**ARDUINO: PRACTICAS WIFI**

**Instalación de la libraría ESP-Wifi.**

En la carpeta Wifi está incluida la librería que utilizaremos para hacer las prácticas.

Para incluir la librería abrir el IDE de arduino y seleccionar en el menú:

*Programa🡪 Incluir Librería.--> Añadir librería .ZIP*

Buscar el zip de la librería donde hayas descargado las prácticas:

*..\Wifi\Libreria WifiESP\ WiFiEsp-master.zip*

La librería dispone de varios ejemplos de uso.

* [ConnectWPA](https://github.com/bportaluri/WiFiEsp/blob/master/examples/ConnectWPA/ConnectWPA.ino) - Demonstrates how to connect to a network that is encrypted with WPA2 Personal
* [WebClient](https://github.com/bportaluri/WiFiEsp/blob/master/examples/WebClient/WebClient.ino) - Connect to a remote webserver
* [WebClientRepeating](https://github.com/bportaluri/WiFiEsp/blob/master/examples/WebClientRepeating/WebClientRepeating.ino) - Make repeated HTTP calls to a webserver
* [WebServer](https://github.com/bportaluri/WiFiEsp/blob/master/examples/WebServer/WebServer.ino) - Serve a webpage from the WiFi shield
* [WebServerAP](https://github.com/bportaluri/WiFiEsp/blob/master/examples/WebServerAP/WebServerAP.ino) - Serve a webpage from the WiFi shield starting a local Access Point
* [WebServerLed](https://github.com/bportaluri/WiFiEsp/blob/master/examples/WebServerLed/WebServerLed.ino) - Turn on and off a led from a webpage
* [UdpNTPClient](https://github.com/bportaluri/WiFiEsp/blob/master/examples/UdpNTPClient/UdpNTPClient.ino) - Query a Network Time Protocol (NTP) server using UDP

**Mini introducción a los módulos Wifi basados en ESP8266.**

Los módulos basados en ESP8266 se comunican con el arduino a través del puerto serie.

Las comunicaciones están basadas en comandos AT.

Para manejar el módulo wifi se crea un conexión por puerto serie y se envían los comandos AT, la respuesta a los comandos se recibe también por el puerto serie.

Al final se adjunta una lista de comandos AT soportados por los módulos wifi ESP8266 (en castellano) y en la carpeta Wifi hay un pdf con un manual del fabricante con información detallada (en Ingles)

Para facilitar el manejo de los módulos existen varias librerías que permiten abstraernos de los comandos AT y manejar el módulo wifi mediante las clases que la librería proporciona.

Los módulos wifi vienen configurados a una velocidad desconocida, normalmente115200 baudios. En los arduino con un único puerto serie hardware, este puerto se utiliza para depuración y por tanto se tiene que emular otro puerto serie que será el que se comunicará con el módulo wifi. Los puertos serie emulados no funcionan bien a altas velocidades por lo que es necesario bajarla. La velocidad recomendada es 9600.

**Clases de la librería WifiESP**

La mayor parte de los métodos de la librería standard de Arduino Wifi están disponibles.

Para más detalles ir a [WiFi library page](http://www.arduino.cc/en/Reference/WiFi).

**WiFiEsp class**

* begin() - Not all authentication types
* disconnect() - YES
* config()
* setDNS() - NO (no AT command available)
* SSID() - YES
* BSSID() - YES
* RSSI() - YES
* encryptionType() - NO (no AT command available)
* scanNetworks() - YES
* getSocket()
* macAddress() - YES

**WiFiEspServer class**

The WiFiEspServer class creates servers which can send data to and receive data from connected clients (programs running on other computers or devices).

* WiFiEspServer() - YES
* begin() - YES
* available() - YES
* write() - YES
* print() - YES
* println() - YES

**Client class**

The WiFiEspClient class creates clients that can connect to servers and send and receive data.

* WiFiEspClient() - YES
* connected() - YES
* connect() - YES
* write() - YES
* print() - YES
* println() - YES
* available() - YES
* read() - YES
* flush() - YES
* stop() - YES

**WiFiEspUDP class**

The UDP class enables UDP message to be sent and received.

* WiFiUDP - YES
* begin() - YES
* available() - YES
* beginPacket() - YES
* endPacket() - YES
* write() - YES
* parsePacket() - YES
* peek()
* read() - YES
* flush()
* stop()
* remoteIP() - YES
* remotePort() - YES

**Comandos AT soportados por los módulos ESP8266**

La siguiente tabla está actualizada hasta la versión 0.20, que corresponde con el firmware 0.9.4 de diciembre de 2014, la versión (oficial) más avanzada que soportan los módulos ESP8266 con 4 Mbit, que son la mayoría de los disponibles actualmente.

|  |  |
| --- | --- |
| **Orden AT** | **Descripción** |
| AT | Sirve para verificar si el sistema de órdenes AT está operativo |
| AT+GMR | Muestra información sobre el firmware del módulo |
| AT+RST | Reinicia el módulo |
| ATE1  ATE0 | Activa o desactiva, respectivamente, el eco de los comandos AT. Normalmente el eco está activo por lo que se muestra tanto el resultado de la operación como la propia operación. Con esta orden se puede reservar alguna información, la clave del punto de acceso, por ejemplo, mientras se muestra por una consola el resto de la información. |
| AT+GSLP | Entra en modo de reposo (si el hardware lo permite) durante el tiempo indicado en milisegundos con el formato AT+GSLP={tiempo} siendo {tiempo} el tiempo de reposo expresado en milisegundos. |
| AT+CWMODE | Establece el modo de funcionamiento WiFi del módulo. Hay tres disponibles (1) estación, (2) punto de acceso y (3) mixto: punto de acceso y estación. Por ejemplo, para establecer el modo de funcionamiento WiFi del módulo como estación se usaría AT+CWMODE=1 y para consultar el modo actual de funcionamiento se usaría AT+CWMODE? |
| AT+CIOBAUD  AT+IPR | Permite establecer la velocidad de comunicación del módulo (expresada en baudios) según el formato AT+CIOBAUD={velocidad} Utilizada con el formato AT+CIOBAUD? informa de la velocidad actualmente establecida. Los valores permitidos son 9600, 19200, 38400, 57600, 74880, 115200, 230400,460800, 921600 y la configuración por defecto es 115200 En realidad, por algún tipo de error, CIOBAUD no funciona y sí IPR (que no está en la documentación) El inconveniente es que AT+IPR? no está implementado y hay que cambiar la velocidad a ciegas. |
| AT+CSYSWDTENABLE  AT+CSYSWDTDISABLE | Activa o desactiva el watchdog (perro guardián) que permite reiniciar el módulo automáticamente si se produce un error de funcionamiento. |
| AT+CIPSTAMAC  AT+CIPAPAMAC | Permite establecer una dirección MAC o consultar la actual tanto en el punto de acceso (CIPAPAMAC) como en la estación (CIPSTAMAC). Para establecer una nueva dirección MAC se utiliza la orden con el formato AT+CIPSTAMAC={MAC} siendo {MAC} la dirección MAC en el formato 01:34:67:89:AB:CD Para consultar la dirección MAC actual se usa la orden con el formato AT+CIPSTAMAC? |
| AT+CWLAP | Muestra una lista con los puntos de acceso disponibles en el alcance del módulo, indicando para cada uno de ellos la codificación, el SSID, el nivel de señal y la dirección MAC. Las diferentes codificaciones se representan por un número: abierta=0, WEP=1, WPA/PSK=2, WPA2/PSK=3 y WPA/WPA2/PSK=3. Se puede limitar la búsqueda a los puntos de acceso que cumplan ciertas condiciones como el SSID, la MAC o el número de canal indicándolos en la orden. |
| AT+CWJAP | Usando la orden con el formato AT+CWJAP={SSID},{clave} el módulo se conecta al punto de acceso indicado por el SSID usando la clave especificada. Para saber a qué punto de acceso se encuentra actualmente conectado el módulo se puede usar la orden en el formato AT+CWJAP? |
| AT+CWQAP | Desconecta el punto de acceso. Para comprobar si se ha desconectado se puede usar la orden con el formato AT+CWQAP? |
| AT+CWSAP | Configura el modo de funcionamiento del punto de acceso según el formato AT+CWSAP={SSID},{clave},{canal},{cifrado} Los modos de cifrado corresponden con los códigos abierta=0, WEP=1, WPA/PSK=2, WPA2/PSK=3 y WPA/WPA2/PSK=4 |
| AT+CWDHCP | Activa o desactiva el DHCP (asignación dinámica de dirección IP) en el punto de acceso y/o en la estación. El formato de la orden es AT+CWDHCP={modo},{estado} siendo {modo} el código del que se activa (0 para el punto de acceso, 1 para la estación o 2 para ambos) y {estado} activar/desactivar DHCP según se use 1 ó 0 como valor. |
| AT+CIPSTA  AT+CIPAP | Establece la dirección IP de la estación (AT+CIPSTA) o del punto de acceso (AT+CIPAP) usada en el formato AT+CIPSTA={IP} o muestra la actual usada en el formato AT+CIPSTA? |
| AT+CWLIF | Muestra una lista de las direcciones IP de los dispositivos conectados al módulo en modo punto de acceso. |
| AT+CIFSR | Muestra la dirección IP local actual del módulo |
| AT+CIPMUX | Establece el modo de conexión simple o múltiple según se use con el formato AT+CIPMUX=0 ó AT+CIPMUX=1 respectivamente. |
| AT+CIPSERVER | Activa o desactiva el modo de servidor según se use en el formato AT+CIPSERVER=0 (que requerirá un reinicio para hacerse efectivo) o AT+CIPSERVER=1,{puerto} que comenzará a aceptar peticiones en el puerto correspondiente. Para poder establecer el modo servidor debe activarse el modo multi-conexión con AT+CIPMUX=1 |
| AT+CIPSTO | Configura el tiempo de espera (timeout) cuando el módulo funciona en modo servidor con el formato AT+CIPSTO={segundos} Si se usa con el formato AT+CIPSTO? informa del timeout actual. |
| AT+CIPSTART | Inicia una conexión con un servicio. Es necesario indicar el tipo de conexión (TCP/UDP) la dirección IP (o el nombre del servidor, si se tiene acceso a un [DNS](https://es.wikipedia.org/wiki/Domain_Name_System)) y el puerto al que se realiza la conexión con el formato AT+CIPSTART={id},{TCP|UDP},{IP},{puerto} Siendo {id} el número de identificador de la conexión (si se aceptan varias, es decir si se utilizó AT+CIPMUX=1, si AT+CIPMUX=0 no hay que indicarlo) También es posible añadir al final de la orden (separado por comas como el resto de parámetros) el puerto local y el modo si se trata de una conexión UDP. Usada con el formato AT+CIPSTART=? muestra la información anterior relativa a la conexión actualmente en curso. |
| AT+CIPSTATUS | Muestra el estado de la conexión actual en el formato STATUS:{estado} siendo {estado} un valor que representa el estado de la conexión como desconectado por un 4, con una IP asignada por un 2, conectado por un 3 y conectado con IP asignada por un 5. Después del estado genérico detalla la conexión en el formato +CIPSTATUS:{id},{tipo},{dirección},{puerto},{modo} siendo {modo} un valor que representa si el módulo está funcionando como cliente (0) ó como servidor (1) |