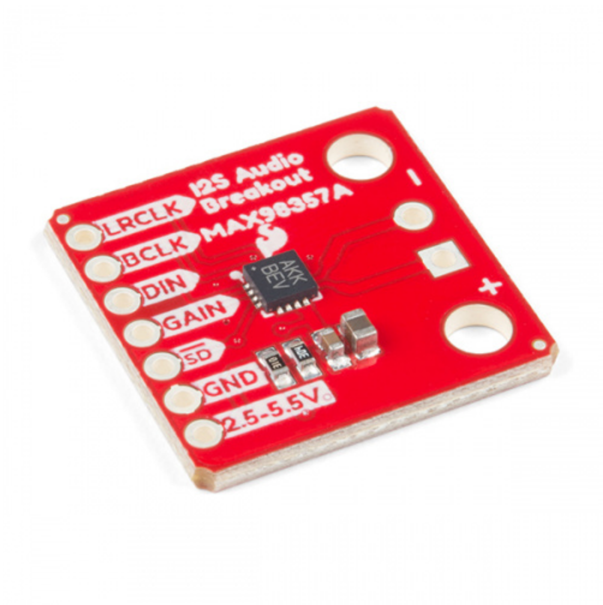
# I2S Audio Breakout Hookup Guide

## Introduction

The I2S Audio Breakout board uses the MAX98357A digital to analog converter (DAC), which converts I2S (not be confused with I2C) audio to an analog signal to drive speakers. The MAX98357A has a built in class D amplifier which can deliver up to 3.2W of power into a 4Ω load. For more information, see the Hardware Overview section below.



SparkFun I2S Audio Breakout - MAX98357A

DEV-14809

## Suggested Tools

You will need a soldering iron, solder, general soldering accessories, screw driver, and hobby knife.

Weller WE1010 Soldering Station

◉ TOL-14734

Solder Lead Free - 100-gram Spool

◉ TOL-09325

Pocket Screwdriver Set

◉ TOL-12891

Hobby Knife

◉ TOL-09200

## Suggested Reading

If you aren't familiar with the following concepts, we recommend checking out these tutorials before continuing.

How to Solder: Through-Hole Soldering

How to Power a Project

This tutorial covers everything you need to know about through-hole soldering.

A tutorial to help figure out the power requirements of your project.

### Switch Basics

A tutorial on electronics' most overlooked and underappreciated component: the switch! Here we explain the difference between momentary and maintained switches and what all those acronyms (NO, NC, SPDT, SPST, ...) stand for.

### ESP32 Thing Hookup Guide

An introduction to the ESP32 Thing's hardware features, and a primer on using the WiFi/Bluetooth system-on-chip in Arduino.

### How to Work w/ Jumper Pads and PCB Traces

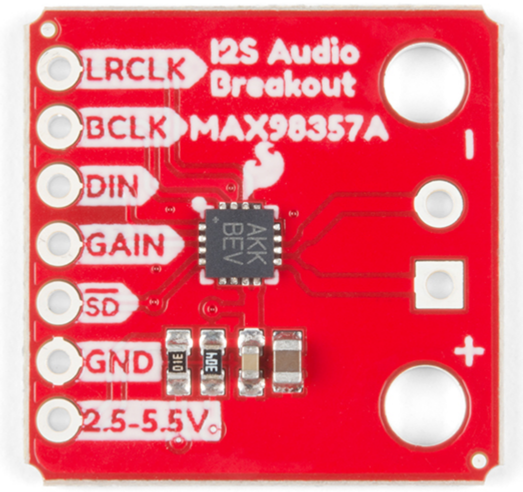
Handling PCB jumper pads and traces is an essential skill. In this tutorial, you will learn how to cut a PCB trace and add a solder jumper between pads to reroute connections. You will also learn how to repair a trace with the green wire method if a trace is damaged.

### ESP32 Thing Motion Shield Hookup Guide

Getting started with the ESP32 Thing Motion Shield to detect movements using the on-board LSM9DS1 IMU and adding a GPS receiver. Data can be easily logged by adding an microSD card to the slot.

## Hardware Overview

The I2S audio breakout converts the digital audio signals using the I2S standard to an analog signal and amplifies the signal using a class D amplifier. The board can be configured to output only the left channel, right channel, or both. For more information about how to configure the board, refer to the Jumper Selection section below.

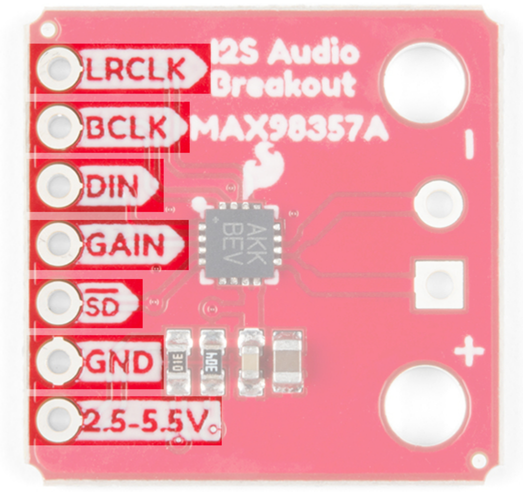


## Board Specs

| Parameter | Description |
|---|---|
| Supply Voltage Range | **2.5V - 5.5V.** |
| Output Power | 3.2W into 4Ω at 5V. |
| Output Channel Selection | Left, Right, or Left/2 + Right/2 (Default). |
| Sample Rate | 8kHz - 96kHz. |
| Sample Resolution | 16/32 bit. |
| Quiescent Current | 2.4mA. |
| Additional Features | Filterless Class D outputs, no MCLK required, click and pop reduction, short-circuit and thermal protection. |

## Pin Descriptions

The SparkFun I2S audio breakout board is fairly simple, requiring only a few pin connections to get the board working.

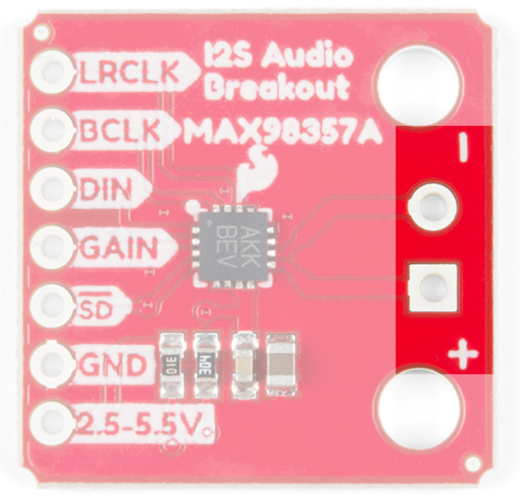


## Inputs

| Pin Label | Description |
|---|---|
| LRCLK | Frame clock (left/right clock) input. |
| BCLK | Bit clock input. |
| DIN | Serial data input. |
| GAIN | Gain setting. Can be set to +3/6/9/12/15dB. Set to +9dB by default. |
| SD | Shutdown and channel select. Pull low to shutdown, or use the jumpers to select the channel output (see jumper selection for more information). |
| GND | Connect to ground |
| VDD | Power input. Must be between **2.5** and **5.5VDC**. |

## Outputs

The output is where you'll connect your speaker.



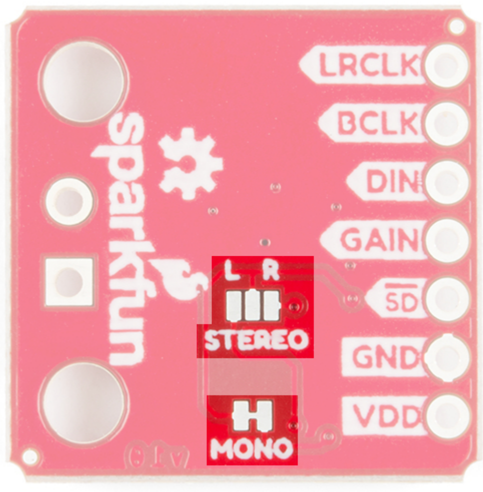
| Pin Label | Description |
|---|---|
| + | Positive speaker output. |
| - | Negative speaker output. |

Speaker wire can either be soldered directly to the output pads, but if screw terminals are more your style, you can use our 3.5mm screw terminals.

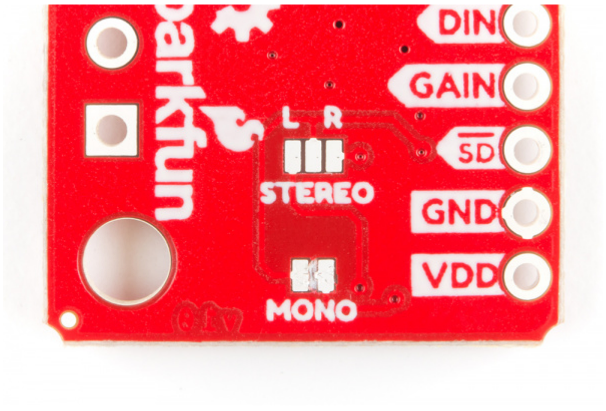
Screw Terminals 3.5mm Pitch (2-Pin)
● PRT-08084

## Jumper Selection

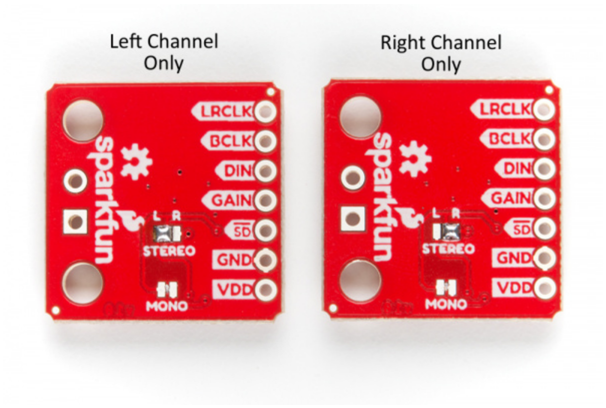
By default the board is configured in “mono” operation, meaning the left and right signals are combined together to drive a single speaker.



If you want a separate speaker for the left and right audio channels you'll first need to cut the mono jumper as pictured below.



To configure the board to respond to a specific audio channel, you'll need to close the stereo jumper as shown below.



## Gain Selection

In addition to being able to select the audio channel output, the gain can also be configured in a few ways. The gain of the amplifier can be configured from as **low as +3dB to as high as +15dB**. While the channel selection can be configured on board, the gain however is controlled externally using the gain pin. By default, the board is configured for **+9dB**, but can be changed using the table below.

| Gain (dB) | Gain Pin Connection |
|---|---|
| 15 | Connected to GND through a 100kΩ resistor. |
| 12 | Connected to GND. |
| 9 | Unconnected (**Default**). |
| 6 | Connected to VDD. |
| 3 | Connected to VDD through a 100kΩ resistor. |

## Examples

**Note:** These example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you have not previously installed an Arduino library, please check out our installation guide.

This board should work with any microcontroller or single board computer that has I2S capable pins. In these examples, we're going to look at a pretty powerful library that allows you to use an ESP32 Thing to play audio from a wide variety of sources. First, we'll play an audio file which is stored in the ESP32's program memory, and once we have that working we'll look at creating a MP3 trigger. The following libraries are needed to run the examples that were originally written for the ESP8266, but also work with the ESP32.

### ESP8266Audio Arduino Library

You'll need to install the ESP8266 Audio Arduino Library, written by Earle F. Philhower, which you can get from the link below. This library will allow you to play a wide variety of audio formats including: AAC, FLAC, MIDI, MOD, MP3, RTTTL, and WAV. To use the library, you can add the library from Arduino by selecting **Sketch** > **Include Library** > **Add .ZIP Library**… and selecting the **.zip** file from wherever you store your file downloads.

**ESP8266 AUDIO LIBRARY (ZIP)**

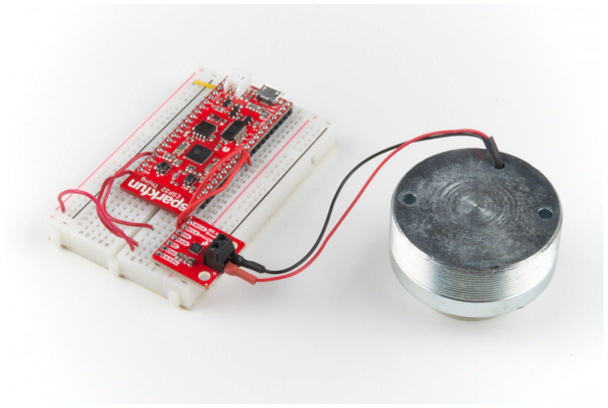
### ESP8266_Spiram Arduino Library

The ESP8266 Audio library depends on the ESP8266 Spiram library, written by Giancarlo Bacchio, which will also need to be downloaded. You can download the library from the link below. Installing the library follows the same process as outlined above.

**ESP8266 SPIRAM LIBRARY (ZIP)**

### First Test

In this first example, we'll run a quick example sketch to make sure the I2S audio breakout board is wired correctly and is working.



## Required Materials

The parts used in this example are listed in the wishlist below. You may not need everything though depending on what you have. Add it to your cart, read through the example, and adjust the cart as necessary.

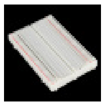
**I2S Audio Breakout Hookup Guide Example 1** SparkFun Wish List

Screw Terminals 3.5mm Pitch (2-Pin)
PRT-08084
Screw Terminals with 3.5mm pitch pins. Comes in 2 or 3 positions and have the really cool feature of slide-locking t…

(2) Break Away Headers - Straight
PRT-00116
A row of headers - break to fit. 40 pins that can be cut to any size. Used with custom PCBs or general custom head…

SparkFun ESP32 Thing
DEV-13907
The SparkFun ESP32 Thing is a comprehensive development platform for Espressif's ESP32, their super-charged …

Jumper Wire Kit - 140pcs
PRT-00124
This is a time saving kit of jumper wires - cut, stripped, and pre-bent for your prototyping pleasure. Included with thi…



Breadboard - Self-Adhesive (White)
PRT-12002
This is your tried and true white solderless breadboard. It has 2 power buses, 10 columns, and 30 rows - a total of …

Surface Transducer - Large
COM-10975
Surface transducers give you the awesome power to turn almost any surface into a speaker. They're essentially jus…

## Hookup Table

The connections that need to be made to the ESP32 are list below.

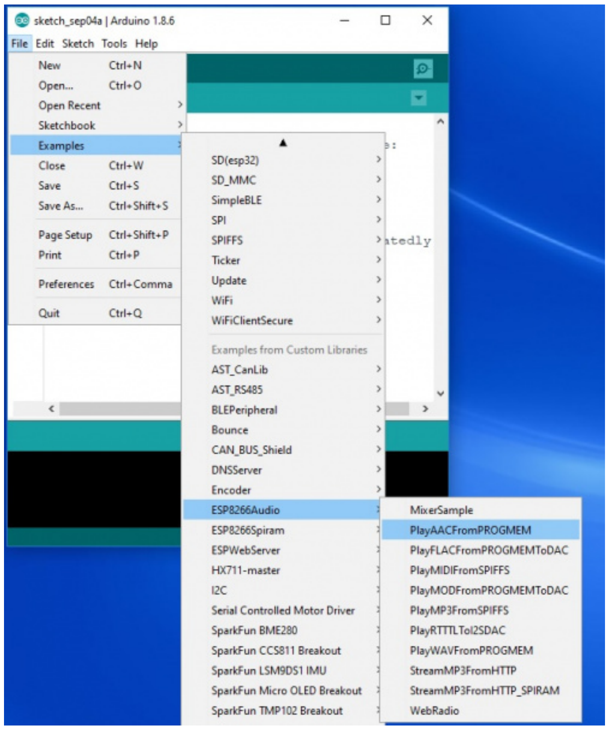
| ESP32 Pin | I2S Audio Breakout Pin |
|---|---|
| VUSB/3V3 | VDD |
| GND | GND |
| GPIO 22 | DIN |

| | |
|---|---|
| GPIO 26 | BCLK |
| GPIO 25 | LRCLK |

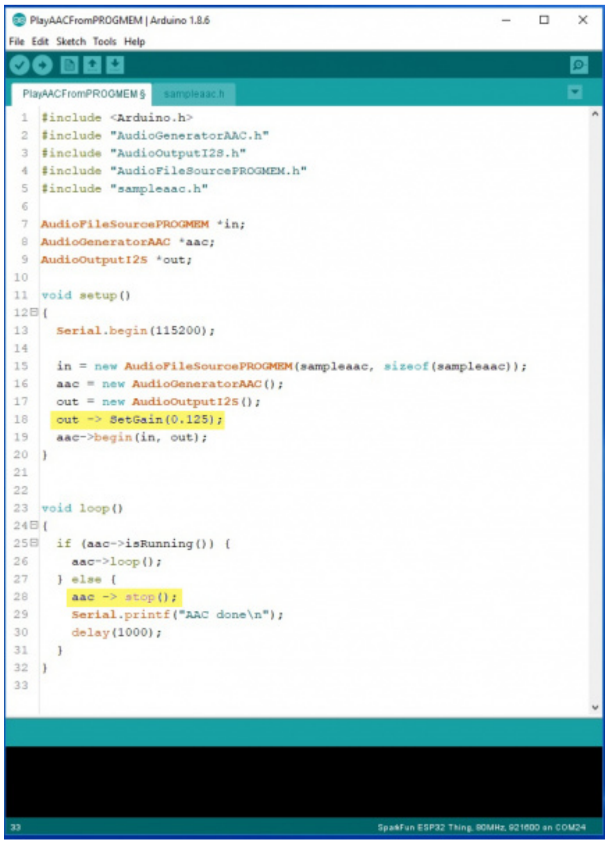
Make sure to also connect a speaker to the I2S audio breakout board's output pins.

## Example Code

We're going to use one of the examples that comes with the library named **"PlayAACFromPROGMEM"**. With the library installed, open the example located in: **File** > **Examples** > **ESP8266Audio** > **PlayAACFromPROGMEM** .



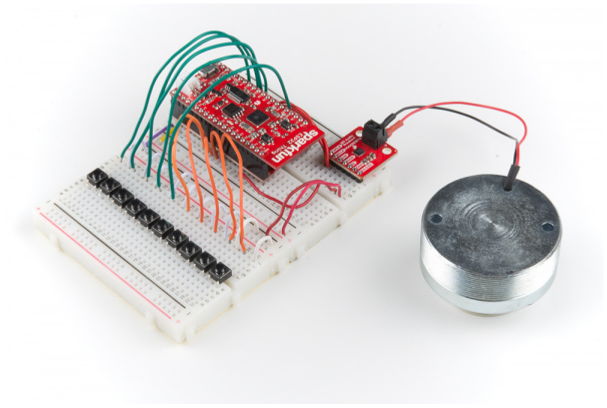
Before we upload the code, we're going to add two lines of code (as highlighted in the image below). The first line is going to adjust the volume, which we add after we initialize the I2S output ( `out = new AudioOutputI2S();` ). After the output is initialized, we're going to add `out -> SetGain(0.125);` . As the name suggests this sets the gain of the output, which takes a floating point number and has a maximum value of 4.0. The second line will reduce hum at the end of the audio clip by adding `aac -> stop();` in the else statement in the main `loop()` . After you upload the sketch to your ESP32, you should hear Homer Simpson's thoughts of perpetual motion machines if everything is working.


PlayAACFromPROGMEM | Arduino 1.8.6
File Edit Sketch Tools Help
PlayAACFromPROGMEM §
sampleaac.h
33
SparkFun ESP32 Thing, 80MHz, 921600 on COM24

```
#include <Arduino.h>
#include "AudioGeneratorAAC.h"
#include "AudioOutputI2S.h"
#include "AudioFileSourcePROGMEM.h"
#include "sampleaac.h"

AudioFileSourcePROGMEM *in;
AudioGeneratorAAC *aac;
AudioOutputI2S *out;

void setup()
{
  Serial.begin(115200);

  in = new AudioFileSourcePROGMEM(sampleaac, sizeof(sampleaac));
  aac = new AudioGeneratorAAC();
  out = new AudioOutputI2S();
  out -> SetGain(0.125);
  aac->begin(in, out);
}


void loop()
{
  if (aac->isRunning()) {
    aac->loop();
  } else {
    aac -> stop();
    Serial.printf("AAC done\n");
    delay(1000);
  }
}
```

## ESP32 MP3 Trigger

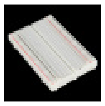
Now that we know the board is working, let's take it up a notch. In this next example, we'll create a MP3 trigger that works similar to our MP3 Trigger.



## Required Materials

For this example, we'll need to add a few more parts to the ones we used in the previous example (including a second breadboard). You may not need everything though depending on what you have. Add it to your cart, read through the example, and adjust the cart as necessary.

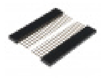
**I2S Audio Breakout Hookup Guide Example 2** SparkFun Wish List



Breadboard - Self-Adhesive (White)
PRT-12002
This is your tried and true white solderless breadboard. It has 2 power buses, 10 columns, and 30 rows - a total of …

(10) Mini Pushbutton Switch
COM-00097

We use these little buttons on everything! These Miniature Single Pole Single Throw switches have a good click to …


ESP32 Thing Stackable Header Set
PRT-14311
These headers are made to work with the SparkFun ESP32 Thing and ESP32 Power Control Shield boards. Each …

SparkFun ESP32 Thing Motion Shield
DEV-14430

microSD USB Reader
COM-13004
This is an awesome little microSD USB reader. Just slide your microSD card into the inside of the USB connector, t…

microSD Card with Adapter - 16GB (Class 10)
COM-13833
This is a class 10 16GB microSD memory card, perfect for housing operating systems for single board computers a…

Before we add code, we'll need some audio files to play. Any MP3 audio file should work, you'll just need to copy them over to your microSD card using a microSD USB Reader. Before ejecting the microSD card from your computer, make sure to relabel the files **TRACKn.mp3**, where n is a integer number between 0-9. The I2S audio breakout board has the same pin connections as the previous example, but this time we're going to change the audio source from PROGMEM to our microSD card. The last step before adding the code below, is to add headers to the ESP32 Thing, as well as the Motion Shield, as outlined in the hookup guide.

```
/* SparkFun I2S Audio Breakout Demo
 * Created by: Alex Wende
 * 8/3/2018
 *
 * Uses a ESP32 Thing to create a MP3 trigger using
 * the I2S Audio Breakout board.
 *
 * Parts you'll need:
 * - I2S Audio Breakout board (https://www.sparkfun.com/products/14809)
 * - ESP32 Thing (https://www.sparkfun.com/products/13907)
 * - Micro SD Breakout (https://www.sparkfun.com/products/544)
 * - A microSD card (https://www.sparkfun.com/products/13833)
 * - Speaker (4-8ohms)
 *
 * The following libraries need to be installed before
 * uploading this sketch:
 * - ESP8266 Audio (https://github.com/earlephilhower/ESP8266Audio)
 * - SRam Library (https://github.com/Gianbacchio/ESP8266_Spiram)
 */

#include <Arduino.h>
#include "AudioGeneratorMP3.h"
#include "AudioOutputI2S.h"
#include "AudioFileSourceSD.h"
#include "driver/i2s.h"
#include <SD.h>

//define trigger pins
#define TRIGGER0 13
#define TRIGGER1 12
#define TRIGGER2 14
#define TRIGGER3 27
#define TRIGGER4 32
#define TRIGGER5 5
#define TRIGGER6 15
#define TRIGGER7 2
#define TRIGGER8 0
#define TRIGGER9 4

//Initialize ESP8266 Audio Library classes
AudioGeneratorMP3 *mp3;
AudioFileSourceSD *file;
AudioOutputI2S *out;

volatile bool playing = 0;
volatile byte loadTrack = 0;

//External Interrupt function with software switch debounce
void IRAM_ATTR handleInterrupt()
{
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > 200)
  {
    //Figure out which switch was triggered, and which track to play
    if(!digitalRead(TRIGGER0)) loadTrack = 1;
    else if(!digitalRead(TRIGGER1)) loadTrack = 2;
    else if(!digitalRead(TRIGGER2)) loadTrack = 3;
    else if(!digitalRead(TRIGGER3)) loadTrack = 4;
    else if(!digitalRead(TRIGGER4)) loadTrack = 5;
    else if(!digitalRead(TRIGGER5)) loadTrack = 6;
    else if(!digitalRead(TRIGGER6)) loadTrack = 7;
    else if(!digitalRead(TRIGGER7)) loadTrack = 8;
    else if(!digitalRead(TRIGGER8)) loadTrack = 9;
    else if(!digitalRead(TRIGGER9)) loadTrack = 10;
    playing = 1;
  }
  last_interrupt_time = interrupt_time;
}

void setup()
{
  Serial.begin(115200);

  //Configure trigger pins to inputs with internal pull-up resistors enabled
  pinMode(TRIGGER0,INPUT_PULLUP);
  pinMode(TRIGGER1,INPUT_PULLUP);
  pinMode(TRIGGER2,INPUT_PULLUP);
  pinMode(TRIGGER3,INPUT_PULLUP);
  pinMode(TRIGGER4,INPUT_PULLUP);
  pinMode(TRIGGER5,INPUT_PULLUP);
  pinMode(TRIGGER6,INPUT_PULLUP);
  pinMode(TRIGGER7,INPUT_PULLUP);
  pinMode(TRIGGER8,INPUT_PULLUP);
  pinMode(TRIGGER9,INPUT_PULLUP);

  //Create interrupts for each trigger
  attachInterrupt(digitalPinToInterrupt(TRIGGER0),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER1),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER2),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER3),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER4),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER5),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER6),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER7),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER8),handleInterrupt,FALLING);
  attachInterrupt(digitalPinToInterrupt(TRIGGER9),handleInterrupt,FALLING);

  out = new AudioOutputI2S();
  mp3 = new AudioGeneratorMP3();

  delay(1000);
  Serial.print("Initializing SD card...");
  if (!SD.begin(33))
  {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  delay(100);
}

void loop()
{
  if(loadTrack) //Load the track we want to play
  {
    //Stop the current track if playing
    if(playing && mp3->isRunning()) mp3->stop();

    if(loadTrack == 1) file = new AudioFileSourceSD("/TRACK0.mp3");
    else if(loadTrack == 2) file = new AudioFileSourceSD("/TRACK1.mp3");
    else if(loadTrack == 3) file = new AudioFileSourceSD("/TRACK2.mp3");
    else if(loadTrack == 4) file = new AudioFileSourceSD("/TRACK3.mp3");
    else if(loadTrack == 5) file = new AudioFileSourceSD("/TRACK4.mp3");
    else if(loadTrack == 6) file = new AudioFileSourceSD("/TRACK5.mp3");
    else if(loadTrack == 7) file = new AudioFileSourceSD("/TRACK6.mp3");
    else if(loadTrack == 8) file = new AudioFileSourceSD("/TRACK7.mp3");
    else if(loadTrack == 9) file = new AudioFileSourceSD("/TRACK8.mp3");
    else if(loadTrack == 10) file = new AudioFileSourceSD("/TRACK9.mp3");

    out -> SetGain(0.08); //Set the volume
    mp3 -> begin(file,out); //Start playing the track loaded
    loadTrack = 0;
  }

  if(playing && mp3->isRunning()) {
    if (!mp3->loop())
    {
      mp3->stop();
      playing = 0;
      Serial.println("Stopped");
    }
  }
}
```

With the code on the board, we can see what the sketch does. You can connect momentary pushbutton switches to each of the trigger pins outlined in the table below, with the other end of the switch connected to ground. Another option is to take a ground wire and touch it to one of the trigger pins. When the pin is pulled down to ground, it triggers the corresponding track to play. If a track is still playing when a new pin is triggered, that track will stop and the new track will play.

| Audio File | ESP32 GPIO Pin |
|---|---|
| TRACK0.mp3 | 13 |
| TRACK1.mp3 | 12 |
| TRACK2.mp3 | 14 |

| | |
|---|---|
| TRACK3.mp3 | 27 |
| TRACK4.mp3 | 32 |
| TRACK5.mp3 | 5 |
| TRACK6.mp3 | 15 |
| TRACK7.mp3 | 2 |
| TRACK8.mp3 | 0 |
| TRACK9.mp3 | 4 |

## Resources and Going Further

Now that we know how to use the I2S Audio Breakout board, it’s time to use it in your own project! For more information on the I2S audio breakout, checkout the links below:

- Schematic (PDF)
- Eagle Files (ZIP)
- MAX98257A Datasheet (PDF)
- Wikipedia: I2S Standard
- GitHub
  - ESP8266 Audio Library Repo
    - ESP8266Audio (ZIP)
  - ESP8266_Spiram Arduino Library Repo
    - ESP8266_Spiram (ZIP)
  - Product Repo
- SFE Product Showcase

Need some inspiration for your next project? Check out some of these related tutorials:

### Build an Auduino Step Sequencer
Make a Step Sequencer using an Arduino, the Auduino firmware, and a handful of hardware.

### MiniGen Hookup Guide
Using the MiniGen, SparkFun's Arduino Pro Mini signal generator shield

WAV Trigger Hookup Guide V11

An introduction to being able to trigger music and sound effects based on buttons, sensors, or switches using the WAV Trigger board.

MIDI Shield Hookup Guide

How to assemble the SparkFun MIDI Shield, plus several example projects.