

Notice d'utilisation

Guide d'installation et d'utilisation de la communication bidirectionnelles d'ESP32 avec le protocole MQTT

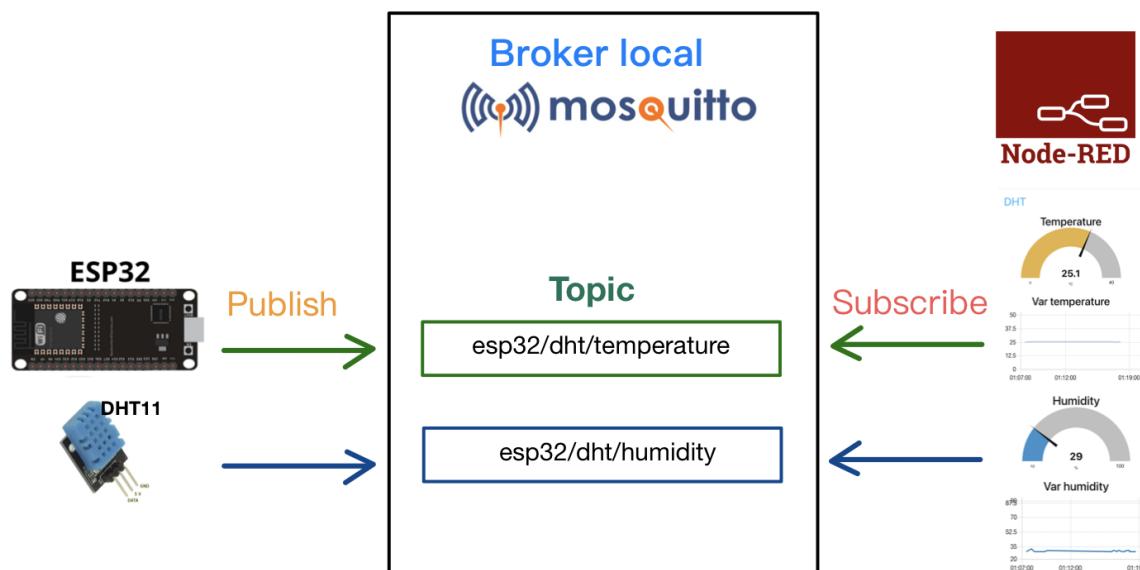
librairie + code + compte et mot de passe = zip

ÉLÉMENT NÉCESSAIRES

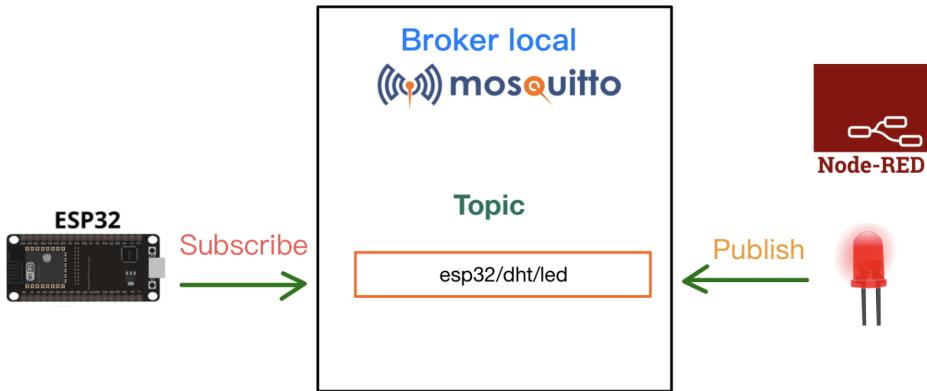
- fichier : **esp_mqtt_node_red.ino**
- **Arduino IDE**
- Cloud : **Node-RED** local
- Broker public : **Mosquitto** local
- **ESP32**
- **DHT11**
- 3 fils électriques
- Un cordon USB - mini USB

ARCHITECTURE

Dans le sens **montant** (uplink) de la communication bidirectionnelle, le capteur DHT11 **mesure** la température et l'humidité de l'environnement en temps réel. l'ESP32 **publie** ces données sur les topics suivants: esp32/dht/temperature et esp32/dht/humidity par WIFI. L'application Node-RED **souscrit** à ces 2 topics donc elle pourra recevoir les données lu par le capteur et les affiche sur le dashboard.



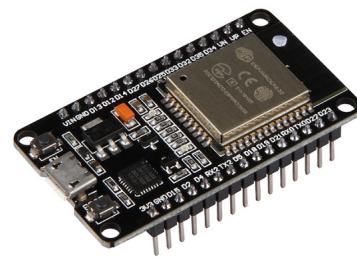
Dans le sens descendant (downlink), l'application Node-RED **publie** une message **on** ou **off** dans le topic esp32/dht/led et l'ESP32 **souscrit** à ce même topic, elle recevra donc le message pour allumer ou éteindre la LED.



DESCRIPTION DES ÉLÉMENS

- **ESP32**

ESP32 est un MCU riche en fonctionnalités avec une connectivité Wi-Fi et Bluetooth intégrée pour une large gamme d'applications. D'ici, on utilise son module Wi-Fi pour publier et souscrire les données.

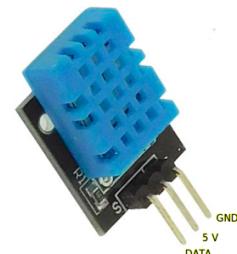


- **DHT11**

DHT11 est un capteur qui permet de mesurer la température et l'humidité, ses spécifications sont :

- Tension de fonctionnement : 3,3V à 5,5V
- Courant de fonctionnement : 0,3mA (mesure) 60uA (veille)
- Sortie : Données série
- Plage de température : 0°C à 50°C
- Plage d'humidité : 20% à 90%
- Précision : Température ±2°C et Humidité ±5%RH.
- Résolution : Température 1°C et Humidité 1%RH

Dans notre cas, on alimente DHT11 à 3,3V.



PRÉPARATION

- **INSTALLATION**

Veuillez installer les logiciels selon le système d'exploitation de votre ordinateur avant de les utiliser :

- IDE **Arduino** dans le site officiel selon votre système opérationnel:
<https://www.arduino.cc/en/software>
- **Node-RED** en local sous MacOS avec : sudo snap install node-red
Utiliser la commande node-red pour démarrer Node-RED dans votre terminal, et Ctrl-C ou fermer la fenêtre du terminal pour arrêter Node-RED.

Source : <https://nodered.org/docs/getting-started/local>

- **Mosquitto** en local sous MacOS avec : brew install mosquitto

Source : <https://mosquitto.org/download/>

Lorsque Mosquitto est installé, il faut configurer mosquitto en modifiant les paramètres du fichier **mosquitto.conf** dans le répertoire /usr/local/etc/mosquitto.

- listener 1883
- allow_anonymous true

Vous pouvez tester l'installation de **Mosquitto** avec la commande de démarrage :

/usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf

Les informations suivants seraient affiché dans le terminal :

```
[Macintosh] as liyunyang in ~
└──→ /usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
1637618700: mosquitto version 2.0.10 starting
1637618700: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1637618700: Opening ipv6 listen socket on port 1883.
1637618700: Opening ipv4 listen socket on port 1883.
1637618700: mosquitto version 2.0.10 running
```

Remarque : s'il y a une erreur : Error: Address already in use , vous pouvez d'abord trouver le PID du processus de mosquitto exécuté : ps -ef | grep mosquitto

Puis terminer ce processus avec : sudo kill PID_Mosquitto.

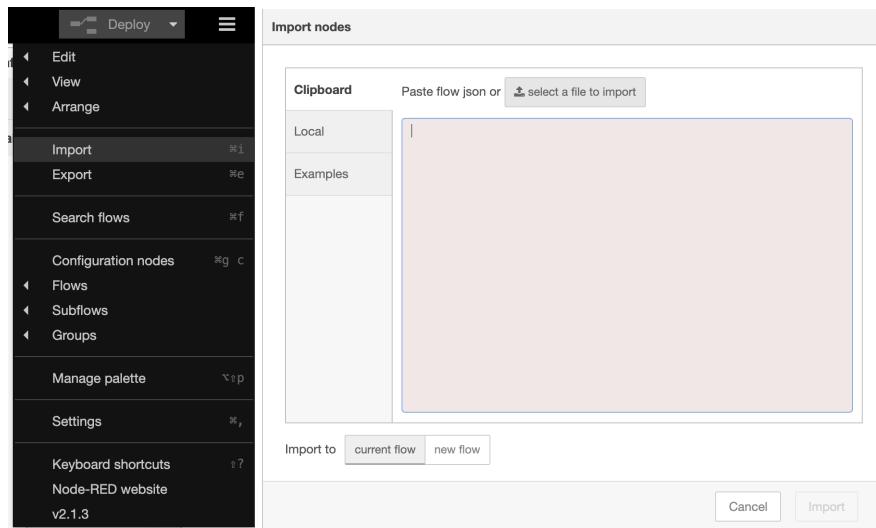
```
[Macintosh] as liyunyang in ~
└──→ /usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
23:04:16
1637618669: mosquitto version 2.0.10 starting
1637618669: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1637618669: Opening ipv6 listen socket on port 1883.
1637618669: Error: Address already in use
23:04:16 exit:1

[Macintosh] as liyunyang in ~
└──→ ps -ef | grep mosquitto
      501 16205      1  0 11:09AM ??          0:11.19 /usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
23:04:29
      501 22499 21758  0 11:04PM ttys001  0:00.00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git
--exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox mosquitto
23:04:38

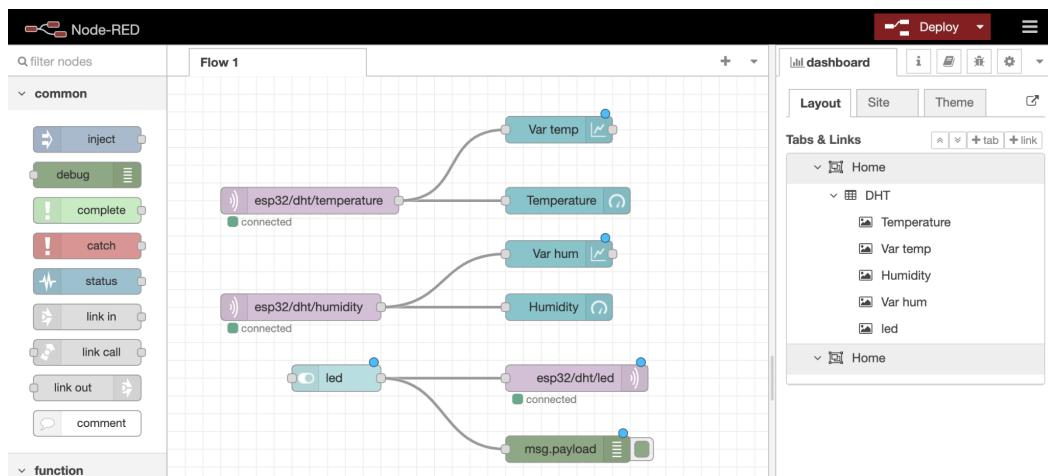
[Macintosh] as liyunyang in ~
└──→ sudo kill 21758
```

- **Création d'une application Node-RED**

Pour créer votre application Node-RED, vous devez ouvrir le lien **IP_Réseau:1880** dans un navigateur avec votre IP du réseau. Puis cliquez sur <Import> en haut à droite du site. Puis cliquez sur <**select a file to import**> pour importer le fichier conlut “**mqtt.json**” dans le dossier.

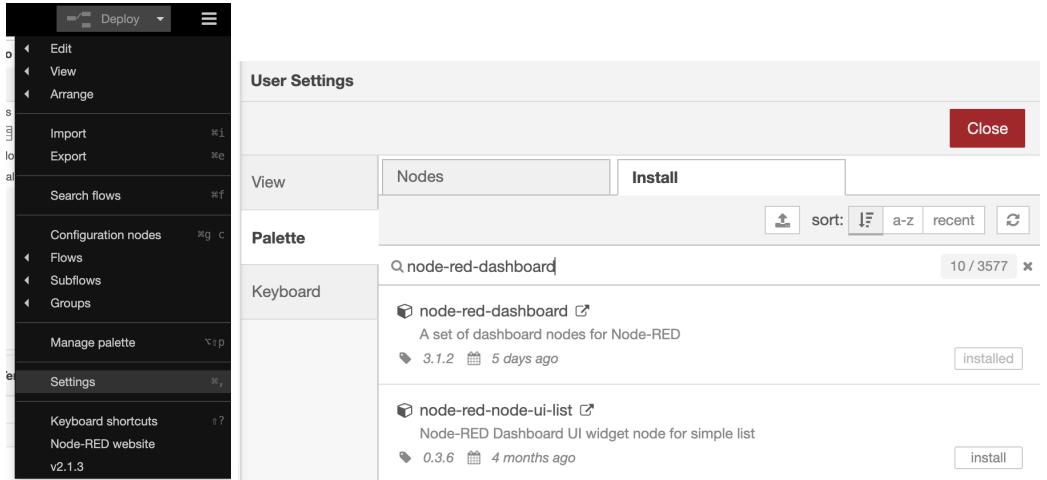


Cliquer sur <Import> en bas de la fenêtre ouverte. Vous auriez un flow comme la figure suivante.



Remarque :

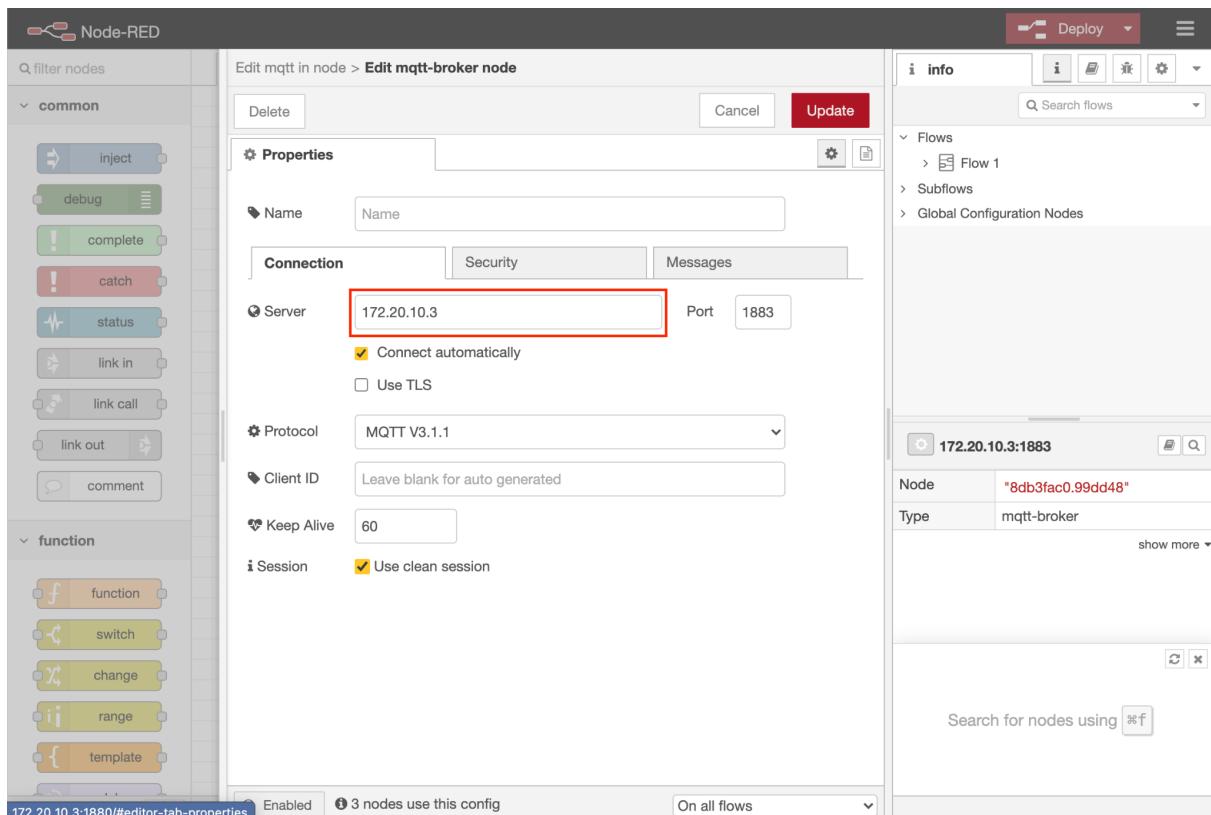
Si vous ne trouvez pas les nodes <gauge> et <chart>, allez sur le bouton à côté de <Deploy>, allez dans **Settings -> Install -> Palette**. Cherchez et installez *node red bashboard*, vous pouvez alors utiliser ces nodes.



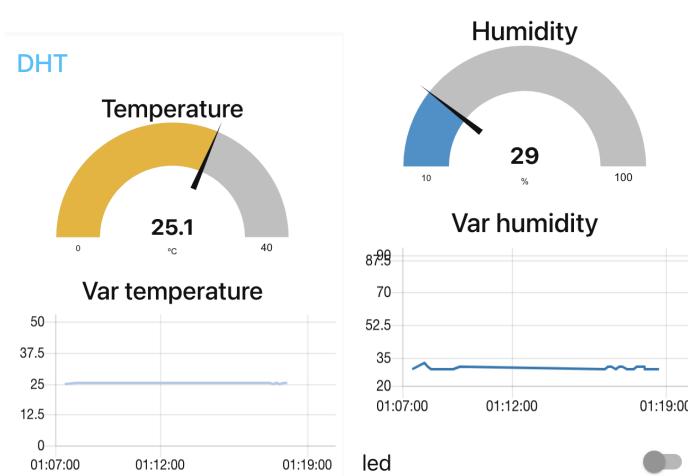
Cliquez ensuite sur le node violet au terme esp32/dht/temperature, puis cliquez sur le petit case du stylo à la ligne de Serveur.

The screenshot shows the Node-RED interface with a flow titled "Flow 1". On the left, the palette contains nodes like inject, debug, complete, catch, status, link in, link call, link out, comment, function, switch, change, range, and template. In the center, a flow is being edited. A node labeled "esp32/dht/temperature" is highlighted with a red box. To its right, another node labeled "esp32/dht/temperature" is connected. Below these nodes is a "led" node. On the far right, the "info" sidebar displays the node's details: Node "5a45b8da.52b0d8", Type "mqtt in", and a note to import a flow by dragging its JSON or using the "File" menu. The "Server" field in the properties panel of the selected node is also highlighted with a red box.

Modifiez l'IP de la case rouge à votre IP du réseau, puis cliquez sur <Update> et ensuite <Done>.



Faites la même manipulation pour les deux autres nodes violets. Puis cliquez sur <Deploy> en haut à droite. Maintenant, vous ouvrez le site **Votre_IP_Réseau:1880** avec /ui à la fin, vous allez trouver des graphiques de la température et de l'humidité comme la figure suivante:

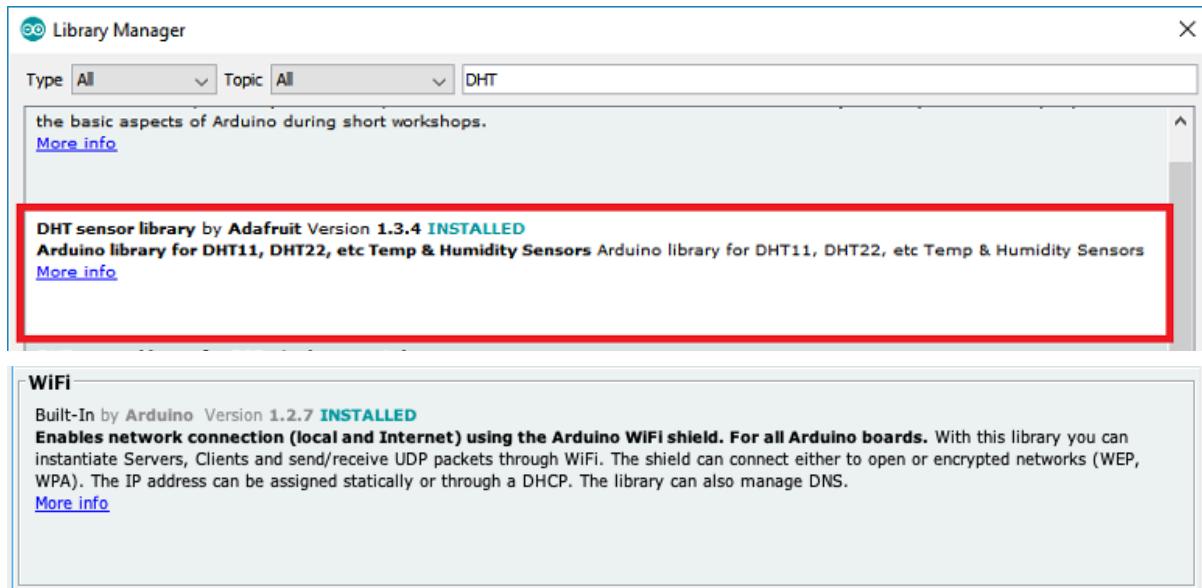


- **Configuration de Arduino**

Démarrer Arduino et déplacez les dossiers **async-mqtt-client-master** sous le dossier des bibliothèques d'installation de votre IDE Arduino.

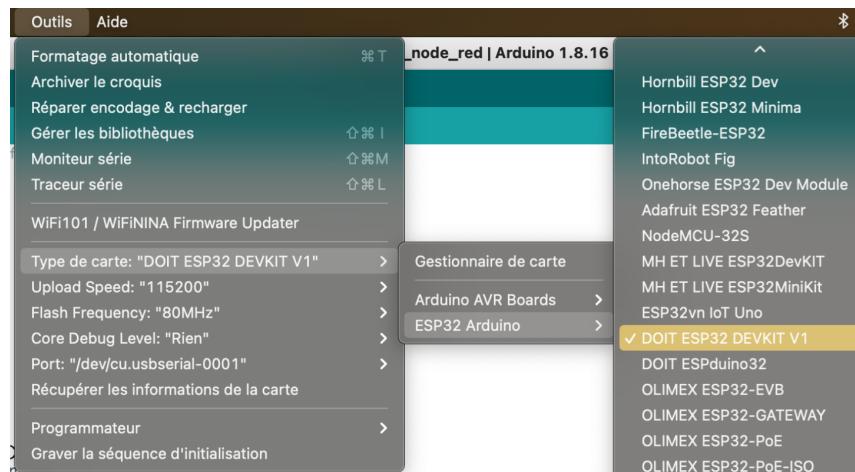
Allez aussi dans **Sketch > Include Library > Manage Libraries**. Le gestionnaire de bibliothèques devrait s'ouvrir. Recherchez "DHT" dans la boîte de recherche et installez la

bibliothèque DHT d'Adafruit. Recherchez et installez aussi la librairie “**WiFi**”, puis réouvrez Arduino.



Vous pouvez également accéder à **Sketch > Include Library > Add . ZIP library** et sélectionnez la bibliothèque que vous venez de télécharger.

Il faut aussi configurer le board du Arduino en **DOIT ESP32 DEVKIT V1**. Tout d'abord, ouvrez la fenêtre des préférences de l'IDE Arduino. Allez dans **Arduino > Préférences**. Ensuite, saisissez le lien suivant dans le champ “**Additional Boards Managers URLs**” : https://dl.espressif.com/dl/package_esp32_index.json . Puis, cliquez sur **OK**. Après, choisir le board comme la figure suivante :



Plus de détaille sur le site :

<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-mac-and-linux-instructions/>

UTILISATION DU SYSTÈME

Mise en marche

➤ Connexion en même réseau

Connectez votre ordinateur avec un réseau privé plutôt qu'un réseau public, ouvrez un terminal et trouver l'IP de ce réseau avec ifconfig, notez-le.

➤ Démarrage de Node-RED et Mosquitto

Vous devez démarrer Node-RED avec node-red dans un terminal, vérifiez que Started flows apparaît dans le terminal comme la figure suivante.

```
[Macintosh] as liyunyang in ~
└── node-red
22 Nov 23:04:11 - [info]

Welcome to Node-RED
=====

22 Nov 23:04:11 - [info] Node-RED version: v2.1.3
22 Nov 23:04:11 - [info] Node.js version: v16.3.0
22 Nov 23:04:11 - [info] Darwin 20.5.0 x64 LE
22 Nov 23:04:11 - [info] Loading palette nodes
22 Nov 23:04:12 - [info] Dashboard version 3.1.2 started at /ui
22 Nov 23:04:12 - [info] Settings file : /Users/liyunyang/.node-red/settings.json
22 Nov 23:04:12 - [info] Context store : 'default' [module=memory]
22 Nov 23:04:12 - [info] User directory : /Users/liyunyang/.node-red
22 Nov 23:04:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Nov 23:04:12 - [info] Flows file : /Users/liyunyang/.node-red/flows.json
22 Nov 23:04:12 - [info] Server now running at http://127.0.0.1:1880/
22 Nov 23:04:12 - [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 Nov 23:04:12 - [info] Starting flows
22 Nov 23:04:12 - [info] Started flows
```

Puis ouvrez un autre terminal et démarrez Mosquitto avec

`/usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf`

Vous aurez les informations comme la figure suivante.

```
[Macintosh] as liyunyang in ~
└── /usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
1637618700: mosquitto version 2.0.10 starting
1637618700: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1637618700: Opening ipv6 listen socket on port 1883.
1637618700: Opening ipv4 listen socket on port 1883.
1637618700: mosquitto version 2.0.10 running
```

Si le Node-RED est bien connecté avec Mosquitto, alors vous auriez les informations du Node-RED et de Mosquitto comme les figures suivantes :

```
[Macintosh] as liyunyang in ~
└── node-red
22 Nov 23:04:11 - [info]

Welcome to Node-RED
=====

22 Nov 23:04:11 - [info] Node-RED version: v2.1.3
22 Nov 23:04:11 - [info] Node.js version: v16.3.0
22 Nov 23:04:11 - [info] Darwin 20.5.0 x64 LE
22 Nov 23:04:11 - [info] Loading palette nodes
22 Nov 23:04:12 - [info] Dashboard version 3.1.2 started at /ui
22 Nov 23:04:12 - [info] Settings file : /Users/liyunyang/.node-red/settings.js
22 Nov 23:04:12 - [info] Context store : 'default' [module=memory]
22 Nov 23:04:12 - [info] User directory : /Users/liyunyang/.node-red
22 Nov 23:04:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Nov 23:04:12 - [info] Flows file : /Users/liyunyang/.node-red/flows.json
22 Nov 23:04:12 - [info] Server now running at http://127.0.0.1:1880/
22 Nov 23:04:12 - [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 Nov 23:04:12 - [info] Starting flows
22 Nov 23:04:12 - [info] Started flows
22 Nov 23:04:12 - [info] [mqtt-broker:8db3fac0.99dd48] Connected to broker: mqtt://172.20.10.3:1883
22 Nov 23:04:12 - [info] [mqtt-broker:10e78a89.5b4fd5] Connected to broker: mqtt://broker.mqtt-dashboard.com:1883
22 Nov 23:04:57 - [info] [mqtt-broker:8db3fac0.99dd48] Disconnected from broker: mqtt://172.20.10.3:1883
22 Nov 23:05:12 - [info] [mqtt-broker:8db3fac0.99dd48] Connected to broker: mqtt://172.20.10.3:1883

[Macintosh] as liyunyang in ~
└── /usr/local/opt/mosquitto/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
1637618700: mosquitto version 2.0.10 starting
1637618700: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1637618700: Opening ipv6 listen socket on port 1883.
1637618700: Opening ipv4 listen socket on port 1883.
1637618700: mosquitto version 2.0.10 running
1637618712: New connection from 172.20.10.3:60613 on port 1883.
1637618712: New client connected from 172.20.10.3:60613 as nodered_63752aa00db5e87b (p2, c1, k60).
```

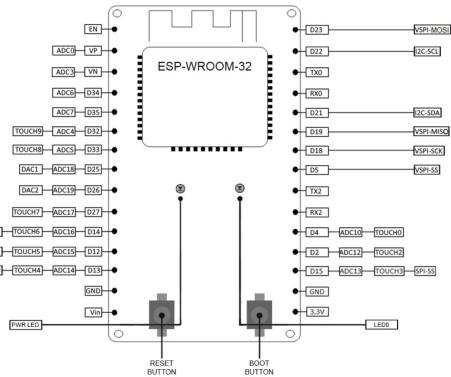
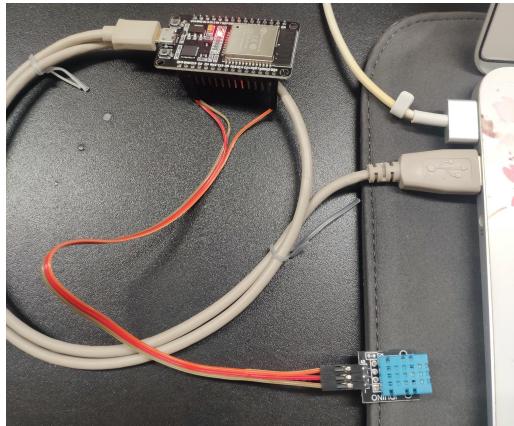
Remarque :

Ouvrez l'application Node-RED avec le site : **Votre_IP_Réseau:1880** (dans la figure ci-dessus, le lien est **172.20.10.3:1880**, 1880 est le port du serveur par défaut).

➤ ASSEMBLAGE

Il faut relier le capteur DHT11 avec l'ESP32 et alimenter l'ESP32 par votre ordinateur avec un cordon USB - micro USB comme la figure suivant:

- File **orange** relier le pin 23 de l'ESP32 et pin DATA de DHT11
- File **rouge** relier le pin 3,3V de l'ESP32 et le pin d'alimentation de DHT11
- File **marron** relier le pin GND de l'ESP32 et le pin GND de DHT11



➤ Programmation ESP32

Déplacez d'abord le dossier **esp_mqtt_node_red** sous le répertoire Arduino. Ouvrez le fichier **esp_mqtt_node_red.ino** dans Arduino, modifiez l'ID de votre WiFi partagé, le mot de passe et l'IP de réseau WiFi.

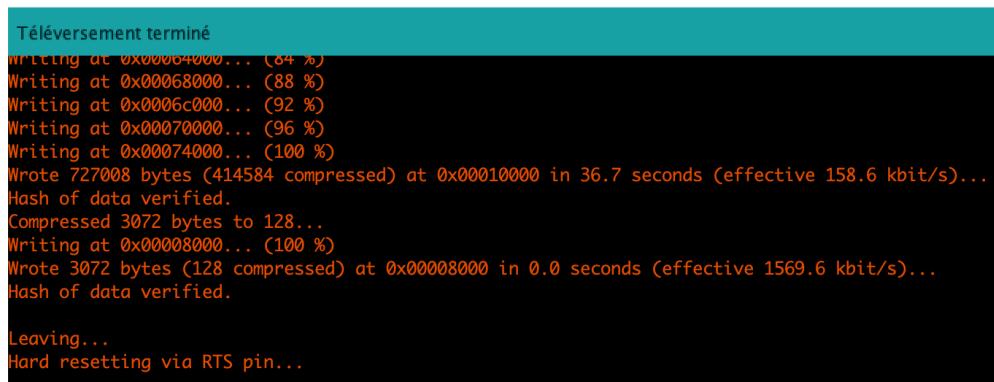
```
// Modifier ID de WIFI et le mot de passe de votre
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "xxxx"

// Mosquitto MQTT Broker avec IP de vote réseau
#define MQTT_HOST IPAddress(172, 20, 10, 3)
```

Maintenant, cliquez sur le 2e bouton pour programmer la carte ESP32



Après le succès de la compilation, il s'affiche **Connecting** dans le terminal. Vous pouvez appuyer sur le bouton **Boot** de l'ESP32 en même temps. Attendez le téléchargement se termine.



Visualisation des données

Vous pouvez cliquer sur le bouton en haut à droite d'Arduino pour consulter les informations de debug.

Moniteur série 

Les données de capteurs et l'information du contrôle de LED seront affichées dans cette fenêtre.

```
01:16:00.828 -> load:0x3fff001c,len:1044
01:16:00.828 -> load:0x40078000,len:10124
01:16:00.828 -> load:0x40080400,len:5856
01:16:00.828 -> entry 0x400806a8
01:16:01.065 ->
01:16:01.099 -> Connecting to Wi-Fi...
01:16:01.203 -> [WiFi-event] event: 0
01:16:01.203 -> [WiFi-event] event: 2
01:16:03.369 -> [WiFi-event] event: 4
01:16:03.406 -> [WiFi-event] event: 7
01:16:03.406 -> WiFi connected
01:16:03.406 -> IP address:
01:16:03.406 -> 172.20.10.4
01:16:03.406 -> Connecting to MQTT...
01:16:05.081 -> Connected to MQTT.
01:16:05.081 -> Session present: 0
01:16:05.081 -> Subscribing at QoS 2, packetId: 1
01:16:05.696 -> Subscribe acknowledged.
01:16:05.729 -> packetId: 1
01:16:05.729 -> qos: 2
01:16:11.095 -> Publishing on topic esp32/dht/temperature at QoS 1, packetId: 2Message: 25.30
01:16:11.095 -> Publishing on topic esp32/dht/humidity at QoS 1, packetId 3: Message: 29.00
01:16:11.856 -> Publish acknowledged. packetId: 2
01:16:12.770 -> Publish acknowledged. packetId: 3
01:16:21.093 -> Publishing on topic esp32/dht/temperature at QoS 1, packetId: 4Message: 25.10
01:16:21.093 -> Publishing on topic esp32/dht/humidity at QoS 1, packetId 5: Message: 30.00
01:16:21.786 -> Publish acknowledged. packetId: 4
01:16:22.581 -> Publish acknowledged. packetId: 5
01:16:31.077 -> Publishing on topic esp32/dht/temperature at QoS 1, packetId: 6Message: 25.10
01:16:31.077 -> Publishing on topic esp32/dht/humidity at QoS 1, packetId 7: Message: 30.00
01:16:31.597 -> Publish acknowledged. packetId: 6
01:16:32.725 -> Publish acknowledged. packetId: 7
01:16:46.249 -> **new_payload: off
01:16:46.249 -> off
01:16:47.784 -> Publish received.
01:16:47.784 -> topic: esp32/dht/led
01:16:47.784 -> qos: 0
01:16:47.784 -> dup: 0
01:16:47.784 -> retain: 0
01:16:47.784 -> len: 2
01:16:47.784 -> index: 0
01:16:47.784 -> total: 2
01:16:47.784 -> on?
01:16:47.784 -> **new_payload: on
01:16:47.784 -> on
01:16:48.714 -> Publish received.
01:16:48.714 -> topic: esp32/dht/led
01:16:48.714 -> qos: 0
01:16:48.714 -> dup: 0
01:16:48.714 -> retain: 0
01:16:48.714 -> len: 3
01:16:48.714 -> index: 0
01:16:48.714 -> total: 3
01:16:48.714 -> off?
01:16:48.714 -> **new_payload: off
01:16:48.714 -> off
```

Allez sur le site de dashboard de Node-RED : Votre_IP_Réseau:1880/ui , vous pouvez visualiser clairement les données de la température et de l'humidity et contrôler le LED.