**Modulo wifi ESP8266-01**

Repositorio : <https://github.com/jalector/ESP8266>

**Introducción a la problemática**

Al intentar programar el módulo wifi ESP8266 desde el IDE Arduino, sin realizar ninguna configuración previa, lo más probable es que se presenten errores y falle la acción.

Algunos de los errores más comunes generados en este caso en el IDE Arduino son:

**“warning: espcomm\_send\_command: cant receive slip payload data**

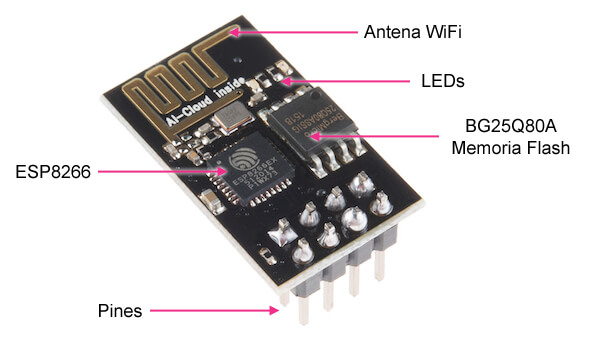
**warning: espcomm\_sync failed**

**error: espcomm\_open failed**

**error: espcomm\_upload\_mem failed”**

Los cuales se presentan debido a que el microcontrolador del Arduino recibe las señales de programación, las cuales no son soportadas por el mismo, ya que las instrucciones enviadas son especiales para el módulo ESP8266 y no son soportadas por Arduino.

**Placa Wifi ESP8266-01**



Tiene disponible dos pines GPIO digitales para controlar sensores y actuadores.

También se puede llegar a utilizar para este uso los pines Rx y Tx si no se utilizan para la comunicación a través del puerto serie. Se puede programar a través de un adaptador serie/USB o con el cableado adecuado, a través de Arduino. Los conectores que vienen por defecto, no permite conectarlo a la protoboard.

Esto dificulta prototipar con este módulo. Sin embargo, lo podemos usar como un dispositivo autónomo o como complemento con Arduino.

**Especificaciones de la placa ESP8266-01**

**Hardware**

* Introducción a la problemática
* Utiliza una CPU Tensilica L106 32-bit
* Voltaje de operación entre 3V y 3,6V
* Corriente de operación 80 mA
* Temperatura de operación -40ºC y 125ºC

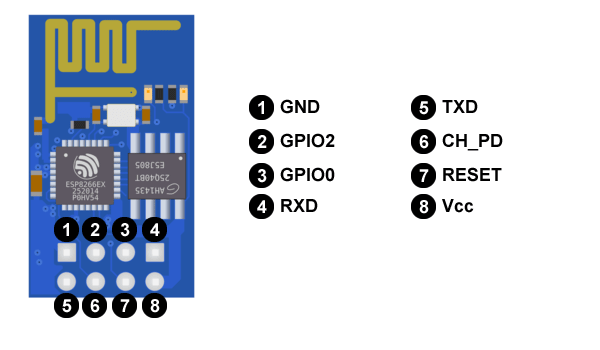
**Conectividad**

* Soporta IPv4 y los protocolos TCP/UDP/HTTP/FTP

**Consumos**

El consumo de energía dependerá de diferentes factores, como el modo en el que se esté trabajando, el ESP8266, de los protocolos que estemos utilizando, de los protocolos que estemos utilizando, de la calidad de la señal WiFi y sobre todo de sí enviamos o recibimos información a través de la WiFi. Oscilan entre los 0,5 μA (microamperios) cuando el dispositivo está apagado y los 170 mA cuando transmitimos a tope de señal.

**Diagrama de placa ESP8266-01**



**Pines:**

1. **Tierra (Conectada a la tierra de Arduino y si se tiene fuentes de alimentación externa, conectar tierra común).**
2. **Pin Digital usado para enviar o recibir señales como los pines de Arduino.**
3. **Pin Digital usado para enviar o recibir señales como los pines de Arduino. Además usado para activar modo de programación del módulo al conectarse a Tierra por medio de una resistencia de 10kOhms.**
4. **Pin de recepción de datos.**
5. **Pin de transmisión de datos.**
6. **Usado para prender y apagar el módulo al desconectar y volver a conectarlo a voltaje. Siempre debe estar conectado. Recibe 3.3V (IMPORTANTE!!! No puede recibir más voltaje porque se puede quemar).**
7. **Usado para reiniciar el módulo al conectarlo y desconectarlo de Tierra.**
8. **Pin que recibe la corriente de voltaje 3.3V (IMPORTANTE!!! No puede recibir más voltaje porque se puede quemar).**

**Justificación de retirar microcontrolador**

Es necesario retirar el microcontrolador de Arduino para que solamente actúe como un puente entre el puerto serial y el módulo wifi, y así poder conectar directamente los pines de transmisión (TX) y recepción (RX) de Arduino con los del módulo Wifi.

Una vez retirado el microcontrolador se puede cargar cualquier sketch en la memoria del módulo sin que Arduino presente alguna interferencia.

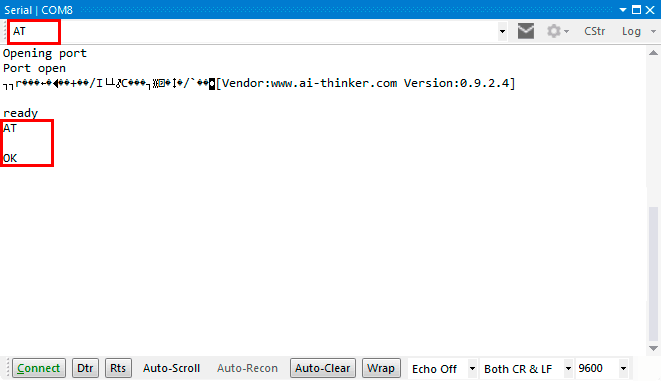
**Pasos para programar placa ESP8266-01**

Habiendo entendido el funcionamiento de la placa ESP8266, haremos uso de su microcontrolador. Para ello en un paso anterior se retira el microcontrolador de la placa arduino.

1. Conectar pin GND (ground) a tierra.
2. Conectar pin VCC a voltaje ( 3.3v ).
3. Conectar pin
4. Conectar pin RX del módulo wifi al pin RX del arduino.
5. Conectar pin TX del módulo wifi al pin RX del arduino.

Una vez realizadas las conexiones se procede a conectar el Arduino (En nuestro caso Arduino-UNO ) en nuestra computadora. Después se seguirán los siguientes pasos:

1. Abrir el programa Arduino si es que no lo tenemos abierto
2. Abrir la consola serial
3. Poner 9600 o 115200 baudios (La frecuencia en la que funciona el módulo puede ser de 9600 o 115200 baudios, esto puede variar en cada módulo ESP8266, en nuestro caso, nos funcionó con 115200 baudios).
4. Poner la consola serial con **Ambos NL & CR** para mostrar información en el monitor serial.
5. Reiniciar placa **ESP8266-01** desconectando y conectando el pin **CH\_PD** a VCC
6. En el monitor serial se apreciara una serie de caracteres y la palabra ready, lo que indica que está listo para recibir los comandos AT.



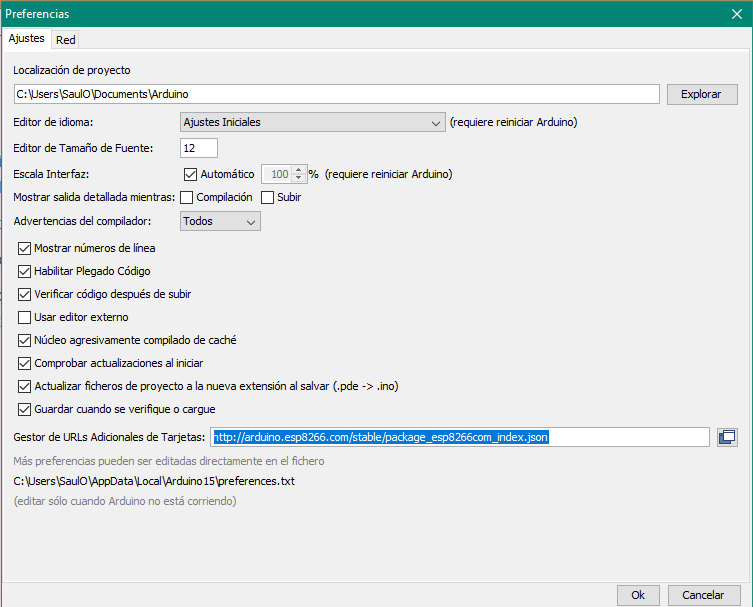
1. Ingresar el comando **AT** y responderá con un **OK** lo que indica que existe comunicación
2. Introducir el comando **AT+CWMODE=3** para poner el módulo en modo estación y access point (Conectarse a una red y crear una red)
3. Verificar que el módulo se encuentre en ese modo introduciendo el comando **AT+CWMODE?** (La respuesta debe de ser 3).

Lista de comandos AT: <https://www.itead.cc/wiki/ESP8266_Serial_WIFI_Module#AT_Commands>

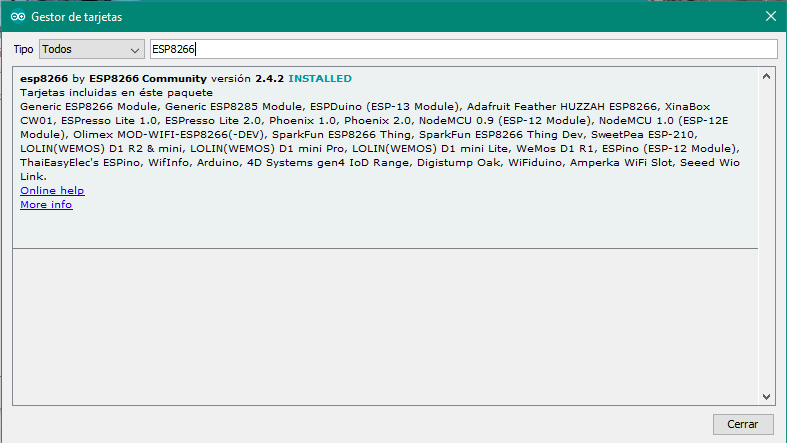
**Configurar el IDE de Arduino para poder programar en la placa ESP8266:**

**Instalar la placa ESP8266-01 en arduino**

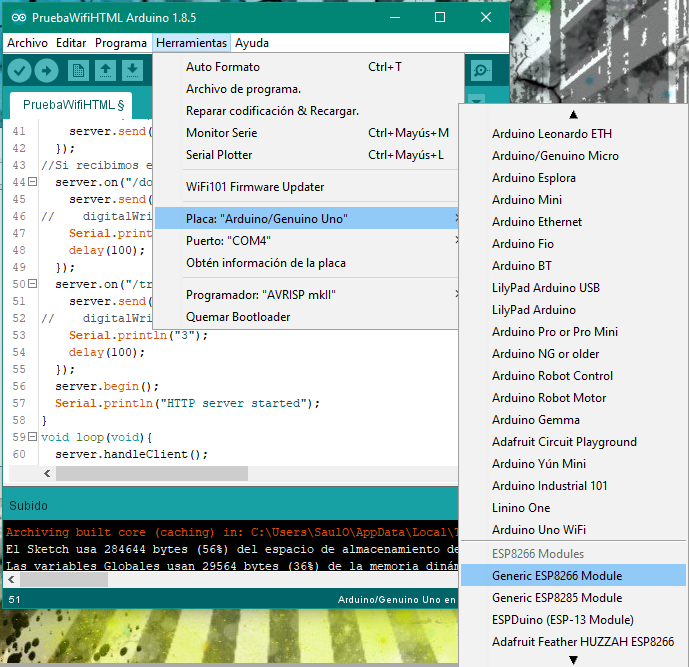
1. En el menú superior ir a: ***Archivo > Preferencias***
2. En gestor de URLs y tarjetas poner insertar lo siguiente y dar click en OK: <http://arduino.esp8266.com/stable/package_esp8266com_index.json>



1. Después ir a ***Herramientas > Placa > Gestor de tarjetas***
2. Buscar **ESP8266**
3. Instalar **ESP8266**



1. Ahora nos vamos a: ***Herramientas > Placa > Generic ESP8266 Module***



**Nota:** Cuando queramos programar el módulo **ESP8266-01** en el IDE se requiere cambiar la placa objetivo, tanto para programar el módulo como para programar el Arduino. Hay que asegurarse que la placa objetivo sea la deseada.

**Subir un sketch al módulo ESP8266-01**

Una vez realizados las configuraciones anteriores, tanto con los comando AT como al IDE Arduino, se podrá sobreescribir el firmware del módulo con un programa que deseemos que se ejecute, el cual contiene las librerías necesarias para conectarse a una red y/o crear una nueva red, además de controlar sensores o realizar acciones con distintos componentes.

Los pasos a seguir son los siguientes:

1. Realizar la conexión explicada en puntos anteriores (la misma que al introducir comandos AT).
2. Abrir el monitor serial.
3. Desconectar el pin CH\_PD de la alimentación de voltaje.
4. Conectar el pin GPIO0 a Tierra por medio de una resistencia de 10kOhms para entrar al modo de programación.
5. Volver a conectar el pin CH\_PD de la alimentación de voltaje, para reiniciar el módulo.
6. Una vez reiniciado el módulo, el monitor serial nos mostrará una serie de caracteres sin sentido pero no ejecutará ningún programa ni mostrará la palabra ready, debido a que estará esperando el programa que será cargado.
7. Se carga el programa en el módulo ESP8266 y una vez cargado se desconecta el pin GPIO0.
8. Se ejecutará el programa cargado en el módulo.

Un ejemplo que pueden probar para cargar un programa al módulo ESP8266 es el siguiente: <https://www.youtube.com/watch?v=F79_EFFDoAA>

Solamente cambien la acción realizada por el módulo en lugar de enviar un tweet.

En Internet vienen varios ejemplos de programas para utilizar el módulo como cliente y/o servidor web.

**Referencias bibliográficas**

<https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>

<https://github.com/esp8266/Arduino/issues/2801>