



Utilisation simple d'un relais pour Wemos Avec un client ESP8266 et un serveur Node-RED

DeltaLab

Espace Maison Milon
2 Place E.Colongin
84600 Grillon

deltalabprototype.fr

Qu'est-ce que DeltaLab ?

DeltaLab est une association 'loi 1901' d'intérêt général, dont l'objectif est la création d'un espace dédié à l'innovation, à la production numérique au prototypage et à l'«expression artistique».

Le principe fondateur de l'association est **d'apprendre à faire soi-même, pour passer de l'idée à l'objet.**

Deltalab se spécialise en **Objets Connectés**, et est en train de créer un vaste «écosystème digital» entre Drôme et Vaucluse, pour répondre à des besoins non-couverts, mettre à disposition ressources et équipements pour usage professionnels et instaurer des partenariats avec les autres structures et initiatives numériques existantes.

Deltalab est aussi un **FabLab** (*Fabrication Laboratory / Laboratoire de Fabrication*), un tiers-lieu de type makerspace où se trouve un atelier qui dispose de machines de fabrication comme des Imprimantes 3D ou des découpeuses Laser.

Deltalab se veut ouvert à tous publics : étudiants, professionnels, associations, inventeurs, designers, artistes, ...

Contexte de cette Documentation

Ce projet est une introduction à l'utilisation de relais via WEMOS.
Il se base sur un shield contenant un seul relais, qui s'emboîte sous la WEMOS.

Table des matières

1. Introduction	04
2. Présentation du Circuit	05
3. Code du client arduino	06
4. Serveur NodeRED	08

I - Introduction

Ce projet a pour but de permettre l'utilisation d'un "relais". Il s'agit d'un switch qui change de position quand il reçoit un message. Dans une utilisation réelle, le relais pourrait être relié à une serrure ou une machine à protéger, à une LED, à un buzzer,...

Le projet se décompose en 2 parties distinctes : le relais et l'esp8266 associée , et le serveur Node-RED .

Les fonctionnalités proposées par ce projet sont les suivantes :

- ◆ Activation d'un relais grâce à du code sur la wemos
- ◆ Activation d'un relais depuis le serveur node-red

Logiciels :

- ◆ Arduino IDE : téléchargeable à : <https://www.arduino.cc/en/main/software>
- ◆ Librairie ESP8266WiFi : à télécharger à <https://github.com/esp8266/Arduino>
Puis à ajouter à Arduino (dézipper, puis dans Arduino : *Croquis* > *Inclure une Bibliothèque* > *Ajouter une Bibliothèque .zip* , récupérez la librairie dans le dossier libraries)
- ◆ ESP8266 : Fichier > Préférences > Gestionnaire de cartes. Entrez l'URL http://arduino.esp8266.com/stable/package_esp8266com_index.json
Puis Croquis>Carte>Gestionnaire de cartes , rechercher ESP8266

II - Présentation du Circuit

Le relais se trouve sur un shield qui est fait pour s'emboîter avec la wemos. Il vous suffit donc de les emboîter en faisant attention au sens (les mêmes pins s'emboîtent ensemble).

III - Codes Arduino

Librairies utilisées :

- PubSubClient - version 2.7.0
- ESP8266WiFi- version 1.0

```
#include <SPI.h>
#include <MFRC522.h>
#include "PubSubClient.h"
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

#define ClientID "rdc_lab" // id du lecteur
#define pin D2

WiFiClient cli;
PubSubClient mqttClient(cli);
const char* ssid = "DeltaLab-Public"; // SSID du point wifi
const char* password = "Milon!Lab"; // mot de passe du point wifi
const char* mqttServer = "192.168.1.45"; // serveur mqtt
const int mqttPort = 1883; // port du serveur à utiliser - défaut : 1883

/* Reconnexion au serveur mqtt si déconnecté */
boolean reconnect() {
  if (mqttClient.connect(ClientID)) {
    mqttClient.subscribe(ClientID);
    mqttClient.loop();
  }
  return mqttClient.connected();
}

/* Réponse du serveur : carte acceptée ou refusée (uniquement si relais local ) */
void callback(char* topic, byte* payload, unsigned int length) {
  String cmd = String((char*)payload);
  if (cmd.substring(0,2).equals("on")) {
    digitalWrite(pin , HIGH);
    mqttClient.publish(ClientID,"ok");
  }
  if (cmd.length() >= 3 && cmd.substring(0,3).equals("off")) {
    digitalWrite(pin , LOW);
    mqttClient.publish(ClientID,"nok");
  }
}
```

```

/*
 * Connexion au WiFi et initialisation du MQTT
 */
void setup() {

  Serial.begin(115200); // Initiate a serial communication
  WiFi.hostname(ClientID);
  WiFi.begin(ssid,password);
  delay(1500);

  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
  }

  mqttClient.setCallback(callback);
  mqttClient.setServer(mqttServer,mqttPort);
  digitalWrite(pin,HIGH);
  delay(1500);
  digitalWrite(pin,LOW);
}

/*
 * Attente d'un message
 */
void loop() {

  delay(1500);
  if (!mqttClient.connected()) {
    reconnect();
  }
  mqttClient.loop();
}

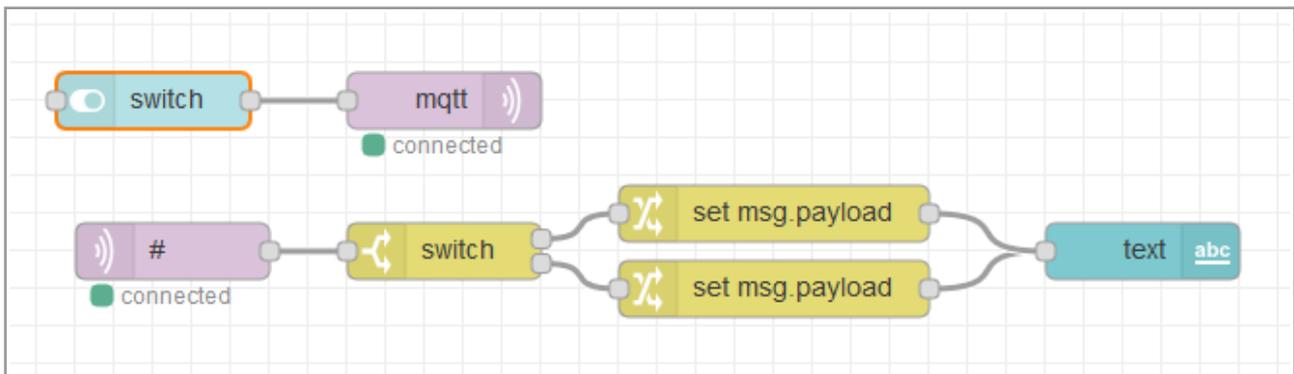
```

Dans arduino, pour choisir la bonne carte , allez dans **Outils > Type de carte**
 On utilise ici une Lolin(Wemos) D1 mini pro pour le lecteur, et une ESP8266 générique pour les relais.Vérifiez le Port et la vitesse d'écriture

Pour téléverser le code sur la carte, allez dans **croquis > Téléverser** , ou cliquez sur la flèche.

IV - Serveur Node-RED

◆ Flow :



◆ Explications

- La node switch permet d'envoyer un message mqtt au relais, pour l'activer ou non
- La node mqtt in permet de récupérer un message de réponse qui permet de savoir l'état du relais. Les nodes set définissent la couleur de l'icone à afficher dans le dashboard selon l'état du relais. La node text affiche l'id du relais ainsi que son état

◆ Pour le refaire

1. Nodes MQTT

- Mqtt in :
 - Serveur local : localhost / 1883
 - topic : # (tous les topics)
- Mqtt out :
 - Serveur local
 - Topic : l'id du relais

2. Node Switch

- condition 1 : msg.payload == on
- condition 2 : msg.payload == off (ou otherwise)

3. Nodes SET

- en haut : set msg.payload to green
- en bas : set msg.payload to red

4. Node Text

- Ui group : créez un nouveau group dans un nouvel onglet.
- label : `{{msg.topic}}`
- value : `<i class="fa fa-square fa-2x"></i>`

