

# Tp ESP32 Tasmota MQTT Node red

[presentation\\_iot\\_v03.pdf](#)

[Diffusion Plan reseaux Castellab Tasmota MQTT node red](#)

[Plan reel reseaux Castellab Tasmota MQTT node red avec MDP](#)

[Installation RaspberryOS Lite 32bits](#)

[arrossage\\_automatique](#)

[Récuperer l'@IP de l'ESP32 via Node-red](#)

[Tasmota LCD](#)

[Heure de l'esp32 dans Node-red](#)

[Date et heure du Raspberry dans Node red](#)

[Installer un serveur MQTT sur un RaspberryPI](#)

[Acceder à votre raspberry depuis internet](#)

[Changer l'image du fond d'ecran du Dashboard de Node-red](#)

[Stockage des données contextuelles dans le système de fichiers](#)

[Notifications sur telephone via l'application ntfy \( open source \)](#)

[Idées pour compteurs avec node red](#)

[Em/Rec 433Mhz](#)

[Installation et utilisation dashboard2](#)

[Installation python dans node-red](#)

[Creation de noeuds qui additionnent des valeurs et ensuite raz](#)

[Dashboard2 : Doc EN](#)

[Changement de mot de passe Node red](#)

[Programmation Chauffage avec le noeud "Ramp-Thermostat"](#)

[Programmation Chauffage avec le noeud "Thermostat-Dynamique"](#)

[Noeuds principaux à installer](#)

## Materiels utilisés

- 1 ESP32 avec son cordon
- 1 PC Fixe ou portable
- 1 Led
- 1 Bredboard
- Des cables Dupont
- 1 Bouton poussoir ou interrupteur
- 1 DHT11
- 1 serveur MQTT fonctionnel - Serveur MQTT de test ==> [fablab37110.ddns.net:1883](https://fablab37110.ddns.net:1883)

OrangePi

[installer un serveur Node-Red sur un Raspberry](#)

## Serveurs Node-red de test



ATTENTION inserer dans le champ MQTT de tasmota et node-red : **fablab37110.ddns.net** ou **86.196.177.176** et ensuite dans le champ "port" mettre **1883** ne pas mettre http://.... ATTENTION I@IP PEUT CHANGER ... Preferez le nom de domaine



**Le mot de passe (nodered) est.... : Demander à GL**

- Plusieurs serveurs Node Red fonctionnel

Nouveau serveur de test NodeRed pour les années 2023-2024-2025-2026-.... : ( @IP externe :82.121.82.186)



Le serveur Node Red ci-dessous doit etre **utilisé comme un serveur de TEST et non pas comme un serveur qui peut servir à long terme ...**



Alerte : MISE A JOUR Terminée.... UTILISER LE SERVEUR NODE-RED [fablab37110.ddns.net:18XX](https://fablab37110.ddns.net:18XX) et vérifier vos flux



SERVEUR NODE RED **ACTIF**

- [fablab37110.ddns.net:1881](https://fablab37110.ddns.net:1881) Pour Xavier
- [fablab37110.ddns.net:1884](https://fablab37110.ddns.net:1884) Pour .....
- [fablab37110.ddns.net:1885](https://fablab37110.ddns.net:1885) Pour Sebastien
- [fablab37110.ddns.net:1886](https://fablab37110.ddns.net:1886) Pour Philippe

- [fablab37110.ddns.net:1887](https://fablab37110.ddns.net:1887) Pour Fernand
- [fablab37110.ddns.net:1888](https://fablab37110.ddns.net:1888) Pour Patrick
- [fablab37110.ddns.net:1889](https://fablab37110.ddns.net:1889) Pour Jean-Luc
- [fablab37110.ddns.net:1891](https://fablab37110.ddns.net:1891) Pour Jean-Luc Test
- [fablab37110.ddns.net:1892](https://fablab37110.ddns.net:1892) Pour Thierry ( cours - demo )
- [fablab37110.ddns.net:1893](https://fablab37110.ddns.net:1893) Pour .....
- [fablab37110.ddns.net:1894](https://fablab37110.ddns.net:1894) Pour .....



Prevoir la reinstallation des noeuds supplementaires utilises...



SERVEUR NODE-RED de secours ne plus utiliser pour l'instant **Sauvegarder vos flux**



SERVEUR NODE RED **INACTIF**

- [castellab.ddnsfree.com:1881](https://castellab.ddnsfree.com:1881) Pour Xavier
- [castellab.ddnsfree.com:1882](https://castellab.ddnsfree.com:1882) Pour .....
- [castellab.ddnsfree.com:1884](https://castellab.ddnsfree.com:1884) Pour .....
- [castellab.ddnsfree.com:1885](https://castellab.ddnsfree.com:1885) Pour Sébastien
- [castellab.ddnsfree.com:1886](https://castellab.ddnsfree.com:1886) Pour Philippe
- [castellab.ddnsfree.com:1887](https://castellab.ddnsfree.com:1887) Pour Fernand
- [castellab.ddnsfree.com:1888](https://castellab.ddnsfree.com:1888) Pour Patrick
- [castellab.ddnsfree.com:1889](https://castellab.ddnsfree.com:1889) Pour Jean-Luc
- [castellab.ddnsfree.com:1890](https://castellab.ddnsfree.com:1890) Pour .....
- [castellab.ddnsfree.com:1891](https://castellab.ddnsfree.com:1891) Pour Jean-Luc
- [castellab.ddnsfree.com:1892](https://castellab.ddnsfree.com:1892) Pour Thierry backup
- [castellab.ddnsfree.com:1893](https://castellab.ddnsfree.com:1893) Pour .....
- [castellab.ddnsfree.com:1894](https://castellab.ddnsfree.com:1894) Pour .....
- [castellab.ddnsfree.com:1895](https://castellab.ddnsfree.com:1895) Pour Gerard Test
- Si mot de passe demandé ⇒ le meme mdp que pc fablab et utilisateur = "admin"



Il est **Primordial** d '**utiliser un serveur node-red par utilisateur et un seul ...**



Il est **IMPERATIF** de **sauvegarder sa configuration Node-red** dans un fichier sur son PC au cas ou ....

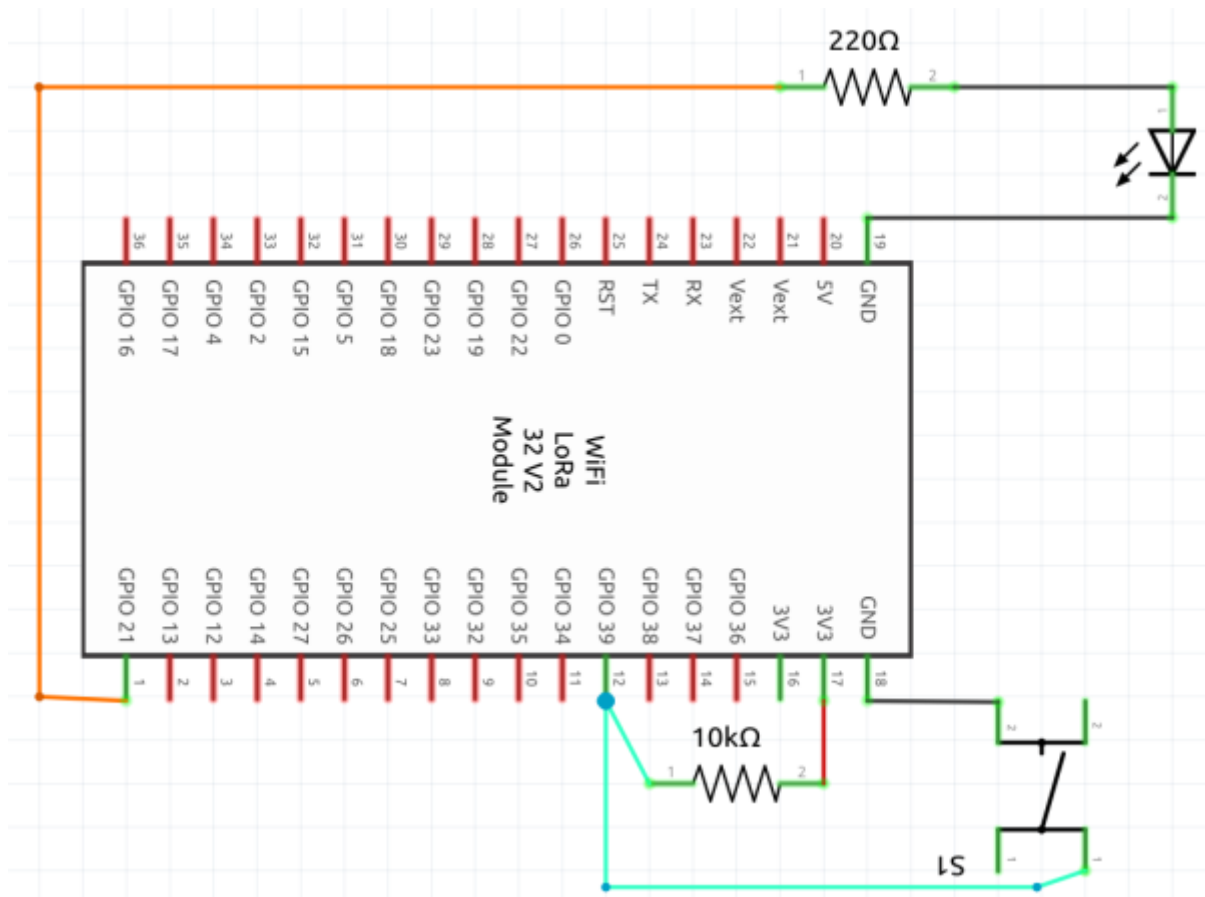
## Sauvegarde configuration Node-red

[sauvegarde configuration Node-red](#)

# Supprimer un Flow dans node-red

Supprimer un flow dans node red

## Schema de raccordement



## Exercice 1 :

### Etapes

- Raccorder l'ESP32 au PC sur un port USB
- Télécharger le binaire Tasmota avec [Web Installer](#) en utilisant Chrome ou Chromium
- Choisir le port COM pour se connecter à l'ESP32
- Sur l'ESP32 appuyer plusieurs secondes sur le bouton "Boot" et en même temps cliquer sur "Install" pour télécharger le firmware Tasmota sur l'ESP32



- Configurer le wifi , Indiquer le SSID et le mot de passe de votre box



Si vous etes au FABLAB et si nous n'avez pas de connexion wifi (castellab), taper dans la console : **Backlog SSID1 Livebox-5200; Password1 MPDCASTELLAB; SSID2 SSIDVOTREBOX; Password2 MDPWIFIVOTREBOX;**

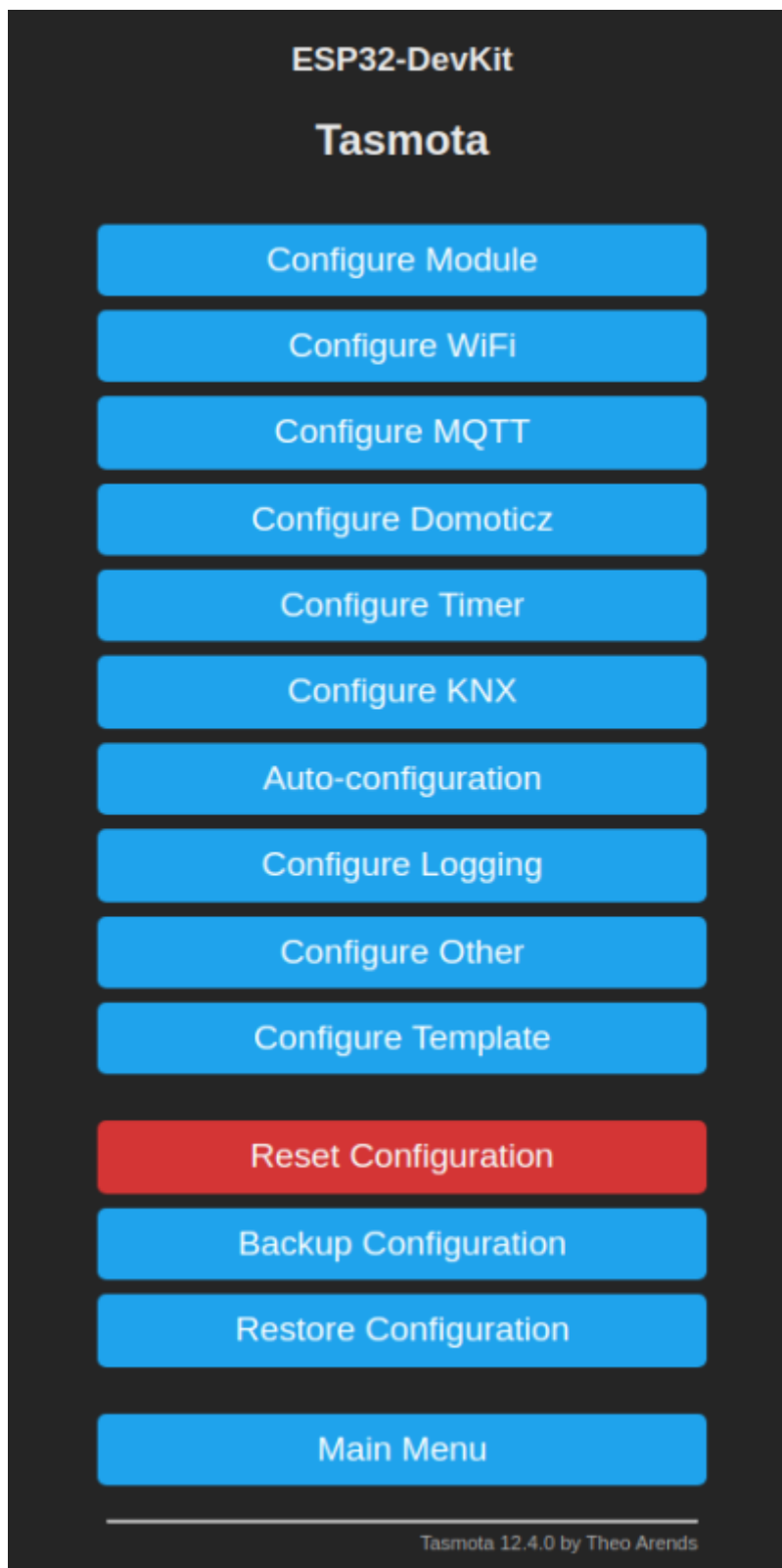
- Connecter vous via le navigateur web **Chrome ou Chromium**, à l'adresse @192.168.xxx.xxx de votre tasmota

(si vous ne connaissez pas @IP de votre tasmota : Vous pouvez utiliser l' [interface de votre box](#) ou le logiciel [nmap](#))

- Une fenetre comme ci-dessous s'ouvrira



- Menu Configuration



- Configurer le serveur MQTT , Exemple : “fablab37110.ddns.net” le port par defaut est 1883



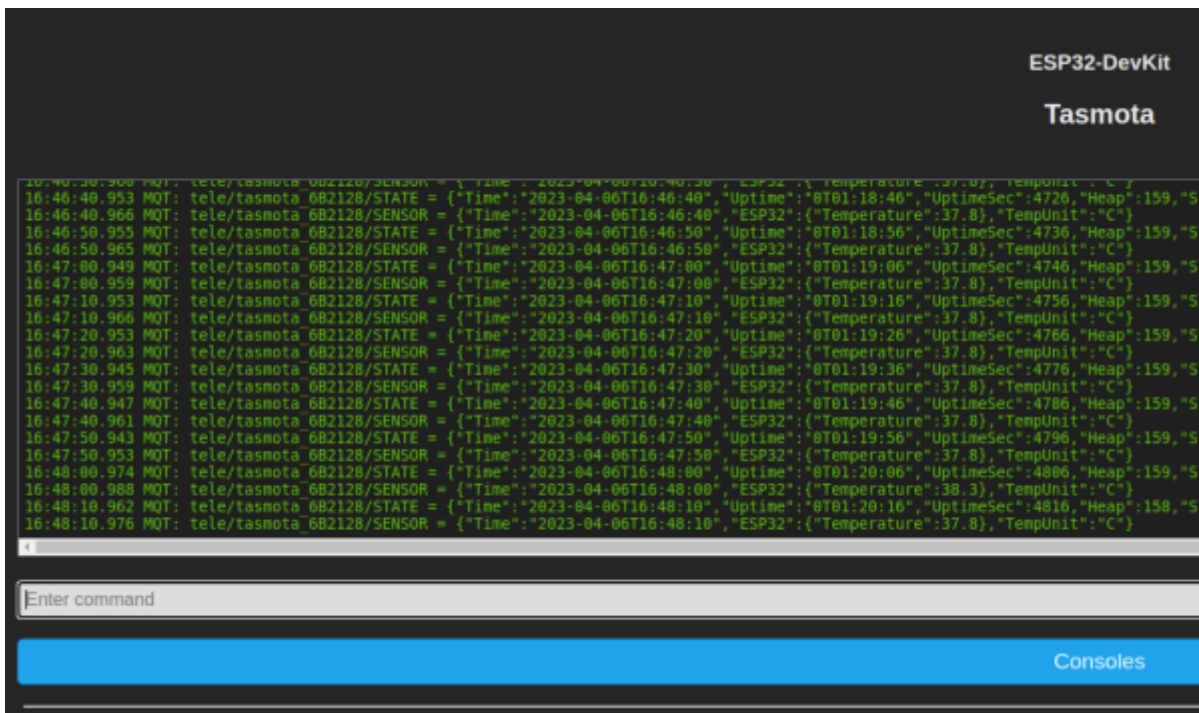
- configurer les modules :



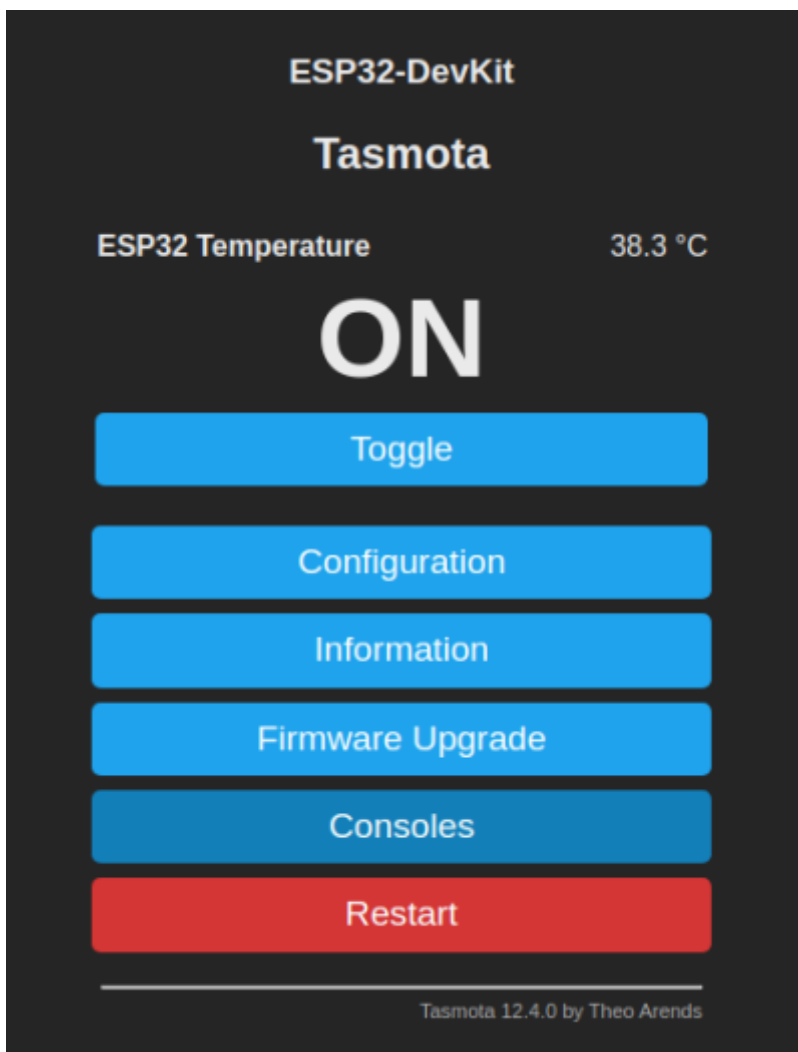
1. - GPIO39 ==> bouton ==> index 1 ( ceci est un exemple on peut configurer d'autres GPIO correspondant à votre ESP32 )
2. - GPIO21 ==> relais ==> index 1 ( ceci est un exemple on peut configurer d'autres GPIO correspondant à votre ESP32 )

- **Ouvrir la console de tasmota** ( revenir au premier ecran) est entrer les commandes :

1. **SetOption146 1** + Entrée( validation de l'envoi de la temperature de l'ESP32) et
2. **teleperiod 5** + Entrée ( definit la periode d'envoi des informations vers MQTT ici tous les 10 s),
3. **SetOption65 1** pour eviter de reinitialiser la configuration sur default d'alimentation,
4. **SetOption1 1** pour default reset intempestif sur appui bouton.



-revenir au premier ecran



- Copier les topic de l'esp32 "stat/tasmota\_6B2128/POWER" ( A partir de la console) pour l'etat

ON/OFF du bouton ( à sauvegarder dans un fichier texte provisoire) Le code 6B2128 depend de l'ESP32, il est différent à chaque ESP32 , reprend une partie de l'adresse MAC de l'ESP32

- Et copier le topic “**tele/tasmota\_6B2128/SENSOR**” pour recuperer la temperature de l'ESP32 Idem le sauvegarder



NOTER les 6 derniers caracteres de l'adresse MAC de votre ESP32 pour le retrouvez plus facilement dans votre configuration et **coller une etiquette sur votre ESP32 avec cette @MAC**

```
16:46:36.908 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:46:36", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:18:46", "UptimeSec": 4726, "Heap": 159, "S1": 159}
16:46:40.966 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:46:40", "Uptime": "0T01:18:46", "UptimeSec": 4726, "Heap": 159, "S1": 159}
16:46:40.966 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:46:40", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:18:46", "UptimeSec": 4726, "Heap": 159, "S1": 159}
16:46:50.955 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:46:50", "Uptime": "0T01:18:56", "UptimeSec": 4736, "Heap": 159, "S1": 159}
16:46:50.955 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:46:50", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:18:56", "UptimeSec": 4736, "Heap": 159, "S1": 159}
16:47:00.949 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:00", "Uptime": "0T01:19:06", "UptimeSec": 4746, "Heap": 159, "S1": 159}
16:47:00.949 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:00", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:06", "UptimeSec": 4746, "Heap": 159, "S1": 159}
16:47:10.953 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:10", "Uptime": "0T01:19:16", "UptimeSec": 4756, "Heap": 159, "S1": 159}
16:47:10.953 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:10", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:16", "UptimeSec": 4756, "Heap": 159, "S1": 159}
16:47:20.966 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:20", "Uptime": "0T01:19:26", "UptimeSec": 4766, "Heap": 159, "S1": 159}
16:47:20.966 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:20", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:26", "UptimeSec": 4766, "Heap": 159, "S1": 159}
16:47:30.963 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:30", "Uptime": "0T01:19:36", "UptimeSec": 4776, "Heap": 159, "S1": 159}
16:47:30.963 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:30", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:36", "UptimeSec": 4776, "Heap": 159, "S1": 159}
16:47:38.959 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:38", "Uptime": "0T01:19:46", "UptimeSec": 4786, "Heap": 159, "S1": 159}
16:47:38.959 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:38", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:46", "UptimeSec": 4786, "Heap": 159, "S1": 159}
16:47:40.947 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:40", "Uptime": "0T01:19:56", "UptimeSec": 4796, "Heap": 159, "S1": 159}
16:47:40.947 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:40", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:19:56", "UptimeSec": 4796, "Heap": 159, "S1": 159}
16:47:50.943 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:47:50", "Uptime": "0T01:20:06", "UptimeSec": 4806, "Heap": 159, "S1": 159}
16:47:50.943 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:47:50", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:20:06", "UptimeSec": 4806, "Heap": 159, "S1": 159}
16:48:00.974 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:48:00", "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
16:48:00.974 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:48:00", "ESP32": {"Temperature": 38.3, "TempUnit": "C"}, "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
16:48:10.962 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:48:10", "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
16:48:10.962 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:48:10", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
16:48:10.976 MOT: tele/tasmota_6B2128/STATE = {"Time": "2023-04-06T16:48:10", "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
16:48:10.976 MOT: tele/tasmota_6B2128/SENSOR = {"Time": "2023-04-06T16:48:10", "ESP32": {"Temperature": 37.0, "TempUnit": "C"}, "Uptime": "0T01:20:16", "UptimeSec": 4816, "Heap": 158, "S1": 158}
```

- Vous pouvez copier d'autres topic pour les inserer dans Node red si besoin

- Vous connecter au serveur Node red ( Au serveur qui vous est destiné pour les tests : fablab37110.ddns.net:18XX )



Attention si vous vous connecter à plusieurs sur le meme serveur Node red , vous allez vous melanger les flux et cela risque de creer des soucis , donc prenez chacun un Node red voir avec Thierry ou Gerard... (Voir le debut du tuto pour les @IP des serveurs node-red)

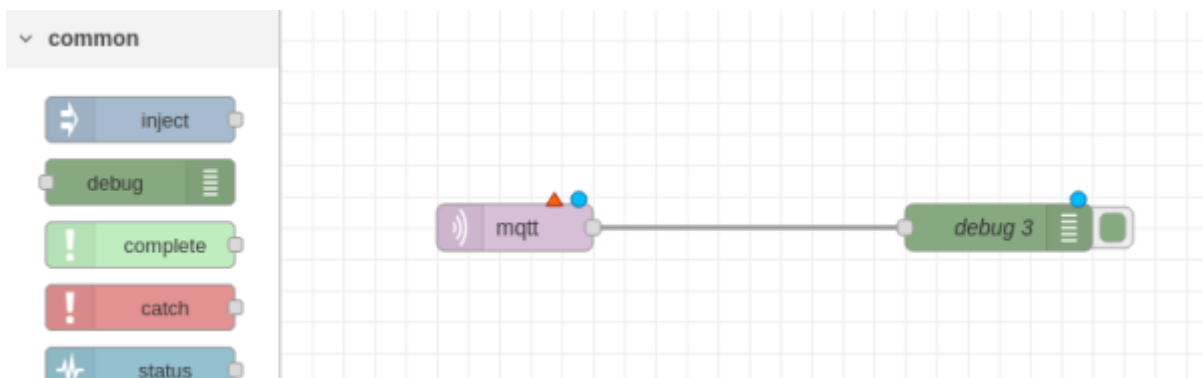
- allez ouvrir une page web comme celle-ci :



- Vous inserez un noeud en le faisant glisser sur la partie centrale. le premier noeud est MQTT IN dans la colonne de gauche section "Network"



- Inserez le noeud "Debug" section "Common" pour visualiser les informations venant de votre ESP32 via MQTT



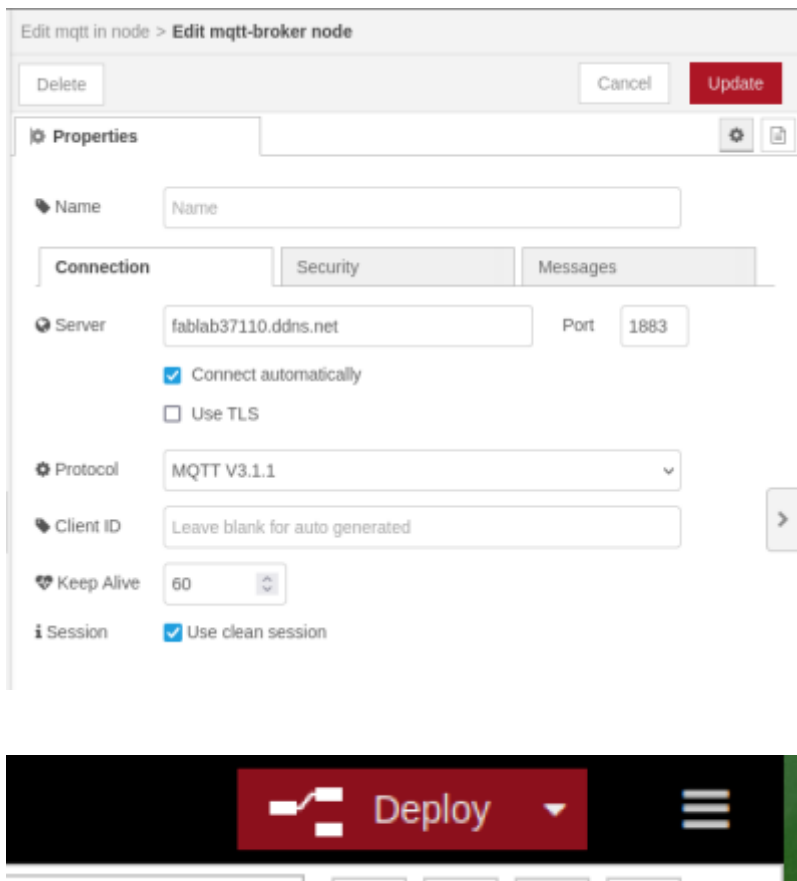
- Il vous faut paramétrer le noeud MQTT IN avec l'adresse du serveur MQTT utilisé par votre Tasmota de l'ESP32 et le topic ( sujet) que vous voulez utiliser

- serveur MQTT => "fablab37110.ddns.net port par default 1883 ( Le crayon pour changer @MQTT)
- Topic à inserer pour le bouton "stat/tasmota\_6B2128/POWER"

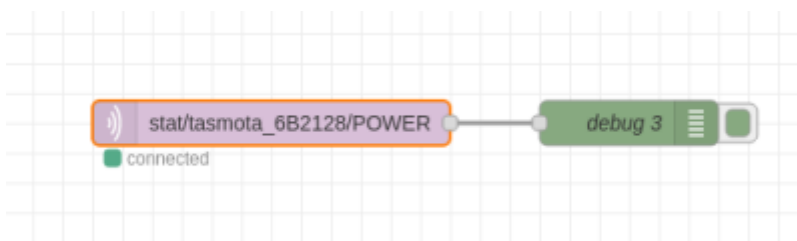
Cliquer sur le bouton "Update" et ensuite bouton "Done" et ensuite le bouton "Deploy" pour valider ( en haut à droite )



Cliquez sur le petit crayon pour avoir cette fenetre et inserer uniquement l'URL de MQTT , le port etant par default 1883



Si les ronds bleu on disparu et aucun triangle orange , c'est qu'il n'y pas d'erreurs et si vous avez l'info " Connected " en dessous du noeud MQTT IN , indique que Node red à bien trouver le serveur MQTT

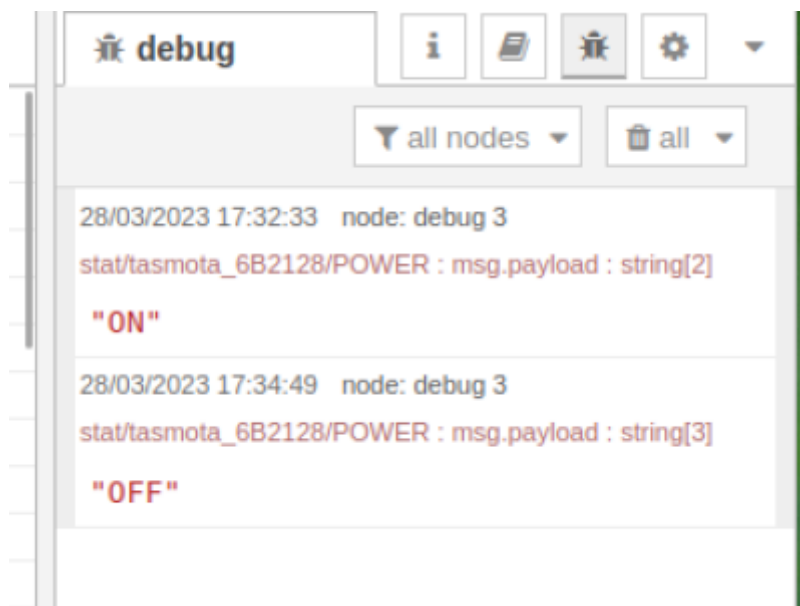


Si vous voulez utiliser un autre topic par exemple la temperature , il faut utiliser un autre noeud MQTT IN

Si vous appuyer 1 fois sur le bouton ( dans la fenetre Tasmota "Toggle" ) , vous devez voir l'information "ON" dans la fenetre node-red "Debug"



Si vous appuyer 1 autre fois sur le bouton , vous devez voir l'information "OFF" dans la fenetre "Debug"



## Visualiser la Led de IESP32 sur Node Red

- 1-Inserer 2 nodes supplementaires :

**node-red-dashboard** , il permet d'avoir un tableau de bord pour visualiser et piloter notre ESP32

et **node-red-contrib-ui-led** , permet d'afficher la valeur de la LED Eteinte = vert, Allumée = Rouge

- 2- Configurer le node de la Led pour l'affichage
- 3- Configurer le node MQTT out et un node switch pour piloter la LED via Node Red

- 
- 1- Inserer les nodes supplementaires ( Cliquez sur le menu en haut à droite à coté de l'icone "Deploy" )



- 2- Cliquez sur **“Manage Palette”** et ensuite l'onglet **“Install”** taper dans la zone de recherche **“dashboard”** et cliquez sur **“install”** à la ligne **“node-red-dashboard”**



- 3- Faire la même chose avec le node **“ node-red-contrib-ui-led ”**
- 4 - Vous devriez avoir dans la colonne de gauche un menu supplémentaire avec le dashboard avec ses nodes correspondants ainsi que le node **“LED”**



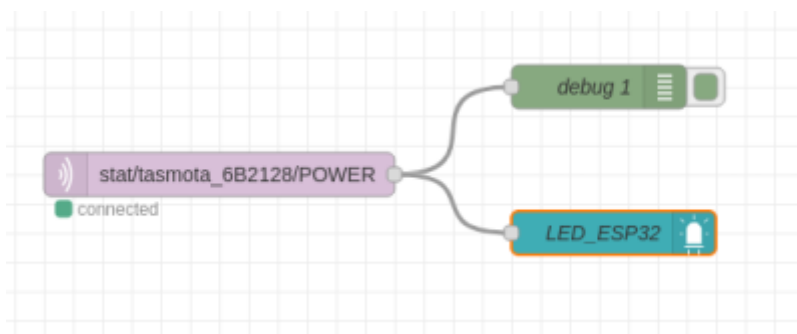
- 5- Inserez le node "LED" et relié le au node "MQTT in" paramétré précédement



- 6- On parametre le node “LED” : Le groupe ( petit crayon) , Le Label ( Nom), La position , La forme(Cercle ou carré), le msg.payload mode String ( ON / OFF), la couleur ( Rouge = Allumée , Blanc = Eteinte )

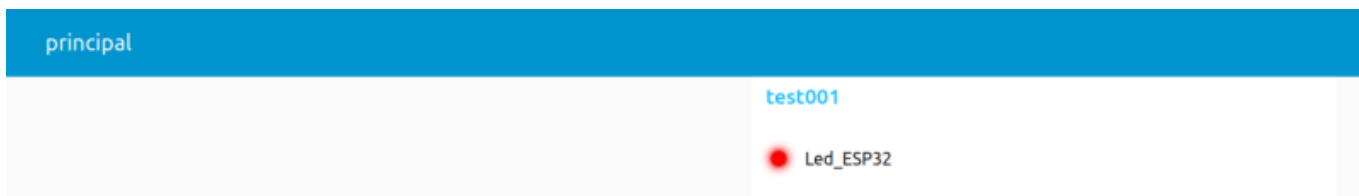


- 7- On parametre le groupe ( Name) et le nom du dashboard ( Tab) en cliquant sur le petit crayon et ensuite on clique sur “Update” et ensuite “Done” et on n'oublie pas de faire “Deploy” ( Enregistrer)



- 8- En ouvrant une nouvelle fenetre pour afficher le Dashboard @iPNodeRed/ui ( exemple : <http://192.168.1.44:1880/ui/>) on affiche le tableau de bord avec la LED qui s'allume ou s'eteint suivant l'etat du bouton de l'ESP32

### Allumé



### Eteint

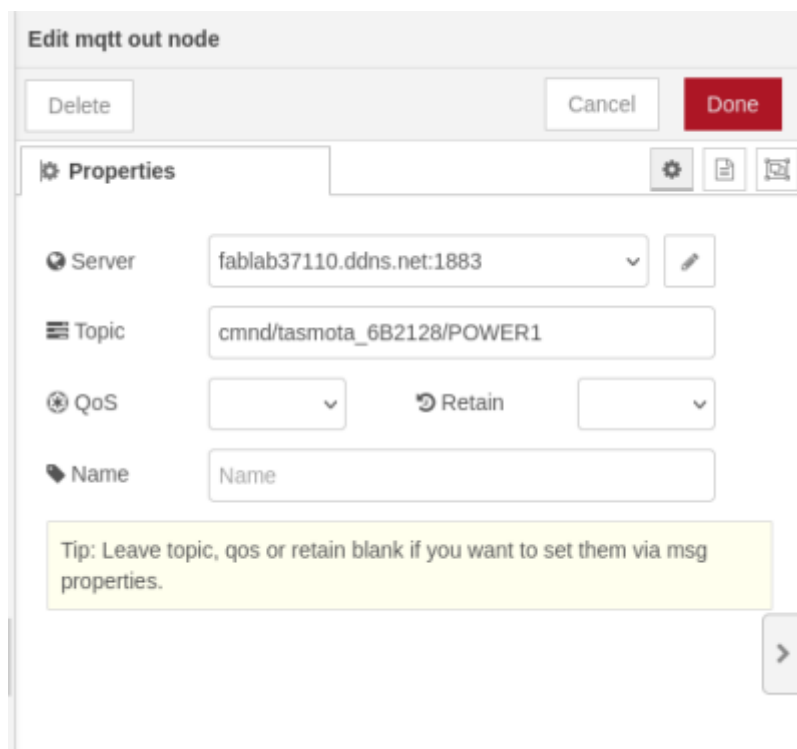


## Piloter la LED de ESP32 et du Dashboard via Node Red

- 1- Inserez 2 nodes : "MQTT out" ( Network ) et "Switch" ( Dashboard) et relié les ensembles ( Sortie Switch sur entree MQTTout)



- 2- Parametrer le node "MQTT out" : **server** (exemple) = fablab37110.ddns.net:1883 ; **Topic** = cmdnd/tasmota\_6B2128/POWER1 (cmdnd = abreviation pour envoyer des infos à tasmota; tasmota\_6B2128 = reference materiel du tasmota esp32 utilisé; /POWER = Bouton ou switch de tasmota; 1 = index 1 de Tasmota )



Cliquez sur le petit crayon ( ligne server) pour avoir cette fenetre et inserer uniquement l'URL de MQTT (pour exemple fablab37110.ddns.net) , le port etant par default 1883

- 3- Parametrer le node "Switch" : le groupe ( ICI test001) , la taille ( Size), Le label( Nom) , On Payload = ON , Off Payload = OFF, Topic = msg.topic ) Cliquer sur "Done" et "Deploy"

- 4- On devrait visualiser "connected" sous le node MQTT out



- 5- On affiche le dashboard ( voir rubrique précédente); En actionnant le "Switch" sur NodeRed on eteint ou allume la LED , et de même avec le bouton de l'ESP32

### Allumé



### Eteinte



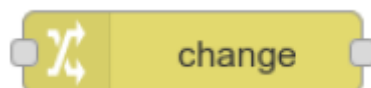
## Exercice 2 :

- Essayer de paramétrer le Dashboard avec la température de l'ESP32 qui s'affiche sur une Gauge et ensuite sur un graphique ( Il faudra utiliser un node "Function" pour obtenir la valeur de la température

[exemple.json](#)

```
var Temp = msg.payload.ESP32.Temperature;  
msg.payload = Temp;  
return msg;
```

- On peut utiliser aussi le noeud : "change" à la place du noeud "Function"





Pour avoir cela :



## La correction

[Correction exercice 2](#)

# Statistiques

Cet page a été consultée : Aujourd'hui: 1 Hier: 3 Jusqu'à maintenant: 1701

From:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

[https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=faire\\_preparation:soireeinfo:tp:start&rev=1767867514](https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=faire_preparation:soireeinfo:tp:start&rev=1767867514)

Last update: **2026/01/08 11:18**

