

PRACTICA 7 : Buses de comunicación III (I2S)

B: Reproducir un archivo WAVE en ESP32 desde una tarjeta SD externa

Código

```
#include "Audio.h"
#include "SD.h"
#include "FS.h"

// Digital I/O used
#define SD_CS      5
#define SPI_MOSI   23
#define SPI_MISO   19
#define SPI_SCK    18
#define I2S_DOUT   25
#define I2S_BCLK   27
#define I2S_LRC    26

Audio audio;

void setup(){

    Serial.begin(115200);

    pinMode(SD_CS, OUTPUT);
    digitalWrite(SD_CS, HIGH);
    SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI);
    SD.begin(SD_CS);
    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(15); // 0...21
    audio.connecttoFS(SD, "Ensoniq-ZR-76-01-Dope-77.wav");
}

void loop(){
    audio.loop();
}

// optional
void audio_info(const char *info){
    Serial.print("info      "); Serial.println(info);
}
void audio_id3data(const char *info){ //id3 metadata
    Serial.print("id3data   "); Serial.println(info);
}
void audio_eof_mp3(const char *info){ //end of file
    Serial.print("eof_mp3    "); Serial.println(info);
}
void audio_showstation(const char *info){
    Serial.print("station    "); Serial.println(info);
}
```

```

}
void audio_showstreaminfo(const char *info){
    Serial.print("streaminfo ");Serial.println(info);
}
void audio_showstreamtitle(const char *info){
    Serial.print("streamtitle ");Serial.println(info);
}
void audio_bitrate(const char *info){
    Serial.print("bitrate ");Serial.println(info);
}
void audio_commercial(const char *info){ //duration in sec
    Serial.print("commercial ");Serial.println(info);
}
void audio_icyurl(const char *info){ //homepage
    Serial.print("icyurl ");Serial.println(info);
}
void audio_lasthost(const char *info){ //stream URL played
    Serial.print("lasthost ");Serial.println(info);
}
void audio_eof_speech(const char *info){
    Serial.print("eof_speech ");Serial.println(info);
}
}

```

Funcionamiento del programa

En primer lugar, haremos uso de tres librerías en las que destacamos la librería Audio.h y SD.h. También tendremos que definir los diferentes pines que usaremos para la recepción y transmisión de datos. Los pines SPI_MOSI, SPI_MISO, SPI_SCK y SPI_CS són los de la tarjeta SD.

```

#include "Audio.h"
#include "SD.h"
#include "FS.h"

// Digital I/O used
#define SD_CS      5
#define SPI_MOSI   23
#define SPI_MISO   19
#define SPI_SCK    18
#define I2S_DOUT   25
#define I2S_BCLK   27
#define I2S_LRC    26

```

A continuación, vamos a declarar una variable del tipo Audio donde vamos a guardar el sonido y sus características. Posteriormente definimos la función Setup donde iniciamos el puerto serie y realizamos la conexión entre el lector de tarjetas y el bus SPI. Para finalizar, configuramos la variable Audio en la que debemos asignarle los pines del bus I2C, ajustar el volumen pertinente y leer el archivo desde el file source.

```
Audio audio;

void setup(){

    Serial.begin(115200);

    pinMode(SD_CS, OUTPUT);
    digitalWrite(SD_CS, HIGH);
    SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI);
    SD.begin(SD_CS);
    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(15); // 0...21
    audio.connecttoFS(SD, "Ensoniq-ZR-76-01-Dope-77.wav");
}
```

Finalmente usamos la función Loop para reproducir el audio.

```
void loop(){
    audio.loop();
}
```

Salida por Terminal

No he podido realizar el montaje ya que no tengo el material suficiente.