

# NodeMCU: 06\_Conectamos nuestro NodeMCU como station a nuestro router de casa (AP)

 [agrportfolioeducativo.blogspot.com/2020/03/nodemcu-06conectamos-nuestro-nodemcu.html](http://agrportfolioeducativo.blogspot.com/2020/03/nodemcu-06conectamos-nuestro-nodemcu.html)

En este caso, nuestro NodeMCU está conectado a un LED como en el anterior. Las conexiones son las mismas.

Por otra parte, ahora necesito la SSID y la contraseña de nuestro router inalámbrico. Ese será nuestro punto de acceso. Una vez conectados, nuestro NodeMCU como cliente estará a la espera de una petición.

Esa petición proviene del navegador de otro dispositivo, como nuestro móvil o nuestro ordenador. En él, escribiremos la IP del NodeMCU al que nos hemos conectado. La IP aparece en el monitor Serie al resetear nuestro NodeMCU.

Se han añadido al programa original una línea para poder escribir acentos (codificación UTF-8) y las etiquetas body.

Además, si abrimos el puerto 80 en nuestro router y hacemos una redirección de las peticiones por este puerto a la IP de nuestro NodeMCU, podremos controlarlo desde el exterior. Se explica para un router determinado en el segundo enlace de la bibliografía.

/\*

---

NodeMCU Server - Conexión en modo Station con un punto de acceso (router) por Dani No [www.esploradores.es](http://www.esploradores.es)

---

Crea una conexión del NodeMCU en modo Station con un punto de acceso que permite

---

encender y apagar un LED conectado a la salida D4 (GPIO2) del módulo NodeMCU.

---

Este código de ejemplo es de público dominio.

---

Modificado por Aurelio Gallardo

---

\*/

---

---

```
#include <ESP8266WiFi.h> //Incluye la librería ESP8266WiFi
```

---

```
const char* ssid = "JAZZTEL_XXXX"; //Indicamos el nombre de la red WiFi (SSID) a la que queremos conectarnos.
```

---

---

```
const char* password = "xxxxxxxxxxxxx"; //Indicamos la contraseña de de red WiFi
```

---

```
WiFiServer server(80); //Definimos el puerto de comunicaciones
```

---

```
int PinLED = D1; // GPIO5 //Definimos el pin de salida - GPIO5 / D4
```

---

```
int estado = LOW; //Definimos la variable que va a recoger el estado del LED
```

---

```
void setup() {
```

---

```
Serial.begin(115200);
```

---

```
pinMode(PinLED, OUTPUT); //Inicializamos el GPIO5 como salida
```

---

```
digitalWrite(PinLED, LOW); //Dejamos inicialmente el GPIO5 apagado
```

---

```
WiFi.begin(ssid, password); //Inicializamos la conexión del NodeMCU con la red WiFi
```

---

```
Serial.printf("\n\nConectando a la red: %s\n", WiFi.SSID().c_str());
```

---

```
while (WiFi.status() != WL_CONNECTED) { // Verifica el estado de la conexión del  
NodeMCU cada 0,5s hasta que conecta
```

---

```
delay(500);
```

---

```
Serial.print(".");
```

---

```
}
```

---

```
Serial.println("");
```

---

```
Serial.println("WiFi conectada"); // Indica que el NodeMCU conectado con la red WiFi
```

---

```
server.begin(); // Inicia el NodeMCU en modo Station
```

---

```
Serial.println("Servidor inicializado");
```

---

```
Serial.printf("IP local del NodeMCU: %s\n", WiFi.localIP().toString().c_str());
```

---

```
Serial.printf("DNS #1, #2 IP: %s, %s\n", WiFi.dnsIP().toString().c_str(),  
WiFi.dnsIP(1).toString().c_str());
```

---

```
Serial.printf("Submascara de red -subnetMask-: %s\n",  
WiFi.subnetMask().toString().c_str());
```

---

```
Serial.printf("Direccion MAC en modo Server -macAddress-: %s\n",  
WiFi.macAddress().c_str());
```

---

---

```
Serial.printf("IP privada del punto de acceso -dnsIP-: %s\n",  
WiFi.gatewayIP().toString().c_str());
```

---

```
Serial.printf("Nombre del NodeMCU como cliente -hostname-: %s\n",  
WiFi.hostname().c_str());
```

---

```
Serial.printf("Connection status: %d\n0=WL_IDLE_STATUS,  
1=WL_NO_SSID_AVAIL, 3=WL_CONNECTED, 4=WL_CONNECTED-FAILED y  
6=WL_DISCONNECTED\n", WiFi.status());
```

---

```
Serial.printf("Nombre de red -SSID-: %s\n", WiFi.SSID().c_str());
```

---

```
Serial.printf("Clave de red -Password-: %s\n", WiFi.psk().c_str());
```

---

```
Serial.printf("Direccion MAC del punto de acceso -BSSID-: %s\n",  
WiFi.BSSIDstr().c_str());
```

---

```
Serial.printf("Intensidad de la conexion con el punto de acceso -RSSI-: %d dB\n",  
WiFi.RSSI());
```

---

```
Serial.printf("\n\nUtiliza esta URL para conectar: http://%s/\n",  
WiFi.localIP().toString().c_str());
```

---

```
}
```

---

```
void loop()
```

---

```
{
```

---

```
// Comprueba si el cliente ha conectado
```

---

```
WiFiClient client = server.available();
```

---

```
if (!client) {
```

---

```
return;
```

---

```
}
```

---

```
// Espera hasta que el cliente envía alguna petición
```

---

```
Serial.println("nuevo cliente");
```

---

```
while(!client.available()){
```

---

```
delay(1);
```

---

```
}
```

---

```
// Lee la petición
```

---

```
String peticion = client.readStringUntil('\r');
```

---

---

```
Serial.println(peticion);
```

---

```
client.flush(); // Descarga los bytes enviados al cliente pero todavía no leídos
```

---

```
// Comprueba la petición
```

---

```
if (peticion.indexOf('/LED=ON') != -1) {
```

---

```
    estado = HIGH;
```

---

```
}
```

---

```
if (peticion.indexOf('/LED=OFF') != -1){
```

---

```
    estado = LOW;
```

---

```
}
```

---

```
//Enciende o apaga el LED en función de la petición
```

---

```
digitalWrite(PinLED, estado);
```

---

```
// Envía la página HTML de respuesta al cliente
```

---

```
client.println("HTTP/1.1 200 OK");
```

---

```
client.println(""); //No olvidar esta línea de separación
```

---

```
client.println("");
```

---

```
client.println("");
```

---

```
client.println("<head><meta charset='UTF-8'></head>"); //Añado la línea para  
codificación de caracteres.
```

---

```
// Así podemos leer los acentos
```

---

```
client.print("<body>"); // Incluyo etiqueta body
```

---

```
client.print("<h1 align=center>El LED está ahora: ");
```

---

```
if(estado == HIGH) {
```

---

```
    client.print("ENCENDIDO</h1>");
```

---

```
    client.print("<img src='https://esploradores.com/Practicas_html/LED_encendido.png'  
alt='Bombilla encendida' style='display:block; margin:auto' width='150px'>");
```

---

```

client.print("<input type='image'
src='https://esploradores.com/Practicas_html/apagar_300x88.png'
style='display:block; margin:auto' width='250px'
onClick=location.href='/LED=OFF'>");

} else {

client.print("APAGADO</h1>");

client.print("<img src='https://esploradores.com/Practicas_html/LED_apagado.png'
alt='Bombilla apagada' style='display:block; margin:auto' width='150px'>");

client.print("<input type='image'
src='https://esploradores.com/Practicas_html/encender_300x88.png'
style='display:block; margin:auto' width='250px' onClick=location.href='/LED=ON'>");

}

client.println("</body>"); // Incluyo etiqueta fin de body

client.println("</html>");

delay(1);

Serial.println("Petición finalizada"); // Se finaliza la petición al cliente. Se inicializa la
espera de una nueva petición.

Serial.println("");

}

```

[view raw NODEMCU como station comunicado con un router AP.ino](#) hosted with ♥  
by [GitHub](#)

## Bibliografía

=====

<https://www.esploradores.com/practica-3-station-conexion-con-access-point/>  
<https://www.esploradores.com/practica-4-station-conexion-desde-cualquier-sitio-del-mundo-con-access-point-encendido-y-apagado-de-un-led/>