

Audio Server + Audio Client

Se describe un sistema Cliente-Servidor para enviar y recibir audio entre 2 sistemas ESP32.

Servidor

El servidor está formado por una módulo ESP32 Doit, un modulo codificador de audio a I2S PCM8108 y un módulo decodificador de I2S a audio PCM5102.

El software está desarrollado en el entorno Arduino compatible UECIDE y puede encontrarse en <https://github.com/ramonlh/Audio-Client-Server>.

El cliente se conecta a una red WiFi que está definida en el código (hay que cambia ésto) en el fichero audioserverUDP.ino:

```
#define WIFI_SSID "myssid"  
#define WIFI_PASSWORD "mypassword"
```

El indicativo de llamada se guarda también en este fichero:

```
#define CALLSIGN "EA4GZI"
```

La dirección IP es fija, 192.168.1.148

```
IPAddress EEip={192,168,1,148}; // 4 bytes, dirección IP
```

Ésto es necesario porque se precisa abrir el puerto **3334** y redirigirlo a la IP 192.168.1.148 en el router para que sea posible acceder al servidor desde internet.

Cuando el servidor arranca y se conecta a la red extrae la dirección IP externa accediendo al servidor “**icanhazip.com**”.

A continuación envía mediante MQTT al servidor “**broker.mqtt-dashboard.com**” los siguientes *Topics*

ubitx/audioserver/EA4GZI/ip	IP externa
ubitx/audioserver/EA4GZI/port	port (3334)
ubitx/audioserver/EA4GZI/rate	rate value (16000)
ubitx/audioserver/EA4GZI/bits	bits value (24)

Cliente

El cliente consiste en un módulo LyraT, basado en un ESP32 y que incluye un codec ES8388, lector de tarjetas SD y varios pulsadores programables.

El software está desarrollado en el entorno ESP-ADF y puede encontrarse en <https://github.com/ramonlh/Audio-Client-Server>.

Los parámetros para la conexión están definidos en un fichero de la tarjeta SD, llamado “config.txt” y que puede editarse con un editor de texto en un PC.

Línea 1	Indicativo (EA4GZI)
Línea 2	SSID

Línea 3	Password
Línea 4	Modo (STEREO/MONO)
Línea 5	Rate (16000)
Línea 6	Fuente de sonido (MIC/AUX)

Ejemplo

EA4GZI
myssid
mypassword
STEREO
16000
MIC

Al iniciarse y después de pulsar el pulsador “***Play***”, el ESP32 lee el fichero, se subscribe al broker MQTT “**broker.mqtt-dashboard.com**” a los Topics, lee los datos recibidos y se conecta a la dirección y puerto indicados mediante el *rate* y *bits* indicados.

ubitx/audioserver/EA4GZI/ip	IP externa
ubitx/audioserver/EA4GZI/port	port (3334)
ubitx/audioserver/EA4GZI/rate	rate value (16000)
ubitx/audioserver/EA4GZI/bits	bits value (24)

Cuando el servidor detecta la nueva conexión entrante empieza a enviar y recibir datos I2S con el audio. El cliente, a su vez, también empieza a enviar y recibir los datos I2S que el codec ES3888 se encarga de decodificar.