

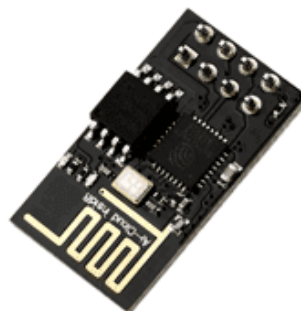
LUIS LLAMAS

Ingeniería, informática y diseño

ESP8266/ESP32 TUTORIALES ARDUINO TUTORIALES ARDUINO INTERMEDIOS

CONECTAR ARDUINO POR WIFI CON EL MÓDULO ESP8266 ESP01

27 MAYO, 2017



¿QUÉ ES EL ESP8266?

El ESP8266 es un **microprocesador de bajo coste con Wifi integrado** fabricado por Espressif. Podemos usar el ESP8266 para conectar nuestros

proyectos de electrónica y robótica con Arduino.

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

En realidad, **el ESP8266 es mucho más que un módulo Wifi** para Arduino. Es un procesador completo, con mucha más potencia que la mayoría de modelos de Arduino. De hecho es uno de los principales "competidores" a los que se enfrenta Arduino.

Existen muchos modelos de placas que integran el ESP8266, y una creciente comunidad de usuarios en torno a este procesador. Trataremos extensamente sobre más detalles y modelos con [ESP8266](#) en la sección de [tutoriales avanzados](#).

Anuncio:

Asus en Alkosto

Anuncio **Verdaderamente poder**
computador Asus. Compra el tu

Alkosto

Comprar

De momento en esta entrada nos limitaremos a emplear **Arduino junto con un módulo ESP01**, un de los primeros en aparecer con el chip ESP8266 y uno de los módulos más sencillos y baratos.

El ESP01 monta la versión más sencilla del ESP8266. Aún así, sigue siendo un procesador de 32bits a 80 Mhz, con 512kB o 1MB de memoria según modelo. Dispone de 2 pines GPIO, SPI, I2C y UART.

En cuanto a comunicación Wifi, el ESP01 tiene comunicación integrada 802.11 b/g/n, incluidos modos Wi-Fi Direct (P2P) y soft-Ap. Incluye una pila de TCP/IP completa, lo que libera de la mayor parte del trabajo de comunicación al procesador.

Podemos usar el ESP01 para dotar de conectividad WiFi a nuestros proyectos con Arduino. Sin embargo, como comentamos con los [módulos de Ethernet](#), la comunicación con Internet puede suponer una carga excesiva para Arduino.



En caso de requerir un uso intensivo de Internet, deberemos

plantearnos otras alternativas como Raspberry Pi, Orange Pi, o

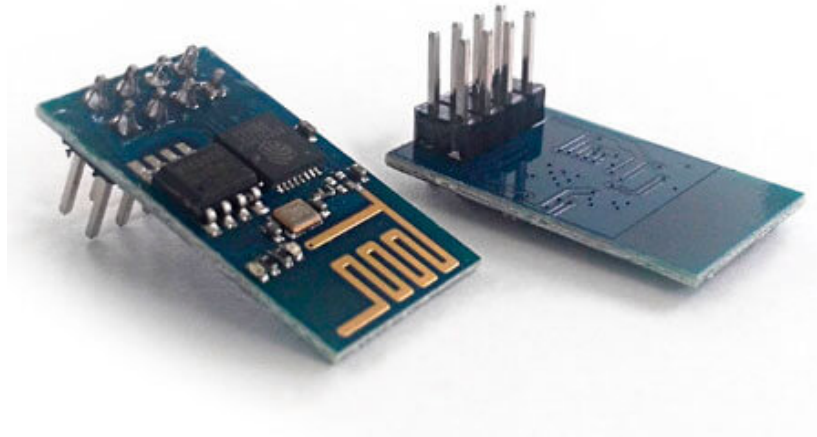
ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Los módulos ESP8266 de tipo ESP01 son muy baratos, ya que como hemos dicho es uno de los motivos de su amplio éxito. Podemos encontrarlos por 1.65€ en vendedores internacionales de eBay o AliExpress.



ESQUEMA DE MONTAJE

La conexión de un módulo ESP01 con Arduino no es complicada, aunque en la práctica van a aparecer ciertas dificultades que finalmente nos harán plantearnos la idoneidad de emplearlo junto a Arduino, o bien optar por una solución basada en ESP8266.

La principal dificultad es la alimentación del ESP01. El ESP8266 y, en particular, el ESP01, tiene una tensión de alimentación de 3.3V. En ningún caso puede alimentarse a una tensión superior a 3.6V, o dañaremos el módulo.

Por otro lado, el consumo del módulo puede sobrepasar los 200mA, sobre todo durante la conexión y arranques. Sin embargo, el regulador de voltaje de 3.3V de Arduino sólo puede proporcionar 50mA (150mA en algunos modelos), lo cual es insuficiente para alimentar el ESP01.

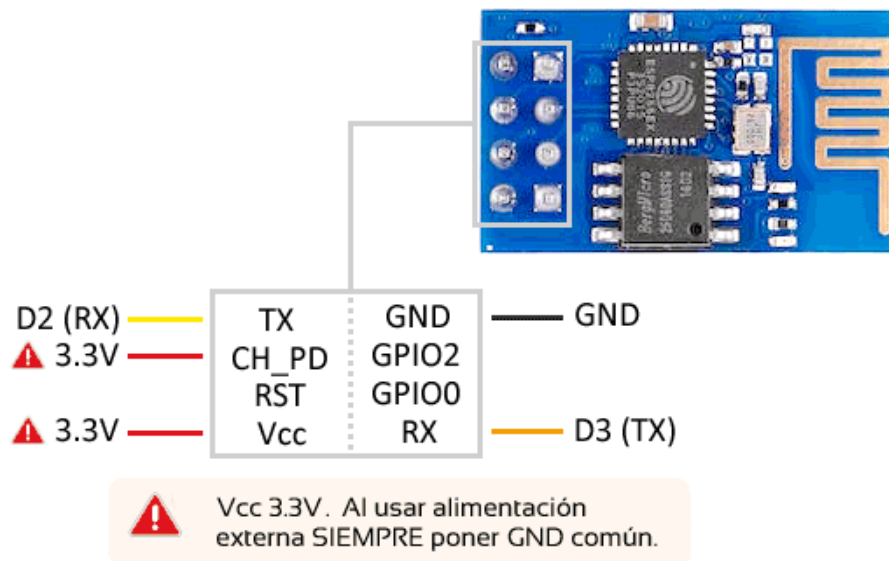
Por tanto, **necesitaremos alimentar el ESP01 con una fuente externa de 3.3V.** De lo contrario experimentaremos continuos cortes y cuelgues durante su funcionamiento, que además pueden reducir la vida

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject



El resto de la conexión no tiene ninguna dificultad. Por un lado tenemos **el pin CH_PD que apaga o enciende el módulo** conectándolo, respectivamente, a Gnd o 3.3V. Por su parte, **el pin RST reinicia el módulo** si se conecta a Gnd. En algunas versiones del módulo podremos dejarlo sin conexión pero, en general, tendremos que conectarlo a 3.3V para que el módulo arranque.

Finalmente, **la comunicación con el módulo se realiza mediante puerto serie**. A diferencia de la alimentación del módulo, que en ningún caso debe exceder los 3.6V, existe un amplio debate sobre si los pines RX y TX son tolerantes a 5V (Es decir, si pueden conectarse directamente a un sistema a un 5V).

En principio, conectar RX y TX a un sistema de 5V no parece dañar la placa de forma inmediata. De hecho, encontramos indicios en la documentación que apuntan a que podrían ser tolerantes a 5V, sin quedar totalmente claro. No obstante, tampoco queda claro si puede reducir la vida útil del componente.



En esta entrada conectaremos directamente los pines de RX y TX. Sin embargo, **si queréis total seguridad deberéis conectar un adaptador de nivel de 3.3V a 5V en los pines RX y TX.**

Para la conexión con Arduino podemos emplear el puerto serie

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

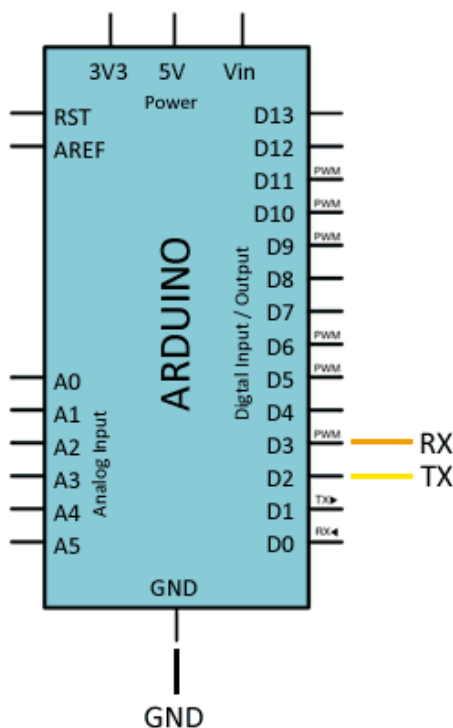
Reject

Por este motivo **frecuentemente se emplea un puerto serie por software**. No obstante, tener en cuenta que esto supone una carga de procesamiento importante para Arduino.



En los esquemas de esta entrada asumimos que estamos usando un puerto serie software con TX en el pin digital 2 y RX en el pin digital 3, pero podríamos definir el puerto en cualquiera otro pin I/O.

Por tanto, la conexión vista desde Arduino sería la siguiente.



En cuanto a las velocidades, el ESP01 puede configurarse a 9600, 19200, 38400, 74880, 115200, 230400, 460800 y 921600. Por defecto suelen venir, según fabricante, a 9600 o 115200.



Si vamos a usar Arduino como intermediario **evitaremos las velocidades de 115200 en adelante** porque la comunicación puede volverse inestable y aparecer errores.

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

La comunicación con el ESP01 con el firmware por defecto **se realiza a través de comandos AT**, que recordemos no son más que comandos de texto enviados por Serial.

Podemos enviar estos comandos por un conversor USB-TTL (FT232, CH340G o CP2102) o, en nuestro caso, usando Arduino y Software serial como adaptador.

PRIMERA PRUEBA

Vamos a hacer la primera prueba de conexión con el ESP01. Para ello conectamos el ESP01 a Arduino como hemos visto en el apartado anterior. Dejamos conectado Arduino al ordenador por USB.

A continuación cargamos el siguiente Sketch en Arduino, que seguramente alguno reconocerá como el programa Serial Loop. Este sketch únicamente actúa "puente" entre el puerto serie hardware conectado con el PC, y el puerto serie Soft conectado al ESP01.

Copiar

```
1. // La velocidad depende del modelo de ESP-01
2. // siendo habituales 9600 y 115200
3. const int baudRate = 9600;
4.
5. #include "SoftwareSerial.h"
6. SoftwareSerial softSerial(2, 3); // RX, TX
7.
8. void setup()
9. {
10.     Serial.begin(baudRate);
11.     softSerial.begin(baudRate);
12. }
13.
14. void loop()
15. // enviar los datos de la consola serial al ESP-01,
16. // y mostrar lo enviado por el ESP-01 a nuestra cor
17. {
18.     if (softSerial.available())
19.     {
20.         Serial.print((char)softSerial.read());
21.     }
```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Una vez cargado el Sketch, encendemos (o reiniciamos) el ESP01. En el Monitor Serie el ESP01 responde con una serie de caracteres que dependen del fabricante y modelo, y finalmente "Ready", indicando que el módulo está listo.

Si ahora escribimos AT, el módulo responderá con "OK", indicando de nuevo que el módulo está listo para su uso.

Si no veis el mensaje inicial finalizado en "Ready" y en su lugar aparecen "caracteres raros", cambiar la velocidad del puerto Serie en el Sketch Serial Loop, y en el Monitor Serie.

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

[Copiar](#)

```
1.  //**** GENERAL ****
2.  // Acknowledge, se recibe "ok"
3.  AT
4.
5.  // Reset, reinicia el módulo.
6.  // Se recibe una cadena de texto que depende del m
7.  AT+RST
8.
9.
10.
11. //**** CONFIGURACION ****
12. // Obtener la velocidad de transmision
13. AT+CIOBAUD?
14.
15. // Cambiar la velocidad de transmision (en el ejemp
16. // Velocidades validas 9600, 19200, 38400, 74880, 1
17. AT+CIOBAUD=9600
18. AT+IPR=9600
19.
20. // Obtener el modo de funcionamiento
21. // 1 Station
22. // 2 SoftAp
23. // 3 Station + SoftAp
24. AT+CWMODE?
25.
26. // Cambia el modo de funcionamiento a 1, 2 o 3
27. // Lo normal es AT+CWMODE=3, por que es el más vers
28. // Tras el cambio es necesario AT+RST
29. AT+CWMODE=mode
30.
31. //**** UNISER A UNA RED WIFI ****
32. // List Access Point
33. // Muestra una lista de las redes wifi disponibles
34. AT+CWLAP
35.
36. // Join Access Point
37. // Unirse a una red wifi existente
38. AT+CWJAP=you ssid, password
39.
40. // Check if connected successfully, or use AT+CWJAI
41. AT+CWJAP?
42.
43. // Obtener la IP del módulo
44. AT+CIFSR
45.
46.
```

ESTA WEB USA COOKIESUtilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject


```
53. // Listar los dispositivos conectados a la red gene
54. AT+CWLIF
55.
56.
57. //**** TCP SERVER ****
58. // Configura un servidor TCP en el puerto 80 (1 siq
59. AT+CIPSERVER=1,80
60.
61.
62. //**** TCP CLIENT ****
63. // Activar multiples conexiones
64. AT+CIPMUX=1
65.
66. // Conectar con el servidor remoto 192.168.1.100 er
67. AT+CIPSTART=4,"TCP","192,168.1.100",80
68.
69. // Configurar el modo de transmisión
70. AT+CIPMODE=1
71.
72. // Enviar data por el canal 4 (en el ejemplo 5 byte
73. AT+CIPSEND=4,5
```

EJEMPLOS AT

Aquí tenemos algunos ejemplos sencillos para realizar acciones básicas con el ESP01 a través de comandos AT.

Copiar

```
1. // Listar las redes WiFi y conectar a una de ella
2. // sustituir SSID y PASSWORD por los parametros de
3. AT+CWLAP
4. AT+CWJAP=SSID,PASSWORD
5.
6.
7. // Establecer conexión como cliente
8. AT+CWJAP=SSID,PASSWORD
9. AT+CIPMUX=1
10. AT+CIPSTART=4,"TCP","google.com",80
11.
12. // Establecer una conexión como servidor
13. realizar un servidor
14. AT+CWJAP=SSID,PASSWORD
15. AT+CIPMUX=1
```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

[Copiar](#)

```
1.  #include "SoftwareSerial.h"
2.  SoftwareSerial softSerial(2, 3); // RX, TX
3.
4.  const int baudRate = 9600;
5.  char* SSID = "tuWifi";
6.  char* password = "tuPassword";
7.
8.  void setup()
9.  {
10.     Serial.begin(baudRate);
11.     softSerial.begin(baudRate);
12.     delay(1000);
13.
14.     softSerial.write("AT+CWJAP=\"");
15.     softSerial.write(SSID);
16.     softSerial.write("\",\"");
17.     softSerial.write(password);
18.     softSerial.write("\r\n");
19.
20.     delay(4000);
21.     softSerial.write("AT+CIPMUX=1\r\n");
22.     delay(2000);
23.     softSerial.write("AT+CIPSERVER=1,80\r\n");
24. }
25.
26. void loop()
27. {
28.     if (softSerial.available())
29.     {
30.         Serial.print((char)softSerial.read());
31.     }
32.     if (Serial.available())
33.     {
34.         softSerial.print((char)Serial.read());
35.     }
36. }
```

USO DEL ESP8266 CON LIBRERÍA

Existen varias librerías que facilitan el uso del ESP8266 junto con Arduino, aunque no tampoco son muy abundantes ya que en general se prefiere programar el propio ESP8266. Estas librería emplean los mismos

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Si queremos usar la librería con el software serial **es necesario entrar en el fichero ESP8266.h** y descomentar la línea.

Copiar

```
1.  #define ESP8266_USE_SOFTWARE_SERIAL
```

Otra mejoras sugeridas para el empleo del ESP8266 con Arduino son incremenar el tamaño del buffer del hardware serial o software serial, para lo cuál es necesario editar el fichero \arduino\hardware\arduino\avr\cores\arduino\HardwareSerial o \arduino\hardware\arduino\avr\cores\arduino\SoftwareSerial y modificar la siguiente línea.

Copiar

```
1.  // En HardwareSerial.h
2.  #define SERIAL_BUFFER_SIZE 64
3.
4.  // En SoftwareSerial.h
5.  #define _SS_MAX_RX_BUFF 64 // RX buffer size
```

Aunque, personalmente, **no recomiendo modificar el tamaño del buffer**. Si el tamaño del buffer se queda corto es porque deberías realizar el tratamiento de los datos en el propio ESP01, o usar otra máquina, pero no modificar el tamaño del buffer.

CLIENTE WIFI - LEER PÁGINAS WEB

En este ejemplo Arduino actúa como cliente, es decir, se conecta a una página web para leerla. Leer una página completa y volcarla por el puerto serie es muy lento, y es una de las muestras de las limitaciones de Arduino frente a un ordenador.

Sin embargo, puede ser útil para Arduino capture información desde un servidor. Por ejemplo, podemos hacer que sincronice la hora, que lea una serie de parámetros de un fichero de texto, que realice una determinada acción si existe un fichero, etc.

Para mostrar en este ejemplo esta capacidad de lectura de datos desde un servidor en Internet vamos a usar www.pasted.co una de

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

'1.2.3.4.5', que simula una serie de parámetros que queremos capturar de un servidor.

El siguiente ejemplo se conecta con esta dirección y realiza la búsqueda del texto 1.2.3.4.5, que muestra por puerto serie. En un ejemplo real emplearíamos estos valores, por ejemplo, para controlar un robot, cambiar los parámetros de medición de una estación, encender o apagar un dispositivo, etc.

Copiar

```
1.  #include "ESP8266.h"
2.  #include <SoftwareSerial.h>
3.
4.  const char* SSID = "myssid";
5.  const char* PASSWORD = "mypassword";
6.  const char* HOST_NAME = "www.pasted.co";
7.  const int HOST_PORT = 80;
8.
9.  SoftwareSerial softSerial(2, 3); // RX, TX
10. ESP8266 wifi(softSerial);
11.
12. void setup(void)
13. {
14.     Serial.begin(9600);
15.
16.     if (wifi.setOprToStationSoftAP()) {
17.         Serial.print("to station + softap ok\r\n");
18.     }
19.     else {
20.         Serial.print("to station + softap err\r\n");
21.     }
22.
23.     if (wifi.joinAP(SSID, PASSWORD)) {
24.         Serial.print("Join AP success\r\n");
25.         Serial.print("IP:");
26.         Serial.println(wifi.getLocalIP().c_str());
27.     }
28.     else {
29.         Serial.print("Join AP failure\r\n");
30.     }
31.
32.     if (wifi.disableMUX()) {
33.         Serial.print("single ok\r\n");
34.     }
35.     else {
```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

```
42. void loop(void)
43. {
44.     uint8_t buffer[800] = { 0 };
45.
46.     if (wifi.createTCP(HOST_NAME, HOST_PORT)) {
47.         Serial.print("create tcp ok\r\n");
48.     }
49.     else {
50.         Serial.print("create tcp err\r\n");
51.     }
52.
53.     char *request = "GET /2434bc64 HTTP/1.1\r\nHost:
54.     wifi.send((const uint8_t*)request, strlen(request));
55.
56.     uint32_t len = wifi.recv(buffer, sizeof(buffer),
57.     if (len > 0)
58.     {
59.         Serial.print("Received:\r\n");
60.         for (uint32_t i = 0; i < len; i++)
61.         {
62.             char c = (char)buffer[i];
63.             if (c == '~')
64.             {
65.                 for (uint32_t j = i + 1; j < len; j++)
66.                 {
67.                     c = (char)buffer[j];
68.                     if (c == '~') break;
69.                     Serial.print(c);
70.                 }
71.                 break;
72.             }
73.         }
74.         Serial.print("\r\n");
75.     }
76.
77.     while (1) delay(1000);
78. }
```

SERVIDOR WIFI - ATENDER PETICIONES

En este ejemplo configuramos el módulo ESP01 para actuar como un servidor Web, es decir, recibir solicitudes por WiFi y responder a ellas.

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

```
6.
7.   SoftwareSerial softSerial(2, 3); // RX, TX
8.   ESP8266 wifi(softSerial);
9.
10.  void setup(void)
11.  {
12.      Serial.begin(9600);
13.      Serial.print("setup begin\r\n");
14.
15.      wifi.restart();
16.      delay(500);
17.
18.      if (wifi.setOprToStationSoftAP()) {
19.          Serial.print("to station + softap ok\r\n");
20.      }
21.      else {
22.          Serial.print("to station + softap err\r\n");
23.      }
24.
25.      if (wifi.joinAP(SSID, PASSWORD)) {
26.          Serial.print("Join AP success\r\n");
27.          Serial.print("IP: ");
28.          Serial.println(wifi.getLocalIP().c_str());
29.      }
30.      else {
31.          Serial.print("Join AP failure\r\n");
32.      }
33.
34.      if (wifi.enableMUX()) {
35.          Serial.print("multiple ok\r\n");
36.      }
37.      else {
38.          Serial.print("multiple err\r\n");
39.      }
40.
41.      if (wifi.startTCPServer(80)) {
42.          Serial.print("start tcp server ok\r\n");
43.      }
44.      else {
45.          Serial.print("start tcp server err\r\n");
46.      }
47.
48.      if (wifi.setTCPServerTimeout(10)) {
49.          Serial.print("set tcp server timeout 10 seconds\r\n");
50.      }
51.      else {
52.          Serial.print("set tcp server timeout err\r\n");
```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

```
59. void loop(void)
60. {
61.     uint8_t buffer[128] = { 0 };
62.     uint8_t mux_id;
63.
64.     uint32_t len = wifi.recv(&mux_id, buffer, sizeof
65. if (len > 0) {
66.     Serial.print("Received from: ");
67.     Serial.print(mux_id);
68.     Serial.print("\r\n");
69.     for (uint32_t i = 0; i < len; i++) {
70.         Serial.print((char)buffer[i]);
71.     }
72.     Serial.print("\r\n");
73.
74.     if (wifi.releaseTCP(mux_id)) {
75.         Serial.print("release tcp ");
76.         Serial.print(mux_id);
77.         Serial.println(" ok");
78.     }
79.     else {
80.         Serial.print("release tcp");
81.         Serial.print(mux_id);
82.         Serial.println(" err");
83.     }
84.
85.     Serial.print("Status: ");
86.     Serial.print(wifi.getIPStatus().c_str());
87.     Serial.println();
88. }
89. }
```

Si ahora en un navegador web introducimos la dirección IP local que se ha asignado al ESP01 veremos que el ESP8266 nos "sirve" una página web, que no es más que el contenido de la solicitud que ha emitido el navegador (en realidad parte de la solicitud).

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

solicitud, lo cuál podemos usar para realizar acciones en el lado del servidor, como veremos en la siguiente entrada.

SERVIDOR WIFI - CONTROLAR SALIDAS DIGITALES

El este ejemplo Arduino actúa también como servidor, pero esta vez queremos que el usuario pueda realizar acciones sobre Arduino a través de la página web que servimos.

Copiar

```
1.  #include "ESP8266.h"
2.  #include <SoftwareSerial.h>
3.
4.  const char* SSID = "myssid";
5.  const char* PASSWORD = "mypassword";
6.
7.
8.  SoftwareSerial softSerial(2, 3); // RX, TX
9.  ESP8266 wifi(softSerial);
10.
11. void setup(void)
12. {
13.     pinMode(LED_BUILTIN, OUTPUT);
14.
15.     Serial.begin(9600);
16.     Serial.print("setup begin\r\n");
17.
18.     wifi.restart();
19.     delay(500);
20.     if (wifi.setOprToStationSoftAP()) {
21.         Serial.print("to station + softap ok\r\n");
22.     }
```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject


```

29.     Serial.print("IP: ");
30.     Serial.println(wifi.getLocalIP().c_str());
31. }
32. else {
33.     Serial.print("Join AP failure\r\n");
34. }
35.
36. if (wifi.enableMUX()) {
37.     Serial.print("multiple ok\r\n");
38. }
39. else {
40.     Serial.print("multiple err\r\n");
41. }
42.
43. if (wifi.startTCPServer(80)) {
44.     Serial.print("start tcp server ok\r\n");
45. }
46. else {
47.     Serial.print("start tcp server err\r\n");
48. }
49.
50. if (wifi.setTCPServerTimeout(20)) {
51.     Serial.print("set tcp server timeout 20 seconds\r\n");
52. }
53. else {
54.     Serial.print("set tcp server timeout err\r\n");
55. }
56.
57. Serial.println("setup end\r\n");
58. }
59.
60. #define wifiWrite(A) wifi.send(mux_id, (uint8_t*) A, sizeof(A))
61. void loop(void)
62. {
63.     uint8_t buffer[128] = { 0 };
64.     uint8_t mux_id;
65.
66.     uint32_t len = wifi.recv(&mux_id, buffer, sizeof(buffer));
67.     if (len > 0) {
68.         Serial.print("Received from: ");
69.         Serial.print(mux_id);
70.         Serial.print("\r\n");
71.
72.         wifiWrite("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n");
73.         wifiWrite("<html>\n<head>\n<title>Luis Llamas\r\n");
74.         wifiWrite("<h2>Salidas digitales</h2>");
75.         wifiWrite("<button onClick=location.href='./?id="

```

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

```
82.     char c = (char)buffer[i];
83.     if (c == '?')
84.     {
85.         if ((char)buffer[i + 6] == '1')
86.         {
87.             digitalWrite(LED_BUILTIN, HIGH);
88.             Serial.println("LED ON");
89.         }
90.         else
91.         {
92.             digitalWrite(LED_BUILTIN, LOW);
93.             Serial.println("LED OFF");
94.         }
95.
96.         break;
97.     }
98. }
99. }
100. }
```

Para ello, servimos una página web con dos botones, que permitirán encender o apagar el LED integrado en Arduino. La extensión de la página que podemos servir es bastante reducida pero, como ya hemos repetido varias veces, si tenéis que servir páginas complejas seguramente deberíais pensar en usar el ESP8266 de forma independiente, o en usar otra máquina.

Al hacer click en los correspondientes botones, la URL solicitada tiene como parámetro '?data=0' o '?data=1'. Estos parámetros se obtienen a través de la URL de la solicitud, empleando el carácter '?' como separador.

En función de que el carácter tras '?data=' sea igual a '0' o '1' se

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject



Si te ha gustado esta entrada y quieres leer más sobre Arduino puedes consultar la sección **tutoriales de Arduino**

4.3

Article Rating



Anuncio:

🔧 Arduino, Arduino intermedio, Comunicación, ESP8266, Internet, Wifi, Wireless

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

NEXT**IMPLEMENTAR UN ARRAY DE TAMAÑO DINÁMICO EN ARDUINO**

TAMBIÉN TE PUEDE GUSTAR**LIBRERÍA ESP-COLOR**

COMUNICACIÓN RF 385, 433 Y 815MHZ CON ARDUINO Y CC1101

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)

CONECTAR UN ESP8266 O ESP32 CON UNA APLICACIÓN EN VUEJS CON VUE NATIVE WEBSOCKETS

CÓMO CONVERTIR CÓDIGO DEL ESP8266 AL ESP32

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)

COMO CAMBIAR LA FRECUENCIA DE UN PWM EN ARDUINO

ÚNETE A LA COMUNIDAD

¿Quieres seguir hablando de los temas que más te gustan con gente como tú?
¿Te has atascado en un proyecto o tienes preguntas que hacer? ¿o quieres
compartir un proyecto con toda la comunidad?



ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Únete a la conversación

28 COMMENTS



oldest ▼

Osqui ⌚ 4 years ago

Hola!

Muchas gracias por estos artículos tan estupendos!

Una pregunta: ¿merece la pena pasarse al nuevo módulo ESP32 en vez de seguir usando el ESP8266?

Gracias!

➡ Reply

luisllamas ⌚ 4 years ago

🗨 Reply to **Osqui**

Depende. El ESP32 es una bestia de procesador pero cuesta el doble que un ESP8266, que aunque sea inferior es un buen procesador.

A medida que el ESP32 baje acabará sustituyendo al ESP8266, pero a día de hoy depende de qué proyectos estés haciendo. Si no vas a usar las funciones de más del ESP32, puedes ahorrarte el sobrecoste.

➡ Reply

Osqui ⌚ 4 years ago

🗨 Reply to **luisllamas**

Mmm, entiendo. ¡Muchas gracias!

➡ Reply

victor mancilla ⌚ 3 years ago

hola muy buen articulo gracias

una consulta respecto a wemos d1 esp 8266

necesito conectar varios sensores analogicos a el pero este

solo cuenta con una entrada analoga existe alguna forma de

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Author

Luis ⌚ 3 years ago|  Reply to [victor mancilla](#)

La opción de usar un Arduino Nano, por ejemplo, es una opción válida. Aunque lo más habitual sería emplear un multiplexor como este <https://www.luisllamas.es/mas-salidas-y-entradas-en-arduino-con-multiplexor-cd74hc4067/>

 Reply**JoseM** ⌚ 3 years ago

Hola:

Muchas gracias por el tutorial, llevo sus buenas horas intentando conectar un arduino uno con el módulo, hasta ahora solo obtengo "?????????????" como salida en el monitor de serie. He intenté conectarlo con un convertidor USB-TTL y no conseguí nada distinto, he probado todas las velocidades desde 4800 a 115200 e intenté con otro módulo y nada ¿tienes alguna idea de que estoy haciendo mal?, agradecería cualquier idea.

 Reply**Carlos** ⌚ 3 years ago|  Reply to [JoseM](#)

Hola. Tienes conectada la linea GND? Me pasó algo similar y yo estaba seguro que estaba todo bien conectado (TX, RX, GND), pero verificando con más atención me encontré con que el GND estaba conectado a un pin que no llegaba a GND en mi protoboard!! Chequealo por las dudas.

 Reply**Dr T** ⌚ 3 years ago

hola, muchas gracias por poner luz en la oscuridad...! igualmente tengo el mismo problema que JoseM. Probé distintas combinaciones de velocidades entre el sketch y el monitor serie, pero en ninguna obtengo el OK.

ESTA WEB USA COOKIESUtilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject


 Reply to [Dr T](#)

Hola, yo tengo el mismo problema, pudiste solucionarlo de alguna manera?
me aparece el "ready" pero al escribir AT, nunca da el OK.

 Reply

Author

Luis  2 years ago

 Reply to [Polko](#)

Puede que venga con otro firmware cargado.
Podéis o restituir el original, o directamente pasar y reprogramar el ESP8266 con el programa que queréis, y pasar de los comandos AT

 Reply

Juancho  3 years ago

Hola!

Muy bueno el artículo!

Me sabrías decir hasta qué distancia puedo conectar dos módulos ESP8266 01 entre sí?

 Reply

Nacho  3 years ago

Hola Luis, excelente post, muy claro! Te hago una consulta, veo que para el servidor usas una llamada recv. Entiendo que esto es bloqueante, por lo que el hilo del programa quedará esperando allí hasta tener alguna comunicación entrante. Al no contar con hilos de ejecución multiples (multi-threadings) en Arduino, ¿Sabes como se manejan este tipo de cuestiones? Por ejemplo, si tuvieramos una aplicación que esté sensando otro tipo de interfaces que le den ordenes al arduino (como pulsadores), y también quisiéramos recibir comandos por medio de wifi. No se si me explico. Muchas gracias, Saludos!

 Reply

Luis Eli  3 years ago

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

 Reply**Nacho** ⌚ 3 years ago

Muchas gracias por el artículo.

¿Hay alguna forma de configurar el módulo ESP01 para que cree una red wifi a la que conectarse?

 Reply**Angela** ⌚ 3 years ago

Gracias,

Al descomentar la línea #define

ESP8266_USE_SOFTWARE_SERIAL y cargar el código nuevamente me sale el error:

serv:10: error: no matching function for call to 'ESP8266::ESP8266(SoftwareSerial&)'

 Reply**Master pro** ⌚ 2 years ago

excelente tutorial una pregunta como puedo acceder al servidor sin estar conectado a la misma red que el esp8266?

 Reply**Mike** ⌚ 1 year ago Reply to [Master pro](#)

Como en todos los casos en que hay que acceder desde internet a una conexión local hay que hacer un port forwarding en el router a la IP del módulo

 Reply**Aaron** ⌚ 2 years ago

Hola

Muchas gracias por el tutorial

Cuántos dispositivos se podrían conectar a la red wifi que se crea?

Gracias

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

, Raspberry Pi y muchos otros temas con tus publicaciones.
Te pregunto en los sketch de este artículo figura una librería que no tengo instalada y me da error. Desde donde la puedo instalar? la misma es #include "ESP8266.h"
Muchas gracias.

➤ Reply

Antonio Rueda Treviño ⌚ 2 years ago

🗨 Reply to [Edgardo Mancini](#)

Buenas localizaste la librería?

➤ Reply

larry aguilera ⌚ 2 years ago

Hola Luis,
Muchas Gracias por su portal es de gran ayuda!

Quisiera saber cual es el microcontrolador con Wifi que tenga al menos dos entradas analógicas y que sea mas económico que el arduino UNO wifi?

➤ Reply

Ricardo ⌚ 1 year ago

Buenas, tengo un arduino recogiendo datos con un Max485 de un inversor fotovoltaico.

La cuestión es que quiero poner un ESP-1 y entrando por ip, leer los datos que recoge arduino.

Por mas que busco por la red mas me lio y no se por donde empezar

➤ Reply

Author

Luis ⌚ 1 year ago

🗨 Reply to [Ricardo](#)

Hola. Prueba con la sección del ESP8266, tienes muchos tutoriales que pueden ayudarte en tu proyecto. Un saludo!

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

Consulta: He conectado un Arduino UNO al ESP01 y en Arduino cargué el sketch de la primer prueba que publicas. El problema es que sólo obtengo caracteres extraños (??>?!? ¿???R??0??N?2%K?\?×??ϕ??t??+:?4??A?) He revisado varias veces las conexiones y están correctas. Lo comunico a 9600 y el módulo está alimentado por una fuente de 3.3V. Es necesario programar el módulo o se supone que con esta prueba debiera funcionar sin tocar nada? Desde ya agradezco de antemano cualquier ayuda

➡ Reply

Hernán Gómez Vélez ⌚ 1 year ago

Buenos días, es posible conectar este módulo ESP01 con un sensor de temperatura GY-906 MLX90614 ESF, toda vez que el GY-906 viene con confección I2C. Los conectores SCL y SCA que son a 5V del GY_906 podrían quemar la placa ESP01. Es especialmente importante que sea el módulo ESP01.

MUCHAS GRACIAS.

➡ Reply

Francisco Javier ⌚ 1 year ago

Buenas noches Luís:

Intento poner en marcha una D1 mini y no lo consigo. He probado con todas las velocidades posibles y no consigo una respuesta ready, y por supuesto no hay respuesta a comandos AT. ¿Qué puedo hacer?

➡ Reply

Author

Luis ⌚ 1 year ago

🗨 Reply to **Francisco Javier**

Pásate por la sección del ESP01

<https://www.luisllamas.es/esp8266-esp32/>

Encontrarás tutoriales mucho más modernos (e interesantes) que esta que fue de las primeras, para sacar partido a ese pequeño D1 mini 😊

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject

enciende el led testigo, pero en el serial monitor no tengo ninguna respuesta. Alguna pista de que puede estar mal?

➡ Reply

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#) [Aceptar](#)

[Reject](#)



Más información



MQTT EXPLORER, UN CLIENTE GENÉRICO PARA MQTT

7 JUNIO, 2021



OPENROBOT V2, SISTEMA MODULAR DE PCB PARA PROYECTOS DE ELECTRÓNICA

20 MAYO, 2021



M5PAPER, UN DISPOSITIVO BASADO EN ESP32 CON PANTALLA EINK 4.7"

9 MAYO, 2021

LUIS

on Controlar un servo con Arduino

Hola! Gracias por tu comentario. Lo que indicas de la segunda línea no lo entiendo....

LUIS

on Comunicación inalámbrica a 2.4Ghz con Arduino y NRF24L01

Jajaja, ¡no hombre, no te llames tonto! Preguntar no te hace tonto (¡opinar de algo...

Copyright ©2018, Todos los derechos reservados

[Aviso legal](#) [Política de cookies](#)

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en [Leer más](#)

Aceptar

Reject