

# MÓDULO 5: "ESP8266"

---

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS



# ¿Qué es el ESP8266?

---

- El ESP8266 es un **microprocesador de bajo coste con Wifi integrado**.
- Antes del ESP8266, las opciones disponibles para conectar un Arduino a Wifi (como el Wifi Shield) eran prohibitivamente caras. La aparición del ESP8266 supuso una pequeña revolución al ser el primer dispositivo realmente barato que proporcionaba conectividad WiFi.
- 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.

# ¿Qué es el ESP8266?

---

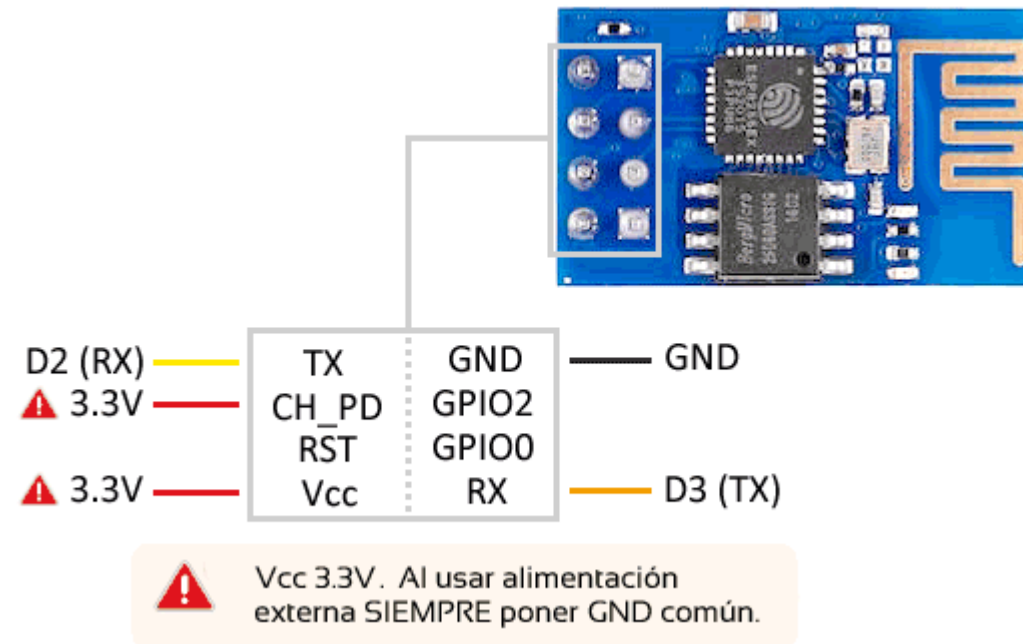
- Nos limitaremos a emplear **Arduino junto con un módulo ESP01**, un de los primeros en aparecer con el chip ESP8266 y uno de los módulos más sencillos y baratos.
- El ESP01 monta la versión más sencilla del ESP8266. Aún así, sigue siendo un procesador de 32bits a 80 Mhz, con 512kB o 1MB de memoria 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, la comunicación con Internet puede suponer una carga excesiva para Arduino.

# Esquema de montaje

---

- La conexión de un módulo ESP01 con Arduino no es complicada, aunque en la práctica van a aparecer ciertas dificultades. **La principal dificultad es la alimentación del ESP01.** El ESP8266 y, en particular, el ESP01, tiene una tensión de alimentación de 3.3V.
- 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 útil del ESP01.

# Esquema de montaje



- 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.

# Esquema de montaje

---

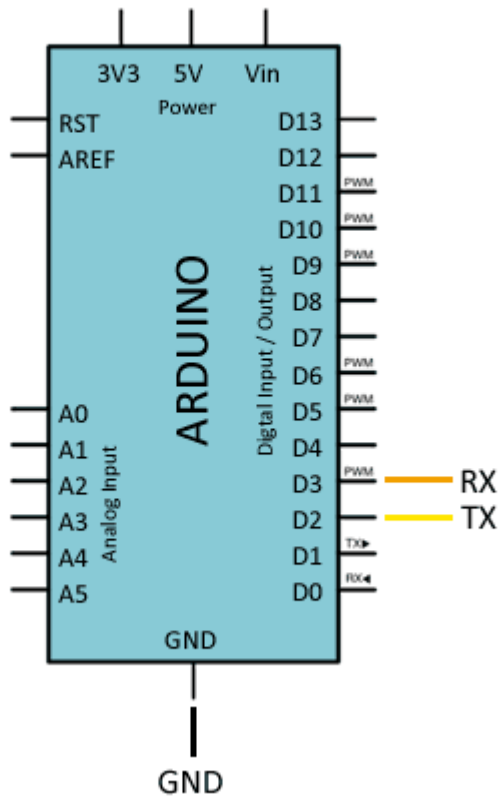
- Finalmente, **la comunicación con el módulo se realiza mediante puerto serie**. A diferencia de la alimentación del módulo, que en ningún caso debe exceder los 3.6V, existe un amplio debate sobre si los pines RX y TX son tolerantes a 5V (Es decir, si pueden conectarse directamente a un sistema a un 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.
- Para la conexión con Arduino podemos emplear el puerto serie normal, pero esto implica que tendremos que desconectar el ESP01 cuando queramos programar Arduino, y no podremos usar el puerto serie para la comunicación con el PC.

# Esquema de montaje

---

- 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.

# Esquema de montaje



- 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.



# Ejemplos de código

---

- Uso del ESP8266 con comandos AT
- 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.

# Ejemplos de código

---

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

# Ejemplos de código

---

- **Primera prueba**

- 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.
- A continuación, un listado de algunos de los comandos AT disponibles para el ESP8266 (archivo \*.ino).

- **Ejemplos AT**

- Dejo en la carpeta ejemplos sencillos para realizar acciones básicas con el ESP01 a través de comandos AT (archivo \*.ino).
  - Ejemplo, ESP01 desde un Arduino para actuar como servidor.

# Ejemplos de código

---

- **Primera prueba**

- 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.
- A continuación, un listado de algunos de los comandos AT disponibles para el ESP8266 (archivo \*.ino).

- **Ejemplos AT**

- Dejo en la carpeta ejemplos sencillos para realizar acciones básicas con el ESP01 a través de comandos AT (archivo \*.ino).
  - Ejemplo, ESP01 desde un Arduino para actuar como servidor.

# Uso de 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 comandos AT que hemos visto anteriormente y gestionan las respuestas recibidas, evitándonos a nosotros el trabajo.
- Una de las más completas es la librería ITEADLIB Arduino WeeESP8266, disponible en este enlace.
- Si queremos usar la librería con el software serial es necesario entrar en el fichero ESP8266.h y descomentar la línea.

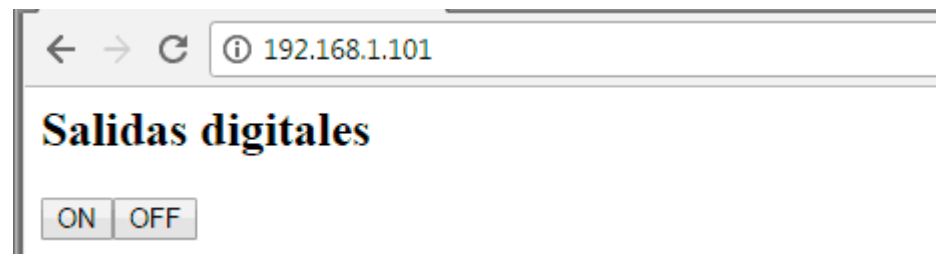
# Uso de ESP8266 con librería

---

- Si queremos usar la librería con el software serial es necesario entrar en el fichero ESP8266.h y descomentar la línea.
  - `#define ESP8266_USE_SOFTWARE_SERIAL`

# 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.
- 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.



# Servidor WiFi – Controlar salidas digitales

---

- 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 enciende o apaga el LED integrado, y se muestra la acción por el puerto serie.