



FACULTY
OF INFORMATICS
Masaryk University

Comunicación MQTT ESP8266 NODEMCU

Integrantes

Pilatasig Jorge

Socasi Bryan

Viera Katherine

Yáñez Danilo

Objetivos

- Entender el funcionamiento de las comunicaciones “machine-to-machine” el cual emplea un protocolo ligero, tipo publicador/suscriptor, denominado MQTT ESP8266 NODEMCU junto con diferentes actuadores y sensores.
- Identificar las características de la tarjeta de desarrollo ESP8266 NODEMCU

Conceptos

Internet de las cosas (IoT)

Es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet. Es, en definitiva, la conexión de internet con mas objetos que con personas.



Conceptos

Comunicación MQTT

MQTT (Message Queue Telemetry Transport), es un protocolo usado para la comunicación machine-to-machine (M2M) en el "Internet of Things". Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empujados con pocos recursos.

Conceptos

Arquitectura MQTT

Dentro de la arquitectura de MQTT, es muy importante el concepto “topic” o “tema” en español ya que a través de estos “topics” se articula la comunicación puesto que emisores y receptores deben estar suscritos a un “topic” común para poder entablar la comunicación. Este concepto es prácticamente el mismo que se emplea en colas, donde existen publicadores (que publican o emiten información) y unos suscriptores (que reciben dicha información) siempre que ambas partes estén suscritas a la misma cola.

Conceptos

ESP8266

El ESP8266 es un chip Wi-Fi de bajo coste que funciona mediante el protocolo TCP/IP. Su punto fuerte es el disponer de una conexión Wi-Fi en un microcontrolador, además se puede programar directamente con el entorno de Arduino con lo que es el chip perfecto para desarrollar nuestras aplicaciones de IoT.



Conceptos

NodeMCU

Es un proyecto Open-Source para el desarrollo de un modelo sencillo de integrar la IoT (Internet of Things), para ello se desarrollan modelos hardware y software que facilite el desarrollo de programas y aplicaciones basados en WiFi.

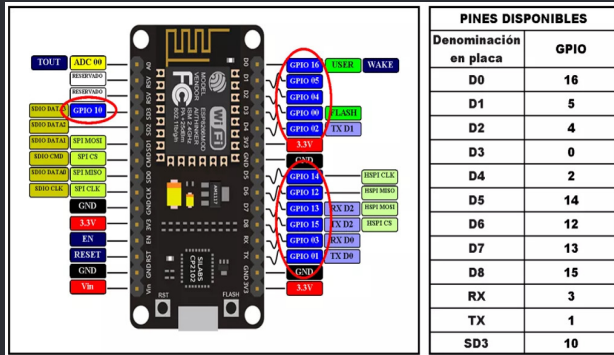


ESP8266

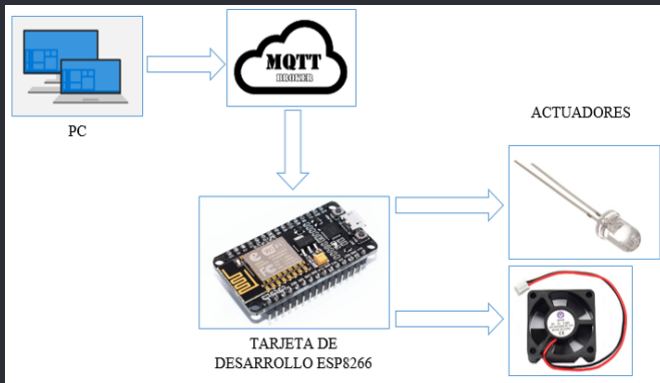
Ventajas

- Programación sencilla a través del Micro-USB
- Alimentación a través del USB
- Terminales (pines) para facilitar la conexión
- LED y botón de reset integrados
- Su programación se hace totalmente transparente, al no requerir ningún cambio en sus pines para la programación, y disponer de conexión USB al igual que Arduino.
- La propia placa dispone de un regulador de alimentación, así como un chip USB-Serial para la comunicación del ESP8266 directamente desde el USB del ordenador.
- Facilidad de desarrollar aplicaciones de IoT debido a la gran comunidad y el gran número de software compatible y librerías existentes para la programación del ESP8266.

Pines del ESP8266 NodeMCU



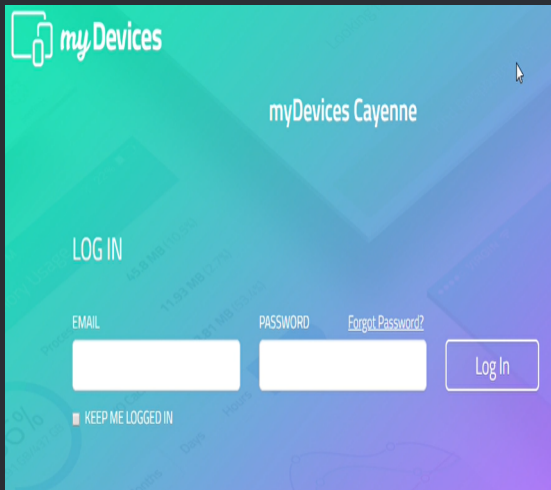
Funcionamiento



Funcionamiento

Inicio en Cayenne

<https://mydevices.com/cayenne/features/remote-monitoring/>



The image shows the login interface for myDevices Cayenne. The background is a teal-to-blue gradient with faint, semi-transparent images of IoT devices and data charts. In the top left corner is the myDevices logo, consisting of a stylized icon of two connected devices and the text "myDevices". To the right of the logo, the text "myDevices Cayenne" is displayed. Below this, the words "LOG IN" are centered. There are two input fields: one for "EMAIL" and one for "PASSWORD". To the right of the password field is a link that says "Forgot Password?". Below the email field is a checkbox labeled "KEEP ME LOGGED IN". To the right of the password field is a "Log In" button with a white border and rounded corners.

myDevices

myDevices Cayenne

LOG IN

EMAIL

PASSWORD

[Forgot Password?](#)


☐ KEEP ME LOGGED IN

Log In

Funcionamiento

Inicio en Cayenne


Devices & Widgets



DEVICES


- Single Board Computers >
- MicroControllers >
- Sensors ▼
- Actuators ▼
- Extensions ▼
- LoRa ▼

Single Board Computers




Raspberry Pi


Microcontrollers



Arduino



SparkFun ESP8266
Thing Dev Board



Generic ESP8266

Funcionamiento

Datos para la conexión con el servidor Cayenne

Connect your Device

SELECT YOUR BOARD

SparkFun ESP8266 Thing Dev Board

STEPS TO CONNECT

- A. Install the [Arduino IDE](#) and add [Cayenne MQTT Library](#) to Arduino IDE.
- B. Install the [ESP8266 board package](#) to Arduino IDE. [show me how](#)
- C. Install required FTDI driver on your computer

MQTT USERNAME:

845cf9e0-66cf-11e9-888e-af3db3cb8c63

MQTT PASSWORD:

66c1209cee2a1c3c14032315965d60b1cfd3af2

CLIENT ID:

81f39390-6796-11e9-933e-cf08617625ed

MQTT SERVER:

mqtt.mydevices.com

MQTT PORT:

1883

NAME YOUR DEVICE (optional):

SparkFun ESP8266 Thing Dev Board

Explicación del código

Librerías necesarias para la conexión con Cayenne

```
#include <CayenneMQTTESP8266.h>
```

Parámetros necesarios para la conexión a la red wifi

```
char red[] = "HUAWEI Y5 2017";  
char passwordWifi[] = "1234567890";
```

Parámetros necesarios para la conexión al servidor de Cayenne

```
char usuario[] = "123ac340-6618-11e9-933e-cf08617625ed";  
char password[] = "5b8d6bf500c30debfdcle7f66476dcefa093271b";  
char clienteID[] = "a6bd1660-661f-11e9-9dd7-93e53f1934d3";
```

Explicación del código

Declaración de las variables de los “actuadores” definidos en los pines de comunicación del ESP8266 NodeMCU

```
const int led = 14;  
const int ventilador = 12;
```

Inicio del método setup() que contendrá la configuración de los pines en el modo de OUTPUT (Salida), y los parámetros declarados anteriormente para realizar la conexión con la red wifi y el servidor de Cayenne

```
void setup() {  
  pinMode(led, OUTPUT);  
  pinMode(ventilador, OUTPUT);  
  Cayenne.begin(usuario, password, clienteID, red, passwordWifi);  
}
```

Explicación del código

Declaración de puertos virtuales a los cuales se conectara el ESP8266 NodeMCU para controlar los actuadores

```
CAYENNE_IN(1)
{
    int canalLed = getValue.asInt();
    digitalWrite(led, canalLed );
}

CAYENNE_IN(2)
{
    int canalVentilador = getValue.asInt();
    digitalWrite(ventilador, canalVentilador);
}
```


Creación de los Widgets en Cayenne

The screenshot displays the Cayenne web interface for creating a widget. The interface is divided into several sections:

- Left Sidebar:** Contains an "Add new..." button and a list of devices: "ESP8266", "ESP8266 MQTT", and "SparkFun ESP8266 Th...". Below this is a search bar labeled "Search Devices".
- Devices & Widgets Panel:** Features a "Search" bar and a list of device categories: "Single Board Computers", "MicroControllers", "Sensors", "Actuators", "Extensions", and "LoRa". At the bottom, there is a "Custom Widgets" section and a "CAYENNE API" link.
- Widget Selection Grid:** Displays various widget examples with icons and labels:
 - Value:** Display Widget Example (thermometer icon, showing 74°)
 - Line Chart:** Display Widget Example (line graph icon)
 - Gauge:** Display Widget Example (gauge icon)
 - 2 State:** Display Widget Example (green square with '1' icon)
 - Button:** Controller Widget Example (button icon)
- Enter Settings Panel:** Contains configuration options for the selected widget:
 - Slider Controller Widget:** A visual representation of a slider.
 - Name:** A text input field containing "Ventilador".
 - Device:** A dropdown menu.
 - Sensor:** A section with a "Data" dropdown menu and a "Unit" dropdown menu.
 - Slider Min Value (optional):** A text input field.

Creación de los Widgets en Cayenne

The screenshot displays the Cayenne dashboard's widget configuration interface. On the left, a sidebar menu lists categories: Microcontrollers, Sensors, Actuators, Extensions, LoRa, and Custom Widgets. Below these is a section for the CAYENNE API with a 'Bring Your Own Thing' button. The main area shows a list of widget options: Gauge, 2 State, Button (highlighted in light blue), and Slider. Each widget entry includes a representative icon and a right-pointing arrow. To the right of the widget list is a configuration panel for the selected 'Button' widget. It contains fields for 'Name' (set to 'Led'), 'Device' (a dropdown menu showing 'ESP8266 MQTT'), 'Sensor' type (a dropdown menu showing 'Digital Actuator'), 'Unit' (a dropdown menu showing 'Digital (0/1)'), 'Channel' (a dropdown menu showing '3'), and 'Choose Icon' (a dropdown menu showing a lightbulb icon and the text 'Light'). At the bottom right of the configuration panel is a green button labeled 'Step 1: Code'.

Microcontrollers

Sensors

Actuators

Extensions

LoRa

ADD DATA TO YOUR DASHBOARD

Custom Widgets

CAYENNE API

Bring Your Own Thing

Example

Gauge
Display Widget
Example

2 State
Display Widget
Example

Button
Controller Widget
Example

Slider
Controller Widget
Example

Name
Led

Device
ESP8266 MQTT

Sensor

Data
Digital Actuator

Unit
Digital (0/1)

Channel
3

Choose Icon
Light

Step 1: Code

