



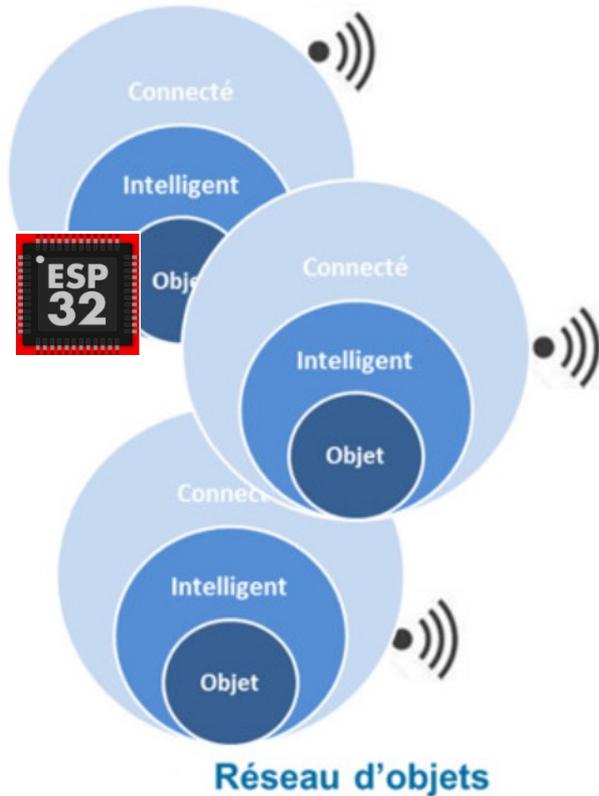
Projet ESP
Présentation IOT

Sommaire

- › C'est quoi l'IOT?
- › L'ESP est un IOT
- › C'est quoi TASMOTA?
- › L'ESP et TASMOTA
- › C'est quoi MQTT?
- › C'est quoi Node-Red?
- › L'ESP avec TASMOTA, MQTT et Node-Red
- › Exemple par un projet

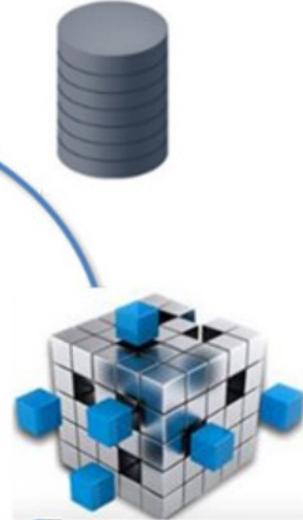
C'est quoi l'IOT

C'est quoi l'IOT? Internet des Objets



APPLICATIF
Fourniture de service

Hébergement
(serveurs, stockage, cloud)

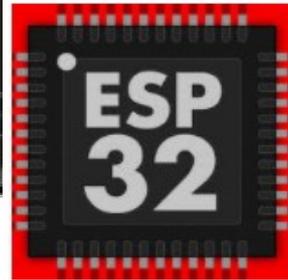
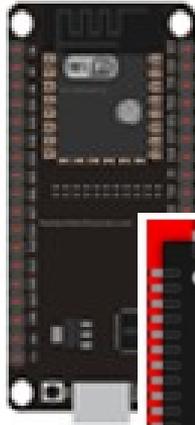


TRAITEMENT
Big data, analyse des données

L'ESP est un IOT

Les IOT: ESP

Capteurs



Actionneurs

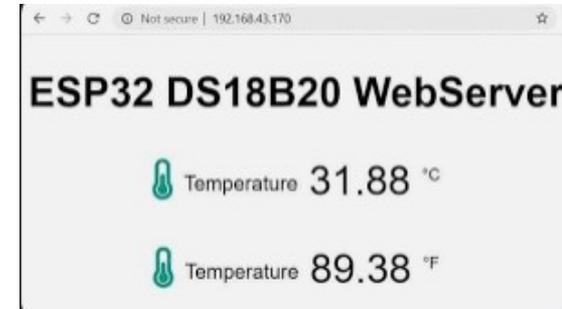


Intelligence IDE

```
sketch_mar10a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```



Portail (WebUI)



Exemple

·1 ESP

- 1 bouton
- 1 sonde DHT11
- 1 buzzer
- IDE

·Code:

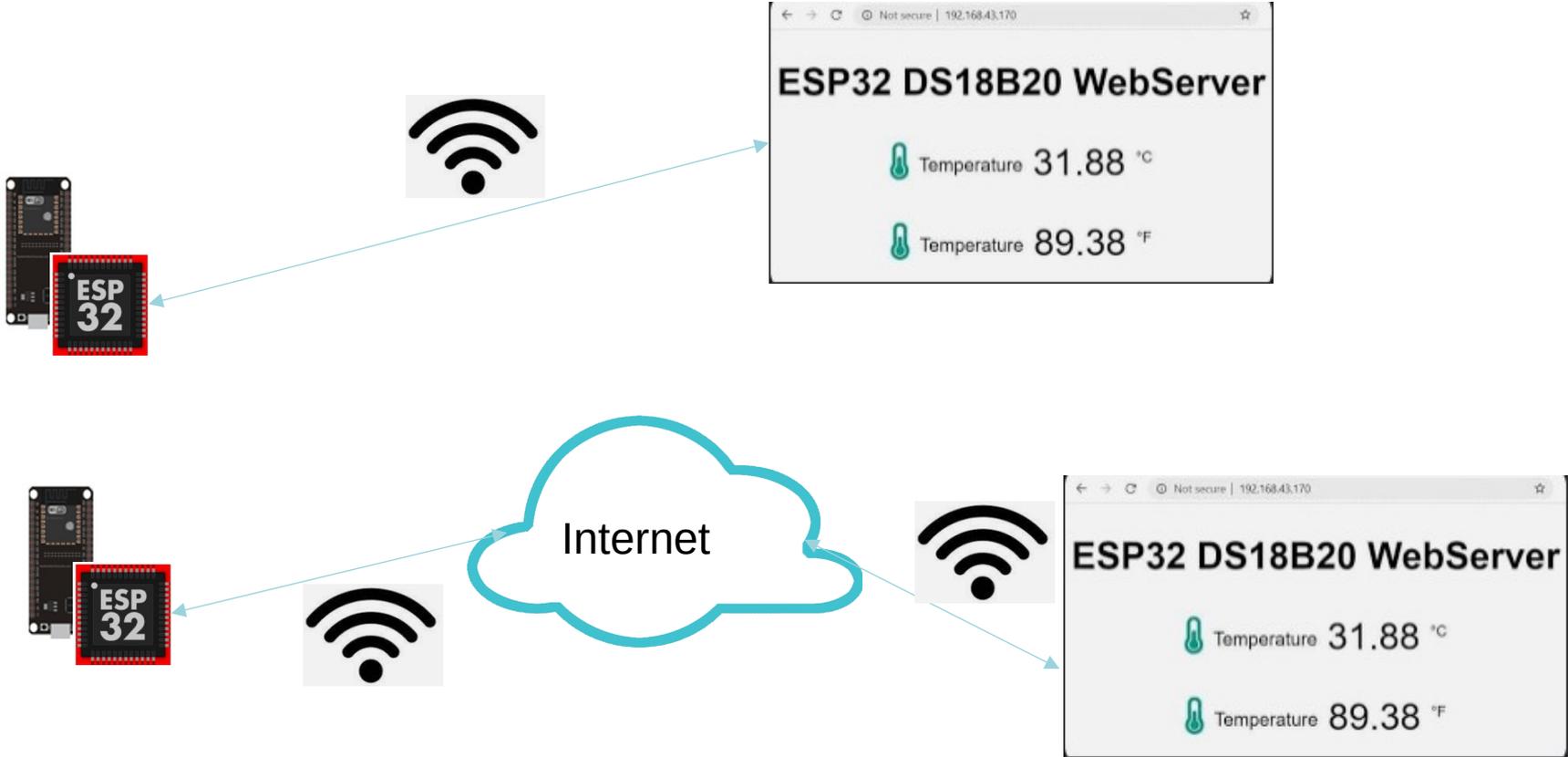
Déclarations capteurs, Actionneurs

Déclaration WIFI

Gestion du DHT11, du bouton, du buzzer

Interface WEB

Les IOT: ESP + IDE



Les IOT: ESP + IDE

Avantages	Inconvénients
Autonomie du projet	Projet lié à 1 IOT
Efficace pour un Projet	Code complexe et long
Sécurité	Maintenance, MAJ
	Peu d'ouverture

C'est quoi TASMOTA?



Unification:

- Un seul firmware pour vos objets ESPXXX
- On peut reflasher des objets du commerce (bridés ALEXA, etc)

Simplification

- Paramétrer au lieu de coder

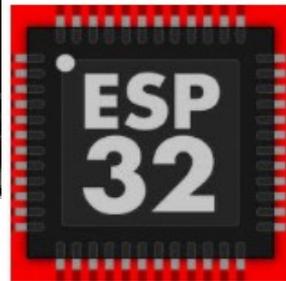
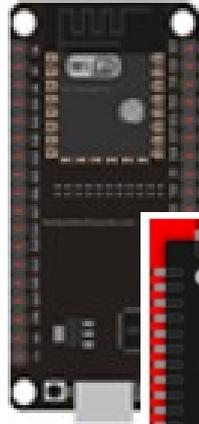
Ouverture

- MQTT

L'ESP avec TASMOTA

Les IOT: ESP + TASMOTA

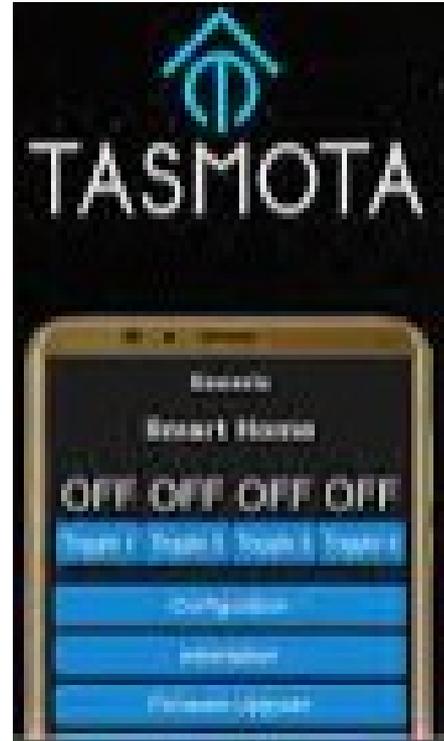
Capteurs



Actionneurs



Intelligence



Exemple

·1 ESP

- 1 bouton
- DHT11
- 1 buzzer

·TASMOTA:

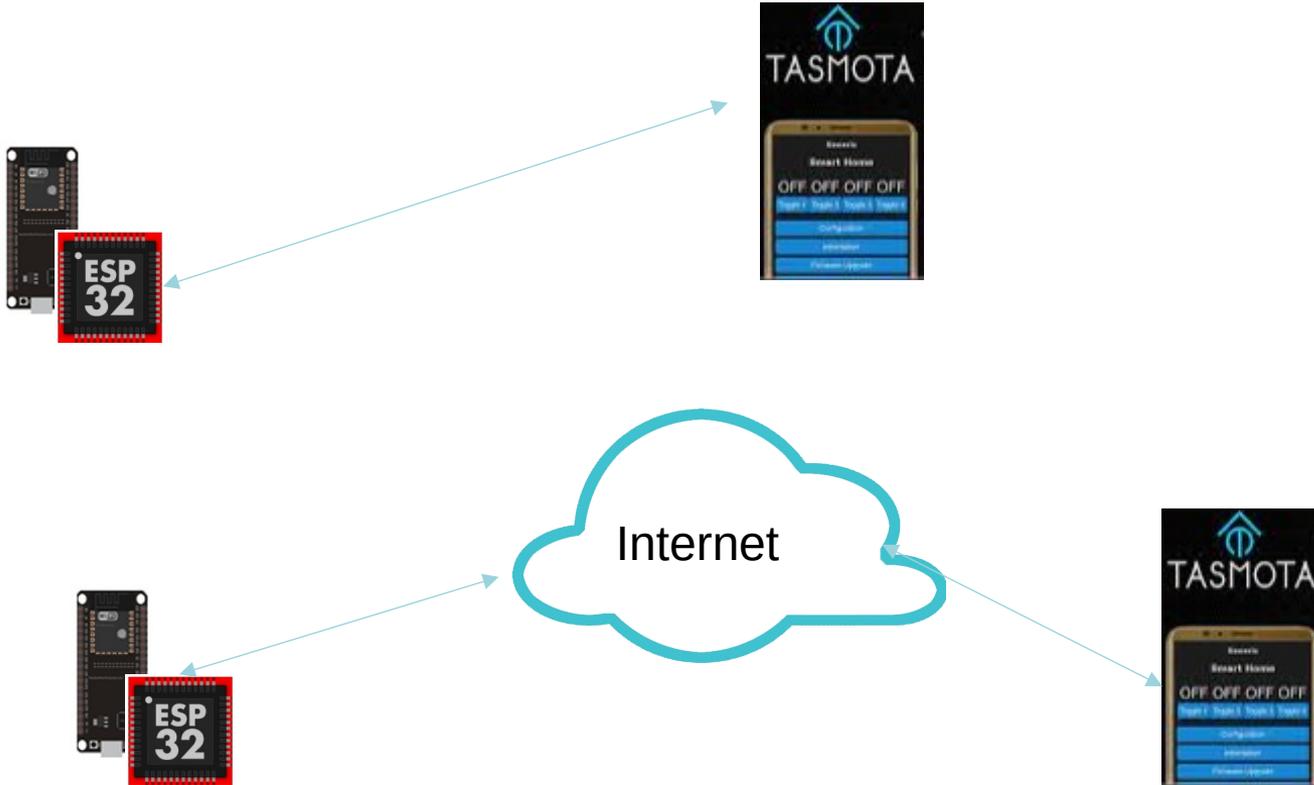
Paramètres DHT11, du bouton, du buzzer

Déclaration WIFI

paramétrages des options

Interface WEB par défaut

Les IOT: ESP + TASMOTA



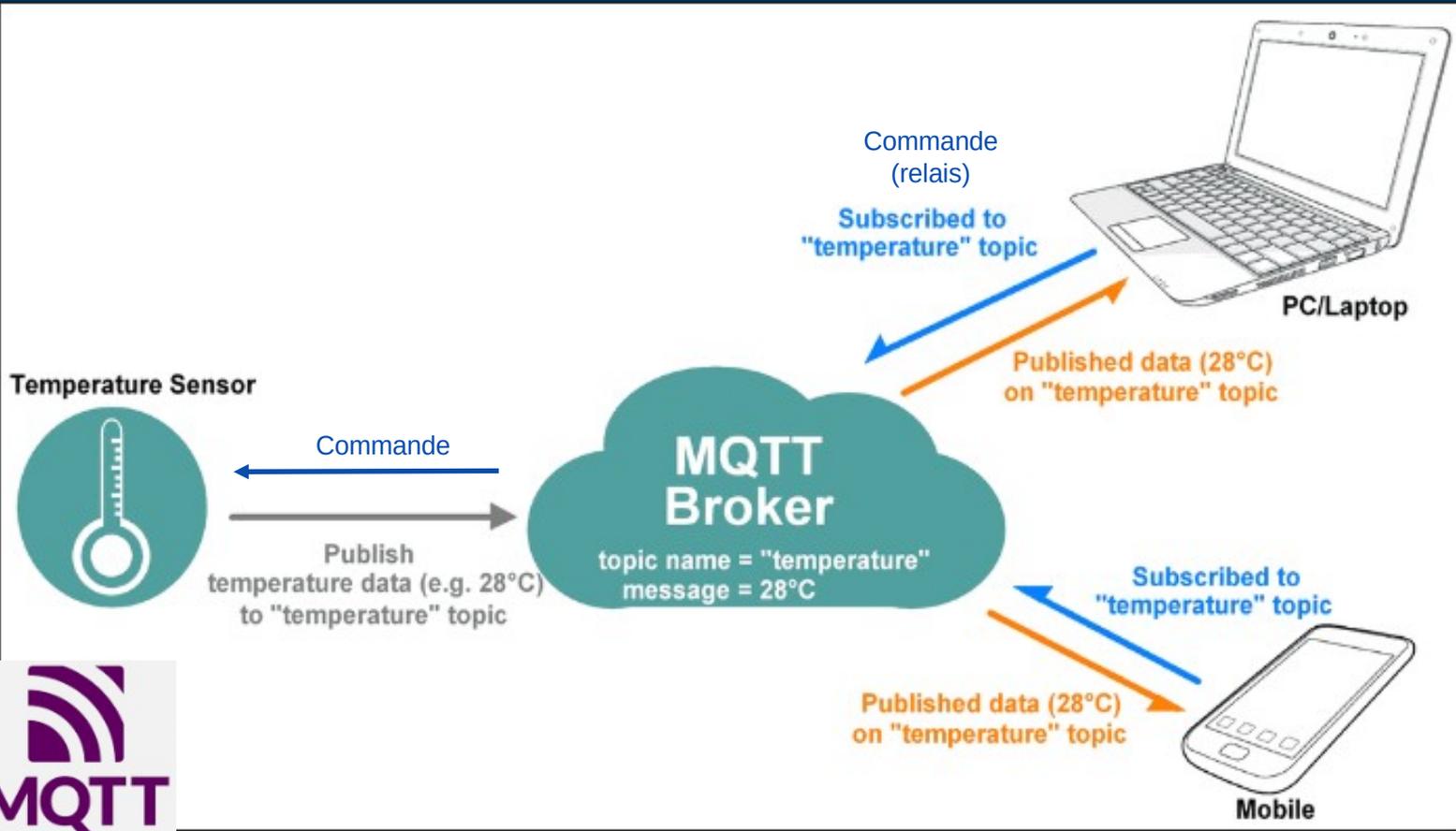
Les IOT: ESP + TASMOTA



Avantages	Inconvénients
Pas de code	Partie intelligente par paramètres et règles
Maintenance, MAJ, évolutions	
Ouverture (MQTT, etc.)	
De base, de nombreuses fonctionnalités (Boutons, capteurs, actionneurs, règles, etc)	

C'est quoi MQTT?

MQTT



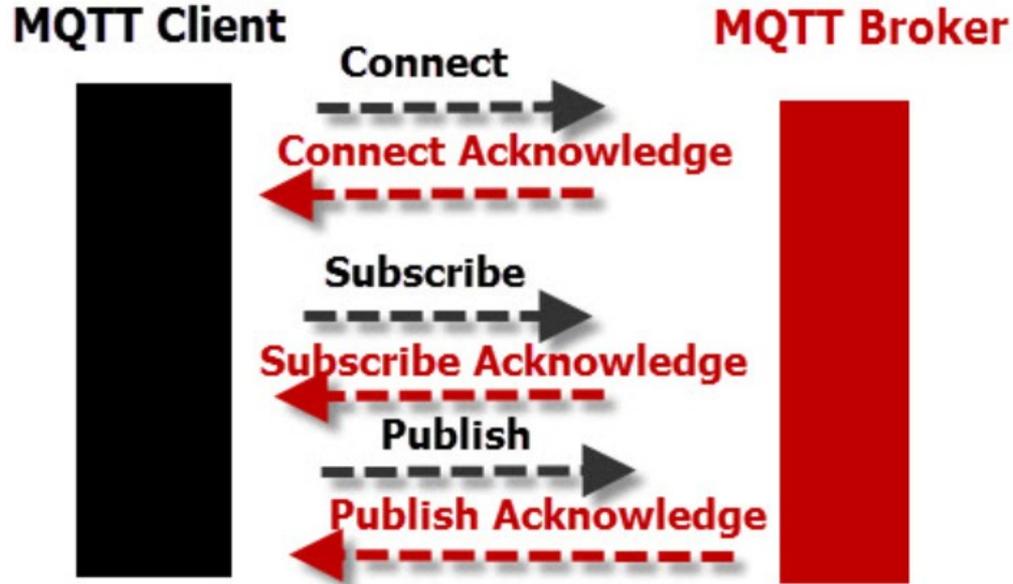
MQTT, c'est quoi?

MQTT est un protocole TCPI/IP de transport de données (messages)

Un Broker MQTT, c'est quoi?

C'est un relais de messages MQTT

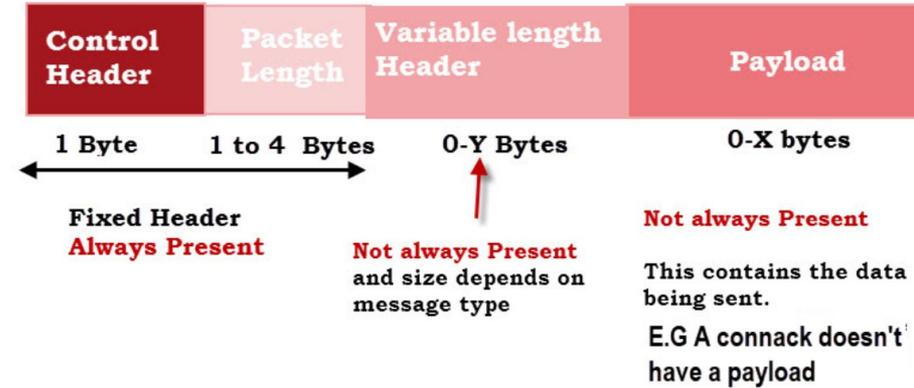
- Connexion,
- Souscription
- Publication



MQTT Client To Broker Protocol

Messages MQTT

- topic
- payload
- (<https://gagnebin.tech/glossaire/le-topic-mqtt>)



MQTT Standard Packet Structure

Un topic (message) MQTT est composé de plusieurs éléments, donc voici les principaux :

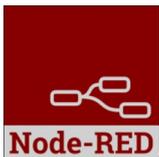
Le topic : L'identifiant dans broker

Le payload : les données à envoyer

Exemple: topic: ESP1/maison/rezdechaussee/entree/

payload: temperature = 19°C, humidité= 50%, etc.

C'est quoi Node-Red?



Low Code

The screenshot displays the Node-RED interface with a workflow for processing PubNub data. The workflow is divided into two main sections: "Subscribe" and "Publish".

Subscribe Section:

- Subscribe** node connects to **PubNub raw data**.
- PubNub raw data** connects to a **json** node.
- The **json** node connects to a **Filter Messages** node.
- The **Filter Messages** node has multiple outputs connected to various data nodes: **mongodb**, **weather data**, **news data**, **command data**, **motion data**, **Tweet "Somebody's Home"**, **traffic signal data**, and **other data**.

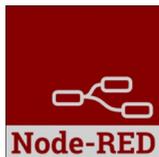
Publish Section:

- Publish** node connects to an **inject** node.
- The **inject** node connects to a function node **f** with the code: `{"news": "Hello World, from Node-RED"}`.
- Another **inject** node connects to a function node **f** with the code: `{"command": "lights_off"}`.
- A third **inject** node connects to a function node **f** with the code: `{"command": "lights_on"}`.
- A fourth **inject** node connects to a function node **f** with the code: `{"command": "disco"}`.
- A fifth **inject** node connects to a function node **f** with the code: `{"command": "disco"}`.
- A sixth **inject** node connects to a function node **f** with the code: `{"command": "disco"}`.
- A seventh **inject** node connects to a function node **f** with the code: `{"command": "disco"}`.
- All function nodes connect to the **iot** node, which is labeled "published".
- The **iot** node connects to a **Tweet mention: InternetOfLego** node.

Library (bibliothèque): A list of nodes is visible on the left, including **delay**, **trigger**, **comment**, **http request**, **tcp request**, **switch**, **change**, **range**, **csv**, **html**, **json**, **xml**, **rbe**, and **social**.

Code: The word "Code" is written in red in the center of the workflow.

Debug: The word "Debug" is written in red above the debug console. The debug console shows a list of messages with their payloads, including "Hello World, from Node-RED", "trafficSignalState:2", "trafficSignalState:3", "trafficSignalState:0", "Wordpress is connected!", "Disco Party!", "trafficSignalState:1", "trafficSignalState:2", "trafficSignalState:3", and "trafficSignalState:0".



Dashboard

127.0.0.1:1880/ui/#/0

Good afternoon, lheb (your feeds today)

Sysinfo

OS	Linux
Hostname	backbox
platform	linux
Arch	ia32

Memory

1 Minute

90.01 Usage

Total Memory	4.076 GB
Free Memory	407.257 MB

24 Hours

%20:%3:%0	%20:%3:%0
-----------	-----------

Arduino

switch

LedState

slider

TwitterPosts

Lastest posts
Everything you need to know about the Switch, Nintendo's unique new video game console
<https://t.co/K3TFp5qaUA>

Sarahah-Notification

Saraha Inbox 0

Weather

weather today Clouds

Location Tunis

detail few clouds

The weather in Tunis at coordinates: 36.82, 10.17 is Clouds (few clouds).

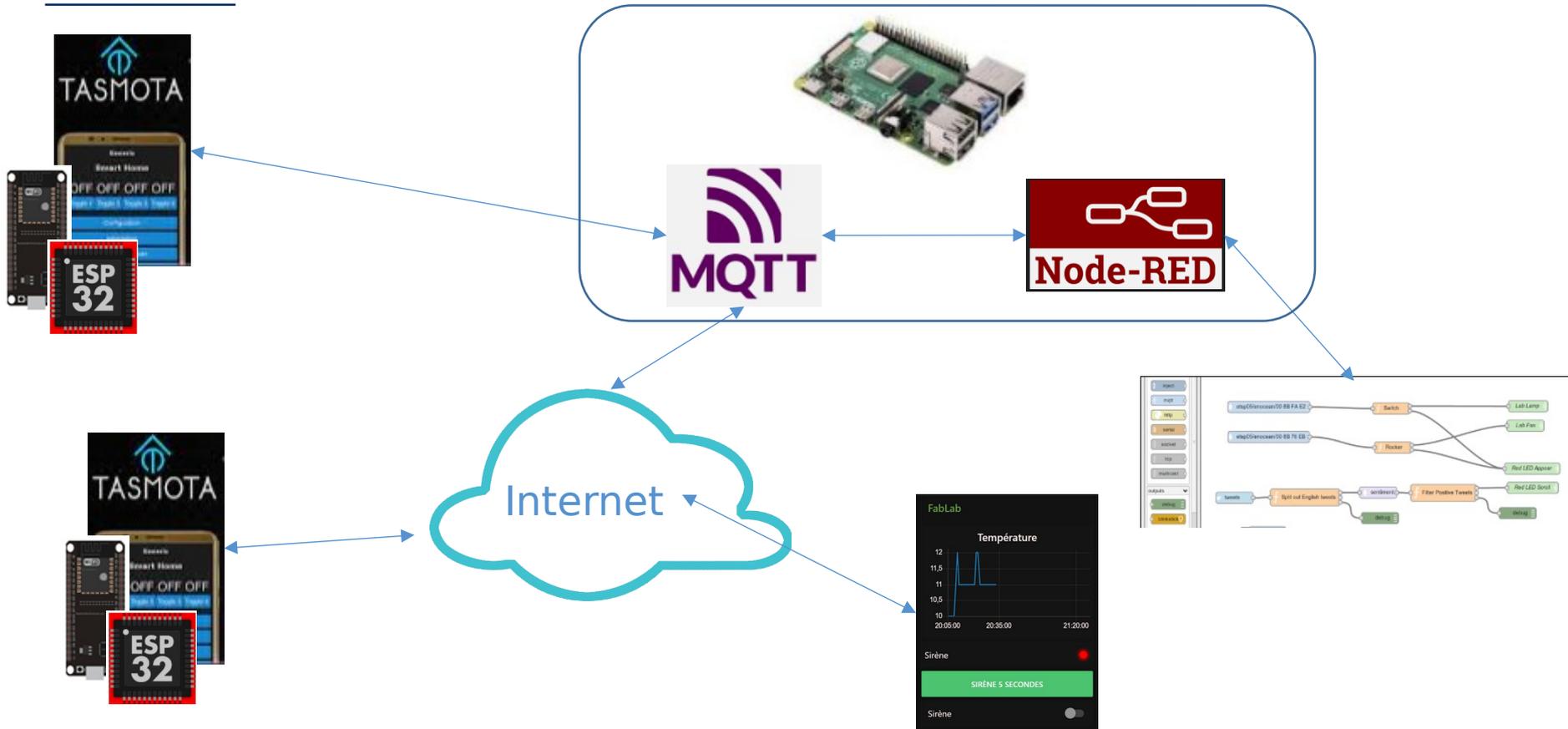
Tweets

Tweet

SUBMIT CANCEL

L'ESP avec TASMOTA, MQTT et Node-Red

Les IOT: ESP + TASMOTA + MQTT + Node-Red



Les IOT: ESP + TASMOTA + MQTT + Node-Red

Avantages	Inconvénients
Low Code	Nécessite un serveur
Rapidité de mise en œuvre du projet	Gestion de la sécurité
Mixer les IOTs	
Puissance de l'intelligence	
Fonctionnalités nombreuses	
Ouverture	
Dashboard	



Projet ESP32 Réalisation de A à Z

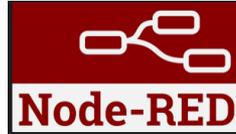
Identification



Nom: Livebox-5200
Password: Castel37110\$



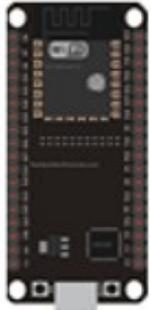
Adresse IP:192.168.1
Port: 1883



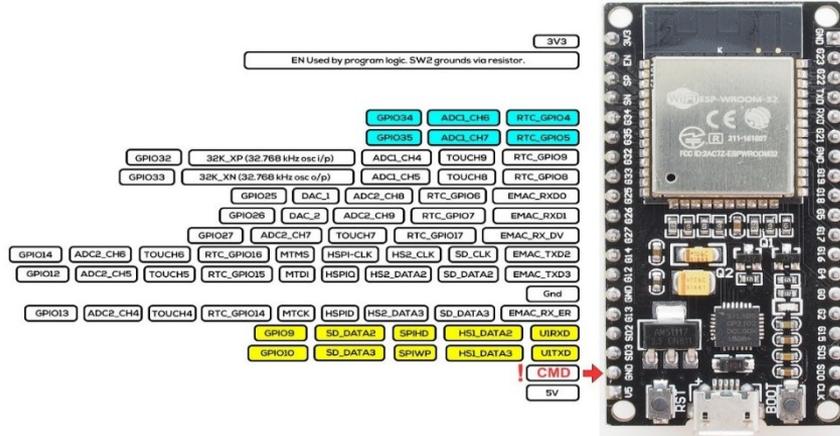
Adresse IP:
Port: 1880

Admin:192:1880
Dashboard: 192.168./ui

Adresse IP:
Nom:



Préparation de l'ESP

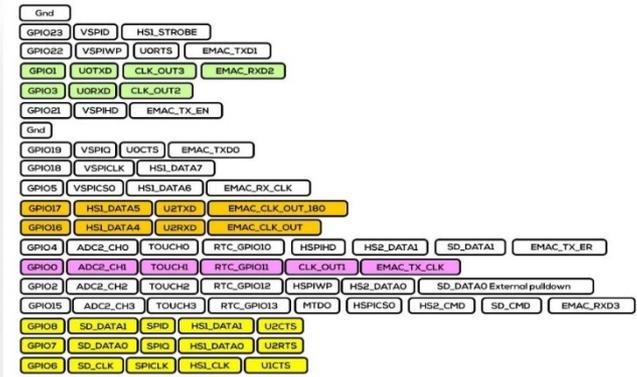


ADC: FSD = 4095 * 1.109V (Because 693mV gave 2550. Is the limit 1.0V?)

DAC: FSD = 255 * 3.19V (Vs = 3.3V). 127 gave 1.63V implying 3.3V FS.

Remapping peripherals:
uart = machine.UART1(L.baudrate=115200.tx=25.rx=26)

Value	Expected	Actual	Error %
10	0.13	0.21	2.4
20	0.26	0.33	2.1
127	1.64	1.63	-0.3
200	2.58	2.53	-1.5
240	3.11	3.01	-3
255	3.3	3.19	-3.3



Used for internal flash, not recommended for other use

Input only. No internal pullup or pulldown.

Used by USB/REPL

GPIO0 has a 5KΩ external pullup. SW0 grounds via 470Ω

Used on ESP32-WROVER-KIT etc to access external SPI RAM

ESP32-DRWD is the chip with embedded 2MB flash and the internal flash is connected to different pins (GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1)

L'ESP

Capteurs Actionneurs

27	Bouton (4)
14	Sirène (1)
19	DHT11



L'ESP: TASMOTA



Tasmota WebInstaller
[https://tasmota.github.io/
install](https://tasmota.github.io/install)

/

Install Tasmota

1. Connect the ESP device to your computer using USB or serial-to-USB adapter
2. Select the firmware variant suitable for your device
3. Hit "Install" and select the correct port or find help if no device found

Tasmota (english) ▼

All ▼

CONNECT

Tasmota Installer powered by ESP Web Tools

L'ESP: Configuration TASMOTA



WIFI

MQTT

Capteurs

Actionneurs

Options

L'ESP: TASMOTA

Tasmota: Les Options

<https://tasmota.github.io/docs/Commands/#set>

set

SetOption1

Set `button multipress` mode to

0 = allow all button actions (*default*)

1 = restrict to single to penta press and hold actions (i.e., disable inadvertent reset due to long press)

SetOption73

Detach buttons from relays and send multi-press and hold MQTT messages instead

0 = disable (*default*)

1 = enable

Example message: `{"Button1":{"Action":"SINGLE"}}`

SetOption146

1 = enable display of ESP32 internal temperature

L'ESP: TASMOTA

SetOption114 Detach switches from relays and send MQTT messages instead
0 = disable (default)
1 = enable
Example result: {"Switch1":{"Action":"ON"}}

SwitchMode(X) 15

<https://tasmota.github.io/docs/Buttons-and-Switches/#swit> SwitchMode 15

Send only MQTT message on switch change. This will stop the switch from controlling power outputs.

```
tele/tasmota/SENSOR = {"Time":"2021-01-01T00:00:00","Switch1":"OFF"}  
tele/tasmota/SENSOR = {"Time":"2021-01-01T00:00:00","Switch1":"ON"}
```

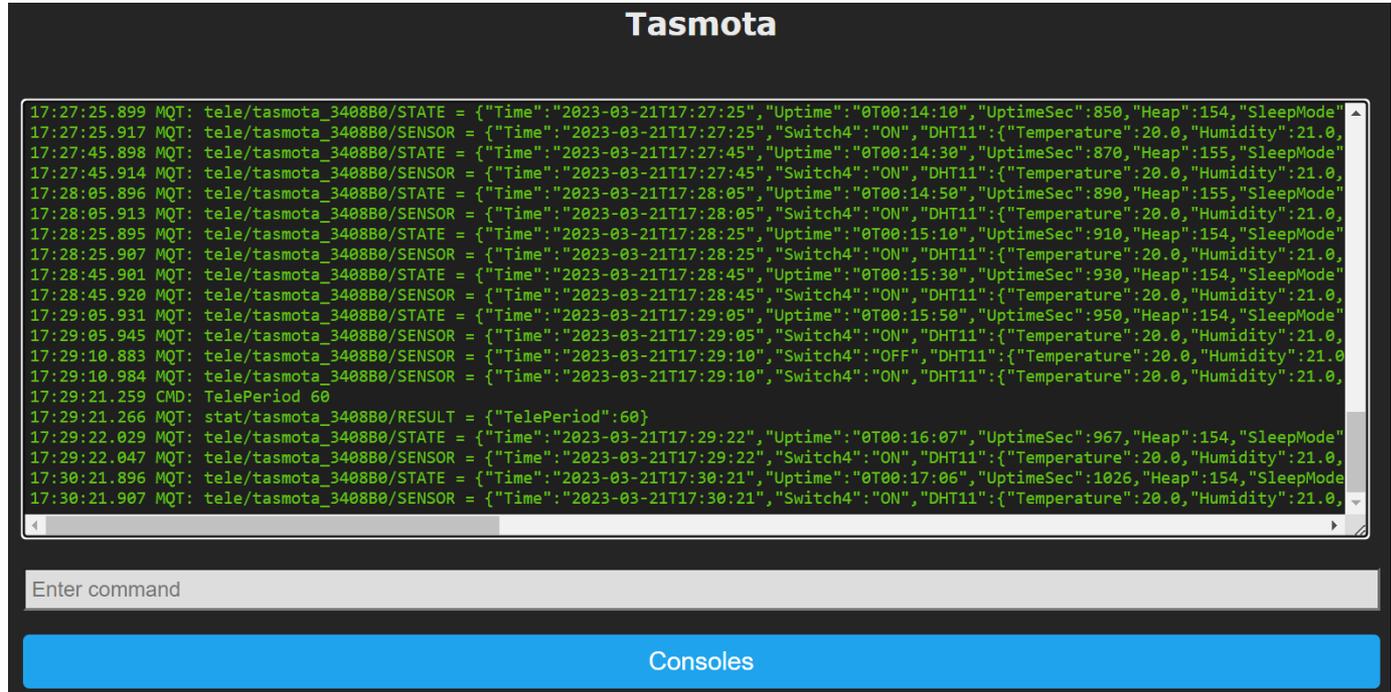
L'ESP: TASMOTA – La console

TelePeriod 60

SetOption146 1

SetOption114 1

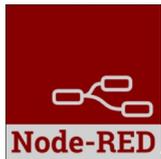
SetOption1 1



The screenshot shows the Tasmota console interface. At the top, the title "Tasmota" is displayed. Below it, a scrollable log window contains a series of MQTT messages. The messages include state updates, sensor readings, and a command execution. The command "TelePeriod 60" is shown as being executed, with the result {"TelePeriod":60}. Below the log, there is an input field labeled "Enter command" and a blue button labeled "Consoles".

```
17:27:25.899 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:27:25","Uptime":"0T00:14:10","UptimeSec":850,"Heap":154,"SleepMode":
17:27:25.917 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:27:25","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:27:45.898 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:27:45","Uptime":"0T00:14:30","UptimeSec":870,"Heap":155,"SleepMode":
17:27:45.914 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:27:45","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:28:05.896 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:28:05","Uptime":"0T00:14:50","UptimeSec":890,"Heap":155,"SleepMode":
17:28:05.913 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:28:05","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:28:25.895 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:28:25","Uptime":"0T00:15:10","UptimeSec":910,"Heap":154,"SleepMode":
17:28:25.907 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:28:25","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:28:45.901 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:28:45","Uptime":"0T00:15:30","UptimeSec":930,"Heap":154,"SleepMode":
17:28:45.920 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:28:45","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:29:05.931 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:29:05","Uptime":"0T00:15:50","UptimeSec":950,"Heap":154,"SleepMode":
17:29:05.945 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:29:05","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:29:10.883 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:29:10","Switch4":"OFF","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:29:10.984 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:29:10","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:29:21.259 CMD: TelePeriod 60
17:29:21.266 MQT: stat/tasmota_3408B0/RESULT = {"TelePeriod":60}
17:29:22.029 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:29:22","Uptime":"0T00:16:07","UptimeSec":967,"Heap":154,"SleepMode":
17:29:22.047 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:29:22","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
17:30:21.896 MQT: tele/tasmota_3408B0/STATE = {"Time":"2023-03-21T17:30:21","Uptime":"0T00:17:06","UptimeSec":1026,"Heap":154,"SleepMode":
17:30:21.907 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:30:21","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,
```

7 MQT: tele/tasmota_3408B0/SENSOR = {"Time":"2023-03-21T17:29:22","Switch4":"ON","DHT11":{"Temperature":20.0,"Humidity":21.0,"DewPoint":-3.0},"ESP32":{"Temperature":30.6},"TempUnit":"C"}



MQTT: souscription

The screenshot shows the Node-RED interface with an MQTT node configuration panel and a debug console. The configuration panel is titled "Edit mqtt in node" and includes buttons for "Delete", "Cancel", and "Done". The "Properties" section contains the following settings:

- Server: MQTTGerbault
- Action: Subscribe to single topic
- Topic: tele/tasmota_628294/SENSOR
- QoS: 2
- Output: auto-detect (parsed JSON object, string or buf)
- Name: Name

The debug console on the right shows a message received at 21/03/2023 12:02:25 from node: debug 29. The message is a JSON object:

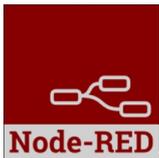
```
tele/tasmota_628294/SENSOR : msg.payload : Object
```

The object contains the following data:

- Time: "2023-03-21T12:02:24"
- Switch1: "OFF"
- Switch2: "OFF"
- Switch3: "OFF"
- Switch4: "OFF"
- Switch5: "ON"
- Switch6: "OFF"
- Switch7: "ON"
- Switch8: "ON"

Under the "ESP32: object" section, the following data is shown:

- Temperature: 55
- TempUnit: "C"



On récupère la température

Edit function node

Delete Cancel Done

Properties

Name GetInternalTemp

Setup On Start On Message On Stop

```
1 var temp = msg.payload.ESP32.Temperature;
2 msg.payload = temp;
3 return msg;
```

debug

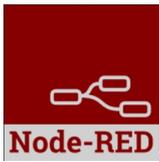
21/03/2023 12:02:25 node: debug 29
tele/tasmota_628294/SENSOR : msg.payload

object

- Time: "2023-03-21T12:02:24"
- Switch1: "OFF"
- Switch2: "OFF"
- Switch3: "OFF"
- Switch4: "OFF"
- Switch5: "ON"
- Switch6: "OFF"
- Switch7: "ON"
- Switch8: "ON"

ESP32: object

- Temperature: 55
- TempUnit: "C"



On affiche la température dans le DashBoard

Edit gauge node

Delete Cancel Done

Properties

Group [Menu] Menu

Size 3 x 3

Type Level

Label Température

Value format {{value}}

Units °C

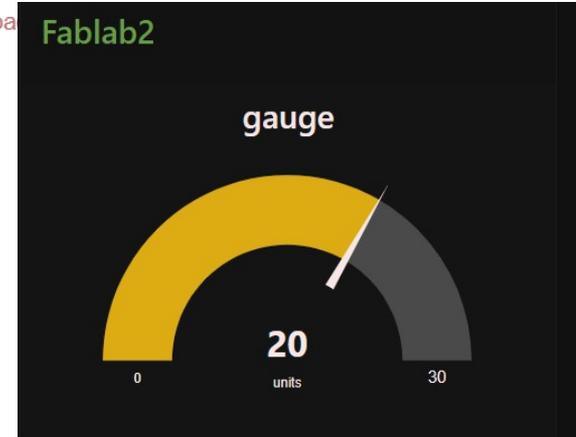
Range min 0 max 100

Class Optional CSS class name(s) for widget

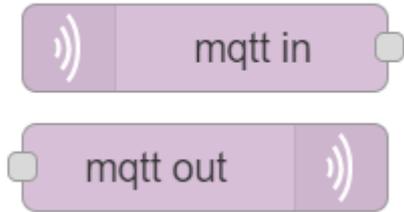
debug

all nodes

```
21/03/2023 18:06:24 node: debug 55
tele/tasmota_3408B0/SENSOR : msg.payload
  object
    Time: "2023-03-21T18:06:21"
    Switch4: "ON"
  DHT11: object
    Temperature: 20
    Humidity: 21
    DewPoint: -3
  ESP32: object
    Temperature: 30.6
    TempUnit: "C"
```



Node-Red



Souscription à notre objet:

Questions / Réponses
