

PRACTICA 7_2 : Reproducir un archivo WAVE en ESP32 desde una tarjeta SD externa

1.Salida del terminal

P7_2 > src > main.cpp > SD_CS

```
6  #define SD_CS 5
7  #define SPI_MOSI 23
8  #define SPI_MISO 19
9  #define SPI_SCK 18
10 #define I2S_DOUT 25
11 #define I2S_BCLK 27
12 #define I2S_LRC 26
13
14 Audio audio;
15
16 void setup(){
17
18     pinMode(SD_CS, OUTPUT);
19     digitalWrite(SD_CS, HIGH);
20     SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI);
21     Serial.begin(115200);
22     SD.begin(SD_CS);
23     audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
24     audio.setVolume(10); // 0...21
25     audio.connecttoFS(SD, "15163726_Hypnocurrency_(Original_Mix).wav");
26     Serial.println("Reproduciendo...");
27 }
28
29 void loop(){
30
31     audio.loop();
32
33 }
34
35 // optional
36 void audio_info(const char *info){
```

PROBLEMS 42 OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE: MESSAGES

```
--- More details at http://bit.ly/pio-monitor-filters
--- Miniterm on COM5 115200,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
016 00:22:57info PSRAM not found, inputBufferSize: 6399 bytes
info buffers freed, free Heap: 286908 bytes
info Reading file: "/15163726_Hypnocurrency_(Original_Mix).wav"
Reproduciendo...
info stream ready
info FormatCode: 1
info DataRate: 176400
info DataBlockSize: 4
info Audio-Length: 51601364
info syncword found at pos 0
info Channels: 2
info SampleRate: 44100
info BitsPerSample: 16
info BitRate: 1411200
```

2.Explicación del código

En este apartado de la práctica 7, se trabaja con especificaciones similares a la anterior, con la diferencia de que ahora se utilizará la comunicación SPI para importar un archivo de audio con formato ".wav" almacenado en una tarjeta SD. Para decodificar los datos del audio se utilizará de nuevo la placa MAX98357 con protocolo I2S. Así pues, se trabajará la comunicación SPI y I2S.

El código comienza con las definiciones de las librerías necesarias en este caso y, de 7 pines necesarios para realizar las comunicaciones SPI y I2S, además de un objeto de la clase Audio, donde posteriormente se explicará para qué se utiliza.

En el setup, inicializamos las líneas de código necesarias para establecer la comunicación SPI mediante el lector de tarjeta SD. Más tarde, se inicializa el Serial y, luego, utilizamos 3 funciones definidas en la clase Audio que habíamos definido antes de programar el setup. La primera función corresponde a "audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT)", donde se especifican los pines de salida I2S. La segunda, "audio.setVolume(10)", se encarga de delimitar el volumen del audio a la hora de reproducirlo y, la tercera, "audio.connecttoFS(SD, "15163726_Hypnocurrency_(Original_Mix).wav")", cuya función es importar el archivo contenido en la tarjeta SD.

En el loop, simplemente se inicializa el loop del objeto audio, que contiene el archivo de música que hayamos almacenado en la SD.

Finalmente, mostramos por el terminal, información técnica del archivo .wav reproducido como: FormatCode, DataRate, DataBlockSize, Audio-Length, Channels, SampleRate, BitsPerSample, BitRate.