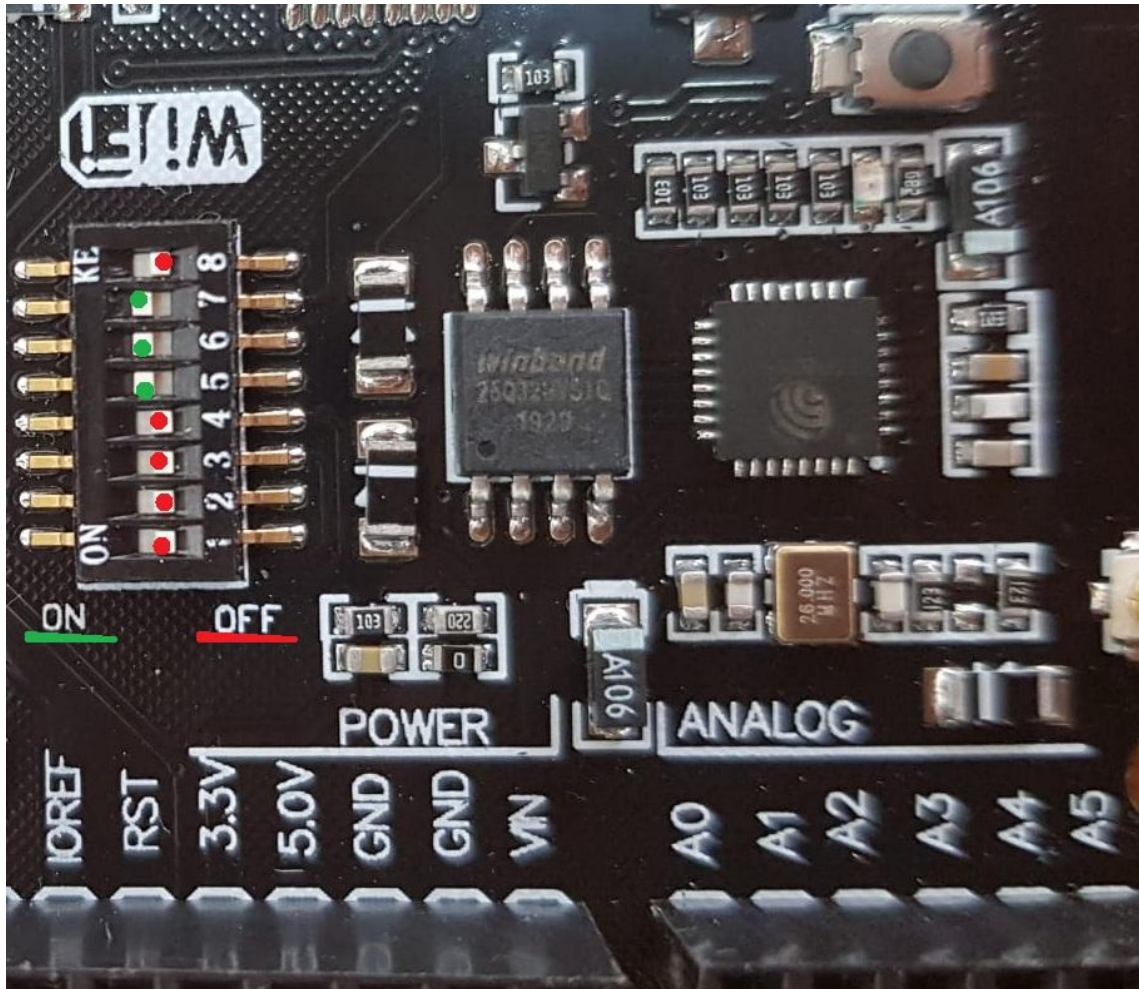


WIFI ARDUINO ESP8266

Hay que tener en cuenta los pines de la placa dejarlos en ON los correspondientes

Para la subida del SKETCH / Programa



El programa que usaré de ejemplo para comprobar la conexión es el siguiente:

(No olvidar cambiar el Nombre y la Contraseña a la de tu Red)

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "TuNombreWifi";
```

```
const char* password = "TuContraseñaWIFI";
```

```
int ledPin = 4; // GPIO4
```

```
WiFiServer server(80);
```

```
void setup() {
```

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

```
  delay(10);
```

```
  pinMode(ledPin, OUTPUT);
```

```
  digitalWrite(ledPin, LOW);
```

```
  // Connect to WiFi network
```

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

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

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

```
  Serial.println(ssid);
```

```
  WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.println("WiFi connected");  
  
// Start the server  
server.begin();  
Serial.println("Server started");  
  
// Print the IP address  
Serial.print("Use this URL to connect: ");  
Serial.print("http://");  
Serial.print(WiFi.localIP());  
Serial.println("/");  
}  
  
void loop() {  
    // Check if a client has connected  
    WiFiClient client = server.available();  
    if (!client) {  
        return;  
    }  
  
    // Wait until the client sends some data
```

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

while(!client.available()){

    delay(1);
}

// Read the first line of the request

String request = client.readStringUntil('\r');

Serial.println(request);

client.flush();

// Match the request

int value = LOW;

if (request.indexOf("/LED=ON") != -1) {

    digitalWrite(ledPin, HIGH);

    value = HIGH;

}

if (request.indexOf("/LED=OFF") != -1) {

    digitalWrite(ledPin, LOW);

    value = LOW;

}

// Set ledPin according to the request

//digitalWrite(ledPin, value);

// Return the response

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

```
client.println("Content-Type: text/html");

client.println(""); // do not forget this one

client.println("<!DOCTYPE HTML>");

client.println("<html>");


client.print("Led pin is now: ");


if(value == HIGH) {

    client.print("On");

} else {

    client.print("Off");

}

client.println("<br><br>");

client.println("<a href=\\\"/LED=ON\\\"><button>Turn On </button></a>");

client.println("<a href=\\\"/LED=OFF\\\"><button>Turn Off </button></a><br />");

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


delay(1);

Serial.println("Client disconnected");

Serial.println("");

}
```

Este Programa usa la placa de servidor mediante la red WIFI y podremos enviar peticiones desde el navegador desde la direccion que nos envíe por el Monitor Serie.

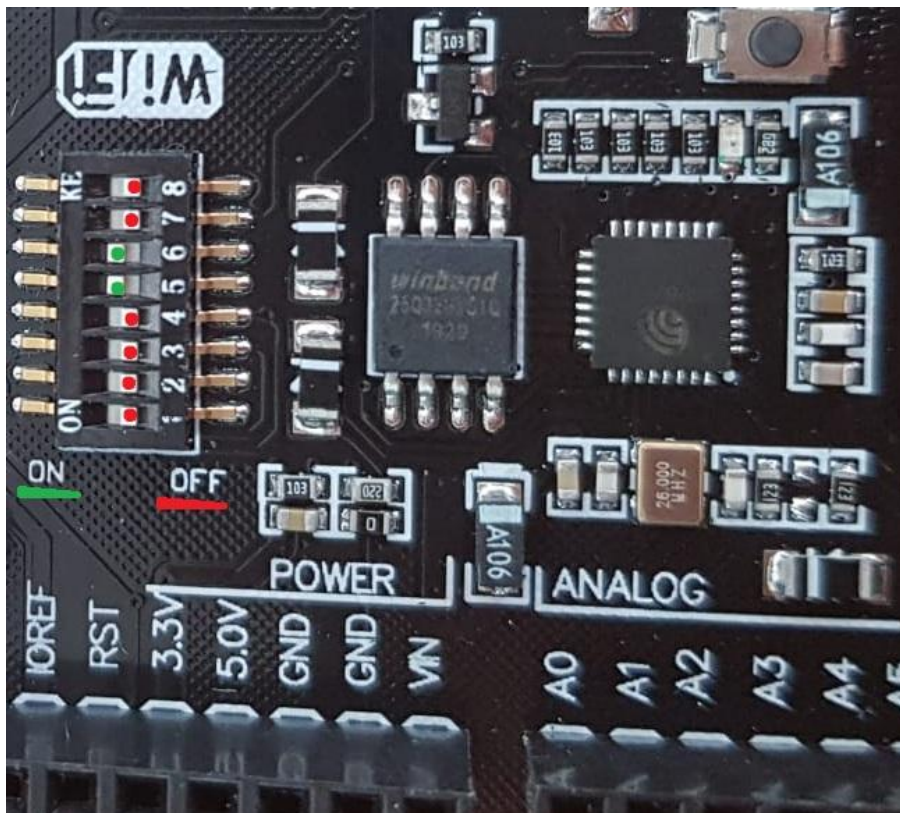
En este caso podremos enviar dos peticiones las de encender y apagar un LED que conectamos en la placa en el puerto 4.

Luego de Cargar el Programa abrimos el Monitor Serie que se encuentra en
HERRAMIENTAS > MONITOR SERIE

Nos aseguramos de tener esta configuración:



Cambiamos el pin numero 7 y conectados por USB presionamos el botón MCU RESET para resetear la placa.



En el monitor serie aparecerá lo siguiente.

```
{1 100} 0d0|  10 #|00 0 0e0b0 c00'e0do'000 c p0dsdr1p0g0  1 00  c n0| 800c00og0 1001' 0  gn 1'  or00' 000' {00o 000' 0;d00d 0| 0' 00g0

Connecting to SCR3
.....
WiFi connected
Server started
Use this URL to connect: http://192.168.43.23/
new client
GET /LED=ON HTTP/1.1      Ip de acceso a la página
Client disconnected        Que nos afirma la conexión wifi

new client
GET /favicon.ico HTTP/1.1
Client disconnected

new client
GET /LED=OFF HTTP/1.1
Client disconnected

new client
GET /favicon.ico HTTP/1.1
Client disconnected

new client
```

Con la dirección de Red proporcionada abrimos el navegador de cualquier dispositivo conectado a la WIFI

