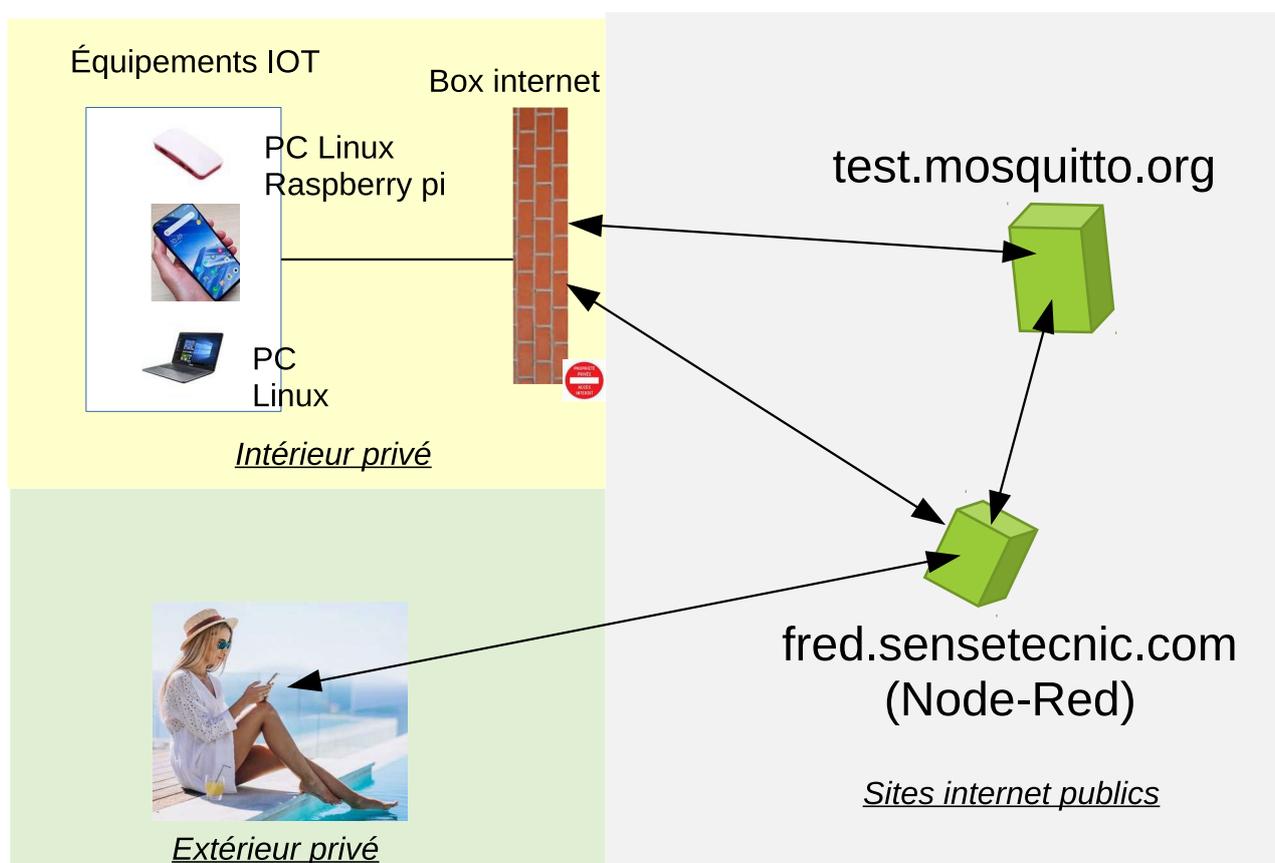


Illustration 7: plateforme pour les TP Node red

La configuration est identique à celle décrite au [chapitre précédent](#) à laquelle est ajouté le site web public : <https://fred.sensetecnic.com/>

3.1.1 Connexion au site Node-Red public : FRED

Les paramètres permettant de se connecter au site node-red: <https://fred.sensetecnic.com/> qui est utilisé dans le cadre de ces travaux pratiques sont :

cliquer sur LOGIN

entrer:

Username

cig_91

Password

cig_91

cliquer sur Sign in

3.1.2 Quelques liens utiles pour découvrir Node-Red

Les liens ci-dessous présentent un ensemble de documents permettant de découvrir et maîtriser l'outil Node-Red :

En Français :

Tutoriel Node-red: <http://silanus.fr/sin/?p=984>

<https://projetsdiy.fr/debuter-arduino-node-red-enregistrer-afficher-mesures-json-port-serie/>

Vidéo you-tube

création d'un Dashboard : <https://youtu.be/N-Cko9s19uQ>

En Anglais :

<http://noderedguide.com/nr-lecture-1/>

<http://noderedguide.com/node-red-lecture-2-building-your-first-flows-15/>

<http://developers.sensetecnic.com/article/introduction-to-node-red/>

<http://developers.sensetecnic.com/guides-and-examples/>

<http://noderedguide.com/tag/fred/>

<https://cookbook.nodered.org/>

<http://developers.sensetecnic.com/article/tutorial-using-fred-cloud-node-red-to-build-an-ai-chatbot-using-ibm-watson/>

3.2 TP1 construire son premier flux

Ref1 <http://noderedguide.com/node-red-lecture-2-building-your-first-flows-15/1>

Ref2: (code): <https://github.com/SenseTecnic/nrguideflows/blob/master/lesson2/2-1-firstflow.json>

Ref3:(ihm attendue) : <http://silanus.fr/sin/?p=984>

3.2.1 Résultats attendus

Les objectifs de ce TP sont les suivants :

- découvrir l'Interface Homme Machine (IHM) de node-red : les nodes, les flows, les onglets : info, debug, dashboard
- découvrir comment créer un flux
- découvrir comment tester un flux (flow)
- découvrir comment enregistrer ou télécharger un flux (flow en Anglais)

3.2.2 Démonstration

3.2.2.1 Découvrir l'Interface Homme Machine (IHM) de node-red

Cliquer sur <https://fred.sensetecnic.com/> pour obtenir dans votre navigateur l'IHM de Node-Red.

Vérifier que l'IHM obtenue correspond bien à la description faite au lien : <http://silanus.fr/sin/?p=984>

3.2.2.2 Découvrir comment créer un flux

Pour construire votre premier flux, Suivez le tuto présenté en [Ref1](#) chapitre : 'Example 2.1 Building your first flow'.

La figure ci-dessous présente l'image du flux construit dans l'IHM de node-red

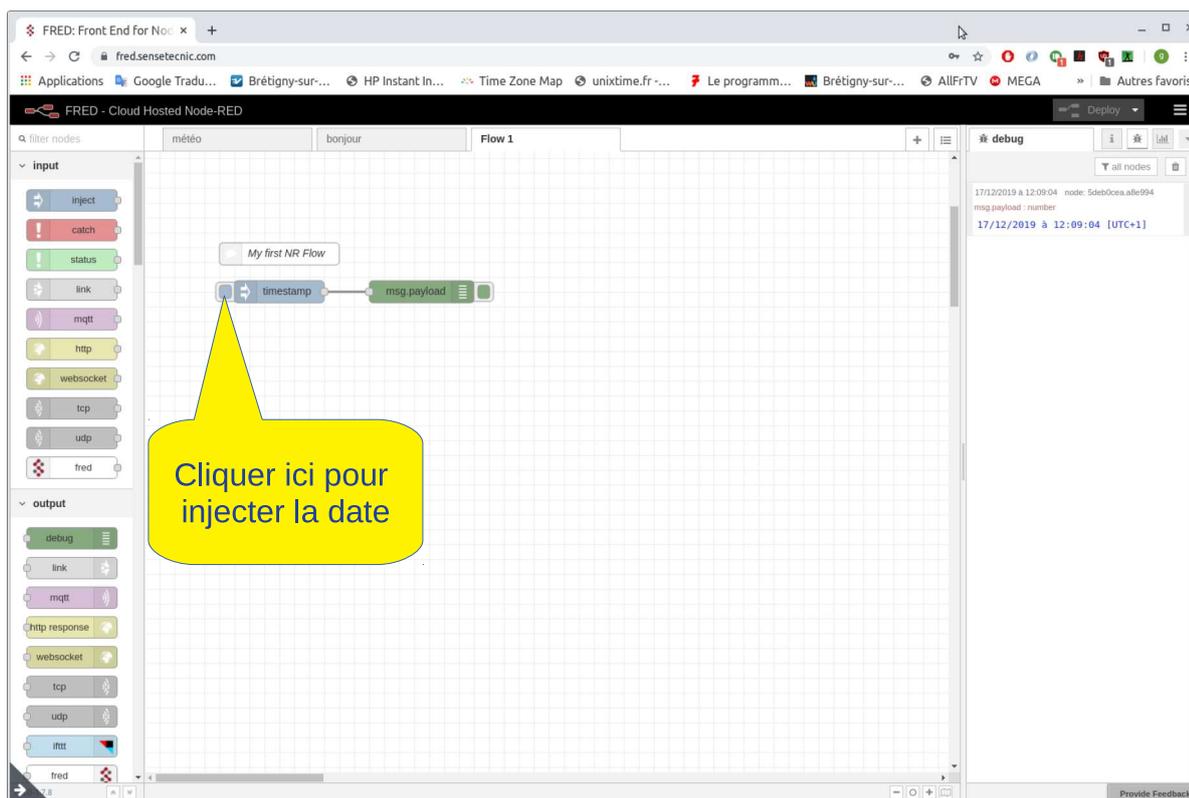


Illustration 8: création d'un flux dans node-red

3.2.2.3 Découvrir comment tester un flux (flow)

Pour tester le bon fonctionnement, cliquer sur le node de type 'inject', intitulé 'timestamp'. Une nouvelle date s'affiche alors dans l'onglet 'debug'.

3.2.2.4 Découvrir comment enregistrer ou télécharger un flux

Se reporter à : [Ref1](#) chapitre : 'Other useful UI features' pour une présentation détaillée.

Les figures ci-dessous présentent comment importer un flux.

L'exportation d'un flux est similaire à l'importation.

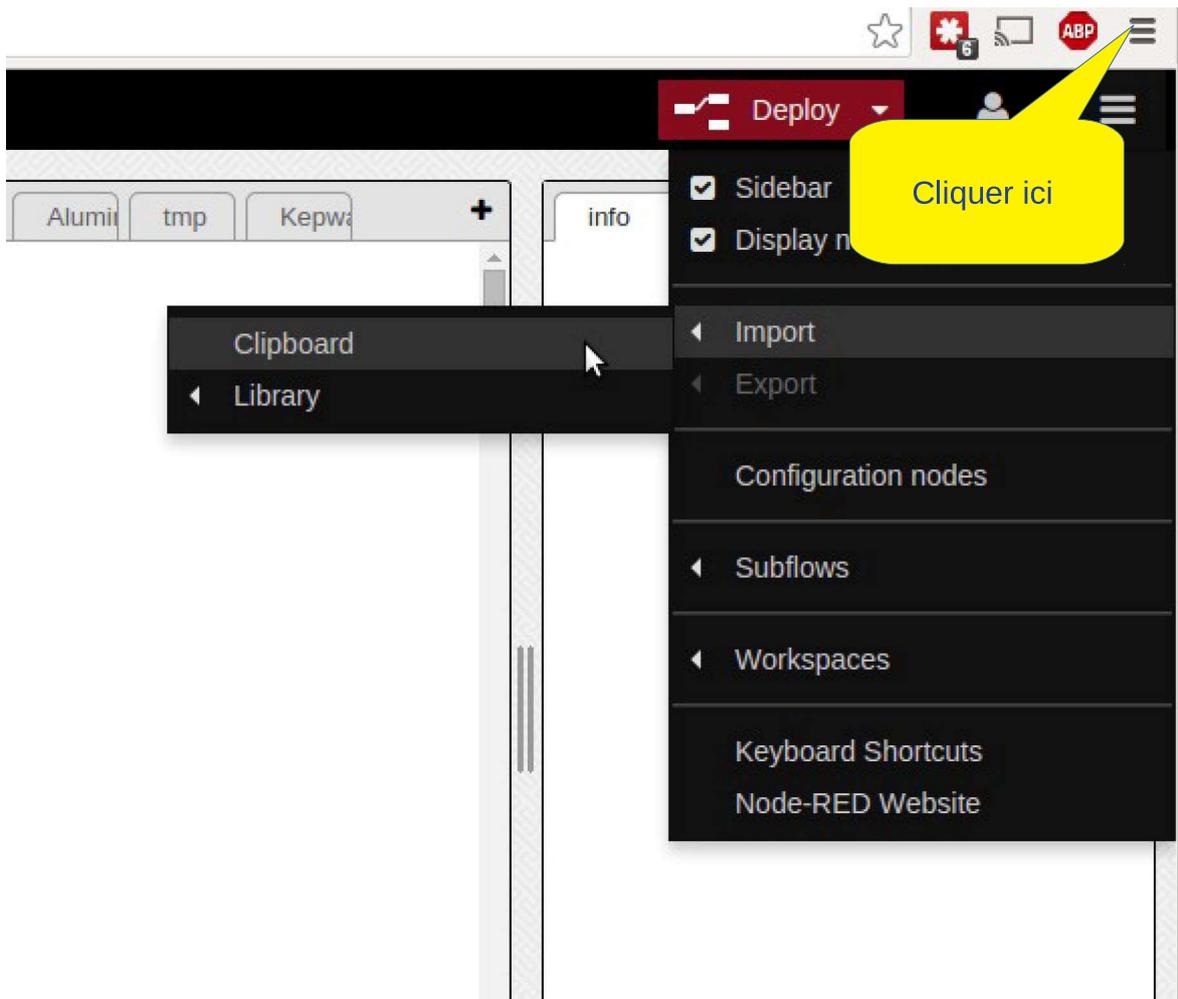


Illustration 9: importation d'un flux Node-Red

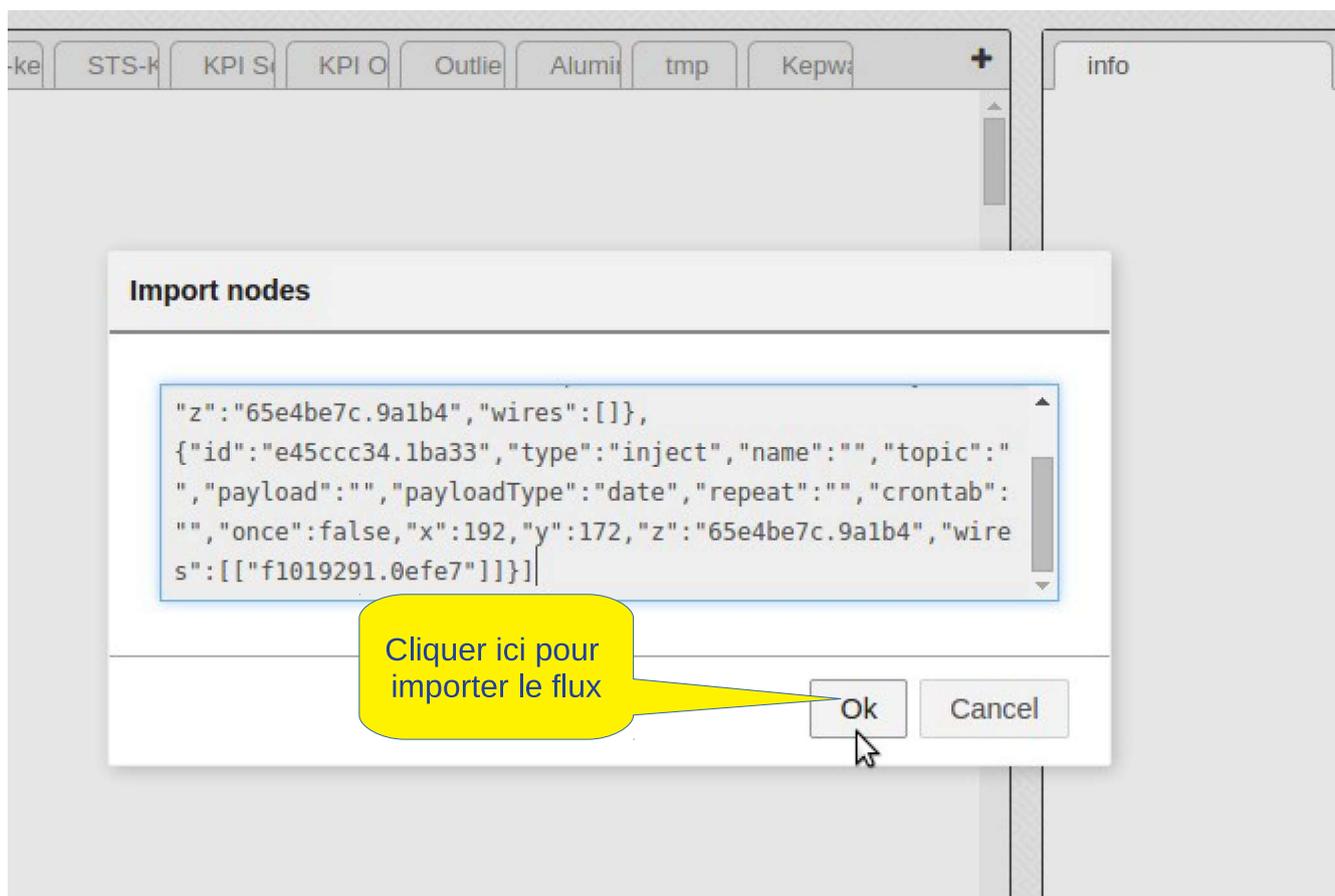


Illustration 10: suite et fin d'importation d'un flux Node-Red

3.2.3 Résultats obtenus

Les résultats obtenus sont conformes aux résultats attendus.

3.3 TP2 météo

3.3.1 Résultats attendus

Les objectifs de ce TP sont les suivants :

- importer un flux intitulé 'météo'
- vérifier son bon fonctionnement

3.3.2 Démonstration

3.3.2.1 Importation du flux météo

Importer dans l'ihm de Node-Red le contenu du fichier '**météo.json**' contenu dans le dossier '[exemple-flux-sur-FRED](#)'

Vérifier que les informations météo sont bien reçues.

3.3.3 Résultats obtenus

Les résultats obtenus sont conformes aux résultats attendus.

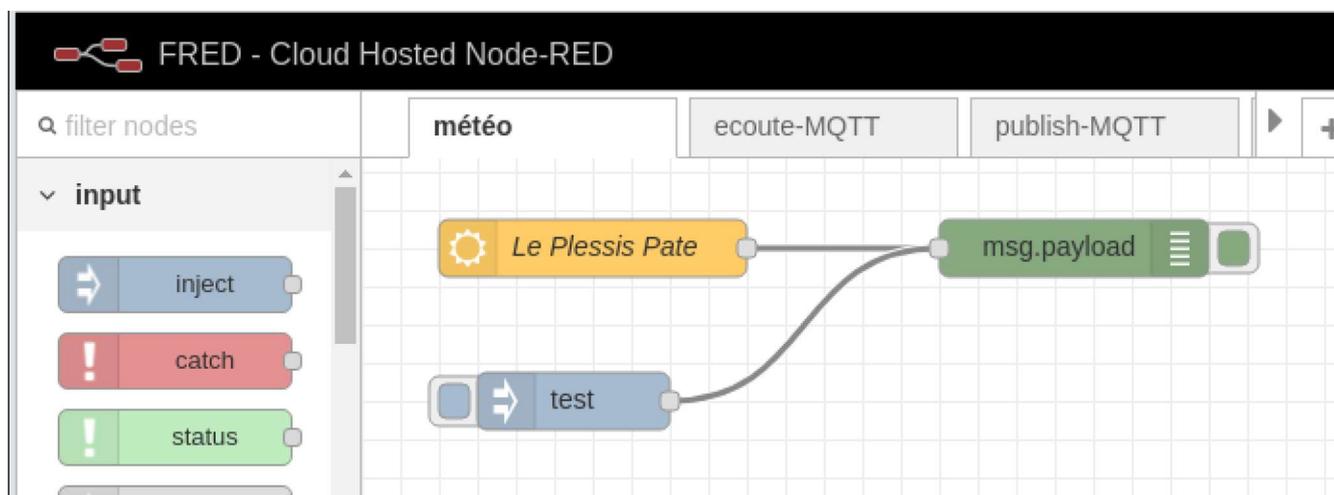


Illustration 11: Flux météo

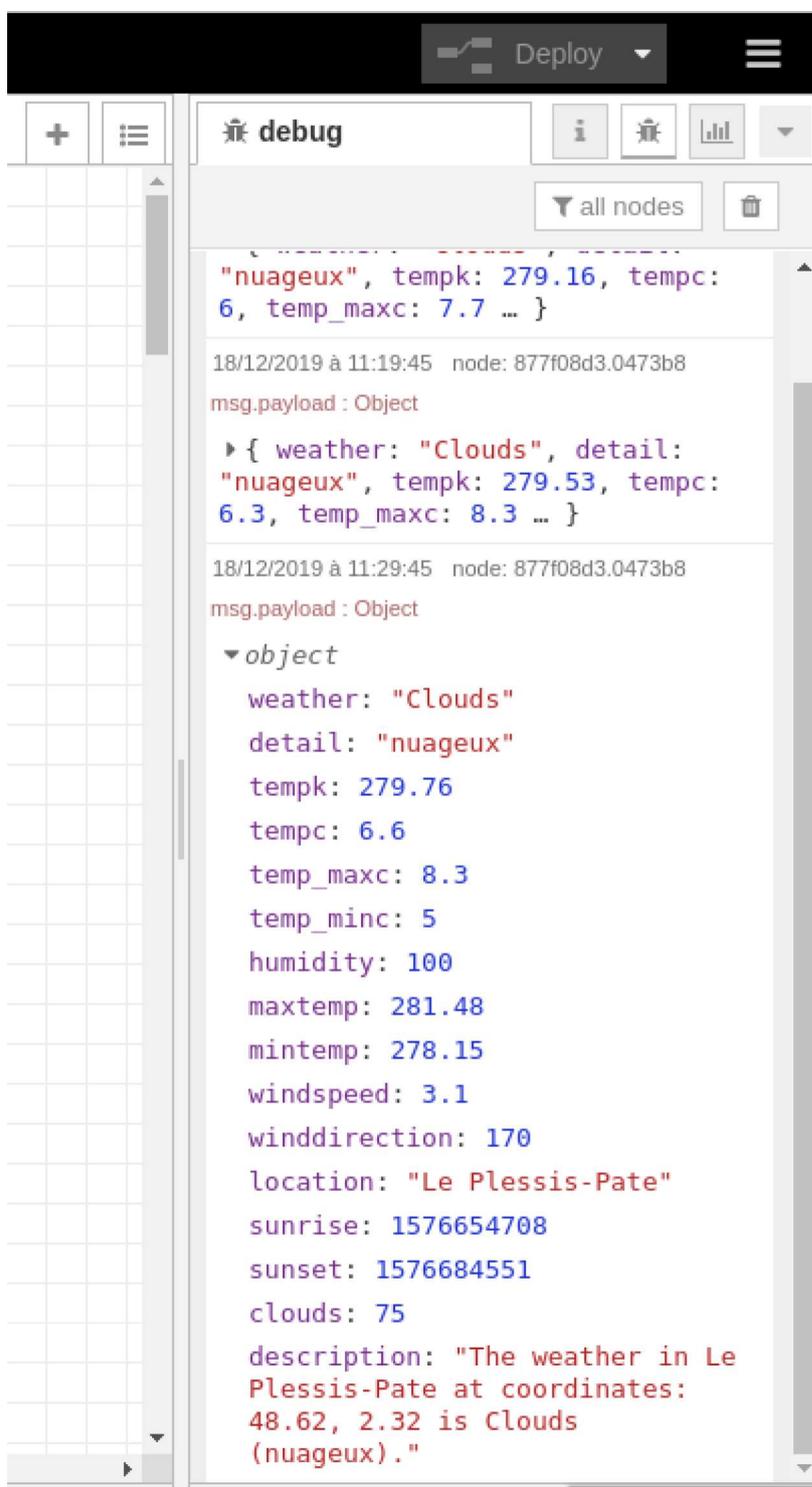


Illustration 12: affichage météo dans la page <https://fred.sensetecnic.com/>

3.4 TP3 slider-gauge

3.4.1 Résultats attendus

Les objectifs de ce TP sont les suivants :

- importer un flux intitulé 'slider-gauge'
- vérifier son bon fonctionnement sur les pages : <https://fred.sensetecnic.com/> et https://cig_91.fred.sensetecnic.com/api/ui/#/0

3.4.2 Démonstration

3.4.2.1 Importation du flux slider-gauge

Importer dans l'ihm de Node-Red le contenu du fichier '**slider-gauge.json**' contenu dans le dossier '[exemple-flux-sur-FRED](#)'

Vérifier le bon fonctionnement. En particulier, vérifier l'affichage du dashboard sur son smartphone.

3.4.3 Résultats obtenus

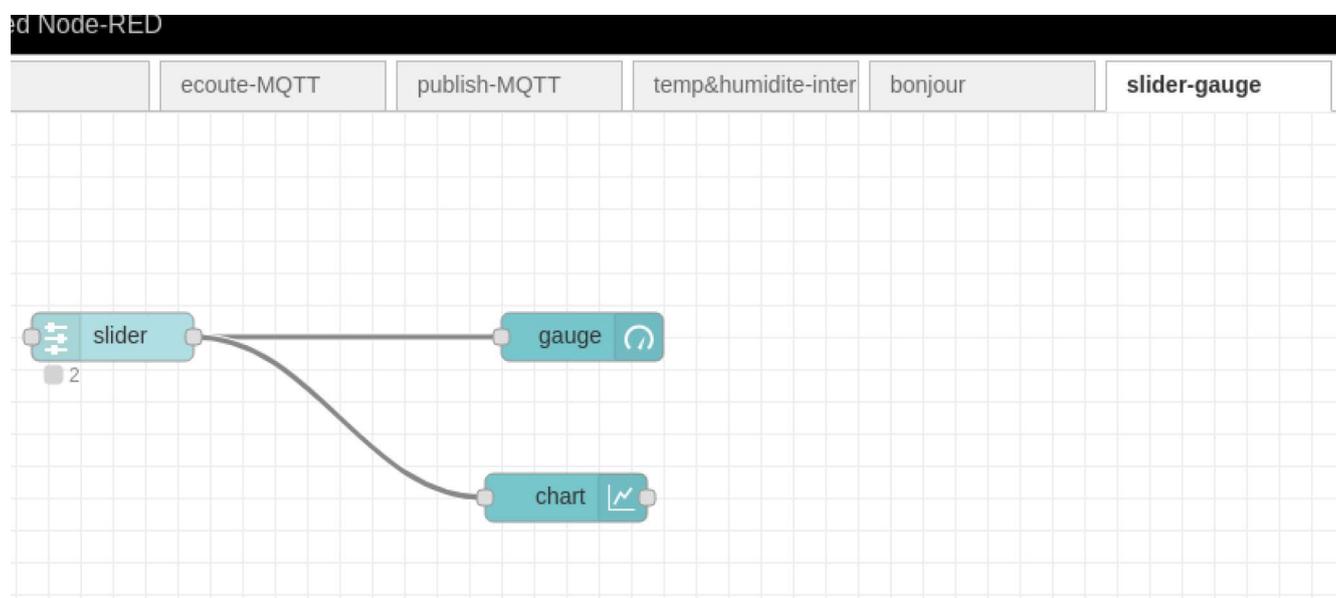


Illustration 13: slider-gauge afficher dans la page: <https://fred.sensetecnic.com/>

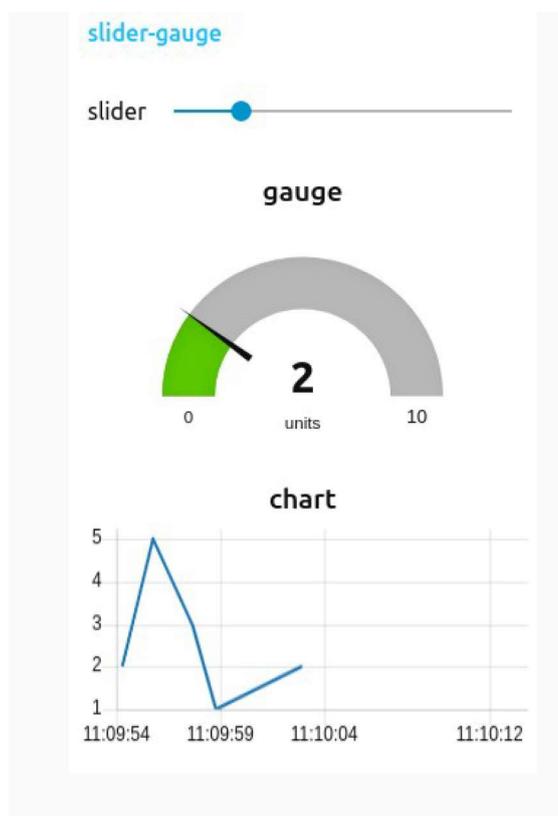


Illustration 14: slider-gauge affiché dans le dashboard en page https://cig_91.fred.sensetecnic.com/api/ui/#/0

3.5 TP4 http-hello

3.5.1 Résultats attendus

L'objectif de ce TP est de démontrer comment développer un mini serveur web avec node-red.

3.5.2 Démonstration

- importer le flux intitulé 'http-hello'

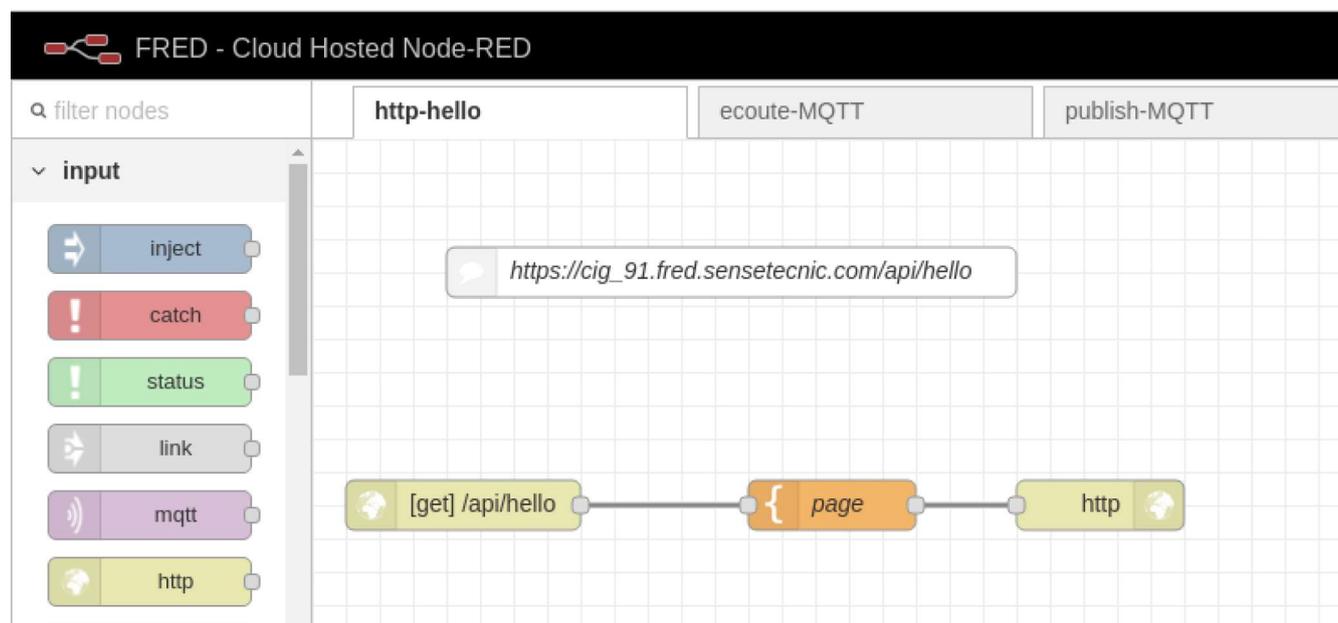


Illustration 15: flux hello-world

Edit template node

Delete Cancel Done

Properties

Name: page

Property: msg.payload

Format: Mustache template

Template Syntax Highlight: mustache

```

1 <html>
2   <head></head>
3   <body>
4     <h1>Hello World!</h1>
5     <h1>Comment ça va bien ?</h1>
6   </body>
7 </html>

```

info

Information

Node	"4a1a4778.bdb7e8"
Name	page
Type	template

show more

Description

Node Help

Sets a property based on the provided template.

Inputs

msg *object*
A msg object containing information to populate the template.

template *string*
A template to be populated from msg.payload. If not configured in the edit panel, this can be set as a property of msg.

Illustration 16: http-hello, configuration du node page

- vérifier son bon fonctionnement en cliquant sur le lien :
https://cig_91.fred.sensetecnic.com/api/hello

3.5.3 Résultats obtenus



Illustration 17: résultat hello-world

3.6 TP5 ecoute-MQTT

3.6.1 Résultats attendus

L'objectif de ce TP est de constituer avec l'outil node-red un client MQTT qui écoute tous les topics dont le nom commence par 'cig/'.

Par exemple,

si depuis l'application MyMqtt sur android, on publie une valeur de température , par exemple 27 degré sur le topic : cig/temperature

alors

cette valeur de température apparaît sur le dashboard . Voir ci-dessous l’affichage attendu.

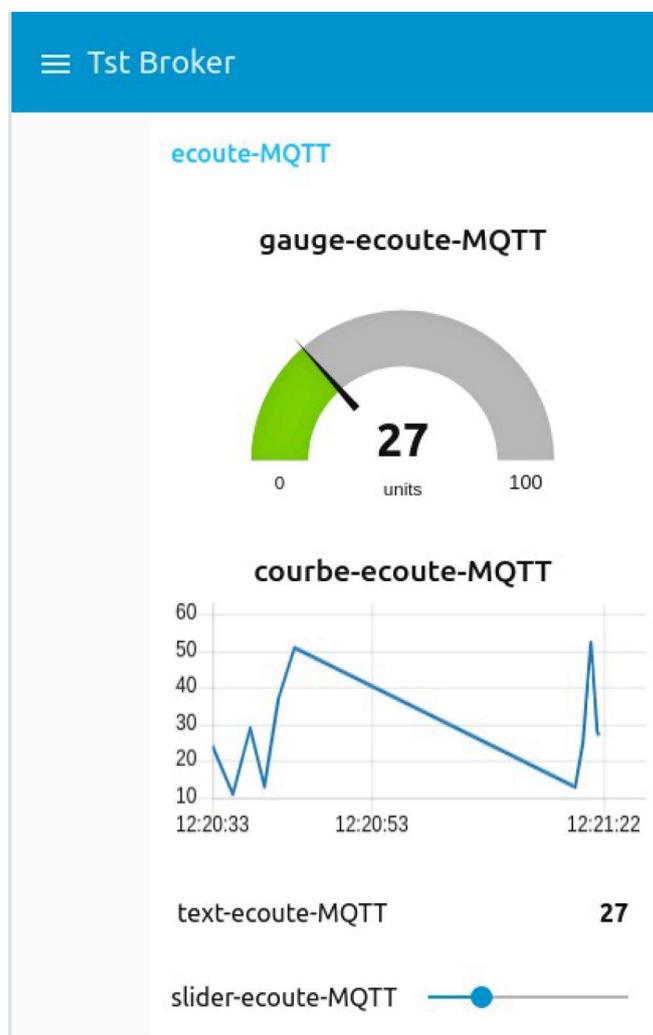


Illustration 18: dashboard ecoute-MQTT

3.6.2 Démonstration

- importer le flux intitulé 'ecoute-MQTT' dans node-red puis,
- publier une valeur de temperature sur le topic cig/temperature en utilisant l'application android MyMQTT.

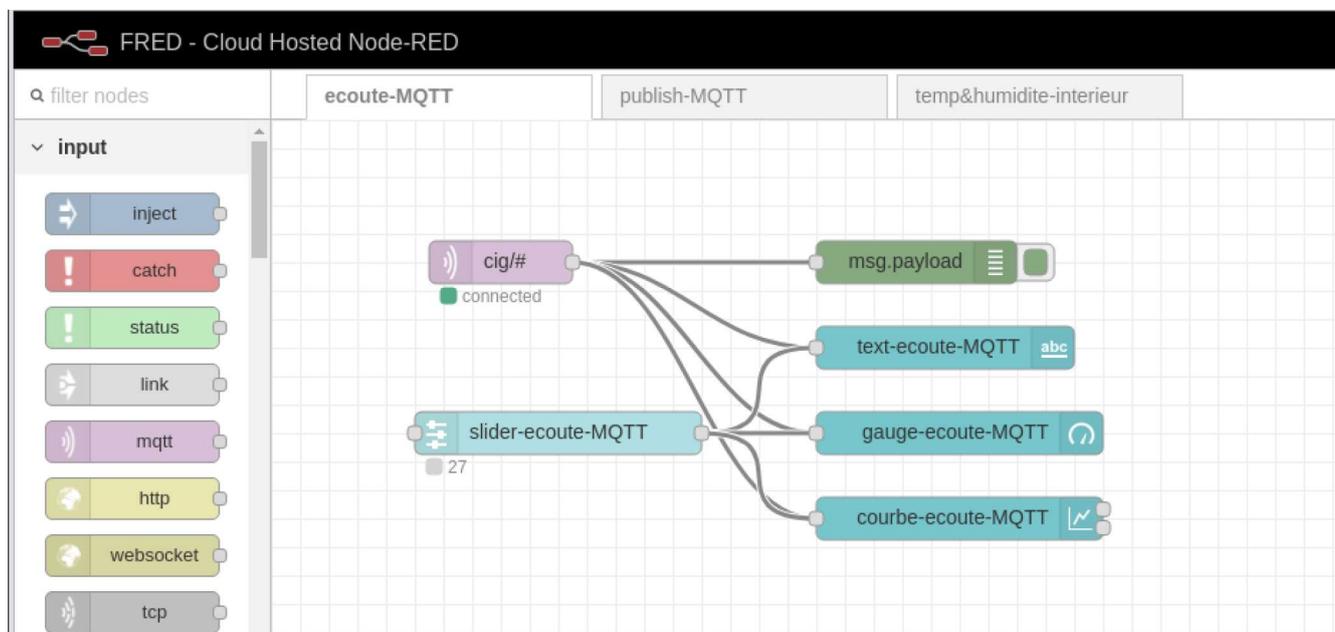


Illustration 19: ecoute-MQTT

3.6.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

3.7 TP6 publish-MQTT

3.7.1 Résultats attendus

L'objectif de ce TP est de constituer, avec l'outil node-red, deux clients MQTT :

- un client qui publie sur le topic cig/temperaure
- un client qui publie sur le topic cig/humidite

Ces deux clients simulent par exemple un objet connecté qui diffuse les deux informations : température et humidité vers d'autres objets connectés au brocker MQTT.

Ces clients nous permettrons d'injecter une valeur de température ou d'humidité en accédant au dashboard associé au flux publish-MQTT à partir d'un navigateur sur smartphone ou sur PC par exemple.

3.7.2 Démonstration

Le flux mise en œuvre dans node red est présenté ci-dessous.

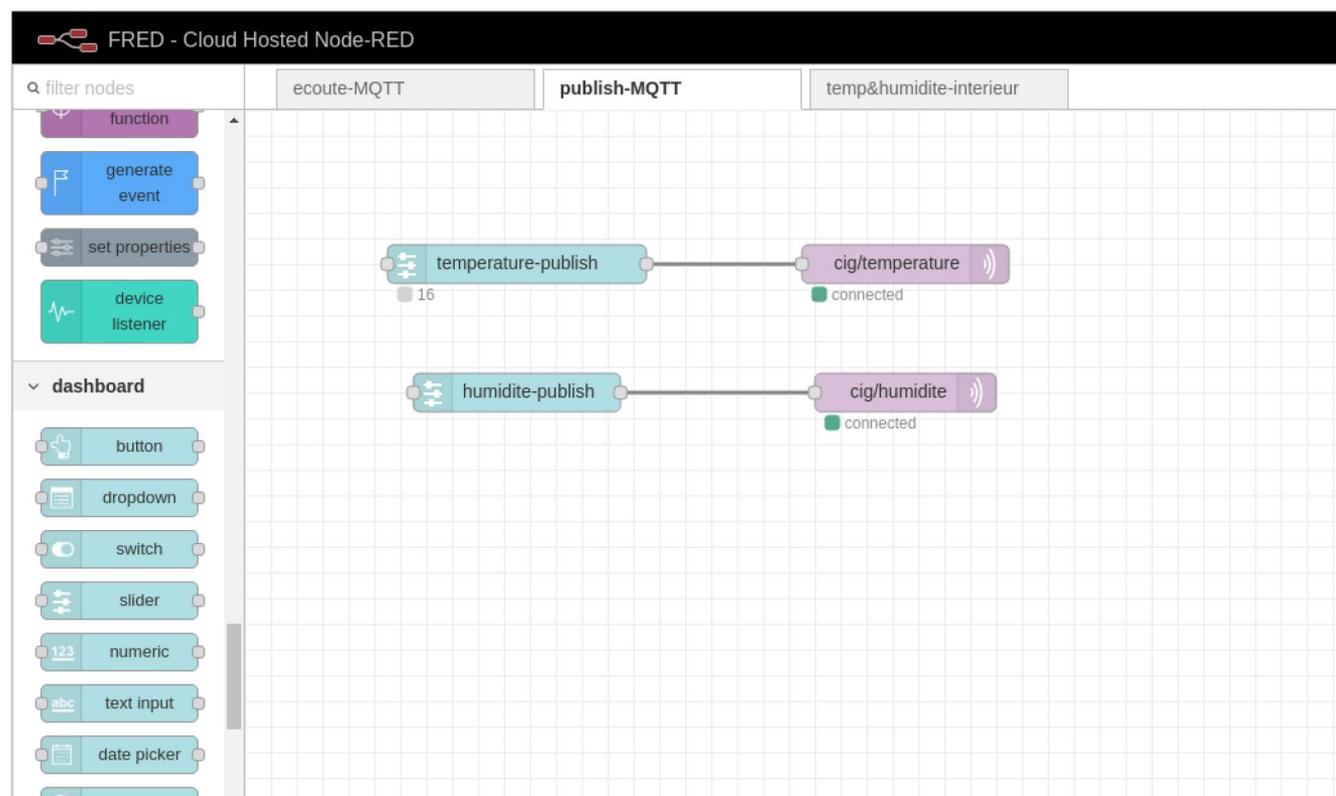


Illustration 20: publish-MQTT

Cliquer sur le lien : https://cig_91.fred.sensetecnic.com/api/ui/#/0 pour obtenir l'écran présenté ci-dessous.

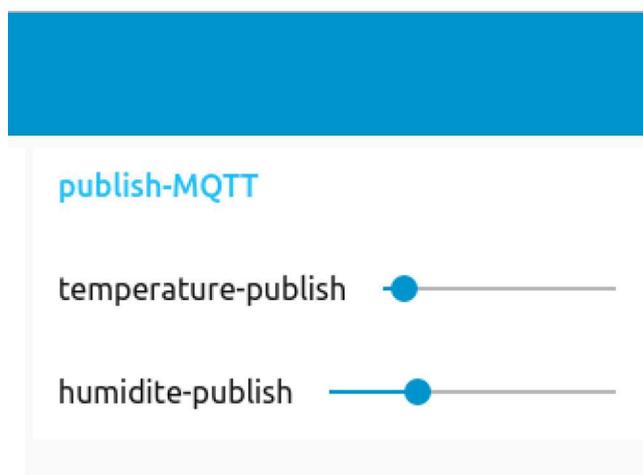


Illustration 21: dashboard publish-MQTT

Visualiser dans l'écran 'debug' situé au lien : <https://fred.sensetecnic.com/> les valeurs que vous publiez en utilisant les sliders 'temperature-publish' et 'humidité-publish'.

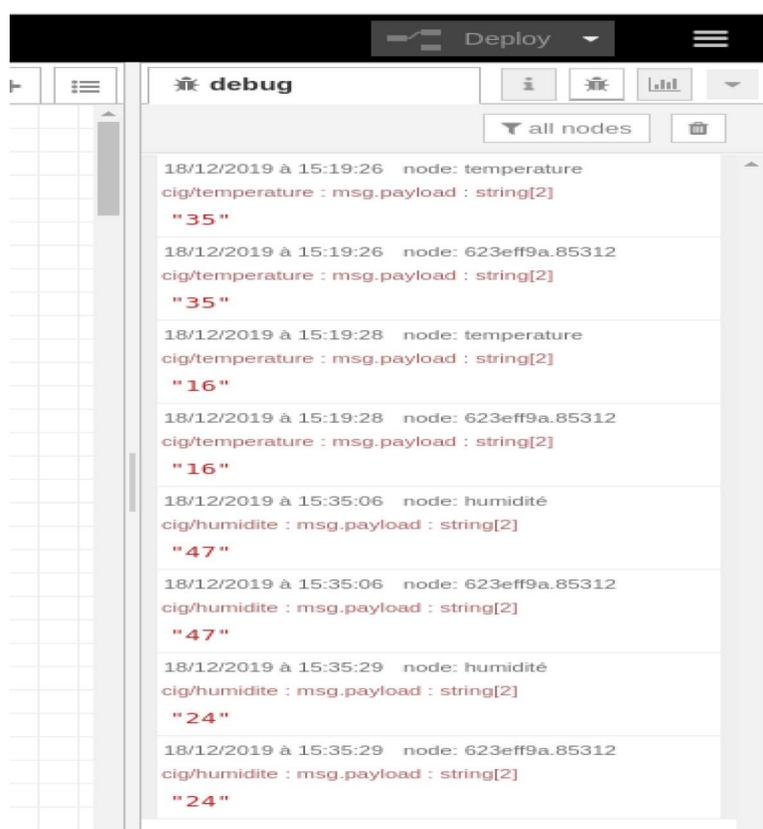


Illustration 22: publish-MQTT-écran 'debug'

3.7.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

3.8 TP7 temp&humidite-interieur

3.8.1 Résultats attendus

L'objectif de ce TP est de constituer, avec l'outil node-red, deux clients MQTT :

- un client qui souscrit sur le topic cig/temperaure

- un client qui souscrit sur le topic cig/humidite

Ces deux clients simulent par exemple un objet connecté qui écoute les deux informations : température et humidité en sortie du broker MQTT.

Ces clients nous permettront de visualiser les valeurs de température ou d'humidité en accédant aux dashboards associés à ces flux à partir d'un navigateur sur smartphone ou sur PC par exemple. Le lien est : https://cig_91.fred.sensetecnic.com/api/ui/#/0

3.8.2 Démonstration

Le flux mise en œuvre dans node red est présenté ci-dessous.

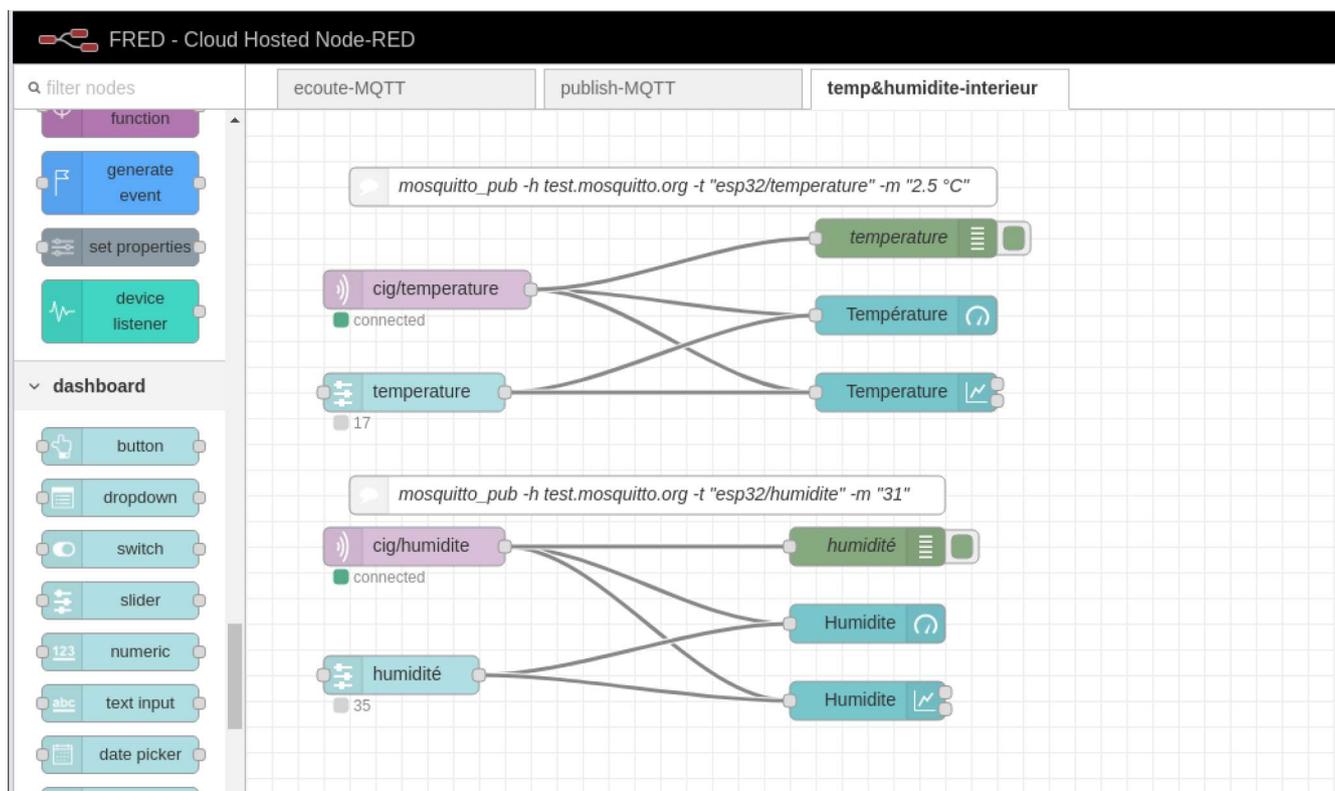


Illustration 23: flux temp&humidite-interieur

Cliquer sur le lien : https://cig_91.fred.sensetecnic.com/api/ui/#/0 pour obtenir les écrans présentés ci-dessous.

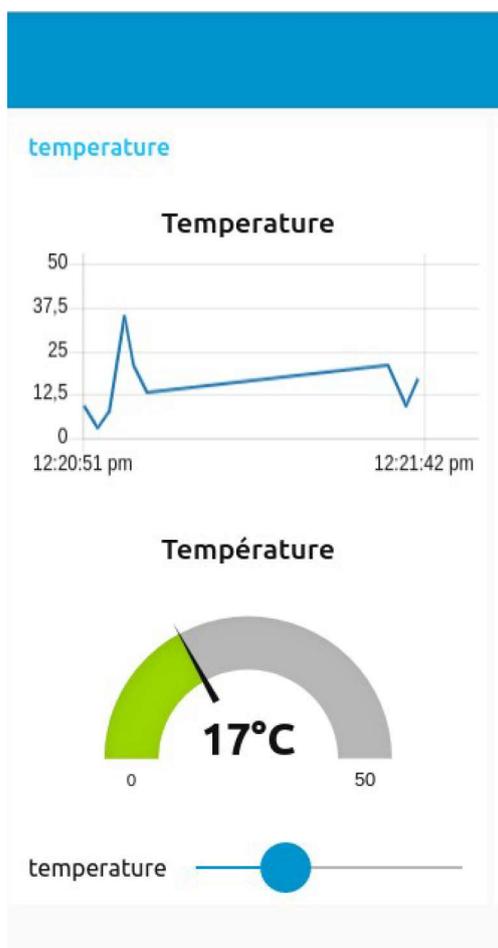


Illustration 24: dashboard 1
temp&humidite-interieur

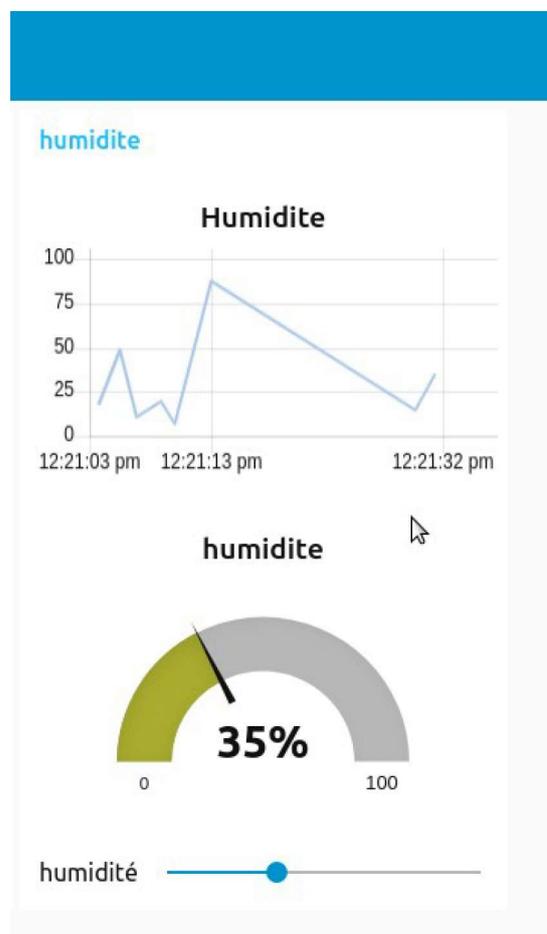


Illustration 25: dashboard 2
temp&humidite-interieur

3.8.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

3.9 TP8 bonjour

3.9.1 Résultats attendus

L'objectif de ce TP est d'utiliser le node 'audio-out' afin de démontrer les possibilités offerte par l'outil node-red.

Lorsqu'on injectera le texte « bonjour » dans le flux node-red, nous entendrons le mot « bonjour » en sortie du haut parleur de notre PC ou smartphone

3.9.2 Démonstration

Mettre en place le flux 'bonjour' en le téléchargeant selon les explications [ici](#).

On obtient le flux sur le site <https://fred.sensetecnic.com/> selon la figure ci-dessous.

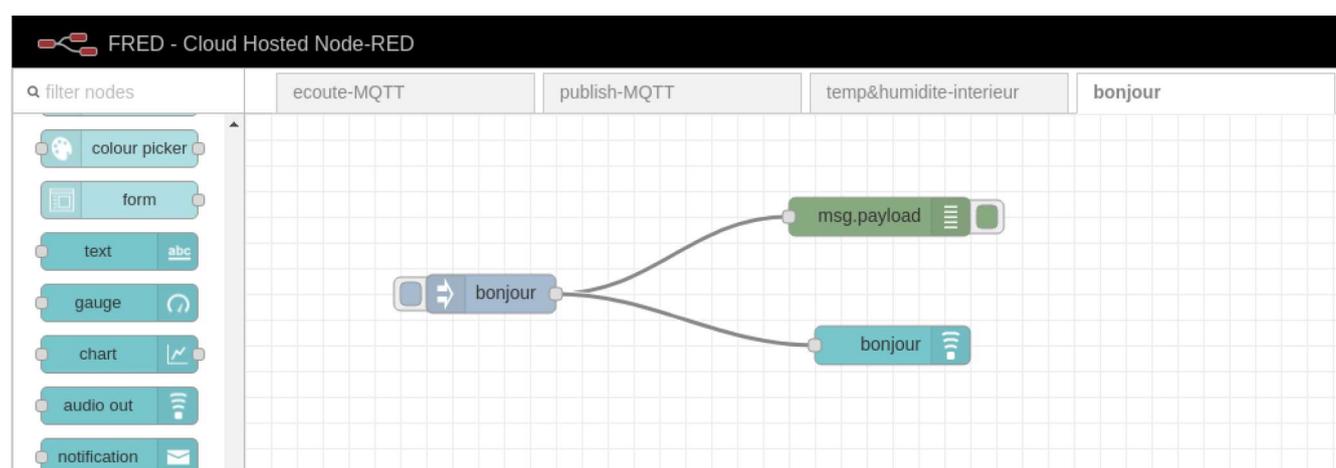


Illustration 26: flux bonjour

Cliquer sur l'entrée du node bonjour. Verifier que le son 'bonjour' sort bien sur le haut parleur

3.9.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

4 TP Node red avec passerelle entre brokers MQTT public et privé

4.1 Configuration

La plateforme utilisée dans ces TP est présentée ci-dessous.

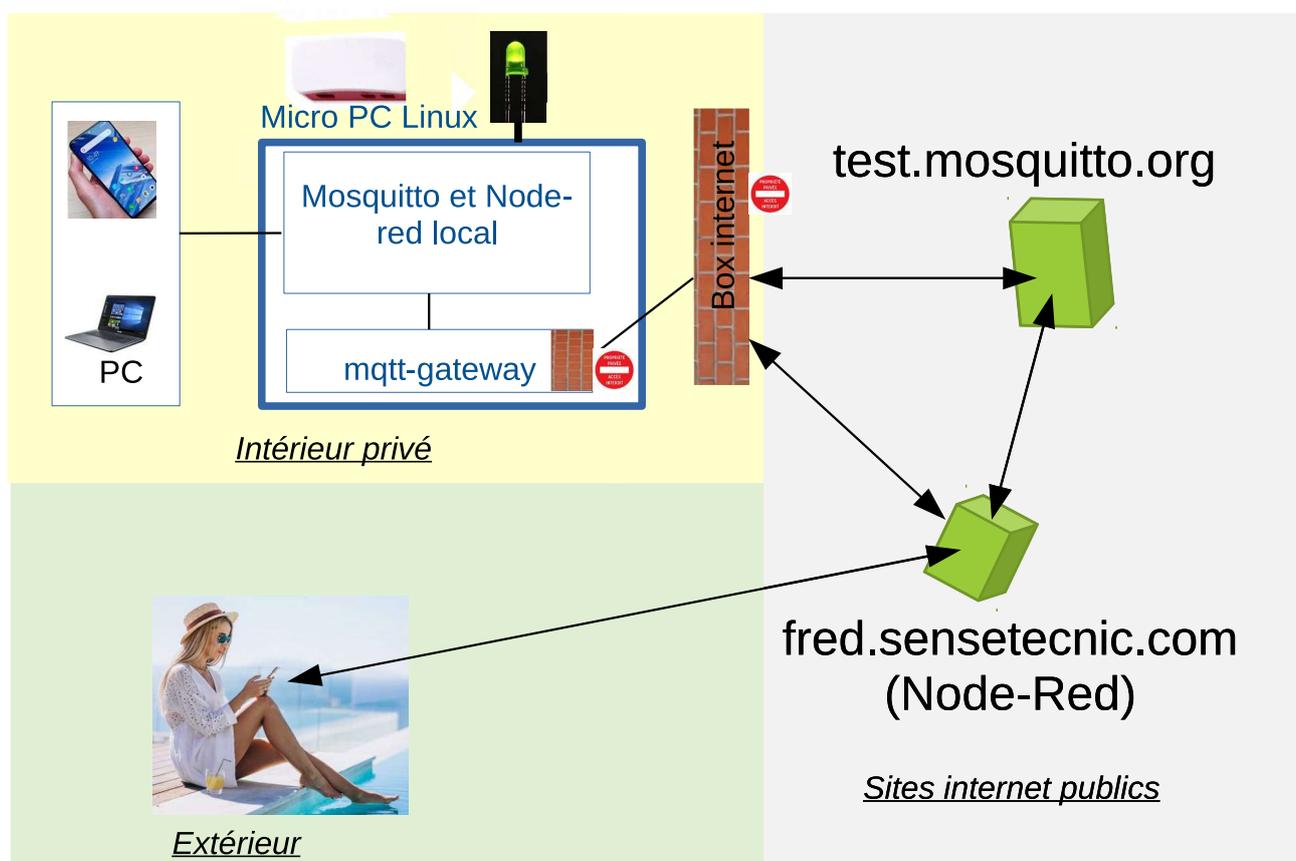


Illustration 27: plateforme pour les TP Node red avec passerelle entre broker MQTT mosquitto

Cette configuration est dérivée de celle du [chapitre précédent](#).

Le micro PC Linux “raspberry pi” héberge un ensemble “Mosquitto et Node-Red local” et une passerelle “mqtt-gateway”. Il est muni d’une “Led Verte” qui représente , par exemple , une commande de volet roulant.

La passerelle “mqtt-gateway” permet la communication entre le broker MQTT local (Mosquitto local) et le broker MQTT public (test.mosquitto.org). Voir figure ci-dessous.

Cette passerelle est à la fois client du broker public et client du broker privé. Ceci est illustré par la figure ci-dessous.

Elle joue un rôle de sécurité en ne laissant passer que les flux autorisés en provenance du broker public.

Dans le cadre de ces travaux pratiques, les flux autorisés en provenance du broker public sont :

- le flux permettant d’allumer et d’éteindre la led verte connectée au micro pc linux (raspberry pi)
- le flux qui commande au raspberry pi d’établir un vpn ngrok pour permettre d’accéder en sécurité aux objets connectés situés en zone privée.

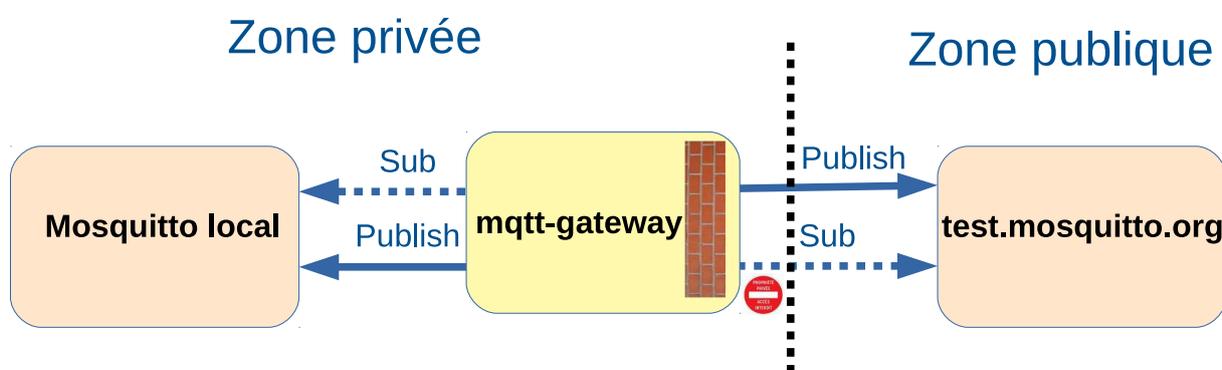


Illustration 28: rôle de la passerelle MQTT

4.2 TP1 Allumer Led Verte en local

4.2.1 Résultats attendus

L’objectif de ce TP est de montrer qu’en utilisant uniquement le module node-red local, il est possible de commander à partir du smart phone ou PC, l’allumage ou l’extinction de la led verte.

Ce TP utilise les éléments situés en ‘intérieur privé’ (local) suivants :

le PC ou le smart phone

le module 'node red' hébergé dans le micro PC linux.

4.2.2 Démonstration

Le flux mis en œuvre dans node red est présenté ci-dessous.

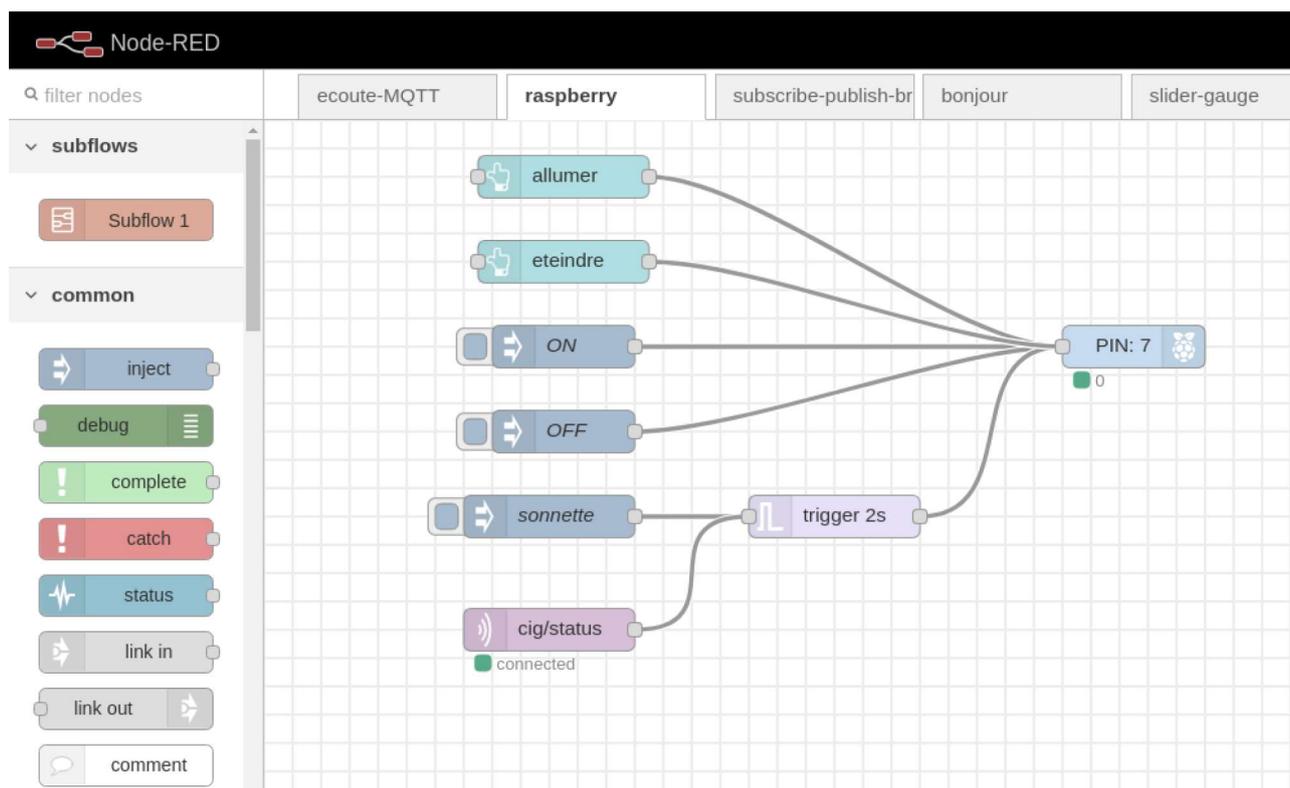


Illustration 29: flux raspberry

Pour commander l'allumage ou l'extinction de la led verte depuis son smarphone ou PC , ouvrir dans le navigateur , la page : <http://192.168.1.27:1880/ui/#!/3> et cliquer sur les boutons. nota : l'adresse IP 192.168.1.27 est un exemple.



Illustration 30: Commande Led Verte depuis la zone privée

4.2.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

4.3 TP2 Allumer Led Verte depuis l'extérieur

4.3.1 Résultats attendus

L'objectif de ce TP est de montrer qu'en utilisant uniquement le Node-Red public : FRED, il est possible de commander à partir d'un smart phone ou PC situé en extérieur, l'allumage ou l'extinction de la led verte.

Ce TP utilise l'ensemble de la configuration présentée [ici](#).

4.3.2 Démonstration

Le flux mis en œuvre dans node red est présenté ci-dessous.

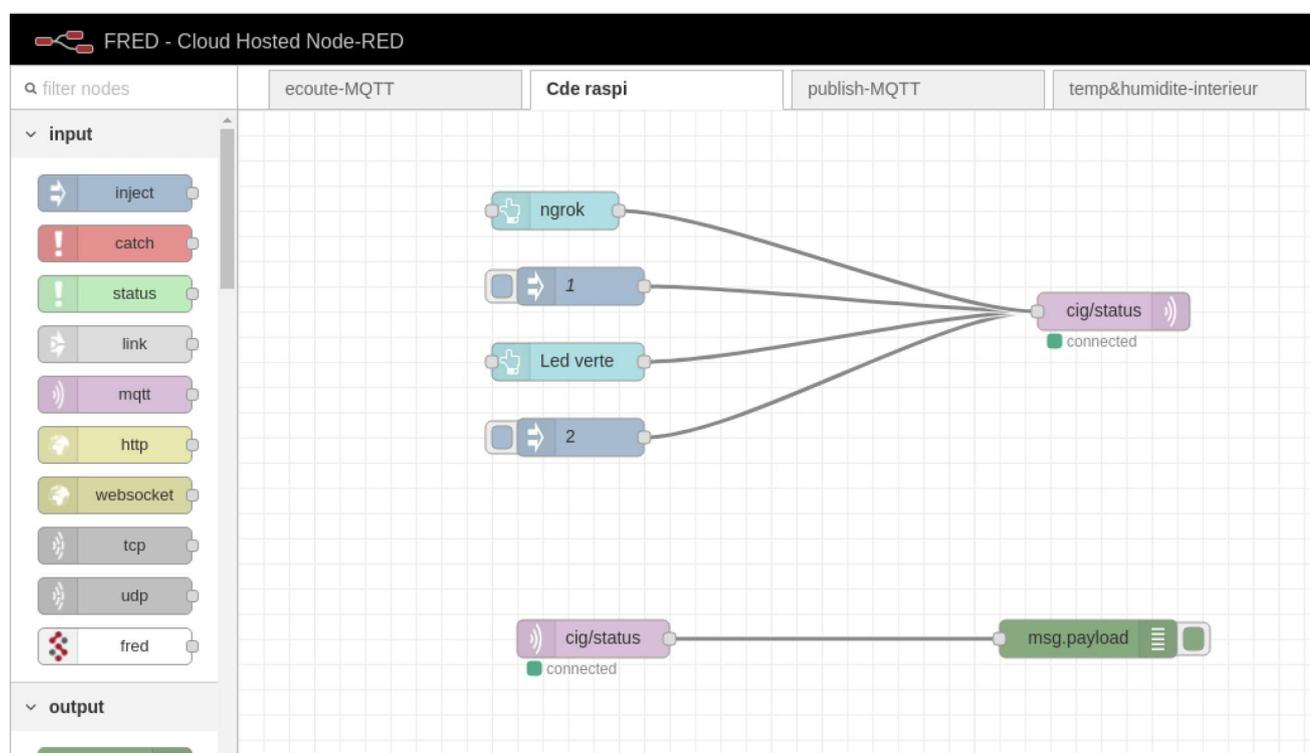


Illustration 31: flux cde raspberry sur le Node-Red public : FRED

Pour commander l'allumage/extinction de la led verte depuis son smarphone ou PC , ouvrir dans le navigateur , la page : https://cig_91.fred.sensetecnic.com/api/ui/#/0 et cliquer sur le bouton 'LED VERTE '.

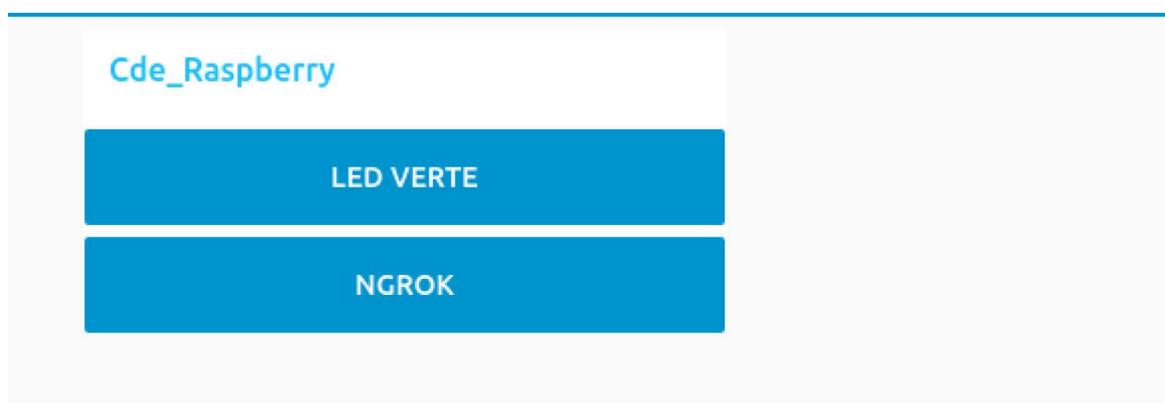


Illustration 32: Commande Led Verte depuis l'extérieur

4.3.3 Résultats obtenus

Le résultat obtenu est conforme aux résultats attendus.

5 TP Sonnette connectée

5.1 Objectifs de ce TP

Ce TP est un exercice de synthèse car il met en ouvre les principaux services qui sont dans le cours 'découverte du monde connecté'.

L'objectif de ce TP est de rendre le service suivant :

1- Je suis à l'extérieur et quelqu'un sonne à ma porte d'entrée qui est verrouillée. Je reçois alors une alerte sur mon smart phone

2- J'utilise ma camera installée à mon domicile pour reconnaître qui a sonné.

3- Si la personne qui sonne est autorisée à rentrer, je lui déverrouille la porte.

Les figures ci-dessous représente les écrans de mon smart phone pour les étapes 1 et 3. L'écran associé à l'étape 2 n'est pas représenté car j'utilise une caméra 'du commerce' non disponible à la date de ce TP.

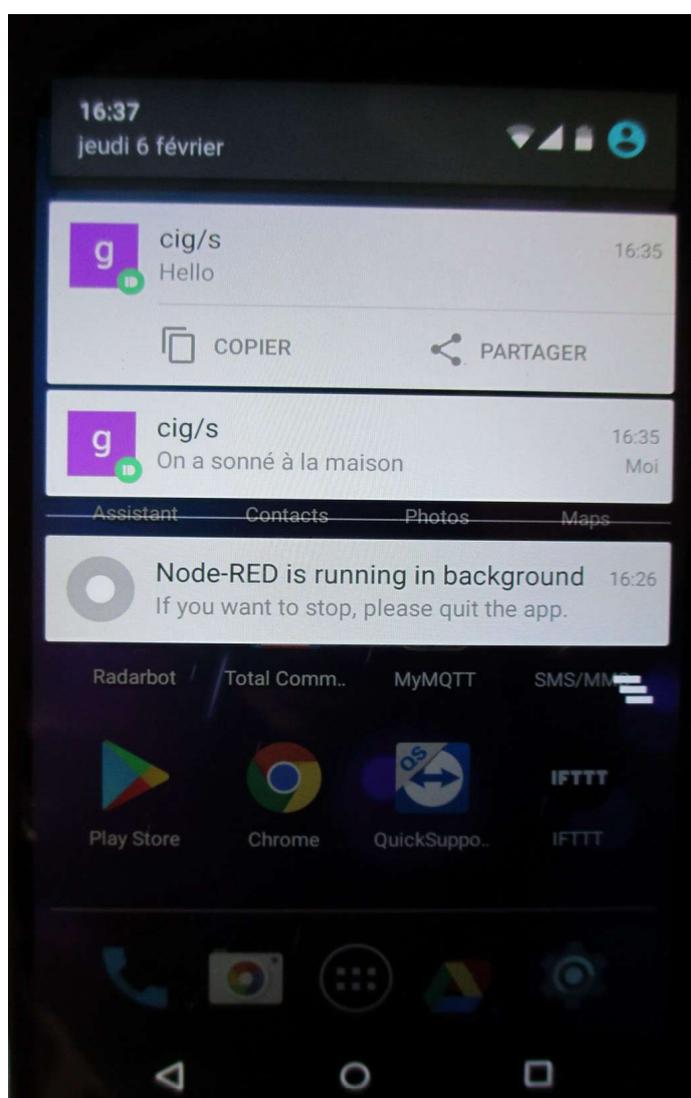


Illustration 33: Quelqu'un sonne à mon domicile !

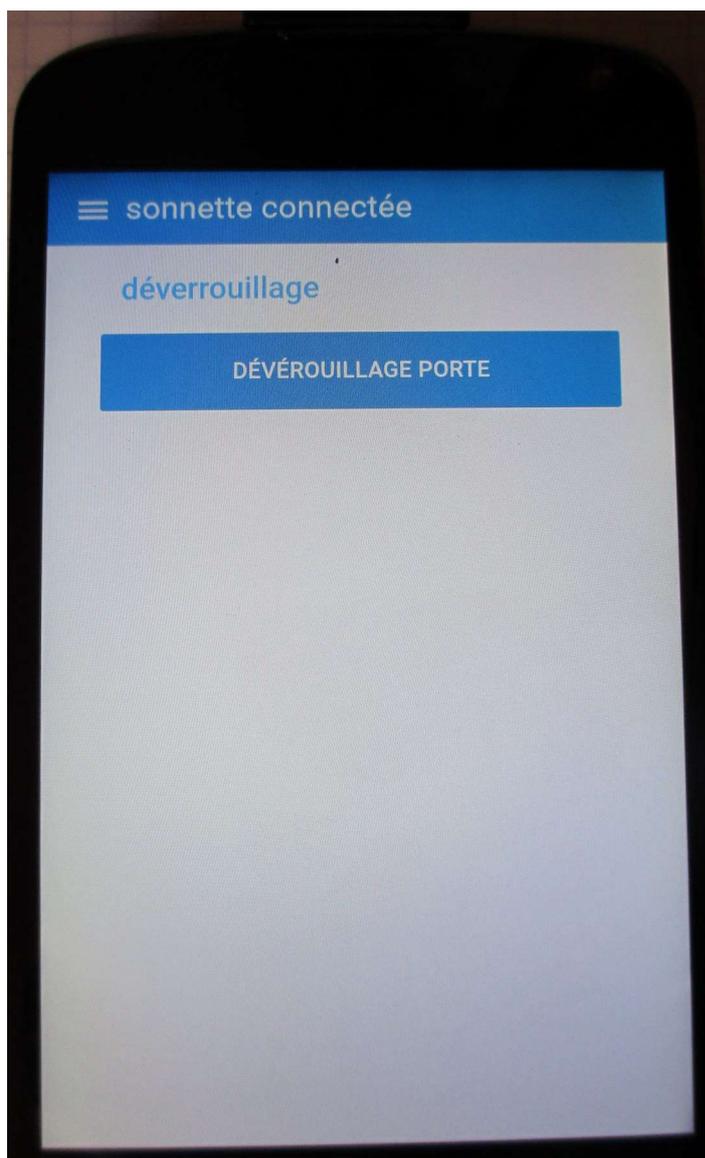


Illustration 34: étape 3: Je déverrouille la porte

5.2 Configuration, scénario et résultats attendus

Les figures ci-dessous présentent le scénario et la configuration utilisée à chaque étape.

5.2.1 Étape 1 : Je suis à l'extérieur et quelqu'un sonne à ma porte d'entrée qui est verrouillée.

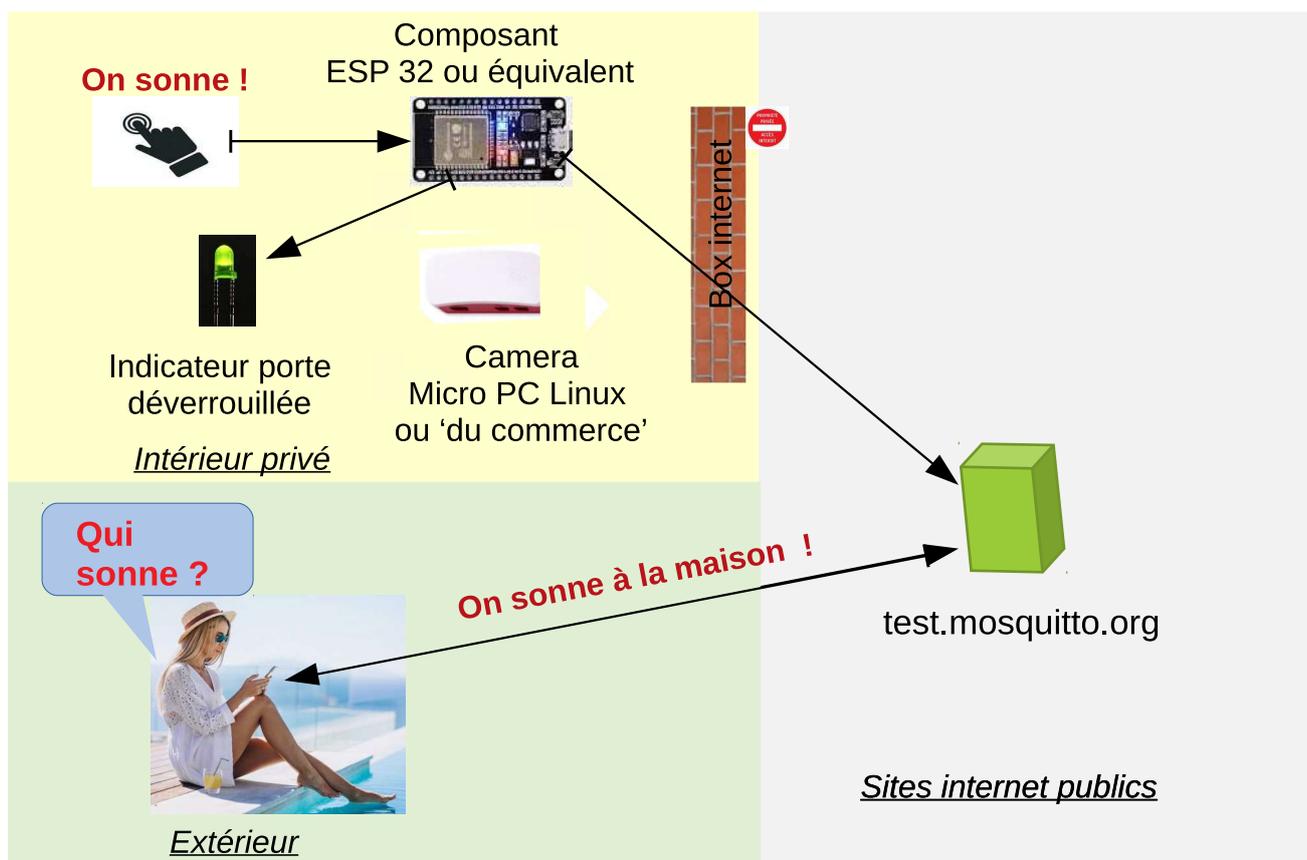


Illustration 35: Quelqu'un sonne à ma porte de mon domicile !

La configuration utilisée dans cette étape est :

En 'Intérieur privé'

- Relié à une carte ESP32 : un bouton poussoir et une Led verte indiquant 'porte verrouillée ou non'. Voir le montage de test [ici](#).

Sites internet publics

- test.mosquitto.org

En 'Extérieur privé'

- un smart phone Android avec l'application 'RedMobile'

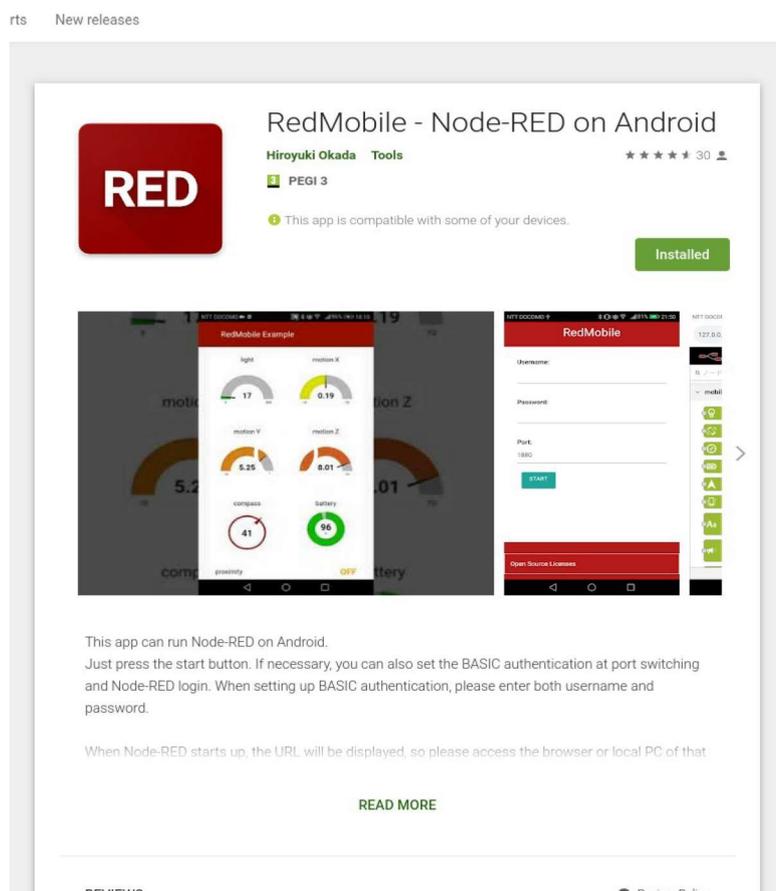


Illustration 36: Application Node Red sur smart phone

Flow node red

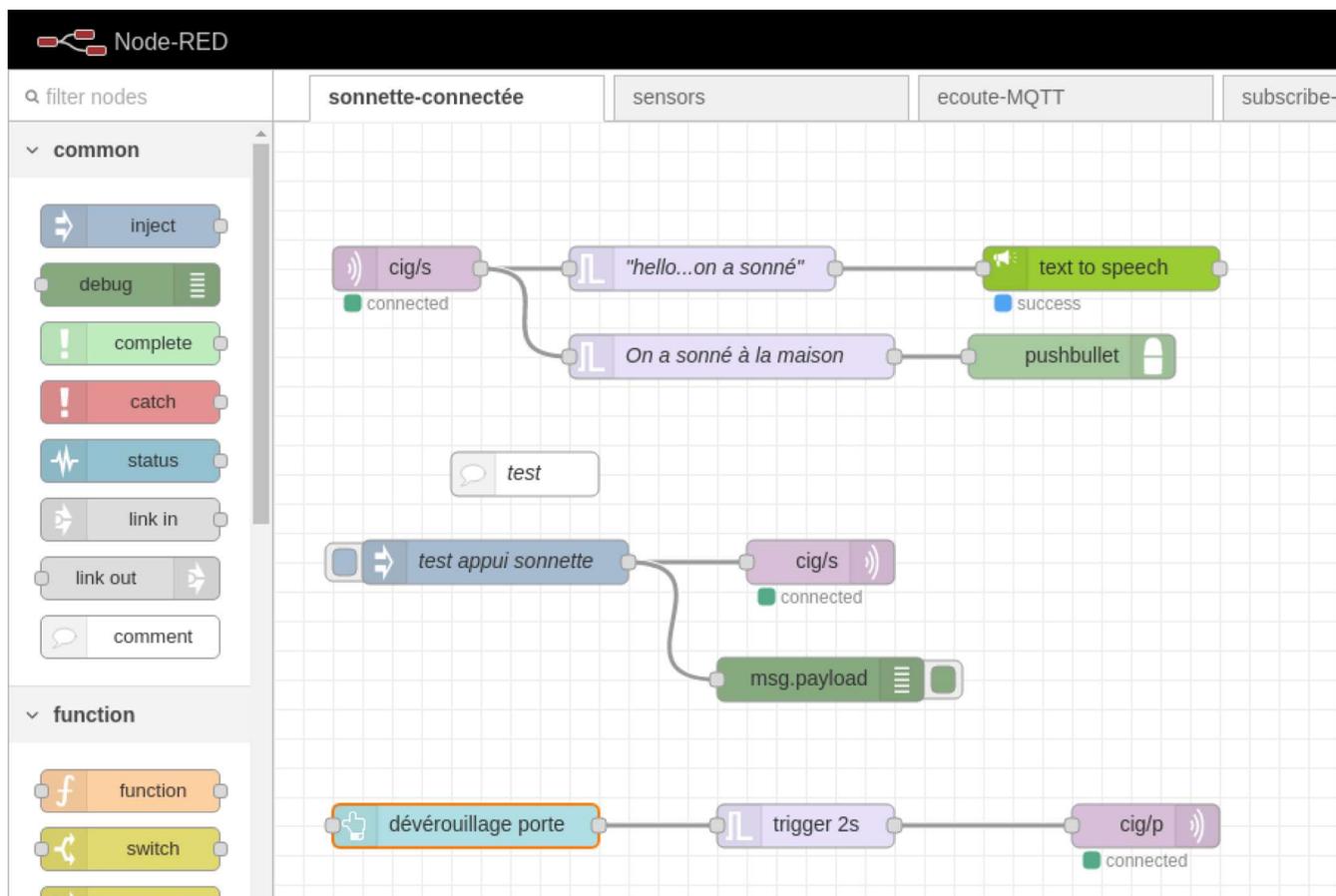


Illustration 37: Flux node-red installé dans l'application NodeRed Mobile

```
[{"id":"badacc20.bdff8","type":"ui_button","z":"281c5f74.3a38","name":"","group":"3015efb4.0ba77","order":4,"width":0,"height":0,"passthru":false,"label":"dévrouillage porte","tooltip":"","color":"","bgcolor":"","icon":"","payload":"p","payloadType":"str","topic":"","x":130,"y":480,"wires":[["4d39387b.9d8e58"]]}, {"id":"3015efb4.0ba77","type":"ui_group","z":"","name":"déverrouillage","tab":"c25775f7.b8b8a8","disp":true,"width":6,"collapse":false}, {"id":"c25775f7.b8b8a8","type":"ui_tab","z":"","name":"sonnette connectée","icon":"dashboard","order":3,"disabled":false,"hidden":false}]
```

Le résultat attendu est:

Lorsque quelqu'un appui sur le bouton poussoir de la sonnette, la personne en extérieur reçoit sur son smart phone : le message vocal : 'Hello...on a sonné, et une notification.

5.2.2 Étape 2 : J'utilise la camera de mon domicile pour reconnaître qui a sonné.

Dans cette version du TP, la caméra n'a pas été utilisée.

5.2.3 Étape 3 : je déverrouille à distance la porte d'entrée de mon domicile.

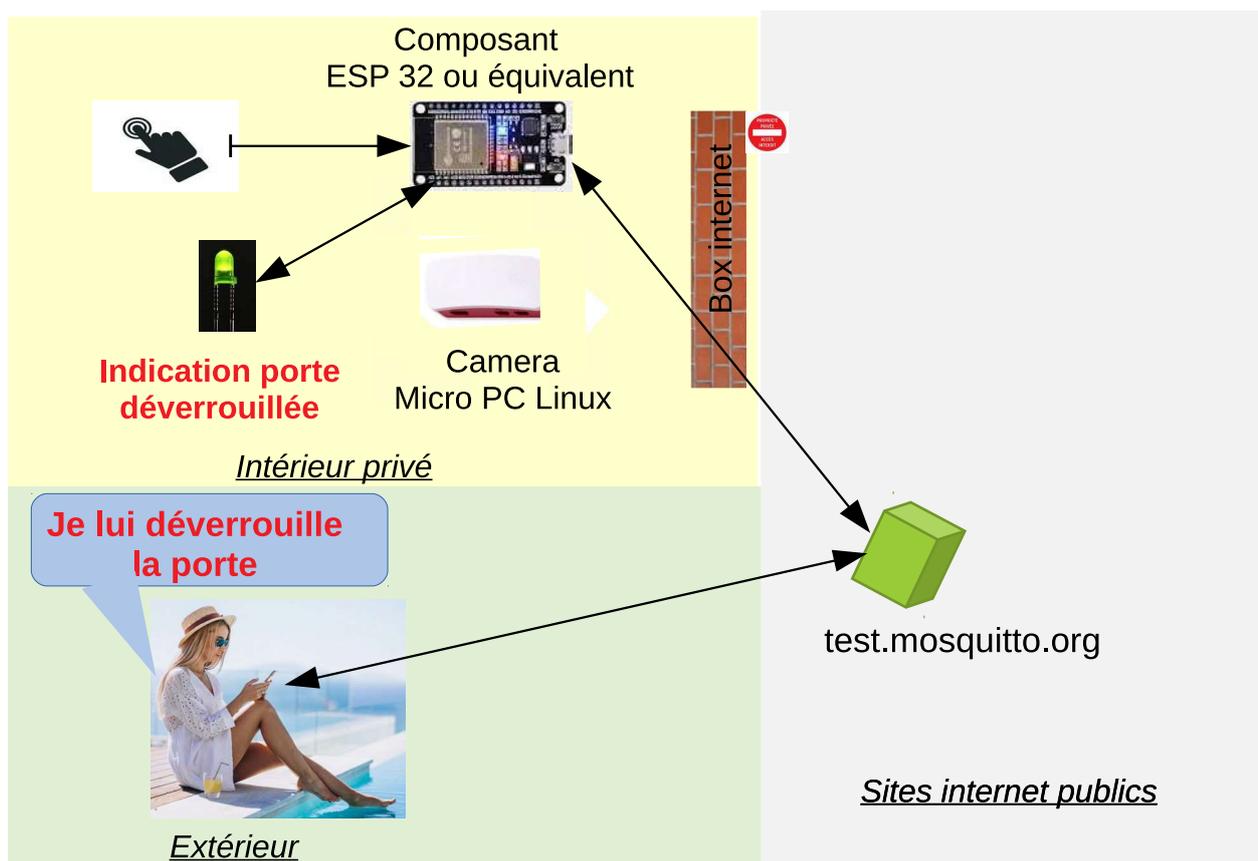


Illustration 38: je déverrouille ma porte à distance

La configuration utilisée dans cette étape est :

Sites internet publics

- test.mosquitto.org

En 'Extérieur'

- Le [bouton poussoir](#) 'deverrouillage porte' sur smart phone.

5.2.4 Résultats obtenus

Les résultats obtenus sont conformes aux résultats attendus

5.2.5 Application de ce TP à d'autres usages

Ce type de TP pourrait être adapté par exemple aux usages suivants :

- recevoir une alerte si le secteur 220 volt est coupé
- recevoir une alerte en cas de détections de présence à mon domicile.

5.2.6 Annexe : câblage de test

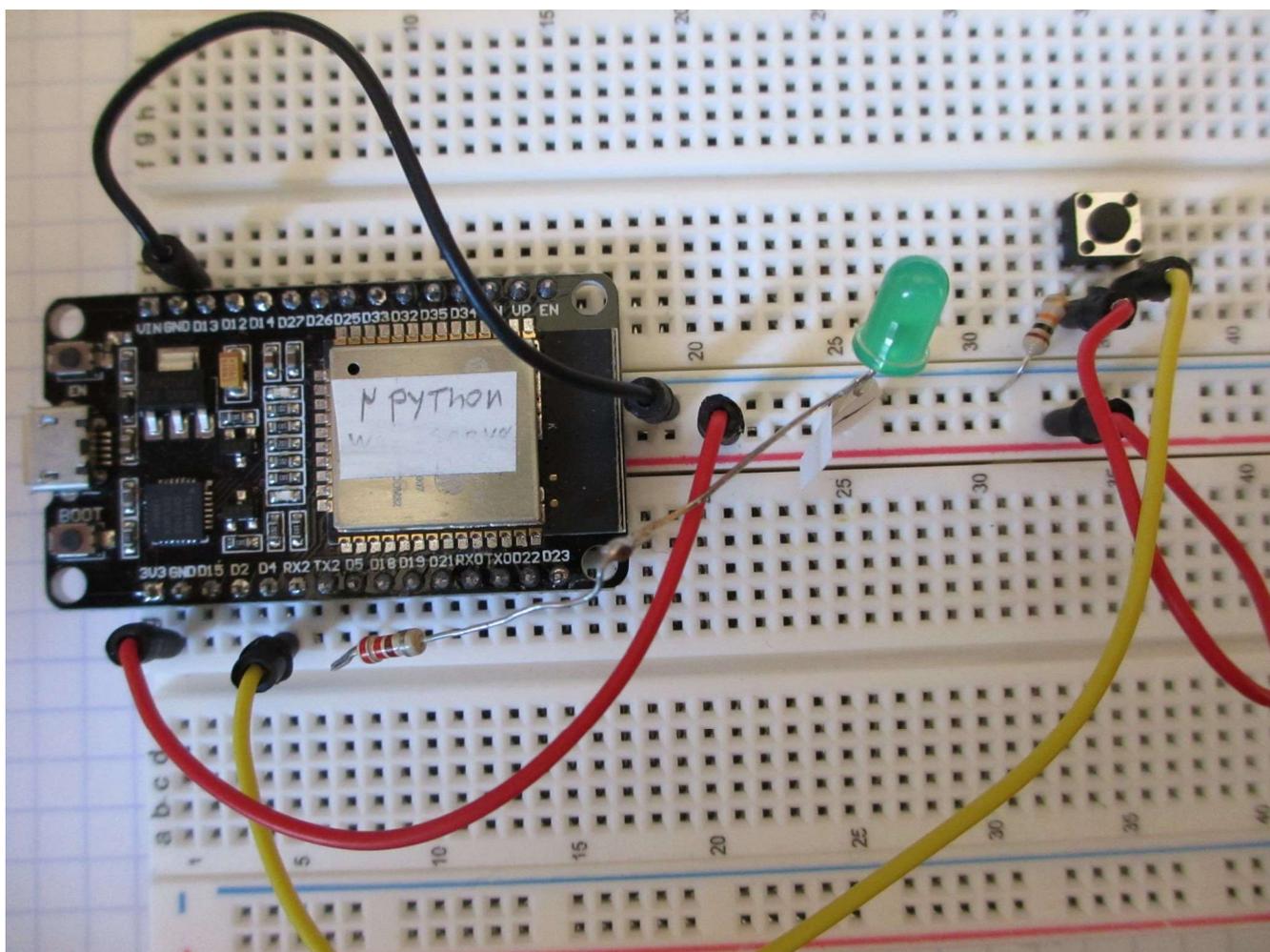


Illustration 39: câblage de test

5.2.7 Annexe : Logiciel installé dans l'ESP32

5.2.7.1 Boot.py :

```
import webrepl
webrepl.start()
```

```
import time
```

```
from simple import MQTTClient
```

```
import ubinascii
import machine
import micropython
import network
import esp
esp.osdebug(None)
import gc
gc.collect()

ssid = 'KNET_0ab3'
password = 'xxxxxxxxxxx'
mqtt_server = 'test.mosquitto.org'
#EXAMPLE IP ADDRESS
#mqtt_server = '192.168.1.144'
client_id = ubinascii.hexlify(machine.unique_id())
topic_sub = b'cig/p'
topic_pub = b'cig/s'

last_message = 0
message_interval = 5
counter = 0

station = network.WLAN(network.STA_IF)

station.active(True)
station.connect(ssid, password)
```

```
while station.isconnected() == False:  
    pass
```

```
print('Connection successful')  
print(station.ifconfig())
```

5.2.7.2 main.py :

```
from machine import Pin  
from time import sleep
```

```
button = Pin(4, Pin.IN)  
led = Pin(5, Pin.OUT)
```

```
def sub_cb(topic, msg):  
    print((topic, msg))  
    # if topic == b'notification' and msg == b'received':  
    # print('ESP received hello message')
```

```
if topic == b'cig/p':  
    print('ESP received on cig/s ')  
    print (msg)  
    if (msg == b'1') :  
        led.value(1)  
    else:  
        led.value(0)
```

```
def connect_and_subscribe():
    global client_id, mqtt_server, topic_sub
    client = MQTTClient(client_id, mqtt_server)
    client.set_callback(sub_cb)
    client.connect()
    client.subscribe(topic_sub)
    print('Connected to %s MQTT broker, subscribed to %s topic' % (mqtt_server, topic_sub))
    return client

def restart_and_reconnect():
    print('Failed to connect to MQTT broker. Reconnecting...')
    time.sleep(10)
    machine.reset()

try:
    client = connect_and_subscribe()
except OSError as e:
    restart_and_reconnect()

while True:
    try:
        client.check_msg()
        button_status = button.value()
    #    print ('toto')
```

```
if button_status == 1:  
    msg = 'b1' #button on  
    print('button ON')  
    client.publish(topic_pub, msg)  
    sleep(1)
```

```
except OSError as e:  
    restart_and_reconnect()
```