play an audio file from SPIFFS using MAX98357A

the following is a guide on how to play an audio file using the MAX98357A amplifier.
for this tutorial, we used the Arduino IDE.

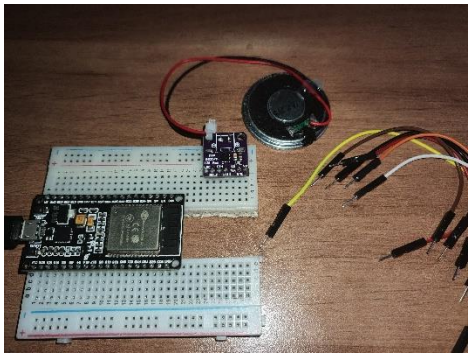
assumptions:

- you have downloaded the Arduino IDE.
- you have configured the IDE to work with the “DOIT ESP32 DIVKIT V1” board.
- you know how to upload a file to esp32’s SPIFFS

a guide to all the steps above can be found in the “bank of knowledge”.

needed material:

- ESP32 microcontroller
- MAX98357A amplifier
- breadboard
- WiFi connection
- 6 wires
- earphones with an aux jack

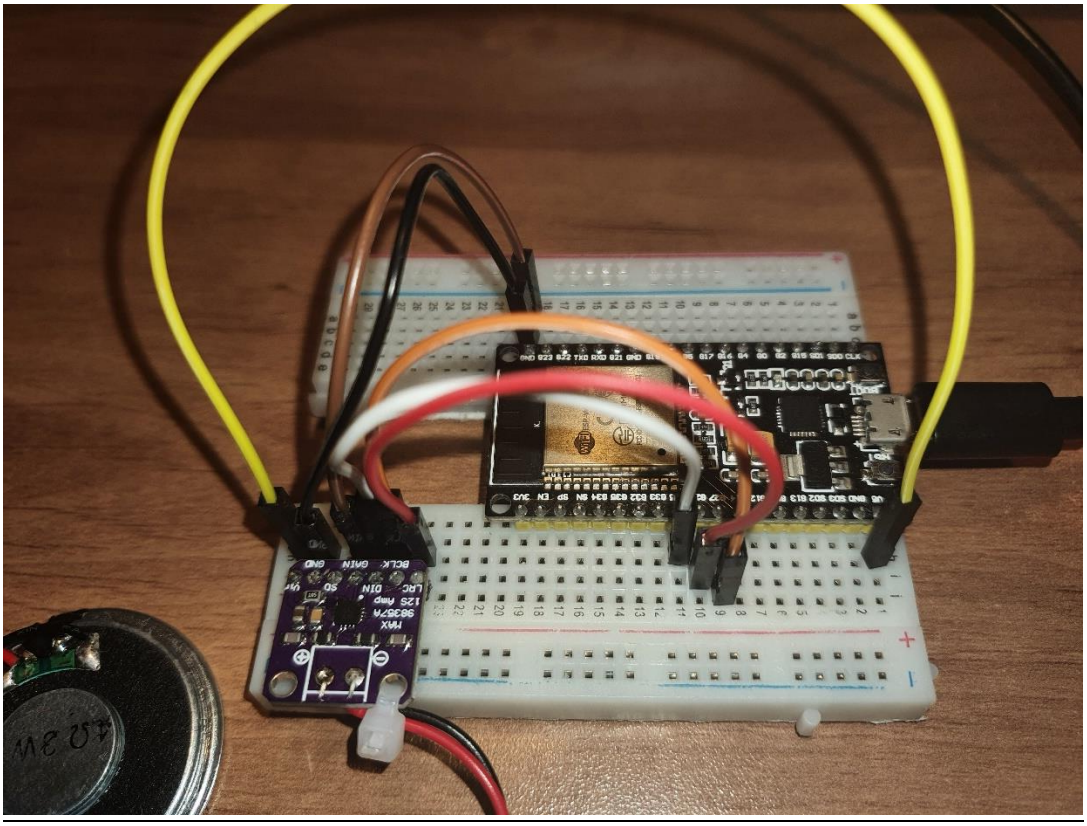


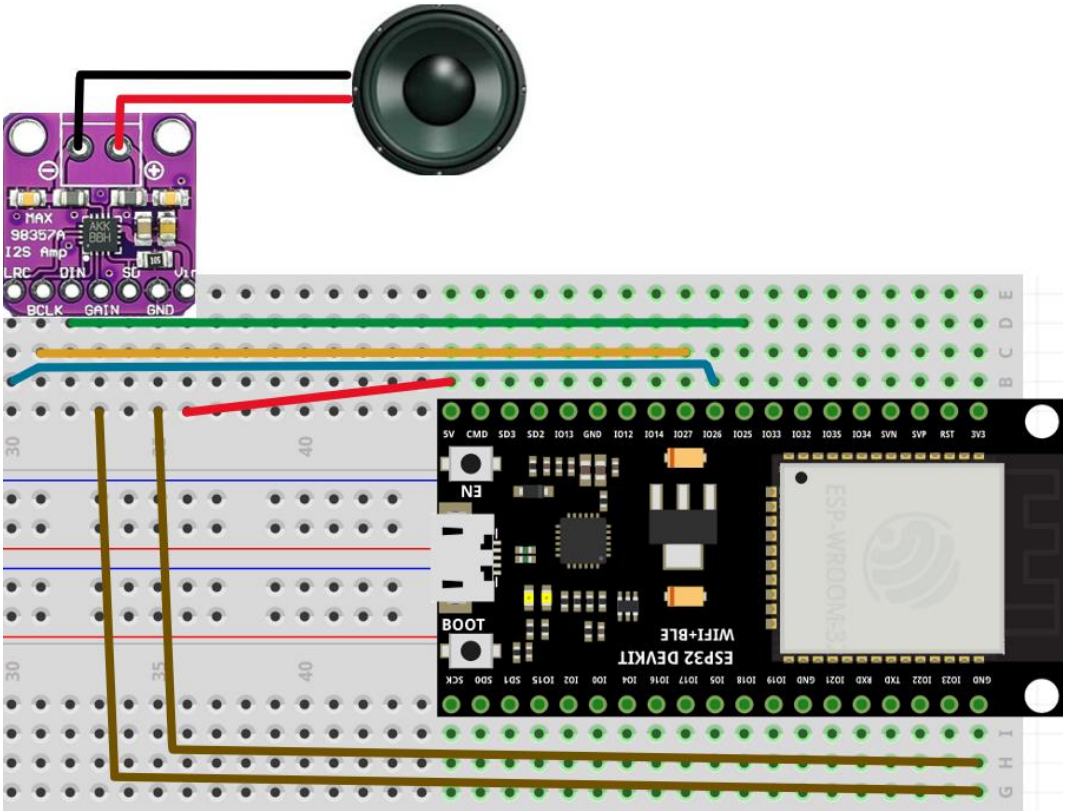
step 1: download the following Github repository as a zip file to your computer:
https://github.com/schreibfaul1/ESP32-audioI2S

step 2: go to the Arduino IDE -> Sketch -> include library -> add .ZIP library -> choose the file you’ve downloaded in step 1.

step 3: setup the wiring as shown in the “audio input and output for microcontrollers” presentation, or as follows:

MAX98357A	ESP32
Vcc	Vcc (preferably 3.3V but can be 5V)
GND	GND
BCK or BCLK	Pin 27 (G27)
DIN	Pin 25 (G25)
LRC	Pin 26 (G26)
GAIN	GND
SD	-




MAX
98357A
I2S Amp
LRC
DIN
SD
Vin
BCLK
GAIN
GND
EN
BOOT
ESP32 DEVKIT
WIFI+BLE

step 4: copy and paste the following code:

```
//#include "Arduino.h" //required for PlatformIO
#include "Audio.h"
#include "FS.h"

// Digital I/O used for I2S DAC amplifier
#define I2S_DOUT      25
#define I2S_BCLK      27
#define I2S_LRC       26

Audio audio;

void setup() {

    Serial.begin(115200);

    if (!SPIFFS.begin(true)) {
        Serial.println("An Error has occurred while mounting SPIFFS");
        return;
    }

    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(20); // range 0...21 - This is not amplifier gain, but controlling
the level of output amplitude.

    audio.connecttoFS(SPIFFS, "123_u8.wav"); // a file with the proper name must be
placed in /data folder and uploaded using "ESP32 sketch upload" plugin

    audio.setFileLoop(true); //this causes the file to play in an endless loop
}

void loop()
{
    audio.loop();
    if (Serial.available()) { // put streamURL in serial monitor
        audio.stopSong();
        Serial.println("audio stopped");
        log_i("free heap=%i", ESP.getFreeHeap());

    }
}
```

NOTE: make sure that the defined Pins match the wiring, and that the file name matches the one you uploaded to SPIFFS.

step 5: connect the ESP32 to your computer, compile and run the code. you might need to press on the “reset” button on your ESP32.

NOTE:

- the following code allows the file to play in a loop. you can change that if you want. while on loop, sometimes the audio plays for 1 or 2 times only for some unknown reason. changing the port which the esp32 is connected to seems to help!

step 6: enjoy 😊