



WiFi / MQTT :

**Exemple d'utilisation simple :
Capteur de mouvement Infra-Rouge**

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 présentation de l'utilisation de MQTT dans un exemple concret : la mise en place d'un capteur de mouvement par Infra-rouges.

Ce projet peut être considéré comme une base pour réaliser une installation plus ambitieuse (batterie de capteurs qui activent divers appareils au passage,...)

Ce projet présente aussi l'ajout d'un écran OLED pour afficher les données en direct.

Table des matières

1. Introduction	04
2. Présentation du Circuit	05
3. Code du client arduino	06
4. Installation sur TTN	08
5. Serveur NodeRED	10

I - Introduction

Ce projet a pour but de présenter l'utilisation de MQTT et de la Wemos D1 en utilisant un exemple basique : un capteur de mouvement par Infra-rouges. Un écran OLED sera emboîté par-dessus la Wemos pour afficher les données en direct.

Le projet se décompose en 2 parties distinctes : le capteur de mouvements , l'écran OLED et la carte Wemos associée et le serveur Node-RED , ici très basique pour la présentation.

Les fonctionnalités proposées par ce projet sont les suivantes :

- ◆ Affichage de présence / mouvement détecté en direct et sur Node-RED.

Logiciels :

- ◆ Arduino IDE : téléchargeable à : <https://www.arduino.cc/en/main/software>
- ◆ Node-RED : à deltalab , entrez l'@ip **192.168.1.45:1880** dans un navigateur
- ◆ Wemos : **Fichier > Préférences > URL du gestionnaire de cartes**. Entrez l'URL http://arduino.esp8266.com/stable/package_esp8266com_index.json
Puis **Outils > Type de Carte > Gestionnaire de cartes** , rechercher **ESP8266** et installer. Choisissez ensuite **Wemos_D1_mini_pro** comme Type de Carte.
- ◆ Librairies :
 - **Librairie ESP8266WiFi** : à télécharger à <https://github.com/esp8266/Arduino>
Puis à ajouter à Arduino (**Croquis > Inclure une Bibliothèque > Ajouter une Bibliothèque .zip**)
 - **Librairie GFX** : pour l'écran
dans Arduino : **insérer une bibliothèque > gérer les bibliothèques**
entrez **GFX** et installez la librairie **Adafruit_GFX_library**
 - **Librairie SSD1306** : pour l'écran
télécharger : https://github.com/mcauser/Adafruit_SSD1306/tree/esp8266-64x48
Puis à ajouter à Arduino (**Croquis > Inclure une Bibliothèque > Ajouter une Bibliothèque .zip**)

II - Présentation du Circuit

Le circuit est simple. La Wemos D1 et l'écran OLED sont fixés sur des *shields* qui s'emboîtent l'une sur l'autre.

Vous devez juste les emboîter dans le bon sens , les mêmes pins les unes sur les autres (3V avec 3V, RST avec RST, ...)

Des soudures sont à prévoir pour installer les bonnes *pins* pour l'emboîtement.

Le capteur infrarouge utilisé ici est un capteur PIR. Il a besoin de 3 pins : + , - et donnée.

Capteur PIR	Wemos
+	5V
-	GND
output	D7

La "vis" jaune permet de régler la durée pendant laquelle le capteur reste activé après une détection. Il suffit de la tourner dans le sens horaire pour augmenter cette durée, et dans l'autre sens pour la réduire. La durée minimale est d'environ 0,3 sec , et la durée maximale peut atteindre 18 minutes.



III - Codes Arduino

Librairies utilisées :

- ESP8266WiFi - version 1.0
- PubSubClient - version 2.7.0
- WiFiClient - version 1.0
- AdafruitGFX - version 1.8.4
- Adafruit_SSD1306 - version 1.1.0 modifiée

Code :

```
#include <Adafruit_GFX.h>
#include "Adafruit_SSD1306.h"
#include "Adafruit_SSD1306.cpp"
#include "PubSubClient.h"
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

#define OLED_RESET 0 // GPIO1

int pirPin = D7;
int val;
String id = "rdc_lab_007";
Adafruit_SSD1306 display(OLED_RESET);

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;

// Reconnexion au serveur mqtt si déconnecté
boolean reconnect() {
  if (mqttClient.connect("PIR_MQTT")) {
    mqttClient.loop();
  }
  return mqttClient.connected();
}
```

```

void setup(){
  Serial.begin(9600);
  WiFi.hostname("PIR_MQTT");
  WiFi.begin(ssid,password);
  pinMode(pirPin, INPUT);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
  }
  mqttClient.setServer(mqttServer,mqttPort);
}

void loop(){
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  val = digitalRead(pirPin);
  Serial.println(val);
  if(val == LOW){
    display.println("RAS");
    display.display();
  }else{
    display.println("OBJECT");
    display.println("DETECTED!");
    display.display();
  }

  //Données pour MQTT
  String contenu = "";
  contenu += id;
  contenu += "-";
  contenu += val;
  char payload[15];
  unsigned int len = contenu.length()+1;
  contenu.toCharArray(payload,len);

  //Envoi sur MQTT
  if(mqttClient.connect("PIR_MQTT")){
    Serial.println(payload);
    mqttClient.publish("PIR",payload);
  }
}

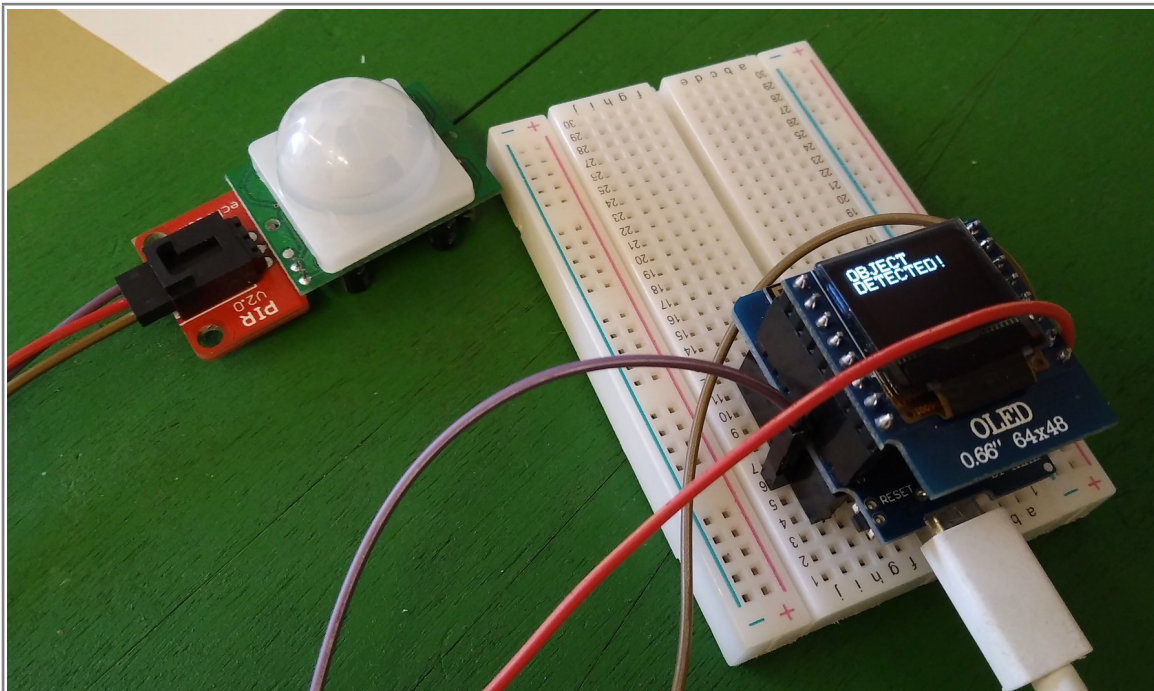
```

Dans Arduino, pour choisir la bonne carte , allez dans **Outils > Type de carte**
On utilise ici une **Wemos_D1_mini_pro**. Vérifiez le Port et la vitesse d'écriture.

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

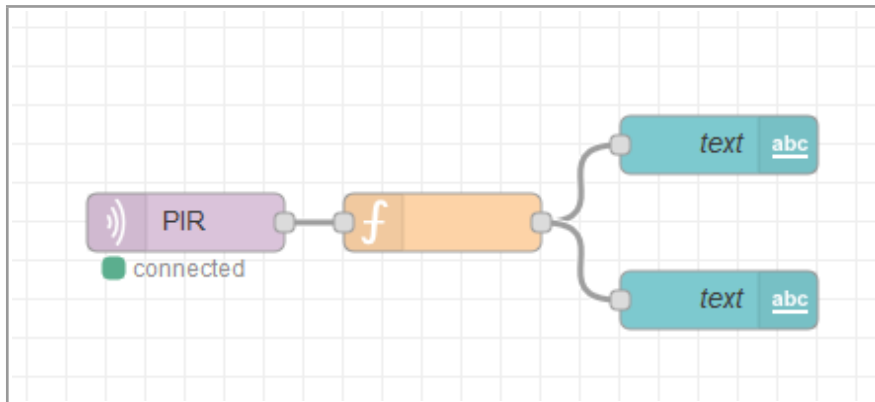
Le capteur Infra-rouge PIR n'a pas besoin de librairie particulière, il fonctionne en effet comme un Switch, il suffit donc de lire la valeur de la pin à laquelle il est relié pour s'en servir.(ici D7)

Aperçu du montage



IV - Serveur Node-RED

◆ Flow :



◆ Explications :

- ❁ Le capteur envoie ses données et MQTT les récupère. Node-RED les lit ensuite grâce à la node **MQTT in**.
- ❁ La node **function** permet de décoder le message et de placer les données dans des champs distincts.
- ❁ Les données sont ensuite récupérées par les nodes **text** qui permettent de les afficher. Celle du haut affiche l'id du capteur (pour savoir lequel c'est), celle du bas affiche une icône d'alerte et un message en cas de présence détectée. Sans présence, le text du bas n'affiche rien. Il vous suffit de changer la valeur des variables **label** et **msg** dans la node **function** pour changer cela.
- ❁ On peut y associer une node **'file'** pour enregistrer les données dans un fichier sur le serveur, ou une node de base de données pour y enregistrer les données choisies.

◆ Pour le refaire :

1. MQTT in

- Server :
 - ip: localhost
 - port : 1883
- Topic : PIR

2. Function Format

code de la fonction :

```
var id = [];  
id = msg.payload.split('-')  
var msg;  
var label;  
if(id[1] == 1){  
    label = "<font color='red'><i class='fa fa-exclamation-triangle fa-3x'></i></font>";  
    msg = "Présence Détectée !";  
}else{  
    msg = "";  
    label = "";  
}  
return {  
    payload:{ capt:id[0], msg:msg, label:label }  
}
```

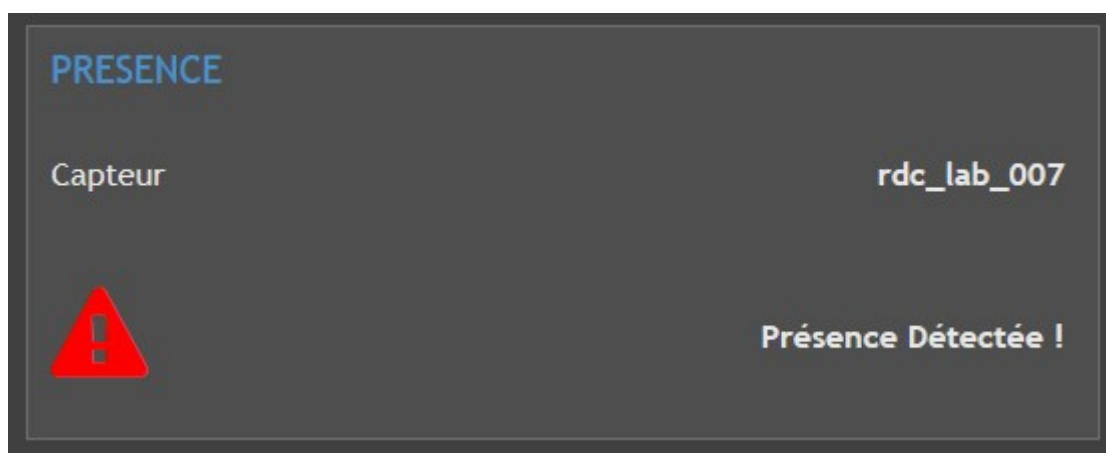
3. Text (id capteur)

- UI group : créez un nouveau group
 - ui_tab : créez une nouvelle
 - width (largeur) : comme vous voulez
- Label : "capteur"
- Value Format : {{msg.payload.capt}}

4. Text (message alerte)

- UI group : le même que pour le text précédent
- Label : {{msg.payload.label}}
- Value Format : {{msg.payload.msg}}

Dashboard :



Pour ce résultat , changez le thème à **sombre** (panneau de droite > onglet dashboard - icone de graphe > theme > dark(sombre)) et réglez la width du groupe et le positionnement des éléments (panneau de droite > dashboard > votre ui tab > layout) Ici , le groupe à une largeur de 12 , le texte du haut à une largeur de 12 et une hauteur de 1 et celui du bas a une largeur de 12 et une hauteur de 2.

