

## SMP:Conectividad Wi-Fi

- **¿Qué son los módulos Wi-Fi? ¿Para qué sirven?**

Son componentes electrónicos que se utilizan en algunos productos, como la placa Arduino Uno R3, para lograr una conexión inalámbrica a internet. En el pasado los módulos Wifi estaban limitados por lo que se llamaba la distancia operativa, mientras que hoy en día estos equipos proporcionan una transferencia de datos efectiva hasta distancias de varios metros.

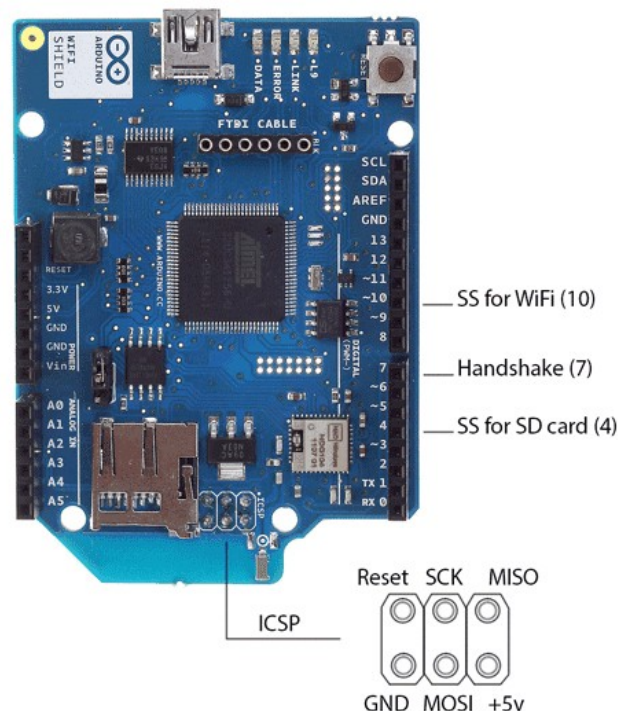
Generalmente los módulos wifi se utilizan para aplicaciones industriales, científicas y comerciales. Por ejemplo, se utilizan para sensores de red, casas inteligentes, IoT, juguetes, electrodomésticos conectados a internet, sistemas de localización en tiempo real, etc.

Hay muchos tipos de módulos wifi y se pueden clasificar por su radio frecuencia, certificación, rango operativo, tipo de encapsulado, etc.

- **Módulos Wi-Fi compatibles con microcontroladores de 8-bits y cómo programarlos.**

### **ARDUINO WIFI SHIELD**

Empezaremos hablando del Wifi Shield de Arduino. Los WiFi Shield de Arduino se trata de una PCB con un microcontrolador propio, la cual se conecta a una placa Arduino para conectar esta a internet de forma inalámbrica. Es compatible con Arduino Uno, Arduino Mega y Arduino Due. Actualmente el Arduino Wifi Shield ya no se utiliza, pues actualmente hay módulos más baratos, placas con Wifi incorporado, y otras muchas opciones que han hecho que ya no merezca la pena utilizar el Arduino Wifi Shield. También hay que destacar que el Wifi Shield de Arduino requiere obligatoriamente una placa Arduino para utilizarlo.



## HARDWARE

El Arduino Wifi Shield permite conectar nuestra placa Arduino a internet de forma inalámbrica. Este se basa en el SIP (System In-Package) HDG204. Este trae un acelerador de encriptación hardware WEP y AES. Luego este puede utilizar los tipo de encriptación WEP y WPA2.

A nivel de microcontrolador el Arduino Wifi Shield tiene un AT32UC3, un microcontrolador con arquitectura AVR de 32 bits, el cual provee una pila de red compatible con los protocolos TCP y UDP.

El Wifi Shield se puede conectar a redes inalámbricas que operen de acuerdo a los estándares 802.11b y 802.11g (2.4GHz, 11Mbps y 54Mbps respectivamente).

Respecto al hardware también comentar que en la placa encontramos una ranura para tarjetas micro-SD, la cual se puede utilizar para servir archivos a la red. El lector de tarjetas microSD es accesible usando la biblioteca SD. Además mencionar que el Wifi Shield necesita 5V para su funcionamiento, que serán proporcionados por la placa Arduino a la que la conectemos.

El Wifi Shield tiene además un puerto Mini-USB, que no es para programar el Arduino enchufado, sino para actualizar el AT32UC3. También tiene un jumper de programación que de nuevo solo se usa para actualizar el firmware.

Finalmente hay una conexión FTDI, la cual permite obtener información de diagnóstico del microcontrolador.

## CONEXIÓN CON ARDUINO

Arduino se comunica con el procesador del Wifi Shield y con la tarjeta SD por medio del bus SPI (Serial Peripheral Interface), a través del conector ICSP (conector de expansión del bus SPI). Esto es en los pines 11, 12 y 13 en la placa Arduino Uno y en los pines 50, 51 y 52 en la placa Arduino Mega. A parte de esto en ambas placas se usa el pin 10 (SS, Slave Select) para activar o desactivar el HDG204 y el pin 4 (MOSI, Master Output Slave Input, aunque el pin se usa como SS) para la tarjeta SD. Destacar que en Arduino Mega el pin 53 que también es SS, se debe mantener como salida, o en caso contrario la interfaz SPI no funcionará, ya que el hardware SPI estaría en modo esclavo en lugar de modo master, lo que haría que la librería SPI.h quedara inoperativa.

Por otro lado el pin 7 se utiliza como “hadshake” entre el Wifi Shield y la placa Arduino, ocn lo que no debe utilizarse.

[https://www.youtube.com/watch?v=dTy0Qg\\_cY6Y&ab\\_channel=Makecademy](https://www.youtube.com/watch?v=dTy0Qg_cY6Y&ab_channel=Makecademy)

## PROGRAMANDO WIFI SHIELD

Para programar la placa Arduino de forma que utilice el Wifi Shield hacemos como siempre, utilizamos el IDE de Arduino para hacer nuestro programa, y para utilizar las funcionalidades del Wifi Shield basta con utilizar la librería <Wifi.h>, para sacar partido de ella.

Por otro lado hemos mencionado que el Wifi Shield dispone también de un lector de tarjetas, al que también podemos acceder en nuestro programa por medio de la librería <SD.h>.

## EJEMPLO

```
#include <WiFi.h>
```

```
char ssid[] = "yourNetwork"; // your network SSID (name)
```

```
char pass[] = "12345678";    // your network password
int status = WL_IDLE_STATUS;    // the Wifi radio's status

void setup() {

    // initialize serial:

    Serial.begin(9600);

    // attempt to connect using WPA2 encryption:

    Serial.println("Attempting to connect to WPA network...");

    status = WiFi.begin(ssid, pass);

    // if you're not connected, stop here:

    if ( status != WL_CONNECTED) {

        Serial.println("Couldn't get a wifi connection");

        while(true);

    }

    // if you are connected, print out info about the connection:

    else {

        Serial.println("Connected to network");

    }

}

void loop() {

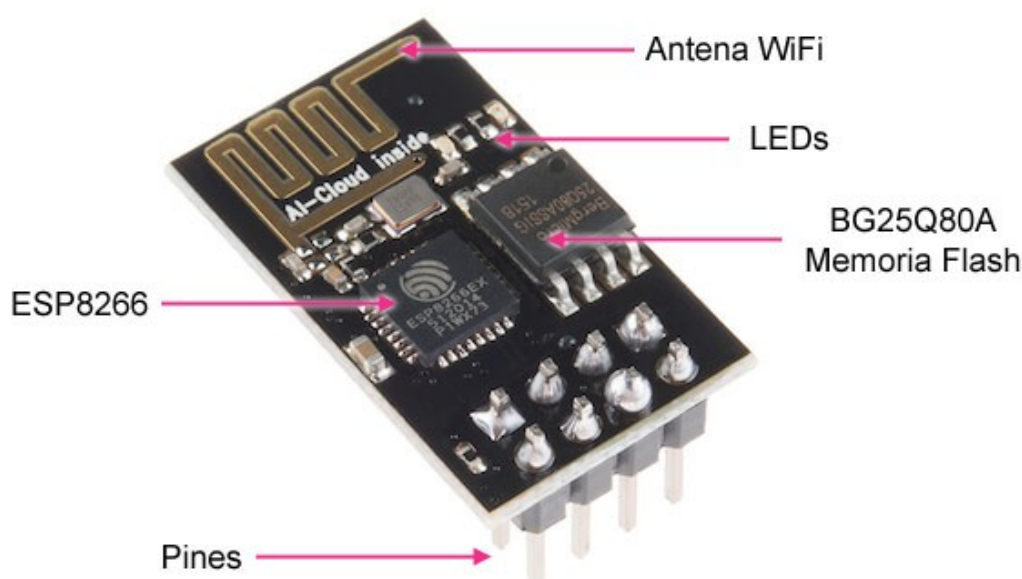
    // do nothing

}
```

## ESP-01

El módulo ESP-01 es un módulo Wifi fabricado por la empresa china Espressif. Es uno de los módulos Wifi más famosos debido a que surge como una alternativa de bajo coste al Arduino Wifi Shield ya mencionado que te permitía conectar tu placa Arduino Uno. El problema inicial que muestra este módulo es que al principio había muy poca documentación, y la mayoría de la misma en chino, y que además era muy difícil de programar, ya que había que programarlo usando comandos AT por medio del puerto serie. Esto hacía que pese a su bajo precio no mereciese la pena utilizarlo con Arduino debido a la dificultad.

Con el tiempo este módulo ha ido evolucionando, y actualmente se puede utilizar como un módulo autónomo, sin necesidad de utilizarlo con Arduino, aunque yo hablaré de cómo usarlo con este. Además con el tiempo ha ido surgiendo más documentación y ya podemos programar este módulo como si de una placa Arduino se tratara. A parte existen otros firmware que funcionan como intérpretes para lenguajes como Python, Javascript o Basic.



## HARDWARE

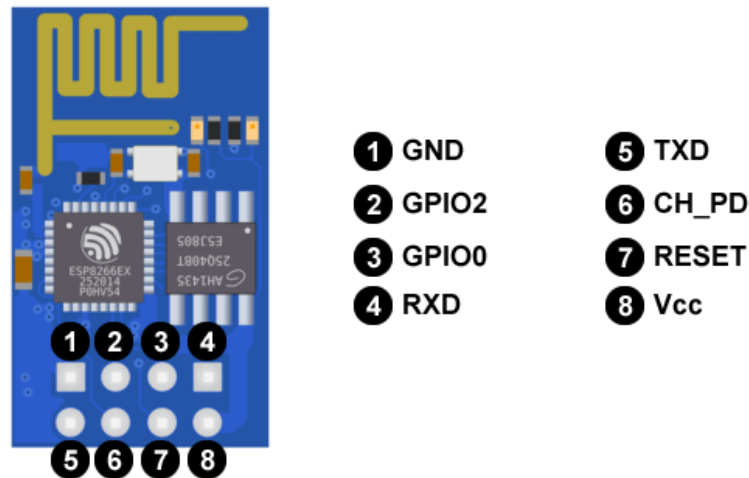
El ESP-01 tiene un microcontrolador ESP8266, que es un microcontrolador de 32 bits, al igual que el AT32UC3 del Wifi Shield. Este microcontrolador tiene las capacidades del Wifi autocontenido, es decir, no requiere de un SIP para el Wifi como el Wifi Shield de Arduino.

Este microcontrolador requiere de un voltaje de operación entre 3V y 3,6V (que es menor que el requerido por el Wifi Shield, de ahí que este sea ideal para IoT y Wreables), en concreto 3,3V es el voltaje adecuado. Una corriente de operación promedia de 80 mA, y soporta IPv4 y protocolos TCP, UDP, HTTP y FTP. Este módulo tiene 8 pines para conectar alimentación, sensores y transmisión del programa.

Tiene una memoria flash donde se pueden cargar los programas (ya hemos dicho que este módulo se puede usar sin una placa Arduino). Ahora bien el ESP8266 no dispone de memoria flash, con lo que tiene que ir aparte.

Tiene también una antena Wifi y unos LEDs que nos informan de si está encendido y de la transmisión de datos.

Soporta los principales buses de comunicación (SPI, I2C, UART-Universal Asynchronous Receiver-Transmitter)



Este módulo tiene dos pines GPIO que son pines entrada/salida de propósito general.

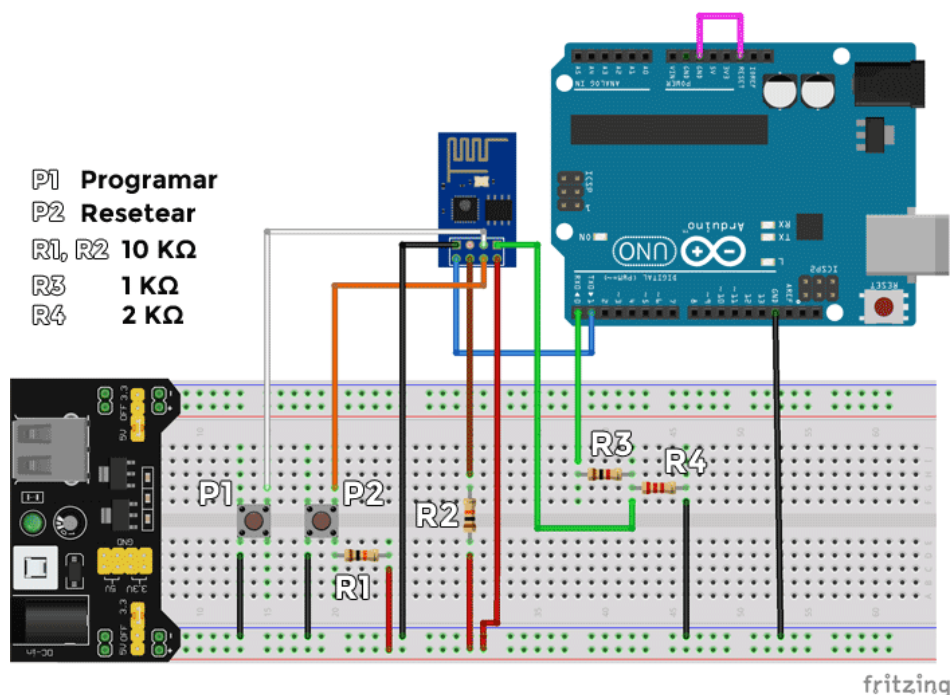
1. GND es la toma de tierra.
2. GPIO2 es una entrada salida de propósito general. Es el pin digital número 2.
3. GPIO0 es una entrada salida de propósito general. Es el pin digital número 0.
4. RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 3.
5. TXD es el pin por donde se van a transmitir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 1.
6. CH\_PD pin para apagar y encender el ESP-01: si lo ponemos a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.
7. RESET pin para resetear el ESP-01: si lo ponemos a 0 V (LOW) se resetea.
8. Vcc es por donde alimentamos el ESP-01. Funciona a 3,3 V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA.

Ya hemos dicho que podemos alimentar nuestro módulo utilizando Arduino UNO, aunque no es lo más ideal, pues aunque Arduino UNO tiene un pin de 3,3V, este puede suministrar un máximo de 50 mA, los cuales se pueden quedar cortos. Se puede hacer y funcionará pero puede fallar.

Lo ideal es utilizar una fuente de alimentación externa, pues este módulo no tiene puerto que permita alimentarlo de forma directa.

## CONEXIÓN CON ARDUINO

Como ya hemos dicho la placa Arduino no es capaz de proporcionar la corriente necesaria, así que lo mejor es complementarlo con una fuente externa.



- El TX del Arduino conectado al RX del ESP-01
- El RX del Arduino conectado a la entrada del divisor de tensión
- La salida del divisor de tensión al RX del ESP-01
- Se hace un puente entre el pin GND y el pin RESET en el Arduino UNO (esto aísla al procesador de los pines GPIO, convirtiéndolo en una pasarela)
- Conectamos todas las GND entre sí
- GND de Arduino
- GND del ESP-01
- La tierra del divisor de tensión
- GND de la fuente de alimentación
- El pin CH\_PD del ESP-01 a 3,3 V con una resistencia de 10 KΩ en serie
- El pin RESET del ESP-01 a 3,3 V con una resistencia de 10 KΩ en serie
- Para habilitar el modo de programación UART, conectamos el pin GPIO0
- Por último quedaría el pin GPIO2 que, como esta placa tiene una resistencia pull-up, lo podemos dejar al aire

## PROGRAMANDO

A diferencia de las placas Arduino, a este módulo hay que indicarle cuando se va a cargar un programa y cuando se va a ejecutar. Para esto ESP-01 dispone de dos modos que se configuran usando los puertos GPIO0 y GPIO2: modo Flash y modo UART.

El modo UART se usa cuando se quiere cargar un programa en el ESP-01, para ello se pone el pin GPIO0 a nivel bajo y el GPIO2 a nivel alto. Como por defecto el GPIO2 está alto basta con dejarlo sin conectar.

El modo Flash se usa para ejecutar los programas, para activarlo hay que poner GPIO0 y GPIO2 a HIGH, que como ya lo están por defecto basta con dejarlos desconectados.

	GPIO0	GPIO2
<b>Modo UART</b> (carga programa)	LOW	HIGH (desconectado)
<b>Modo Flash</b> (ejecuta programa)	HIGH (desconectado)	HIGH (desconectado)

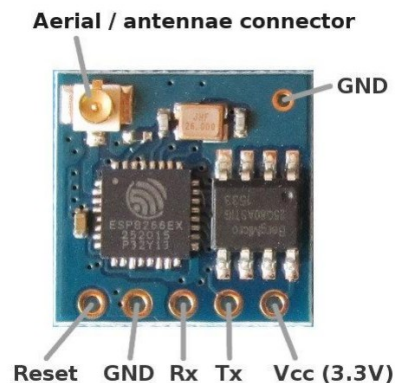
Por último, para programar el módulo se puede utilizar el mismo Arduino IDE, pero hay que descargar los drivers del ESP8266, seleccionar la placa que vamos a programar, en este caso Generic ESP8266 Module y ya podemos programarlo

Para utilizar las funcionalidades Wifi debemos utilizar la biblioteca <ESP8266WiFi.h>

## ESP-XX

A parte del ESP-01 hay otros muchos módulos, los cuales tienen MCU de la serie ESP pero de distintos modelos para distintos usos.

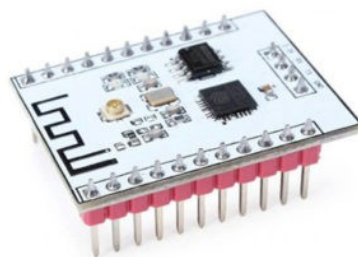
- ESP-05: módulo muy simple. Está destinado a ser un Shield Wifi para Arduino. Es muy fácil de conectar a la protoboard, pero no tiene ningún puerto GPIO accesible.



- ESP-12: Este ha desbancado al ESP-01 como el más utilizado. Este sí tiene puertos GPIO, pero no es fácil de conectar con la protoboard, porque si o sí hay que soldar.



- ESP-201: son ideales para prototipar. Disponen de puertos GPIO, son fácilmente encajables en la protoboard y permite el acople de una antena externa.



Todos los anteriores son compatibles con Arduino, pero aunque se alimenten con 3.3V requieren de una fuente de alimentación externa, ya que Arduino no proporcionar la corriente necesaria en el pin de 3.3V.

## NodeMCU



Vamos a hablar también de NodeMCU, una placa de desarrollo basada en ESP8266. La ventaja que tiene NodeMCU respecto a las placas Arduino con Wifi integrado es que su precio es muy reducido en comparación.

Ya hemos mencionado que el NodeMCU está basado en ESP8266, pero no solo eso sino que concretamente NodeMCU está basado en ESP-12, que a su vez está basado en ESP8266.

## **HARDWARE**

A diferencia de los módulos ESP-XX, NodeMCU trae un conversor serie USB para programar y alimentar la placa a través del USB.

Tiene fácil acceso a los pines, botón de reset y LEDs para indicar estado.

En cuanto a los pines de los que dispone

- 13 pines digitales numerados del D0 al D12
- 1 pin analógico numerado A0
- 3 pines de 3,3V
- 1 pin de 5V (versión V3 2 pines 5V)
- 4 pines de tierra GND (versión V3 5 pines GND)



## **PROGRAMACIÓN**

Al principio esta placa estaba pensada para ser programada en LUA, pero con el tiempo han surgido drivers que han permitido programarla en otros lenguajes de programación, y es más tras la integración de ESP8266 con el entorno de Arduino ha permitido la programación de este usando el IDE de Arduino.

Es igual de simple que programar los módulos ESP antes visto. Podemos utilizar la IDE de Arduino, seleccionar la placa que vamos a programar y ya podemos cargar directamente el programa. Aquí no hace falta como con los módulos ESP-XX indicar la carga y la ejecución sino que trabaja directamente como lo haría una placa Arduino.

## **ARDUINO UNO WIFI R2**

Por último vamos a ver la placa de desarrollo Arduino que trae integrado Wifi. Esta es la Arduino Uno Wifi R2 (hay más). La principal desventaja de esta placa con respecto a los módulos vistos anteriormente y con NodeMCU es que cuesta cerca de 40€, mientras que ninguno de los módulos anteriormente vistos llegaba a los 10€.

## **HARDWARE**

Otra gran diferencia de esta placa con lo ya visto es que no está basada en el ESP8266, sino que tiene un MCU ATmega4809, que es un controlador de 8-bits y por separado tiene un SoC u-blox NINA-W102, que proporciona la funcionalidad Wifi según los estándares 802.11b/g/n y además también proporciona Bluetooth v4.2.



Sino que vemos que tiene un MCU ATmega de 8 bit como estamos acostumbrados a ver y además un SoC que se encarga de las funcionalidades Wifi y Bluetooth que es de 32-bits.

En cuanto a la programación, tenemos que es un placa Arduino, luego claramente es compatible con el entorno de Arduino, luego cargamos y ejecutamos programas de forma normal por medio del USB.

Esta placa para su programación (si se quieren usar funcionalidades Wifi) tiene que usar la biblioteca <WifiNINA.h>, cuya documentación es fácil de encontrar.