

Práctica 5 Comunicaciones WIFI y stack IP con ESP32

5.1 Objetivos de la práctica:

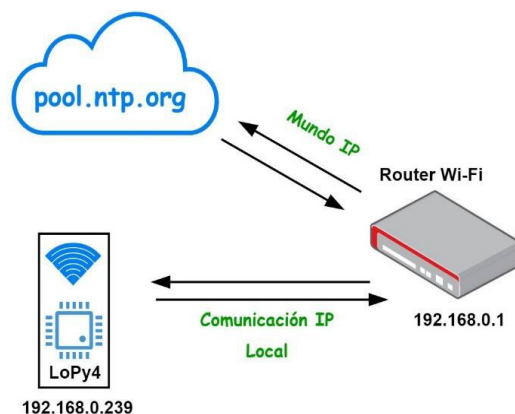
- Poner en práctica los conocimientos de redes inalámbricas WIFI
- Manejar comunicaciones IP a bajo nivel mediante sockets y entre diversas plataformas: PC, Móvil, CLOUD, Sensor
- Manejar protocolos de alto nivel: HTTP, FTP, NTP, MQTT y estándares de interoperabilidad: SENML

5.2 Tareas a realizar:

1. Conéctate a la red wifi del laboratorio o a una creada por el móvil como punto de acceso, extrae tu IP y comprueba la conectividad con Google mediante un ping.

2. Pon en hora el módulo mediante un servidor NTP (Network Time Protocol).

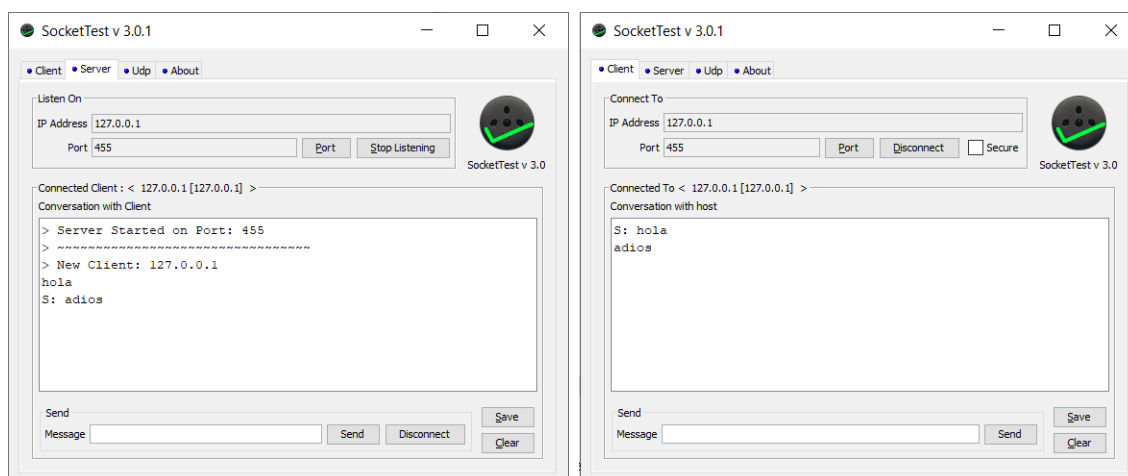
Material de apoyo: <https://lastminuteengineers.com/esp32-ntp-server-date-time-tutorial/>



3. Monta un chat entre dos sockets con una aplicación software PC (SocketTest) o con una aplicación móvil (Simple TCP Socket Tester). A veces el firewall del ordenador no permite las conexiones externas, y es necesario configurarlo correctamente (o desactivarlo).

Material de apoyo: <https://sockettest.sourceforge.net/>

https://play.google.com/store/apps/details?id=com.simplesockettester&hl=en_US



4. Sustituye uno de los extremos por el módulo hardware siendo cliente y envía cada segundo tu hora local.
5. Añade una capa de control de tal modo que cuando se le mande “start” empiece a mandar la hora hasta que se le mande “stop”.

ENSEÑAR 1

6. A partir del código de la práctica 3:
 - a. Adapta el código al ESP32 para que se conecte a la wifi, a un servidor TCP y mande los datos del acelerómetro.
 - b. Modificar el código en Python para recibir los datos a través de un servidor TCP en lugar de usar el puerto serie.

ENSEÑAR 2

Material de apoyo: <https://thatgirlcoder.com/2021/09/01/how-to-create-a-simple-python-tcp-ip-server-and-client/>

7. Monta un servidor WEB que muestre la hora y tenga un botón para resetear la hora a las 0:00.

ENSEÑAR 3

Material de apoyo: <https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/>

8. Basándote en el estándar SenML crea un fichero Json que se genere cada 10 segundos, que contenga datos de temperatura inventados con sus unidades y la marca temporal.

Material de apoyo: <https://datatracker.ietf.org/doc/html/rfc8428>
<https://arduinojson.org/>

9. Súbelos a un servidor FTP con un nombre que sea grupoXX_ddmmss.json.
Usa el servidor FTP del laboratorio (IP: 155.210.150.77, user: rsense, pass: rsense) o móntate uno con la aplicación FileZilla (asegúrate que el firewall permite conexiones entrantes).

ENSEÑAR 4

Material de apoyo:

https://registry.platformio.org/libraries/ldab/esp32_ftpclient
<https://filezilla-project.org/>

10. Sube datos a la nube, en concreto al servicio gratuito proporcionado por Adafruit.

- a. En primer lugar, se debe de crear un usuario en Adafruit, generar una aplicación para obtener un "AIO Key" y crear un feed al que subir datos.

Material de apoyo: <https://learn.adafruit.com/adafruit-io-basics-feeds>
<https://io.adafruit.com/>

- b. Sube datos desde navegador (<https://www.apirequest.io>) utilizando POST HTTP según documentación.

Material de apoyo: <https://io.adafruit.com/api/docs/?shell#create-data>

- c. Usa librería de Adafruit IO para subir datos al feed usando MQTT.

Material de apoyo: https://github.com/adafruit/Adafruit_IO_Arduino

- d. Usa la librería para suscribirte al feed y comprueba que recibes actualización al escribir desde el navegador.

ENSEÑAR 5

Material de apoyo: <https://github.com/plapointe6/EspMQTTClient>

5.3 Hitos evaluables de la práctica:

1. Enseñar el control del sistema mediante la comunicación serie y explicar la comunicación por wifi.
2. Enseñar la representación de las tres componentes de los datos de aceleración recibidos en una sola gráfica.
3. Enseñar y explicar cómo se genera la página web y su funcionamiento.
4. Enseñar el funcionamiento y mostrar el fichero Json generado y el recibido en el servidor.
5. Enseñar la comunicación bidireccional desde el navegador hacia el ESP32 y viceversa.