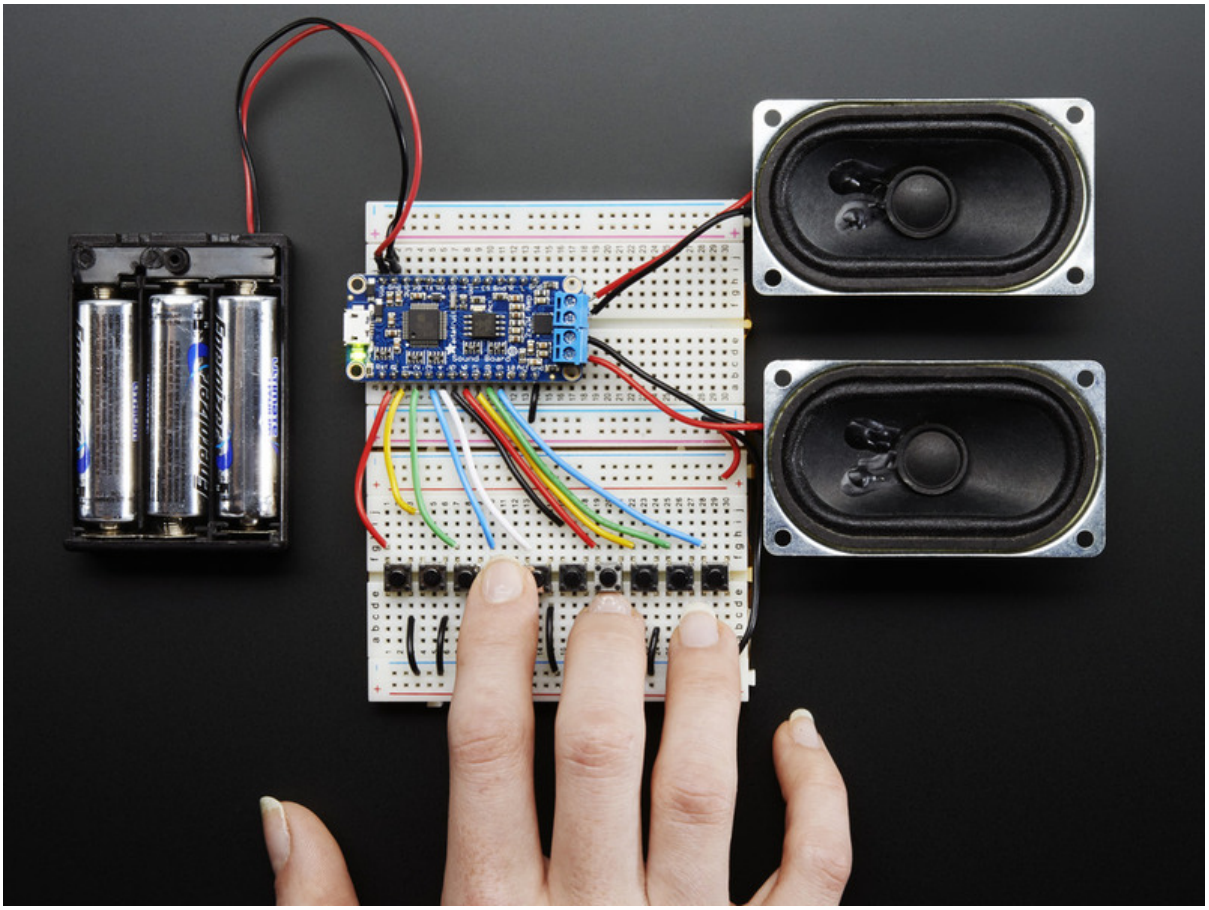
Adafruit Audio FX Sound Board

Created by lady ada



https://learn.adafruit.com/adafruit-audio-fx-sound-board

Last updated on 2024-03-08 02:00:56 PM EST

Table of Contents

Overview 5

- Amplifier or Headphone out
- Trigger Effects

Tour 9

- Headphone Output Type
- Stereo Amplifier Type

Pinouts 12

- Power Pins
- UART Pins
- Volume Trigger pins
- Audio Outputs
- Trigger Pins
- Other Pins

Copying Audio Files 16

Powering it 18

- USB power pack
- Powering with Amplifier
- Wiring a battery pack to the Vin + GND pins
- Using Vin JST Connector
- Using a Lipoly Backpack

Triggering Audio 24

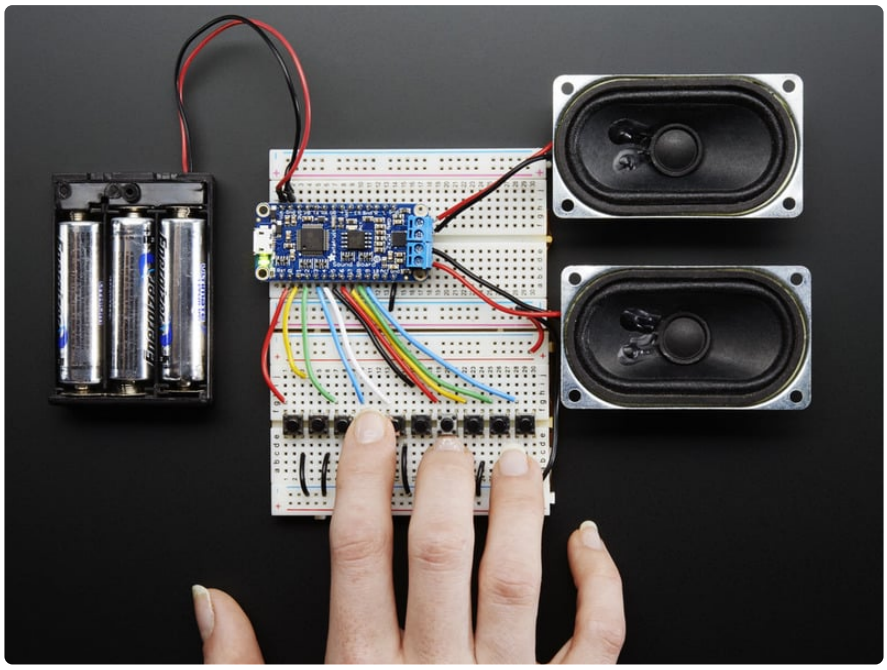
- How many triggers are there?
- How long does it take for audio to play once I've triggered the pin?
- Trigger Types
- Basic trigger - Tnn.WAV or Tnn.OGG
- Hold Looping Trigger - TnnHOLDL.WAV or TnnHOLDL.OGG
- Latching Loop Trigger - TnnLATCH.WAV or TnnLATCH.OGG
- Play Next Trigger - TnnNEXT#.WAV or TnnNEXT#.OGG
- Play Random Trigger- TnnRAND#.WAV or TnnRAND#.OGG
- Wire up Buttons

Serial Audio Control 28

- Arduino Library
- Load Demo Sketch
- General Usage
- Commands
- IDLE mode commands
- List Files
- Volume up and down
- Play track by Number
- Play track by name
- PLAY mode commands
- Pause & Unpause
- Stopping Playback
- Current playback time
- File size and remaining

Advanced Triggering 37
• Trigger Order
• Basic foreground & background
• Advanced Background/Foreground
Creating Audio Files 40
How Much Music? 41
• Compressed or No?
• Stereo or Mono
• Bit Rates / Sample Rates
• Some examples
•
• Uncompressed WAV
• Compressed Ogg Vorbis
Troubleshooting 43
• Doesn't show up as a drive on my computer
• I can't get audio to play when I trigger the pins
F.A.Q. 44
Downloads 46
• Disk Images
• Datasheets & Files
• Firmware
• Schematic and Fabrication Print
• Headphone Out version
• Amplifier Version
• Mini Version

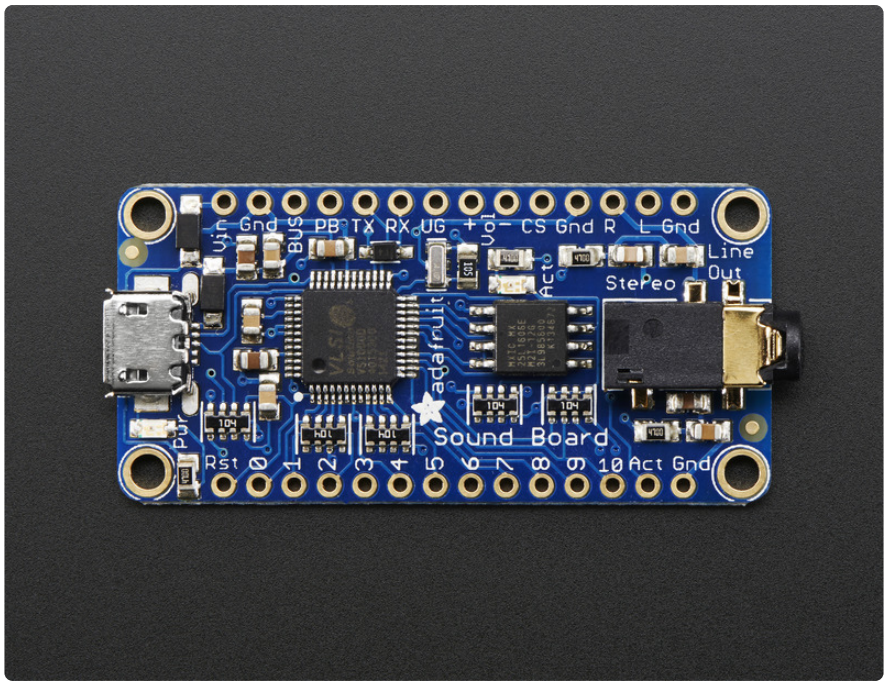
Overview

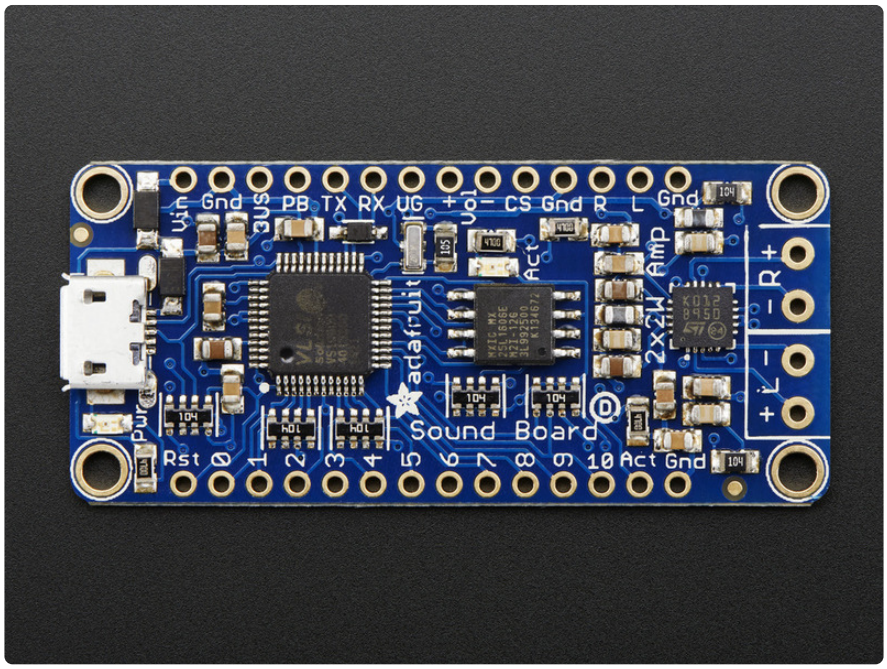


Would you like to add audio/sound effects to your next project, without an Arduino+Shield? Or maybe you don't even know how to use microcontrollers, you just want to make a sound play whenever you press a button. What about something that has to be small and portable? You are probably feeling a little frustrated: it's been very hard to find a simple, low cost audio effects trigger that is easy to use and does not require any programming

UNTIL NOW!

Don't get me wrong, I love the MP3 Music Maker shield, and our Wave Shield is a dependable classic. But you still need to get an Arduino involved. There's all sorts of tricks with ISD chips or recordable greeting cards, but they never sound any good. So after a lot of engineering and tinkering we've come out with the Adafruit Sound Board, the easiest way ever to add audio effects to a project!





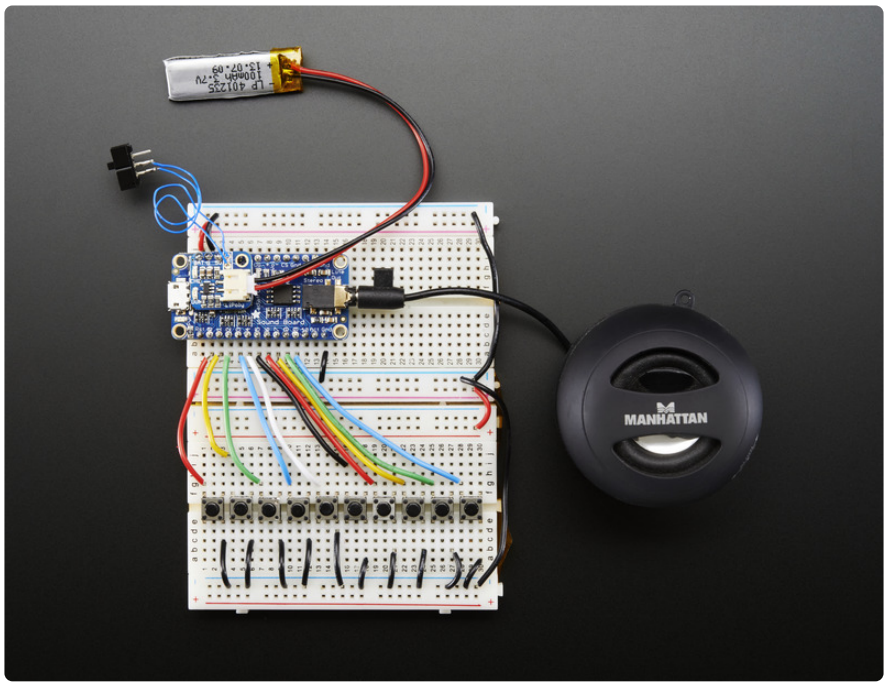
The Sound Board has a lot of amazing features that make it the easiest thing ever:

- **No Arduino or other microcontroller required!** It is completely stand-alone, just needs a 3 to 5.5VDC battery
- **Small** - only 1.9" x 0.85"
- **Built in storage** - yep! you don't even need an SD card, there's 2MB or 16MB of storage on the board itself, so you can store up to 15 minutes of quality compressed audio. Double that if you go with mono instead of stereo
- **Built in Mass Storage USB** - Plug any micro USB cable into the Sound Board and your computer, you can drag and drop your files right on as if it were a USB key

- **Compressed or Uncompressed audio** - Go with compressed Ogg Vorbis files for longer audio files, or uncompressed WAV files
- **High Quality Sound** - You want 44.1KHz 16 bit stereo? Not a problem! The decoding hardware can handle any bit/sample rate and mono or stereo
- **11 Triggers** - Connect up to 11 buttons or switches, each one can trigger audio files to play
- **Stereo line out** - There's a breakout for both left and right channels, at line level, so you can always hook up to any kind of stereo or powered speaker
- **Five different trigger effects** - by changing the name of the files, you can create five different types of triggers which will cover a large range of projects without any programming

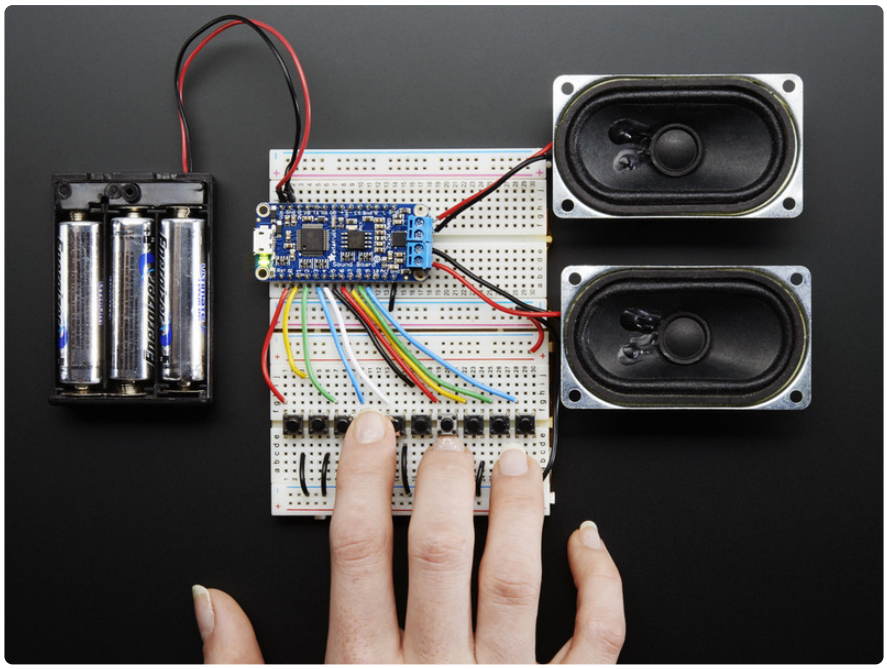
Amplifier or Headphone out

We have two versions, one with a headphone jack so you can connect to a stereo or powered speaker set



And a version with built in class D stereo amplifier, you can have 2 x 2.2W of output audio power, driving 4 or 8 ohm speakers.

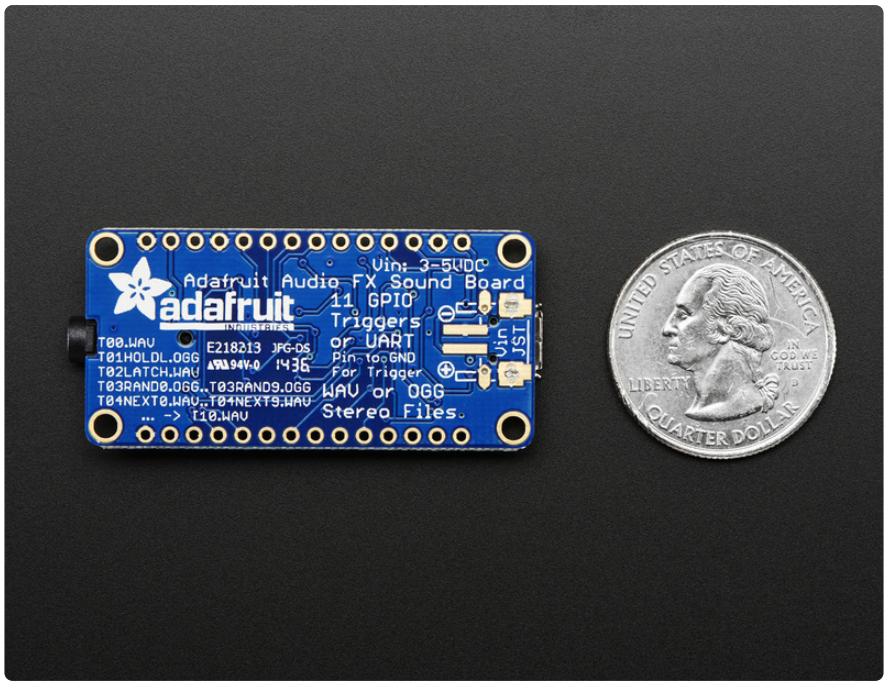
The amplified version of the board comes with two terminal blocks. They may be blue or black.



Trigger Effects

What do we mean by trigger effects? Well, depending on your project you may need to have audio play in different ways. We thought of the five most common needs and built it into the Sound Board so you just rename the file to get the effect you want.

1. **Basic Trigger** - name the file Tnn.WAV or Tnn.OGG to have the audio file play when the matching trigger pin nn is connected to ground momentarily
2. **Hold Looping Trigger** - name the file TnnHOLDL.WAV or .OGG to have the audio play only when the trigger pin is held low, it will loop until the pin is released
3. **Latching Loop Trigger** - name the file TnnLATCH.WAV or .OGG to have the audio start playing when the button is pressed momentarily, and repeats until the button is pressed again
4. **Play Next Trigger** - have up to 10 files play one after the other by naming them TnnNEXT0.WAV thru TnnNEXT9.OGG. Will start with #0 and each one on every momentary button press until it gets through all of them, then go back to #0
5. **Play Random Trigger** - just like the Play Next trigger, but will play up to 10 files in random order (TnnRAND0.OGG thru TnnRAND9.OGG) every time the button is pressed momentarily

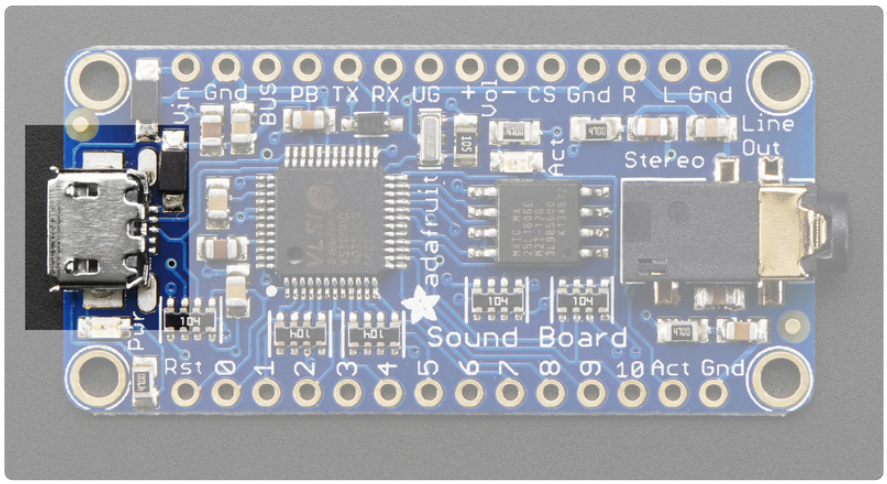


The sound board is designed to be simple: it does not have polyphonic ability, can't play MP3's, isn't reprogrammable or scriptable, and you can't have any other kind of trigger type. However, there's a good chance the project you want to make will work great.

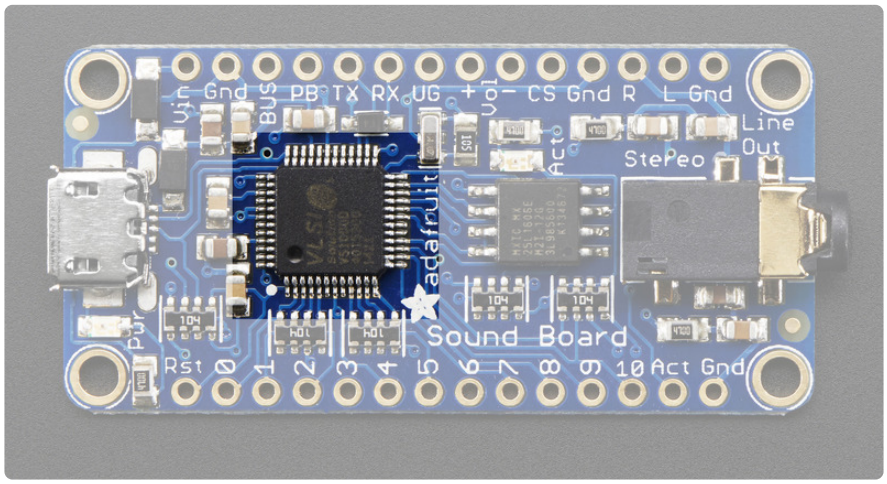
We designed this board specifically for people who wanted to make props, costumes, toys, and other small portable projects. Check out the tutorial for all the powering options, you can power from 3-5VDC so a 3xAAA battery pack (http://adafru.it/727) or a LiPoly battery will work well. You can even use our LiPoly backpack to fit on top for an all-in-one rechargeable effects board (http://adafru.it/2124)

Tour

We'll go over each pin on the board in the next page, right now let's look at the basic elements of the Sound Board so you have an idea of what each piece does.

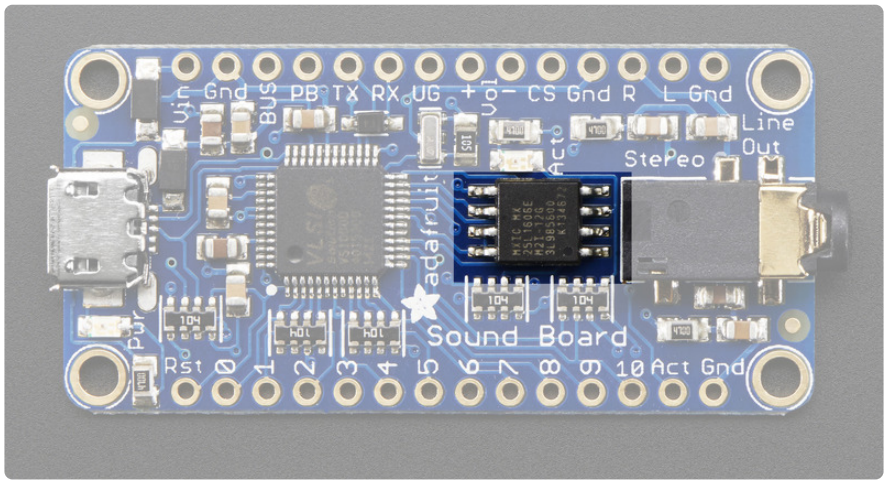


Starting from the left is the micro B USB jack. You'll need a standard Micro B USB cable to connect this to your computer. You do need a computer (any will do) in order to save files onto the Sound Board flash storage



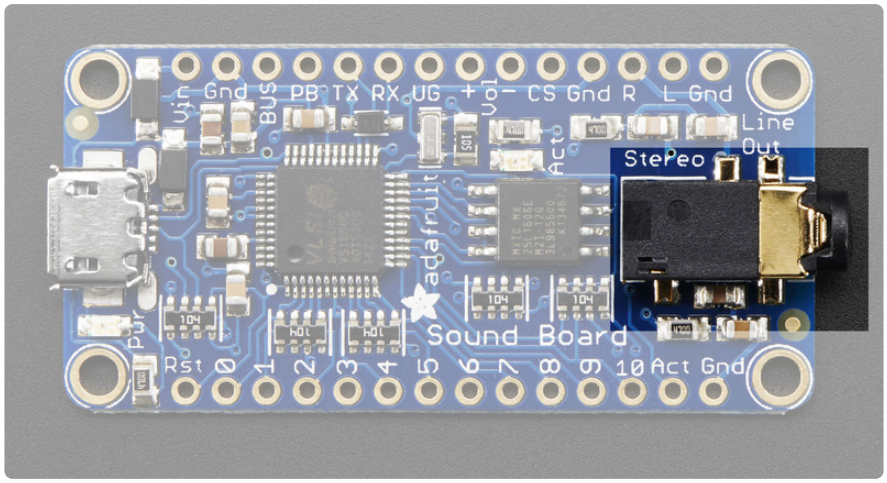
This big chip is the audio decoding engine. It has OGG and WAV decoding capability and can listen to the trigger pins to know when to play each file. It has great audio quality, much better than trying to have a microcontroller generate the audio on its own.

It cannot decode MP3 files - but that makes it a lot less expensive since MP3 decoding requires a patent license where as OGG and WAV are free for any kind of use.



This is the onboard flash chip. It's basically like a little SD card, but soldered directly onto the PCB. These chips can store a few megabytes. This doesn't seem like much but for most audio effects, you don't need hours of music. You can store a couple minute's worth of compressed OGG on a 2M flash chip.

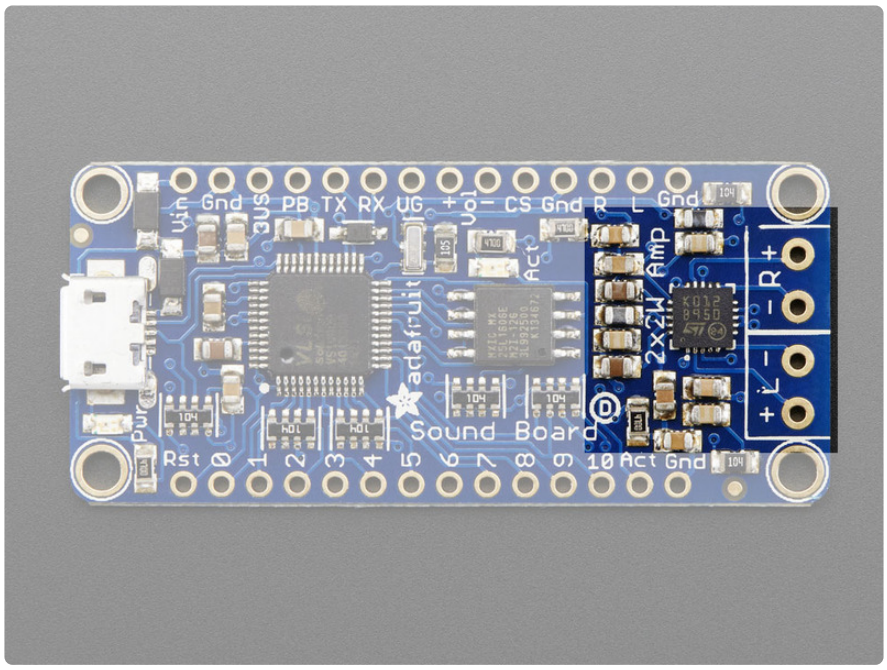
Headphone Output Type



All the way to the right is the headphone jack. This is for connecting to a pair of headphones or into a powered speaker. It cannot drive an 8 ohm or 4 ohm speaker. Instead, an amplifier will have to be used (basically, just like your phone or pocket MP3 player needs an external powered/amplified speaker)

The outputs are 'live level' (about 1Vpp) and have DC blocking capacitors so it can be connected to any kind of amplifier.

Stereo Amplifier Type



All the way to the right is the Class D stereo amplifier. This is for connecting to a pair of speakers! The outputs are bridge-tied-load (BTL) so do not connect R+ to L+ and R-

to L- to get more power, it could damage the chip. If you only need one speaker, simply connect the channel you want.

The amplifier can drive 8 ohm or 4 ohm speaker, up to 2.8W (10% distortion, 4 ohm) and 2.2W (1% distortion, 4 ohm)

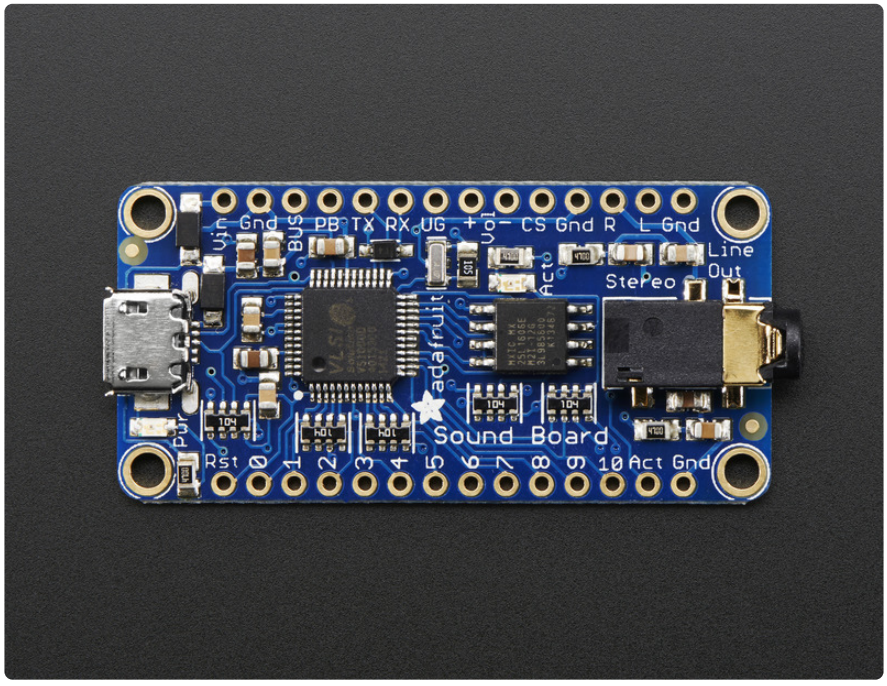
4 ohm speakers will be louder than 8 ohm

Powering from 5V will be louder than 3V, so if you need more oomph, you can power from 4 x AA rechargeable batteries, or 3 x AA Alkaline, or a USB battery pack. It's pretty damn loud tho!

Pinouts

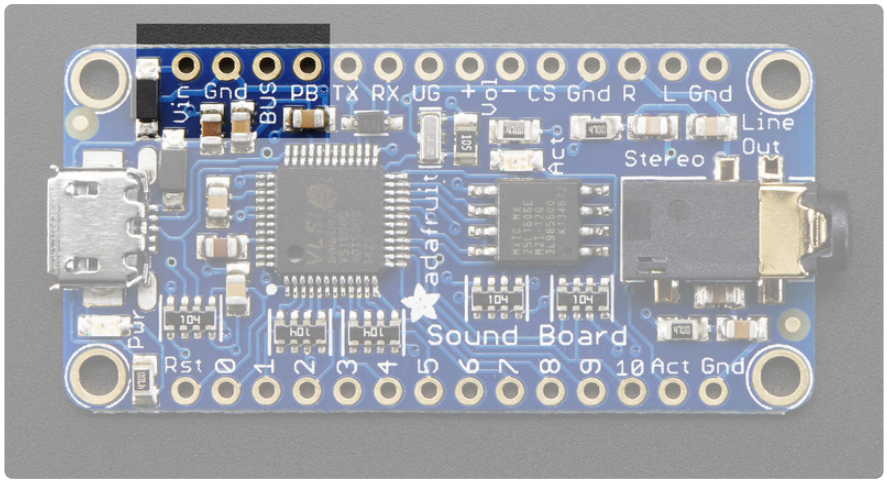
The top and bottom pin sets are the same for the headphone and amplifier out version of the sound board, we just use the headphone version for the photos below

Please note you can only connect a powered amp to the headphone jack output or L / R pins. You cannot connect the amplifier version output to another amplifier. The built in amp means they are for direct-to-speaker connection only.



The Sound Board has a lot of pins, but we can group them into chunks, lets go through each kind of group in order to understand all the stuff you can do.

Power Pins

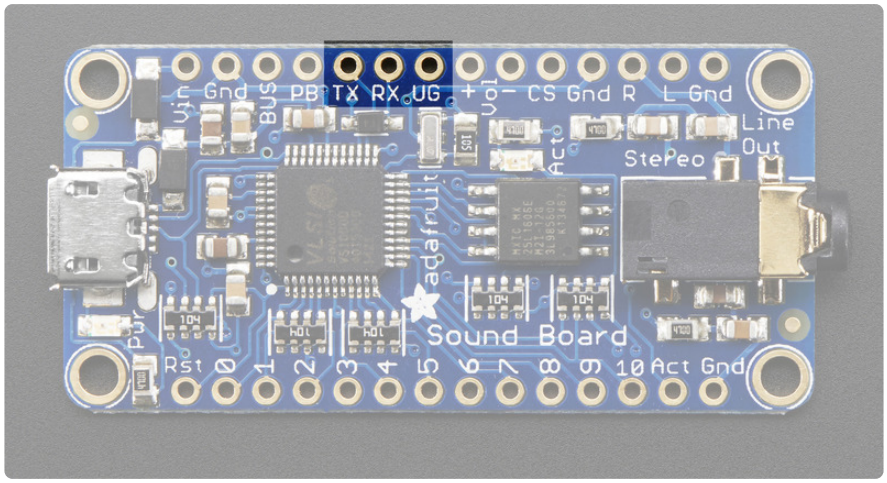


There are a couple power pins in the top. This is what you'll use to power the Sound Board in your project.

- **Vin** - This is the primary 'battery' power input pin. Power with 3-5.5 VDC
- **GND** - there's a couple ground pins but we suggest this one for power input. The others can be used for signal grounds
- **BUS** - this is the 5V that comes from the USB connector. We break this out in case you want to use it to say charge a LiPoly battery (See the Powering It page!)
- **PB** - this is the 'Power Button' pin, which we use for testing. Just leave it disconnected for use.

In general, you'll only want to power through the Vin and GND pins. The Vin input pin is polarity protected and can use 3V-5.5V DC input. If you want to use a larger voltage, you'll need another regulator that can bring the voltage down to 3-5V DC

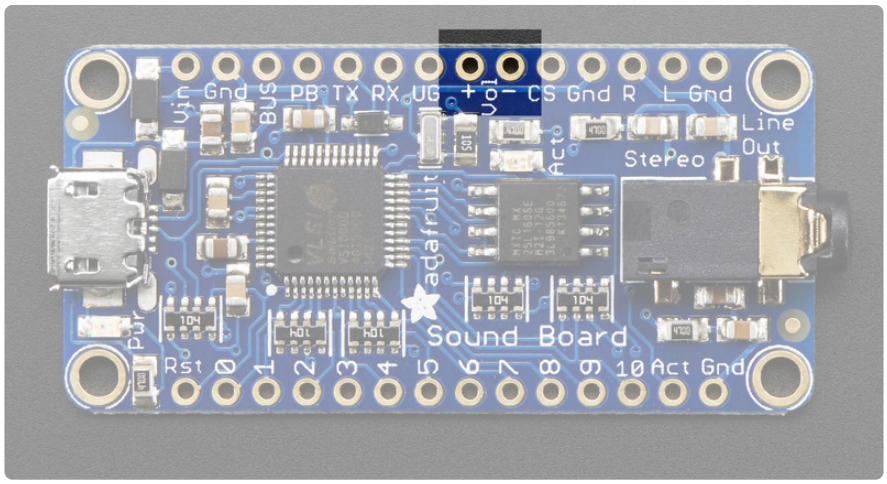
UART Pins



If you want to control the Sound Board over UART Serial, you can do so by using the TX, RX and UG pins.

- **TX** - this is the serial OUT from the board. 3.3V logic
- **RX** - this is the serial INTO the board. We add some level shifting so you can use 3-5V logic.
- **UG** - the UART/GPIO selector pin. Pulled high for default GPIO trigger mode. Tie to ground and reset the board for UART mode.

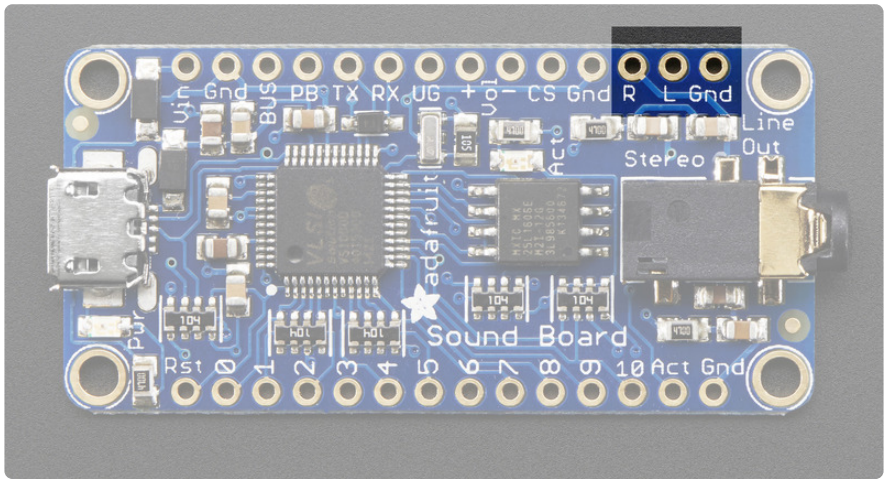
Volume Trigger pins



These are two extra inputs that can be used to adjust the volume in GPIO trigger mode only. Connect buttons that go from each pin to ground, when the button is pressed, the volume will decrease or increase

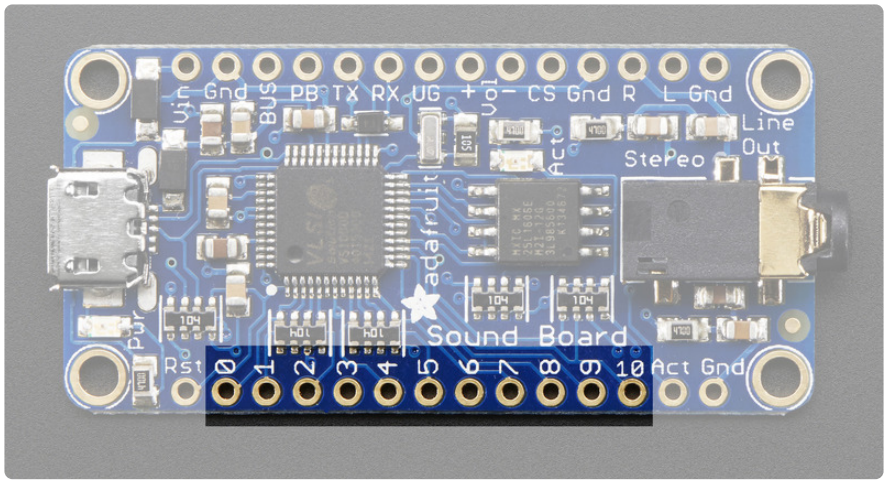
The volume is **not** stored during power cycling or resets, so it will have to be re-applied after restarts.

Audio Outputs



If you don't want to use the headphone jack, connect your amplifier to the Right, Left and Ground pins. These are line level outputs for headphones of about 30 ohms, not for big speakers! There are output DC blocking capacitors so you can safely connect directly to any kind of amplifier

Trigger Pins

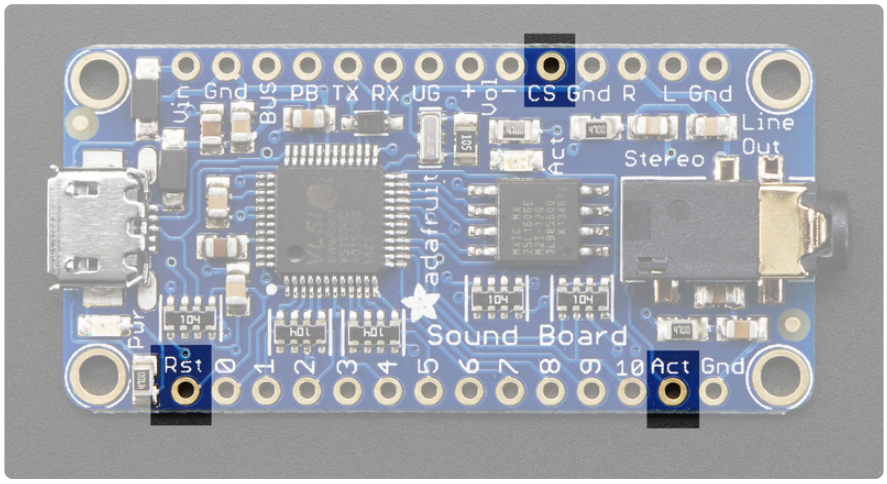


Finally, we're at the trigger pins! These are the 11 inputs that can be used to trigger audio playing. We'll go over how to set the file for each pin in a later page.

Each pin has a pullup resistor, use a button/switch to connect to ground when pressed in order to 'trigger' it. The board has debouncing logic built in.

Note: Trigger Pins are meant to be connected to switches to ground and not voltage levels.

Other Pins



We have a few other pins you may be curious about

- **Rst** - this is the reset pin, you probably don't need to use this pin but when tied to ground it resets the board.
- **Act** - this is the Activity pin, which goes low when an audio file is played
- **CS** - this is the chip select line for the onboard flash chip, do not connect anything to it! We use it for programming the chip during manufacture!

Copying Audio Files

We'll start by copying some audio files over to the Sound Board. To make it really easy, we have a pack of files ready for you. They're not terribly interesting (just spoken words) but will help you test the triggers.

Download the zip file here and uncompress it on your computer:

Demo_Tracks.zip

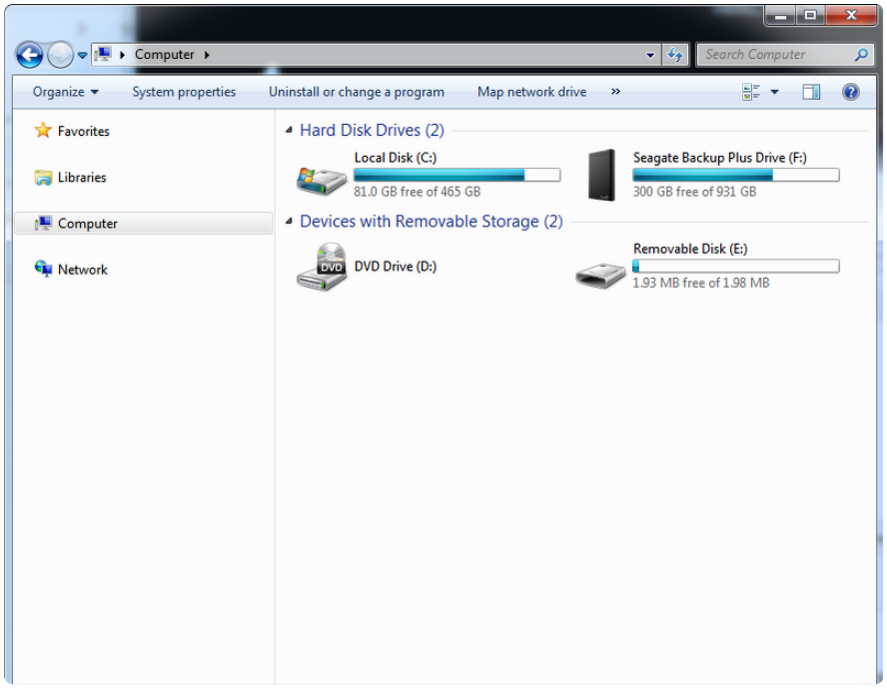
https://adafru.it/e0s

Now plug in the sound board into your computer, it will show up as a new USB key. No drivers are required!

Note that WAV files typically play faster than OGG files, as they are uncompressed and it takes a tiny but possibly noticeable delay in playing OGG

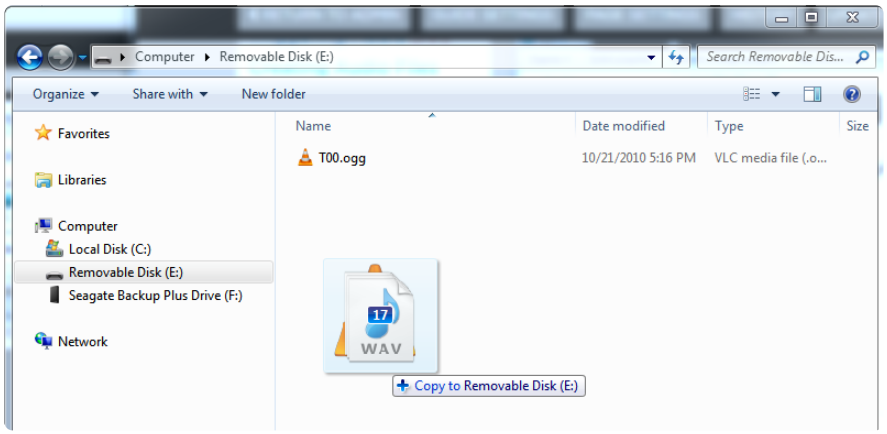
files when the board decompresses the file. So you may want to convert files to WAV for fastest playback.

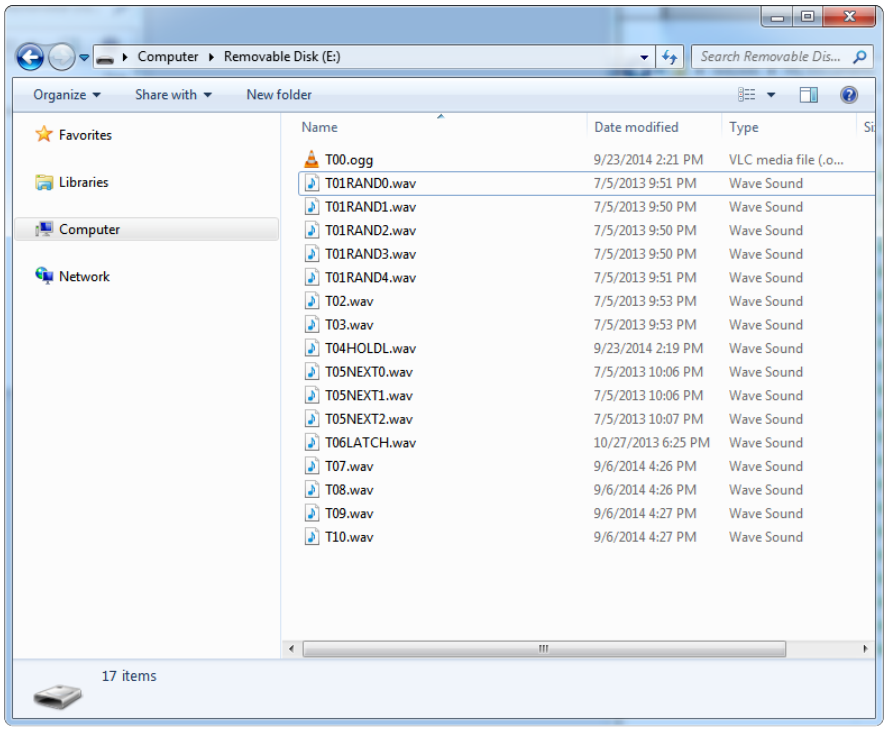
Please use a known good data+power USB cable. With battery packs one gets cheap "power only" cables that will not work to connect via USB to copy sound files.



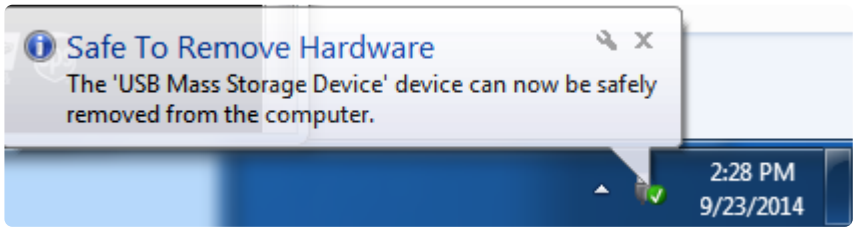
There may already be a file on there, that's from our testing procedure. You can delete or overwrite it.

Copy over all the new files into the new disk drive





Then eject the disk properly to make sure the data was completely written



Powering it

You won't be able to play audio while the board is connected to USB, so to actually trigger audio you'll need to unplug the USB cable and power it another way. There are many ways to power the Sound Board, you'll have to go with whatever it easiest for you.

USB power pack

This is by far the simplest way to power and doesn't even require any soldering! Simply connect the board micro USB to a USB power pack. Since there are no data lines on a battery pack, it won't try to start up as a USB key. Just connect with any microUSB cable. These are large but it will power the Sound board + amplifier for a looooong time

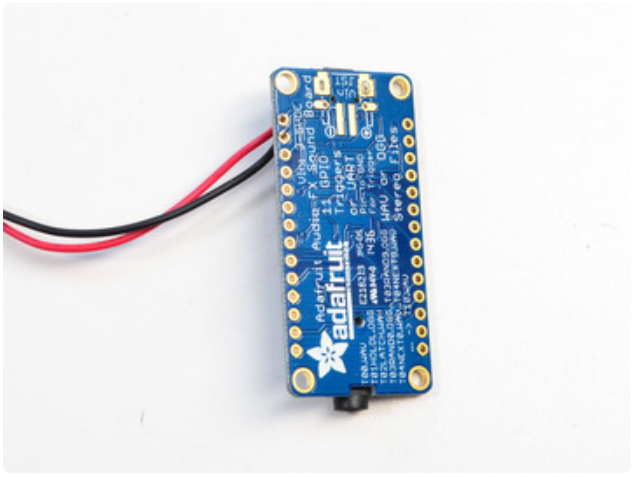
The down-side of using these is that some power packs automatically turn off when they don't detect a lot of current draw immediately. Its impossible to tell ahead of time whether your power pack will work without testing, but if your project doesn't make music sounds almost immediately, it can be an issue

Powering with Amplifier

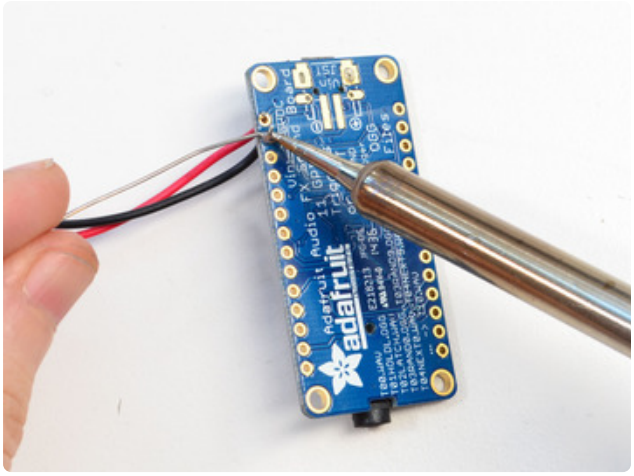
The 2x2W amplifier version of the Sound Board is a pretty beefy 2x2W amplifier. If you are using 2 x 4 ohm speakers, that means you could be seeing as much as 2 Amp spikes of current. (In reality, assume spikes of 1A since audio isn't a DC load). Power with good charged batteries like AA's or even AAA's. For lipoly batteries, a 500mAh battery is minimal! Go for 1200mAh or more if you plan on playing it loud

Wiring a battery pack to the Vin + GND pins

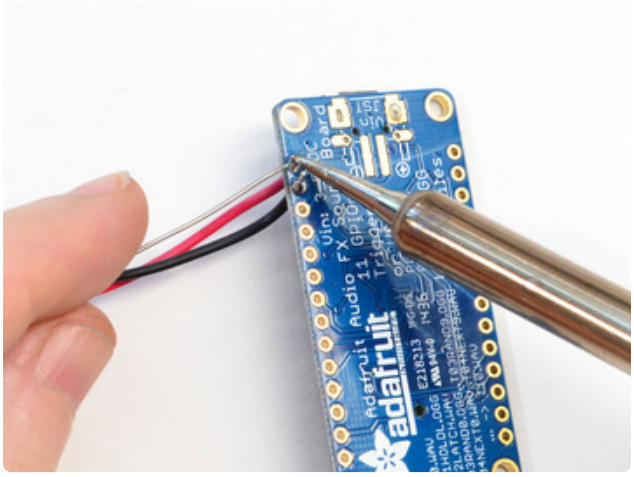
You can use a AA or AAA battery pack and wire it up directly to the Vin an GND pins. We recommend 3xAA or 3xAAA alkaline or rechargeable as ideal. You can use 4xAA or 4xAAA but in that case, make sure you've got rechargeable batteries, as the higher voltage alkalines may be a bit much for the board.

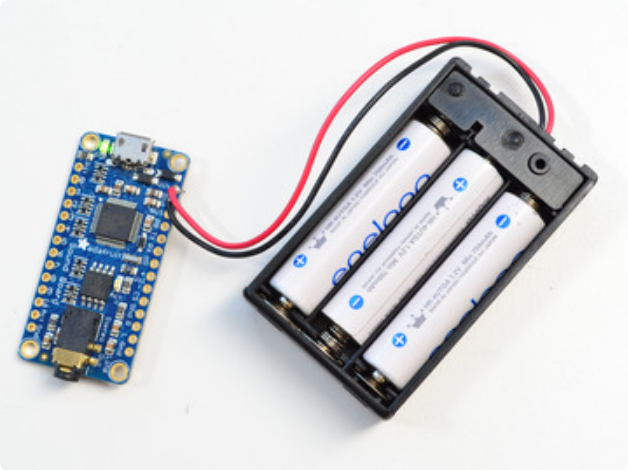


For most battery packs the red wire is the positive wire, connect that to Vin. The black wire is ground, connect that to GND.



Solder both wires securely. You **cannot** just twist the wires into place!

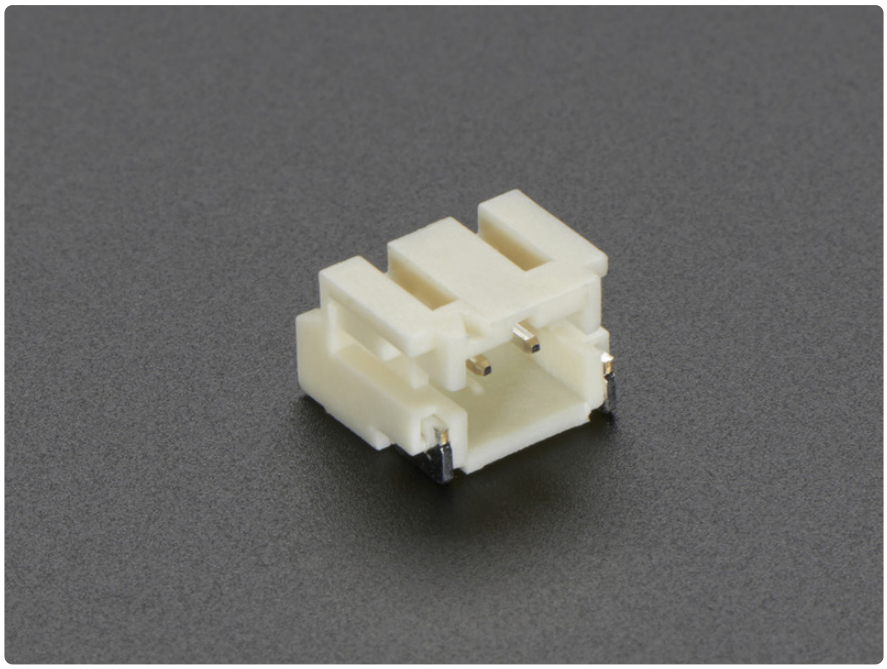




The nicer battery packs have switches, or you can just remove the batteries when not in use. If the green light is lit, it's powered up!

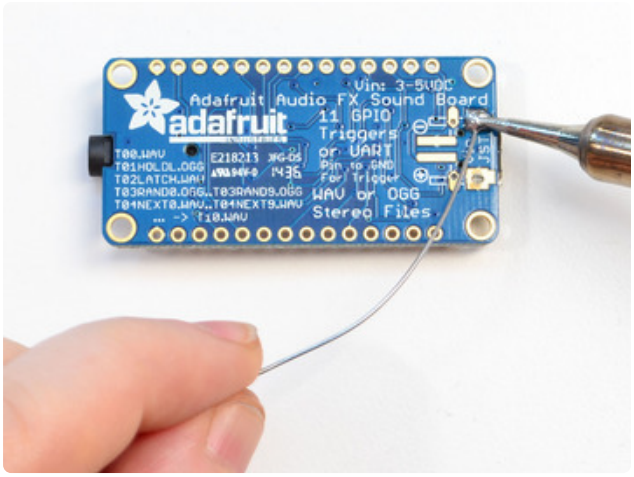
Using Vin JST Connector

If you don't want to solder wires directly, you can solder on a JST PH 2-pin connector (http://adafru.it/1769) to the back. We don't include this on the board because its a bit bulky and if the JST is in place you can snap a battery in or out easily. The connector is the same as the Vin/GND pins on the side so use one or the other!

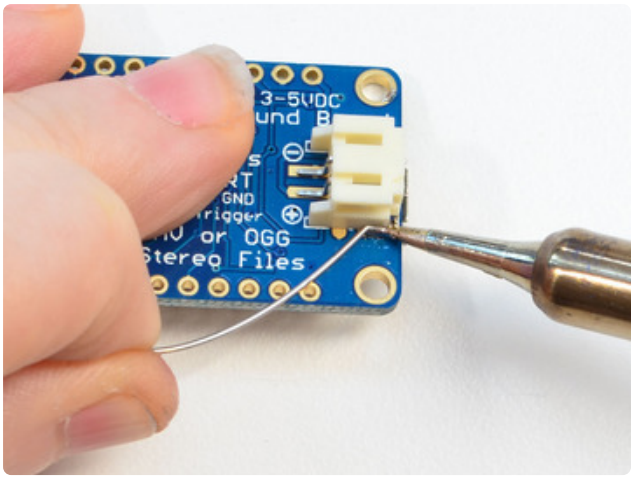


All our Lipoly batteries come with a JST (https://adafru.it/e0v), and our 3xAAA on/off pack also does (http://adafru.it/727).

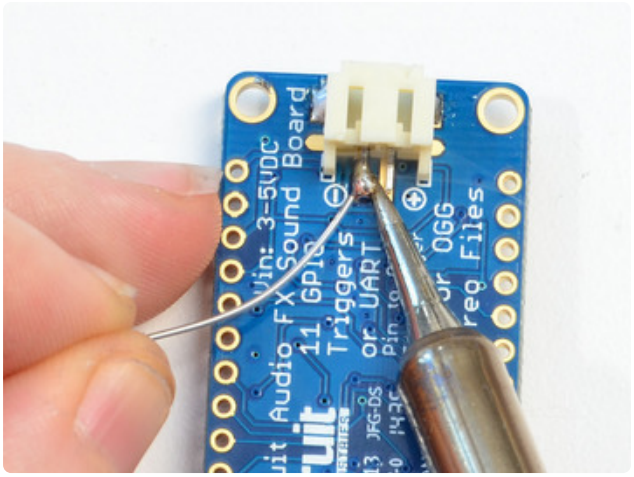
Or, connect your custom battery with a JST cable (http://adafru.it/261)

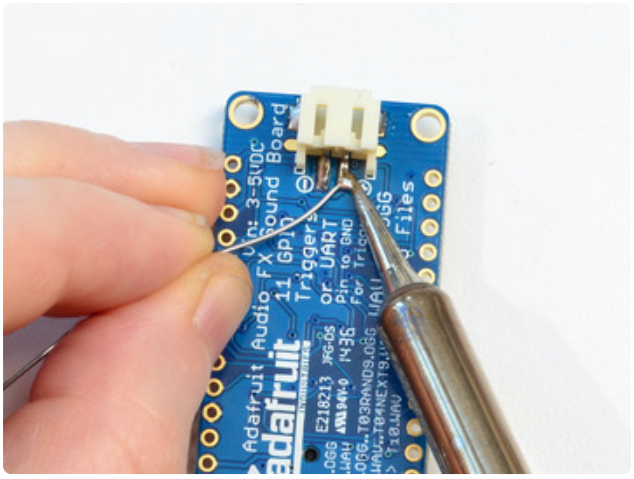


Start by melting some solder onto one of the side pads of the JST footprint on the bottom

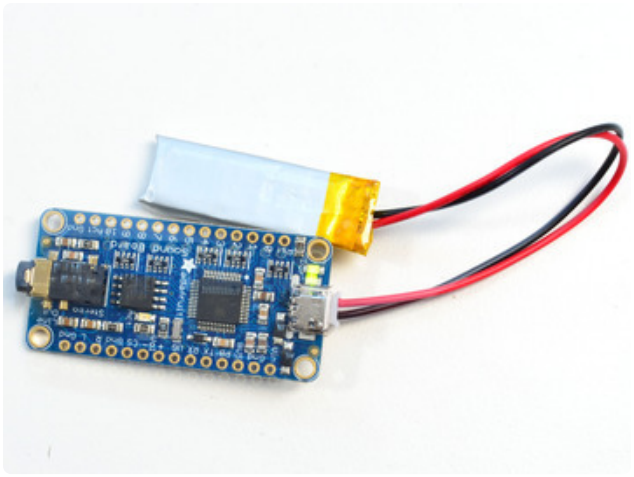


Place the JST and solder that pad onto the connector to tack it in place





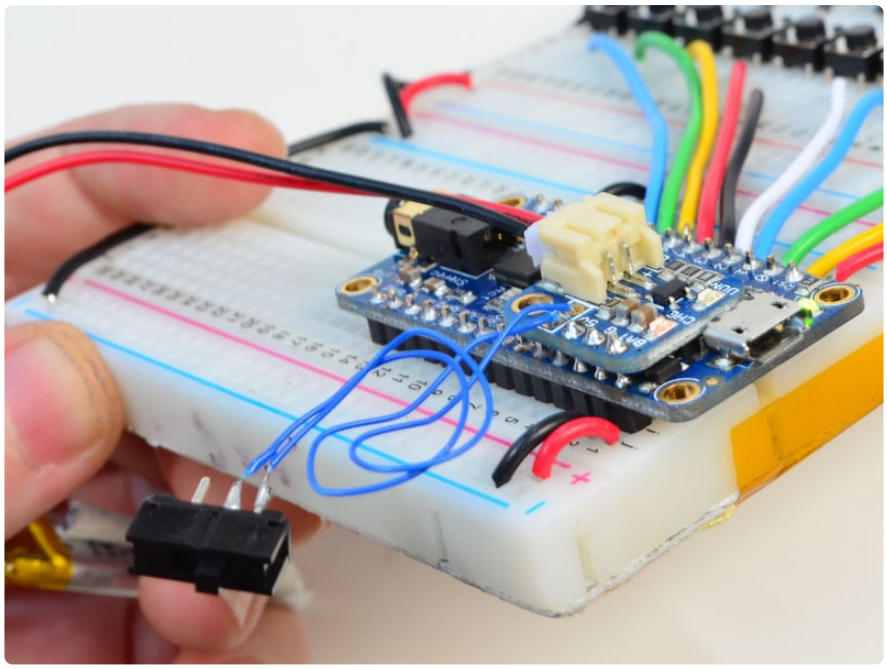
Solder all four pads - two mechanicals on the side, and two power coming out the end.



Plug in your battery and look for the green light!

Using a Lipoly Backpack

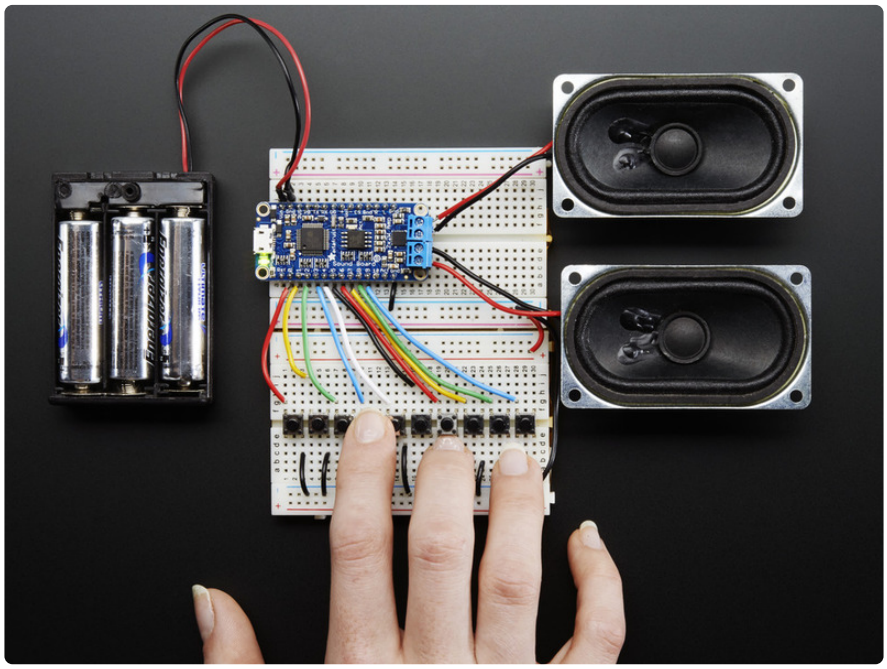
We designed a Lipoly backpack for the Pro Trinket (http://adafru.it/2124) but you can also use it with the Sound Board! Use the extra long header included to solder it over the USB jack, it connects to Vin, GND and BUS



The Lipoly backpack lets you plug in a lipoly battery of any size and recharge it over the MicroUSB connector, so it's basically an all-in-one portable + rechargeable system.

With the backpack, you can easily add an on/off switch, as well. (http://adafru.it/805)

Triggering Audio



OK **finally** you are ready to play some audio. You can use the board in two ways, either in 'Trigger mode' (default) or 'Serial mode' (more advanced usages, see the next page)

This page will talk only about using it in trigger mode since that's what we think most people will do.

How many triggers are there?

There are **eleven trigger pins** - named #0 thru #10

You don't have to use them all! We just had lotsa pins so we made them all available.

Each trigger bin is an input that we recommend using with a pushbutton or tactile button or other kind of switch. **When the # pin is connected to GND for more than about 125 milliseconds it will trigger**! There is a 100K pull up resistor on each one, so you do not need any extra resistors or pullups.

Remember, you don't have to use a mechanical button or switch - you can use conductive thread, tilt switches, two pieces of tinfoil, the output from some other electronic thingy, just anything that will send a ground signal to the pin.

How long does it take for audio to play once I've triggered the pin?

Good question! It matters whether you are using WAV or OGG. Compressed audio takes a little more time to get going.

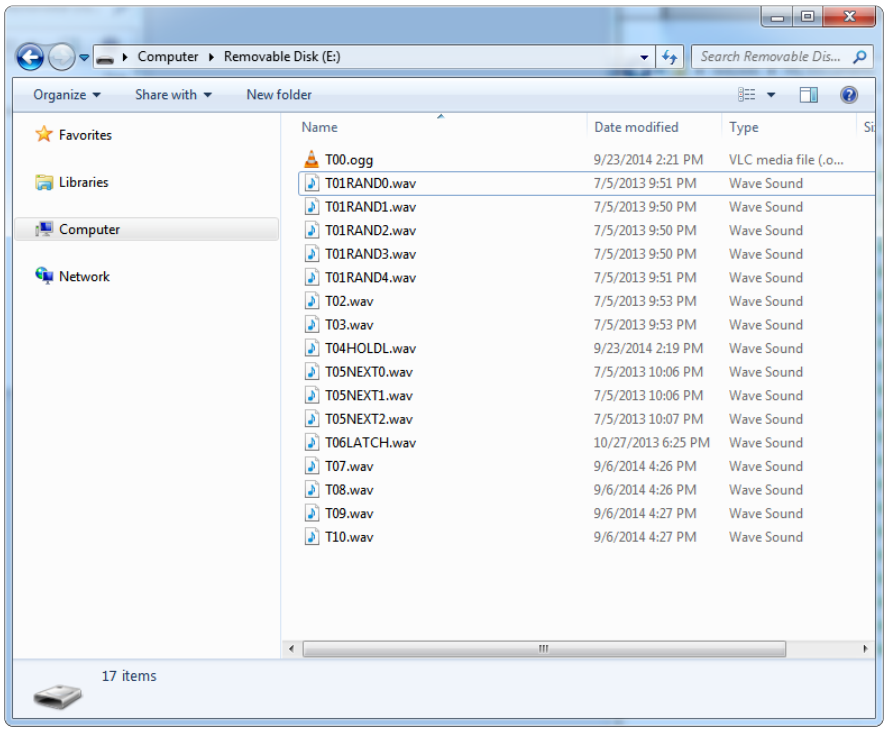
From the moment the SFX board sees a ground level on the pin, it takes ~120ms to play a WAV file and ~200ms to play an OGG file. These are within 'instant feedback' expectation (https://adafru.it/e4r)

If 'repeating' a file by keeping the button held down, or doing a latching trigger type, there's a ~20ms delay (imperceptible) between WAV replays and ~120ms between OGG replays (noticable if the audio is meant to perfectly loop).

Trigger Types

There are a lot of different ways you may want to play your sound effects. Normally a microcontroller would be required to get exactly what you want, but the Sound Board is pretty smart and has the ability to **play audio a couple different ways depending on the file name**

There's no code or firmware involved, only the file name of the audio file!



Let's understand this by going through the five types:

Basic trigger - Tnn.WAV or Tnn.OGG

The first type is the 'basic trigger' - when the button is pressed, audio plays. The entire file is played from beginning to end once.

To enable this trigger, name the file **Tnn.WAV** or **Tnn.OGG** where nn is the trigger #. For example, if you want to use pin #0, the file could be called **T00.WAV** (that's two zeros after the T), if you want to use pin #6, **T06.OGG** - all the way up to **T11.WAV**

Hold Looping Trigger - TnnHOLDL.WAV or TnnHOLDL.OGG

This is a more complex trigger. Instead of pressing once the button to play, it plays ONLY when the button is held down. Great for "hold the button down to play the ray gun blaster sound effect" Call the file **T02HOLDL.WAV** for example

As long as the trigger pin is connected to ground, it will continue to play the same track on repeat. If you want a perfectly smooth transition between the end and beginning, we suggest WAV files, as OGG decompression takes a few milliseconds and has a noticeable delay.

Latching Loop Trigger - TnnLATCH.WAV or TnnLATCH.OGG

This is a little like the Hold Looping trigger but you do not need to keep the button held down. Instead, press the button once to start the looping effect, then press it again to stop.

This is maybe good for if you want a continuous effect without having to keep the pin held down. Call the audio file **T08LATCH.OGG** for example

If you want a perfectly smooth transition between the end and beginning, we suggest WAV files, as OGG decompression takes a few milliseconds and has a noticeable delay.

Play Next Trigger - TnnNEXT#.WAV or TnnNEXT#.OGG

Let's say you want to have one button but many different sound effects. For example, a stuffed animal that has a squeeze sensor trigger. It would say different things each time it is squeezed. For this kind of effect, use the Play Next Trigger.

This trigger is basically like the basic trigger, one button press per play, but you can have multiple effects on one pin

You can have up to 10 audio files triggered on one pin, they will play in order. For example, if you're using pin #3, the files would be named **T03NEXT0.WAV**, **T03NEXT1.WAV**, **T03NEXT2.OGG** etc. up to **T03NEXT9.WAV**

Just make sure it starts with #0, and put as many as you like up to #9. You do not need to use all 10 '# slots' up. If a number is missing, like **T03NEXT3.WAV** doesn't exist, it will automatically play #0 again.

Play Random Trigger- TnnRAND#.WAV or TnnRAND#.OGG

OK so you like the Play Next mode but you don't want to have it always in the same order? Use Play Random mode. You can have up to 10 audio effects, from say **T07RAND0.OGG** up to **T07RAND9.OGG**.

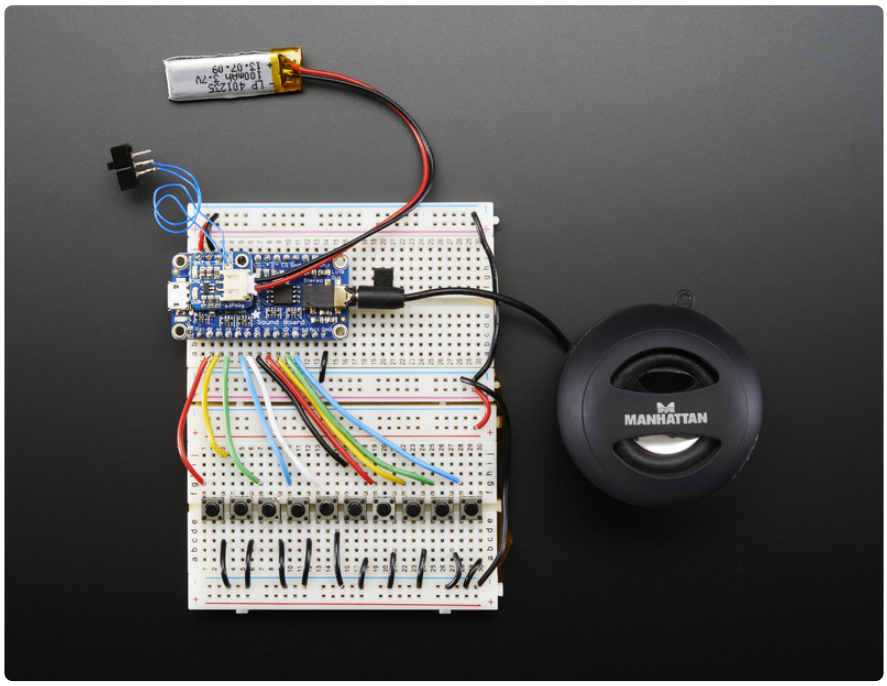
When the button is pressed, a 'random' track will be played.

Please note, this is not 'cryptographic quality' randomness :) In fact, itwill play through all of the tracks at least once (but in any order) before repeating.

Wire up Buttons

If you grabbed the demo tracks from the "how to upload files" page, you can try each of the different kinds of trigger modes

For this demo, I am using a breadboard and small 6mm tactile buttons, but you can use a wire to touch from a trigger pin to GND with headphones plugged into the audio jack



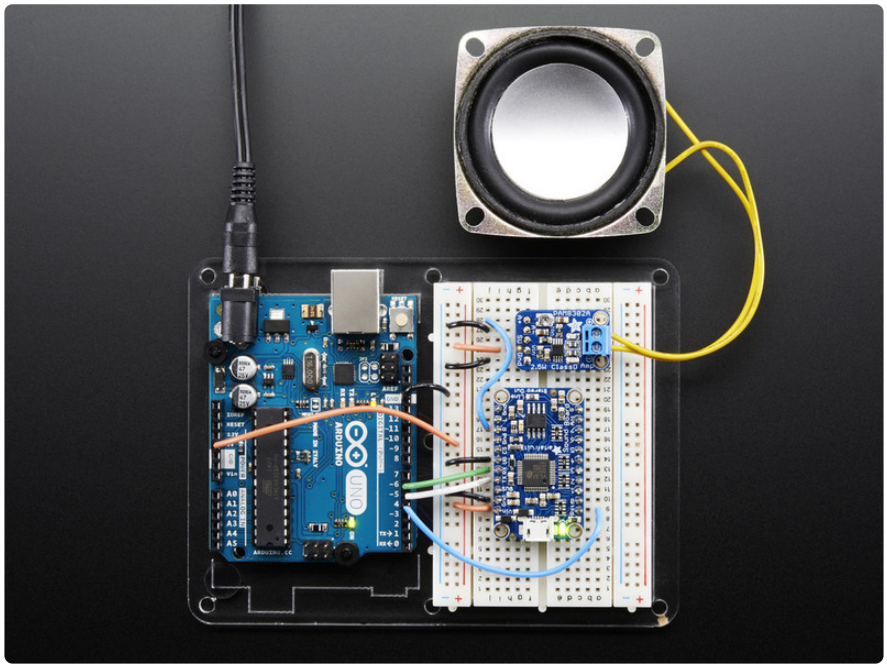
You can see each button connects to the trigger pin, and then the other side to ground. For this demo I'm using the Lipoly backpack but you can power any other way. When buttons are pressed, the audio files on the board get triggered!

Serial Audio Control

If for some reason you don't want to use the trigger mode, say you need some more complex audio playing, you can use serial audio control. This allows any microcontroller with 9600 baud TTL UART to send commands to the module.

All Adafruit Soundboards can be used in either UART mode or GPIO (button) mode - but not at the same time!

We'll demonstrate with an Arduino, but you can use any microcontroller with a little adaptation

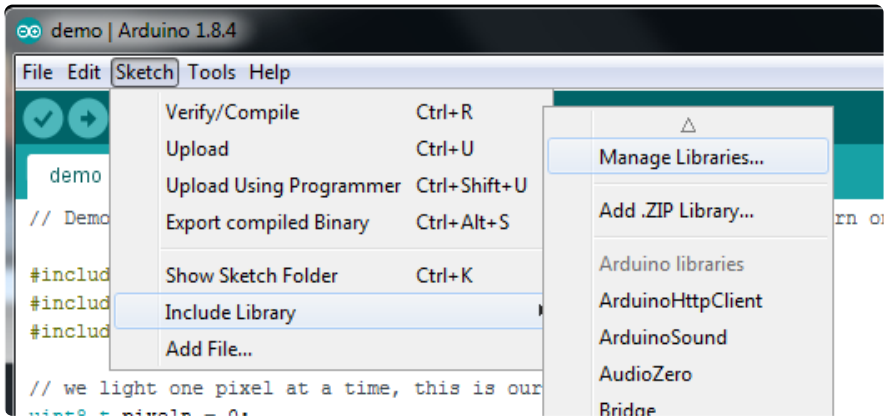


Arduino Library

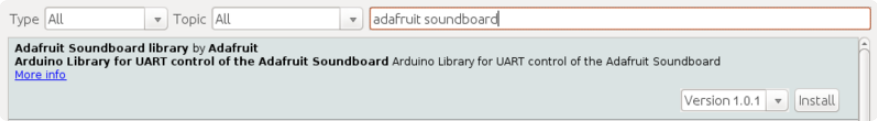
If you have an Arduino, you can wire it as seen above (**UG** to **Ground**, **TX** to **#5**, **RX** to **#6** and **RST** to **#4**) to control the sound board via our simple menu example.

To begin interacting with the sound chip, you will need to download the Adafruit_Soundboard Library (https://adafru.it/eyF) from the Arduino library manager.

Open up the Arduino library manager:



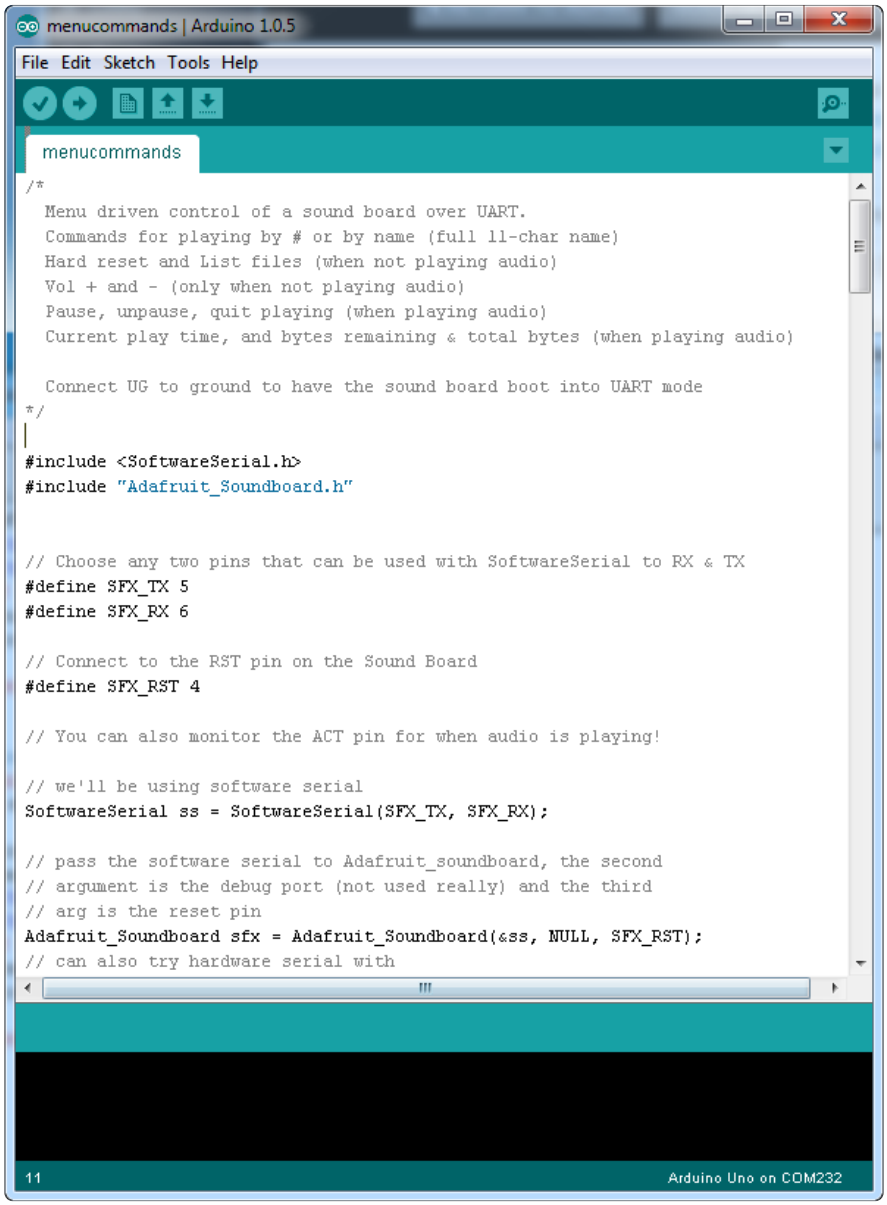
Search for the **Adafruit SoundBoard** library and install it



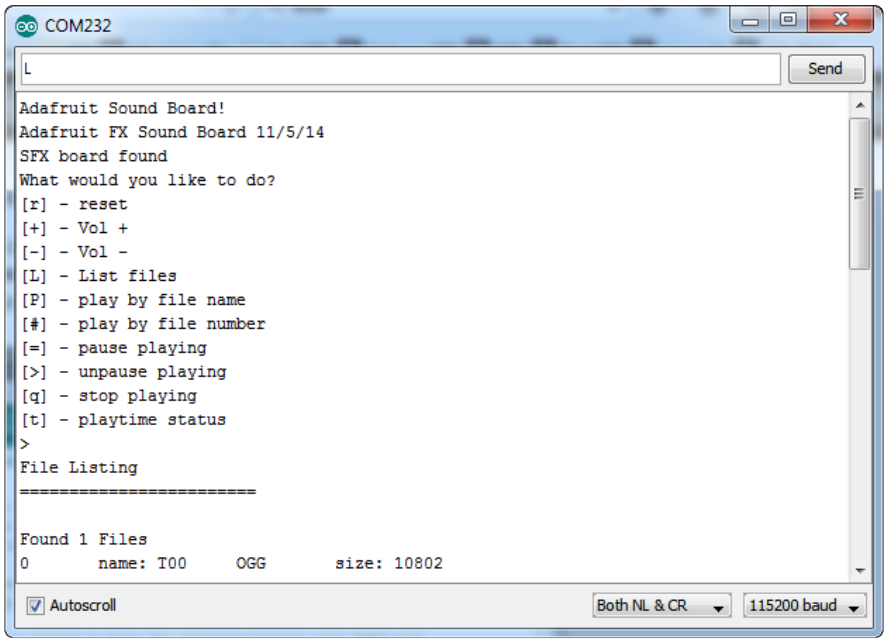
We also have a great tutorial on Arduino library installation at: http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (https://adafru.it/aYM)

Load Demo Sketch

Now you can open up **File->Examples->Adafruit_Soundboard->menu** and upload to your Arduino wired up to the breakout



Then open up the serial console at 115200 baud to interact!



General Usage

For connections we recommend at a minimum

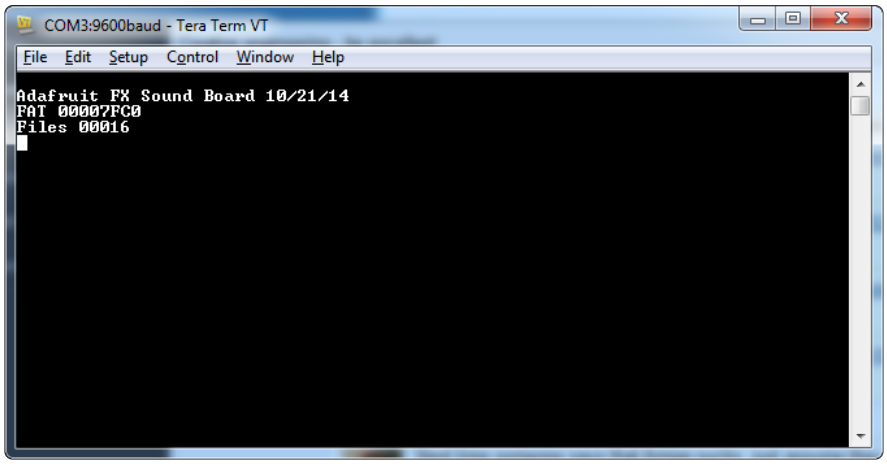
- Connect **UG** to **GND** (to start the sound board in Uart mode)
- Connect **RX** to the data-output pin from the microcontroller into the sound board
- Connect **TX** to the data-output pin to the microcontroller from the sound board
- Connect **RST** to another microcontroller pin, when toggled low, it will reset the sound board into a known state

If you want to know when sound is being played, the **ACT** pin is **LOW** when audio is playing - this output also controls the red ACT LED

You can use 3.3V or 5V logic on the **RX** pin, it has a level shifter. TX is output at 3.3V which can be read from a 3.3V or 5V logic. For the **RST** pin, it's 3.3V and has a pullup. To reset the board, set the microcontroller pin to low, then an output, then back to an input like so:

```
digitalWrite(reset_pin, LOW);
pinMode(reset_pin, OUTPUT);
delay(10);
pinMode(reset_pin, INPUT);
delay(1000); // give a bit of time to 'boot up'
```

After reset, the sound board will print out a bunch of info



First line tells you when this firmware was written (10/21/14) and the name of the board - the name is the same for all the SFX boards right now, but you may have a different firmware date

The second line will read **FAT** and then a 8 digit hexadecimal number, which will tell you the size of the partition that all the files are on in 512 byte sectors. E.g. here 0x7FC0 = 32,704. 32704 * 512 bytes = 16,744,448 (16 MB)

Third line will read **Files** and then a 5 digit decimal # of files that are on the Soundboard. here, we see it has 16 files available.

Commands

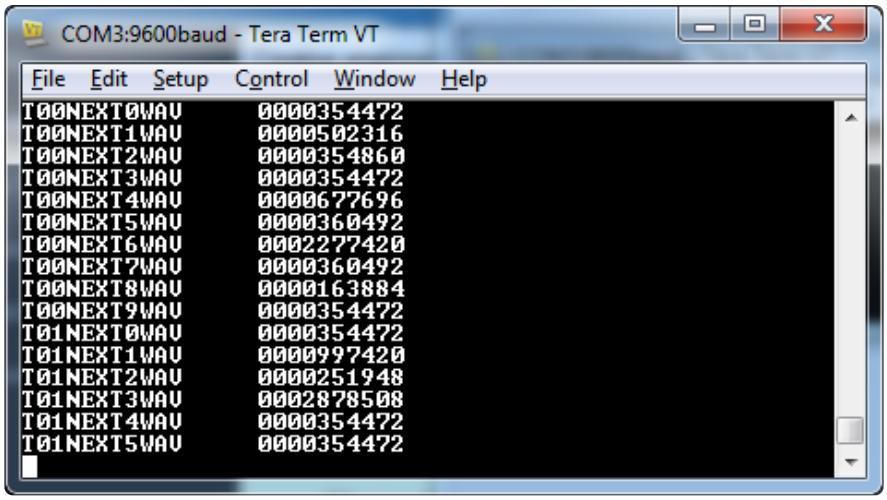
There are a few commands, in two 'sets' - one set is commands that can be run in IDLE mode, and the other is commands that can be run in PLAY mode.

IDLE mode commands

These are commands you can use when audio is not being played!

List Files

- **L** - Send "L\n" (L plus new line) to list files



files will be listed in following format:

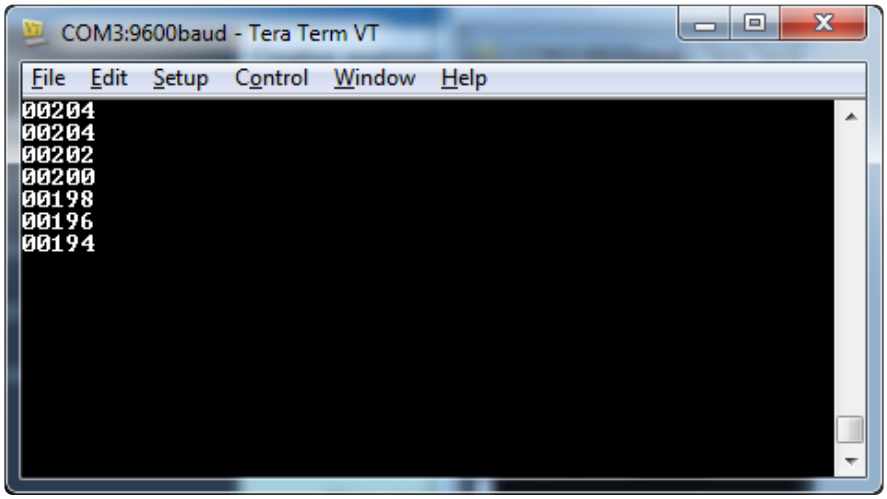
FILENAMEEXTfilesizeinbytes

Where "FILENAMEEXT" is the full 8.3 file name of each file, with no dot, if you have a file with shorter than 8 characters, it will be padded with spaces.

The filesize is in bytes, and is padded with zeros as well

Volume up and down

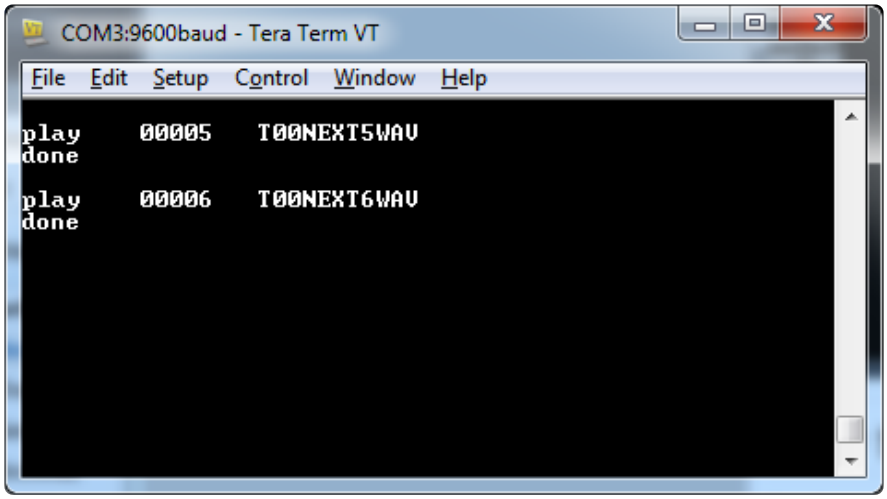
You can adjust volume from 0 (silent) to 204 (loudest) in 2-increments. To increase the volume, send +\n (plus symbol plus new line). to decrease volume send -\n (minus symbol plus new line) and it will reply with the new volume (5 ascii characters, two of which are going to be zeros and then three digits of volume, plus a new line)



Play track by Number

Each track has a number with respect to when it was copied onto the SD card, starting with track #0, these are in the order that the tracks are printed out when you send **'L'** for list tracks.

The fastest way to play a track is by the track number, send **#NN\n** (# symbol, then the number of the track, and end with a new line). E.g. to play the first track (track #0) send **#0\n** and for the 11th track, send **#10\n**



If the file wasn't found, it will reply with **NoFile\n**

If the file is found, the sound board will reply with **play\tNNNNN\tFILENAMEEXT**\n where \t is the tab character and \n is the newline character. NNNNN is the track number, and FILENAMEEXT is the 11-character file name (8.3 without the dot).

When the file is done, it will print **done\n** to let you know.

Play track by name

Maybe you want to just play the track by a name, not number. No problem! You can do that with the **P** command

Send **PFILENAMEEXT\n** - 'P' plus the 11-character file name (8.3 without the dot) then a new line. If the filename is shorter than 8 characters, fill the characters

For example, to play **T00NEXT5.WAV** send **PT00NEXT5WAV**\n and to play **T09.OGG** send **PT09 OGG**\n

If the file wasnt found, it will reply with **NoFile\n** otherwise it will reply as above, with **playNNNNNFILENAMEEXT**\n

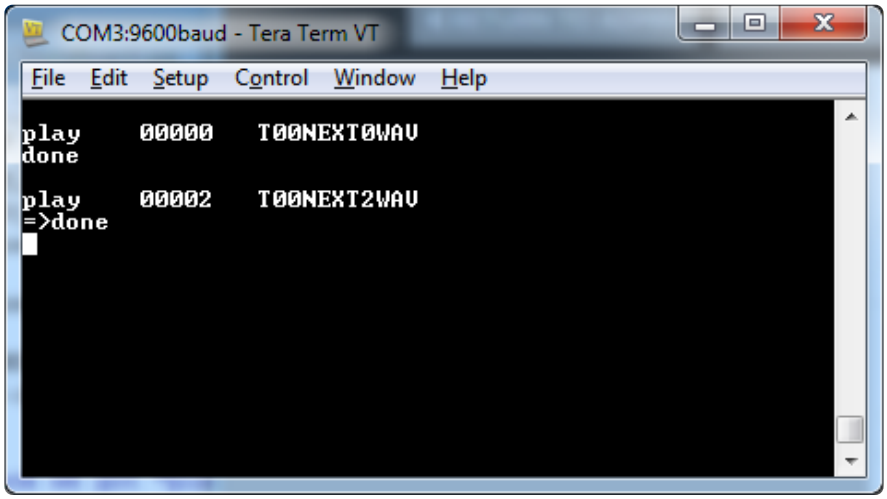
PLAY mode commands

OK now that you are playing audio, you can do stuff during play, these are not available in IDLE mode

Pause & Unpause

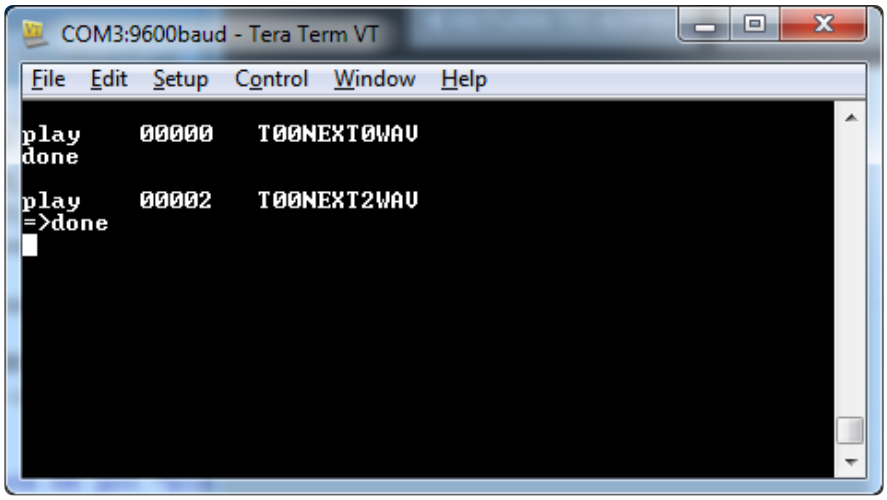
You can pause at any time by sending the = character, no new line required. To restart playback, send the > character

These character will be echo'd back to you on the UART so you know they were received!



Stopping Playback

You can also stop instantly by sending a **q** (for 'quit') and the audio will stop and return back to IDLE mode

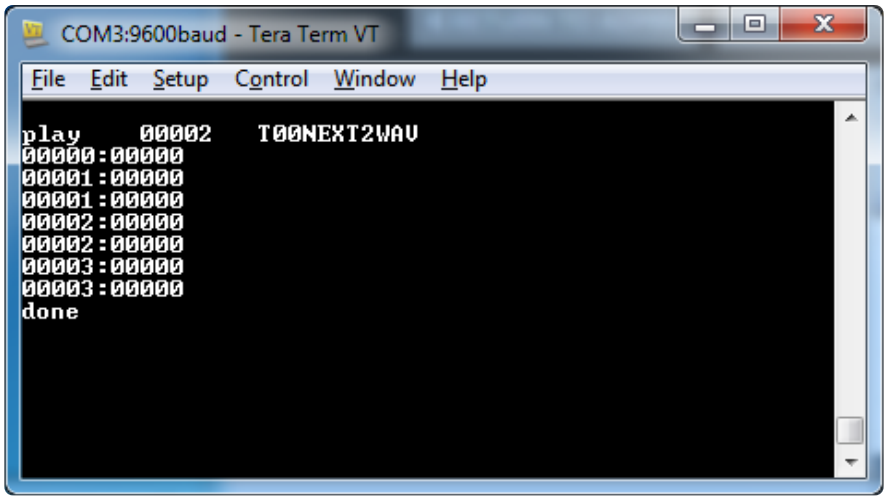


Current playback time

You can query how much time has been spent playing the current track by sending an **t** character

You will get a response which is 5 digits a : and then another 5 digits. The first 5 digits are seconds currently elapsed, the second half is supposed to be the codec's report of total play time but so far we've always just gotten 0's so we suggest just using the first set of digits (current playback in seconds)

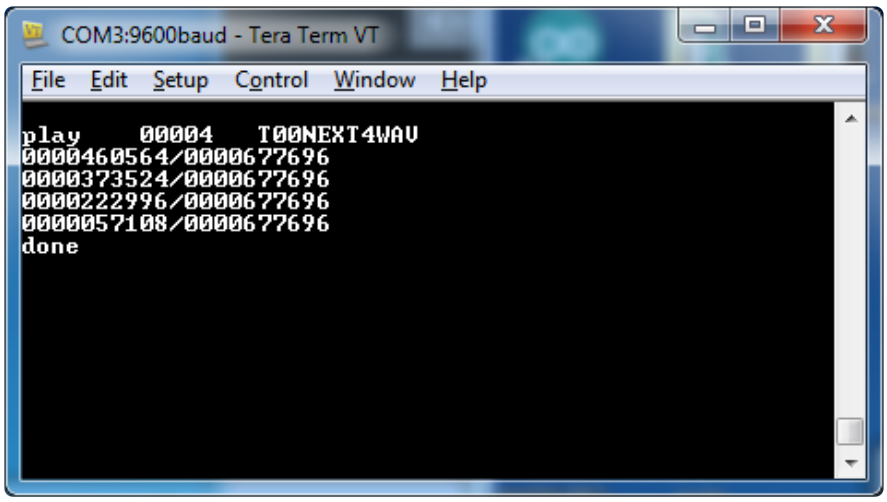
Please note this is a blocking operation. Doing it too much can cause your audio to stutter!



File size and remaining

Finally, you can query the size of the file and how far along you are in it, which is sort of like how much you've got left to play but of course, for compressed OGGs may vary a bit. Still, might be handy!

You can get the file size and remaining bytes by sending an '**s**'



The sizes are reported in decimal bytes, 10 digits long seperated by a /

The first number is remaining bytes and the second number is total size.

Please note this is a blocking operation. Doing it too much/fast can cause your audio to stutter!

Advanced Triggering

Even though the Sound Board is fairly simple and easy to use, you can do some cool stuff with it, if you're willing to get creative

Trigger Order

What happens if you have more than one pin connected to ground/triggered? How does the Sound Board know what to play? Great question!

There are 11 pins, #0 thru #10, the sound board looks at each pin, in order starting with #0 and if it sees that pin connected to ground checks if it can find a **Tnn** file to play

That means, if you have pin #10 held down and then pin #0 held down, on the next loop #0 will always get played first, then #10

You can use this quirk to create background effects and foreground effects

Basic foreground & background

For this example, we'll be using this set of WAV files that have some Darth Vader effects. Unzip and load these onto your sound board

darthvaderSFX.zip

https://adafru.it/e9V

Ok now that you're done, you can see we have one file called **T10HOLDL.WAV** - this is a file that will play on loop as long as pin #10 is held down. There's also a couple **T00RANDx.WAV** files, these are files that will be played in random order when pin #0 is held down

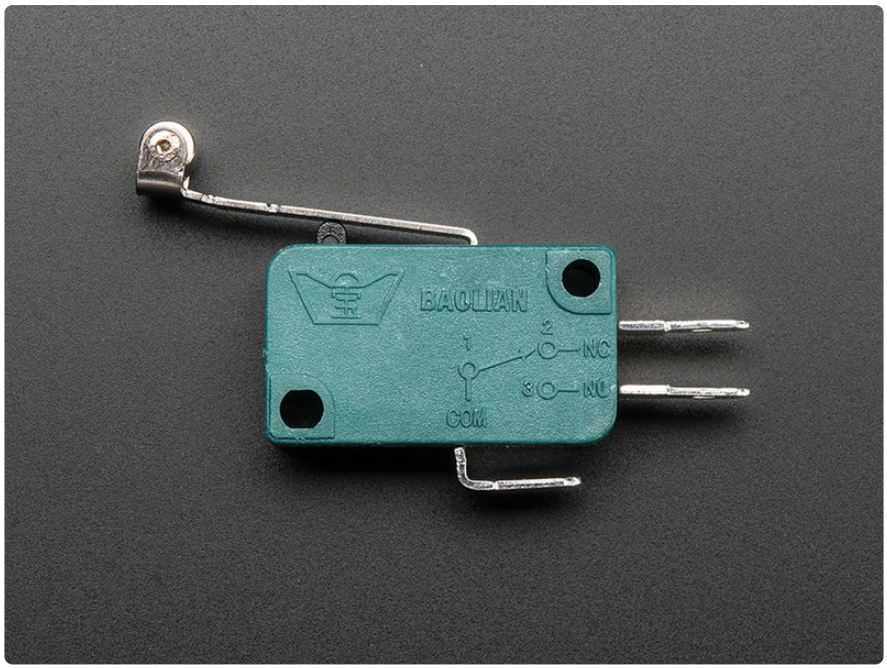
Power the sound board and connect a wire from pin #10 to ground (not a switch!) you'll notice that the Darth Vader breathing sound is now playing on loop. If you then have a button between pin #0 and ground and press it, a voice clip will play on the next loop cycle (once the breathing has ended)

If you have a very short background sound loop, say one that loops evey 1-2 seconds, then this is probably good enough. If you have a long loop like this one, where it takes 3 seconds to loop it can be annoying because you have to wait until the sound is done to have the foreground noise.

Advanced Background/Foreground

The problem here is that you can't release the looping sound since its tied to ground. You could have one button held down all the time or try a latching loop but it's a lot easier if you just use a switch that has NC (normally closed) and NO (normally open) contacts - sometimes called a DPST (double pole single throw) switch

Most low cost switches and buttons do not have a normally-closed output, so make sure to check the specifications. This switch is a good example of one with both, and it even has nice markings!



In this case, we connect the COM (common) pin to ground, then the normally-closed to pin #10 and the normally open to pin #0

With this setup, pin #10 is triggered normally, and when the lever is pressed, #10 is released and #0 is pressed. That way the #10 background loop stops immediately, and the #0 foreground sound starts immediately. When the lever is released, it goes back to having pin #10 connected

Here's a list of some of the switches we have at Adafruit that have a NC and NO output

- Micro Switch with Lever (http://adafru.it/819)(momentary)
- Micro Switch with Wire (http://adafru.it/820) (momentary)
- Micro Switch (http://adafru.it/817) (momentary)
- Weatherproof Pushbuttons (http://adafru.it/2093)(momentary)

We also have some buttons/switches that are not momentary, they latch on or off.

- Weatherproof Pushbuttons (http://adafru.it/482) (on/off)
- Slide switch (http://adafru.it/805) (on/off)

Creating Audio Files

Now you know it all works, you'll need to create your own audio files. The sound board does not support MP3 so if you have your audio clips in MP3 format you'll have to convert them to OGG or WAV

You can use either OGG or WAV format, and you can 'mix and match' so some files are OGG and some are WAV

There's some pro's and con's to the two formats:

- **OGG** is compressed, but still sounds great. It uses much less space so if you want to say store a full 2 or 3 minute song, you'll need to go with this. However, it does take a few milliseconds to start playing the file, so if you want to have perfectly looping audio, OGG will have a gap in between each play through
- **WAV** is uncompressed, so its the highest quality. But it takes up a lot of space. Since it's uncompressed, it starts playing instantly, and if you're looping an effect, this will have no discernable gap.

If you want to use your own sound files, you can! Record, sample, remix, or simply download files from a sound file sight, such as freesample.org. Then, to make sure you have the files converted to the proper specifications, check out this guide here (https://adafru.it/s8f) that'll show you how! Spoiler alert: you'll need to make a small, 22Khz (or lower), 16 bit PCM, mono .wav file!

There's tons of software that can generate OGG or WAV, we used this service and it worked very nicely without needing to install any software: http://audio.online-convert.com/convert-to-ogg (https://adafru.it/e0q)

If you don't mind installing software, Audacity can do a lot of conversion and is free software. (https://adafru.it/e0r)

Generating audio files, especially if you want to keep them small, can take a little experimentation: higher bit rates and sample rates will sound 'better' but take more space. You can go with 44.1KHz sample rate which is basically audio CD quality, or down to maybe 8KHz for spoken word or low-rez effects.

Another way you can save space is to convert stereo files to mono. The decoder supports stereo but if you only have a single speaker it doesn't matter if you have stereo output and stereo takes 2x as much space!

Use 8 or 16 bit rates for WAV, 24-bit does not apparently work

How Much Music?

We have SFX boards in both 2MB and 16MB versions. You may be thinking "Wow, that is nothing, I send emails bigger than that on a daily basis!" But you'd be surprised! For sound effects and small embedded audio devices, you need a surprisingly small amount of space.

Depending on whether or not you compress the audio, have stereo or mono, CD quality or 'every day' quality, you can store anywhere up to approx an hour of compressed music (mono 22KHz Ogg Vorbis on the 16MB version)

Compressed or No?

First up, remember you can always have compressed audio which saves you a ton of space. The Sound Board can handle Ogg Vorbis (a sort of royalty-free MP3 format) in any bit rate

Stereo or Mono

Most projects do not need true stereo, if you are only using one speaker or if you don't need two distinct channels, record/convert your audio as Mono! You'll get the same audio from both left and right channels and fit 2x as much music.

Bit Rates / Sample Rates

CD quality audio is 44.1KHz 16Bit, thats 44100 2-byte samples per second. While you can decode audio on the Sound Boards at that rate, often times your speakers or headphones aren't that good anyways.

We suggest sticking with 16 bits, but reducing the sample rate to 22KHz unless you are piping the audio into a quality stereo system or expect users to have fancy headphones plugged in. If you're doing 'voice' type effects, you may be able to go down to 11KHz. Every time you reduce the sample rate, you can double or triple the length of audio.

Some examples

Uncompressed WAV

The absolute biggest files you can generate and play are CD quality uncompressed stereo WAV files, at 44.KHz/16 bit.

Stereo WAV 44.1 KHz 16 Bit - (2 bytes * 2 channels * 44100) = ~175 KB per second, so 2MB can hold 12 seconds, 16MB can hold 90 seconds

Mono WAV 44.1 KHz 16 Bit - (2 bytes * 1 channels * 44100) = ~88 KB per second, so 2MB can hold 23 seconds, 16MB can hold 180 seconds (3 minutes)

Stereo WAV 22 KHz 16 Bit - (2 bytes * 2 channels * 22050) = ~88 KB per second, so 2MB can hold 23 seconds, 16MB can hold 180 seconds (3 minutes)

Mono WAV 22 KHz 16 Bit - (2 bytes * 1 channels * 22050) = ~44 KB per second, so 2MB can hold 45 seconds, 16MB can hold 6 minutes

Stereo WAV 11 KHz 16 Bit - (2 bytes * 2 channels * 11025) = ~44 KB per second, so 2MB can hold 45 seconds, 16MB can hold 6 minutes

Mono WAV 11 KHz 16 Bit - (2 bytes * 1 channels * 11025) = ~22 KB per second, so 2MB can hold 90 seconds, 16MB can hold 12 minutes

Compressed Ogg Vorbis

You can compress or convert any kind of file (like MP3) to Ogg Vorbis and its royalty free!

Since it's compressed, Ogg Vorbis does lose a little quality but its very unlikely that you will notice. The only real tradeoff for Ogg files is that the Codec is a little slower so it takes longer to play files, on the order of 50 milliseconds or so. In general this is not noticable, but if you are looping audio with a TnnHOLDL.ogg file, you'll notice a small gap.

Ogg Vorbis is about 1:5 to 1:10 compression over WAV files, so you can store 5-10x as long audio if you compress it to Ogg, compared to the numbers above

Troubleshooting

Here's some hints to help you with your sound board:

Doesn't show up as a drive on my computer

Check you have a data-capable USB cable, or try another cable just in case.

USB cable - USB A to Micro-B
This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...
https://www.adafruit.com/product/592

On some computers, give it a minute, the first time it sees the drive it may try to read data from it and its a bit slow (we saw this on Windows 7 in particular)

Make sure the green LED near the microUSB port is lit, that tells you it has power, however the red LED should not be lit

I can't get audio to play when I trigger the pins

Is the board connected to a computer USB port? You cannot use it as a USB storage device and play audio at the same time, even if you 'eject' the drive, you can only power it from battery or a power-only USB port like a USB wall adapter.

Do you see the red LED blink or turn on at all when you trigger the pin? You should see something blink/flicker when it plays audio.

Make sure nothing is connected to the CS or UG pins!

In case there's something amiss with your audio files, try using our audio file downloads, remember they go in the base directory of the USB drive, they cannot go in a folder or in a compressed zip!

To trigger, connect a wire from a GND pin to a numbered pin

With headphones in the headphone jack (or speakers connected), plug the board into USB. You should hear a beep/tone when it detects that it is connected to a computer

F.A.Q.

How come 'looping tracks' like TO1LATCH.WAV dont loop in UART mode?

UART mode is **completely manual** - only 'button mode' does the 'random/next/latch' behaviors. that's because if you are controlling via UART you can do whatever you like :)

I am having difficultly triggering GPIO #9 and #10

For reasons we haven't nailed down yet, Trigger pins #9 and #10 require more current to trigger (they are set to inputs in the firmware so that's not it) and might not be possible to do so from a logic output device. Using a switch ought to work!

Can I trigger using a Eurorack Output?

No, the board is set to trigger through switch closures. You could make a circuit to perhaps go from a Eurorack trigger to a switching trigger but that's beyond the scope of this guide.

If I hold down a button in 'play once' Tnn.WAV mode, it will repeat!

That's right, after it is done playing the sound it will check the buttons and find that the one you have held down is pressed so it will play it again - this is normal!

If you want to have it play a sound only once through, even if the button is held down, you can try making a sound with very long empty noise at the end, then use TnnHOLD mode so that it will play an empty noise. Then when the button is released and re-pressed it will start at the beginning. OGG type is best for this since it is compressed

I don't have short delays between loops, whats up with that?

The microconroller on board has to look through the entire filesystem every time it detects a button press, so the key to making loops short is keeping the filesystem clean and small.

Once you have your setup the way you want, make sure the SD card is empty, even reformat it with the 'blank' image if you've copy n pasted files around (https://adafru.it/rc4)! Use WAV format not OGG because WAV plays faster. Also, make sure you are using T00 slot since that is the first button checked.

Why is there a sharp popping sound when the board is first powered?

If you hear this, it is normal. The board does not include any anti-popping circuitry to prevent this initial transient.

Doesn't show up as a drive on my computer

Check you have a data-capable USB cable, or try another cable just in case

On some computers, give it a minute, the first time it sees the drive it may try to read data from it and its a bit slow (we saw this on Windows 7 in particular)

Make sure the green LED near the microUSB port is lit, that tells you it has power, however the red LED should not be lit

---

USB cable - USB A to Micro-B
This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...
https://www.adafruit.com/product/592

I can't get audio to play when I trigger the pins

Is the board connected to a computer USB port? You cannot use it as a USB storage device and play audio at the same time, even if you 'eject' the drive, you can only power it from battery or a power-only USB port like a USB wall adapter.

Do you see the red LED blink or turn on at all when you trigger the pin? You should see something blink/flicker when it plays audio.

Make sure nothing is connected to the CS or UG pins!

In case there's something amiss with your audio files, try using our audio file downloads, remember they go in the base directory of the USB drive, they cannot go in a folder or in a compressed zip!

To trigger, connect a wire from a GND pin to a numbered pin

With headphones in the headphone jack (or speakers connected), plug the board into USB. You should hear a beep/tone when it detects that it is connected to a computer

---

The ADAFRUITSFX folder available space is not showing full capacity?

Some OS's add hidden files for placing various meta information needed for their OS specific features. These may not show up by default and may be taking up space. Try turning on "Show hidden files" in your OS and look for these. Some common ones to look for are:

- **.Trashes** on MacOS

Downloads

Disk Images

If somehow your board file system gets messed up, you can use this image to reburn the original clean FAT12 filesystem on.

On windows:

- Use win32diskimager (https://adafru.it/caT)to write the image onto the board.
- Uncompress the Zip file and write the 2MB or 16MB file.

On mac os:

- Download the "disk image" for the 2MB or 16MB size below and uncompress
- Use the DiskUtiliity on my Mac to "restore" that FAT12 disk from the disk image. Caution: if you use the Disk Utility wrong, you can destroy your OS. **Make sure you restore the disk image to the correct device.** You may need to try 'restoring' twice

Download the compressed 2MB image

https://adafru.it/e7w

Download the compressed 16MB image

https://adafru.it/eDy

When burning an image to the Sound Board, unplug any and all external hard drives so you only have the internal computer hard drive and the Sound Board show up as disks, that will minimize the risk of accidentally destroying your external drive by formatting it

Datasheets & Files

- Datasheet for the VS1000 OGG/WAV decoder (https://adafru.it/e0y)
- EagleCAD PCB Files on GitHub (https://adafru.it/omF)
- Fritzing objects available in Adafruit Fritzing Library (https://adafru.it/aP3)

Firmware

We cannot release the source code (VLSI has it under a non-open license) but here are the binaries you can flash into the 2M/16M disks

This forum thread has some information on how to program the chips https://forums.adafruit.com/viewtopic.php?f=19&t=173168#p855405 (https://adafru.it/doW)

2M110514.BIN

https://adafru.it/EYb

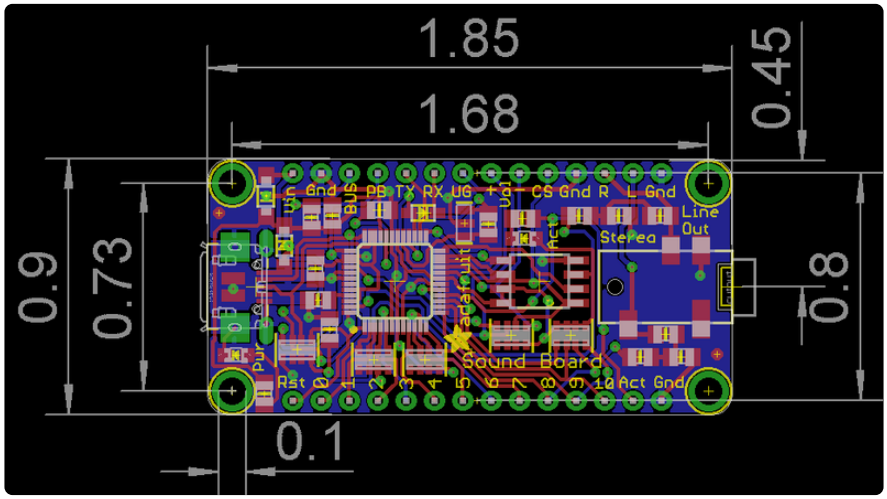
6M110614.BIN

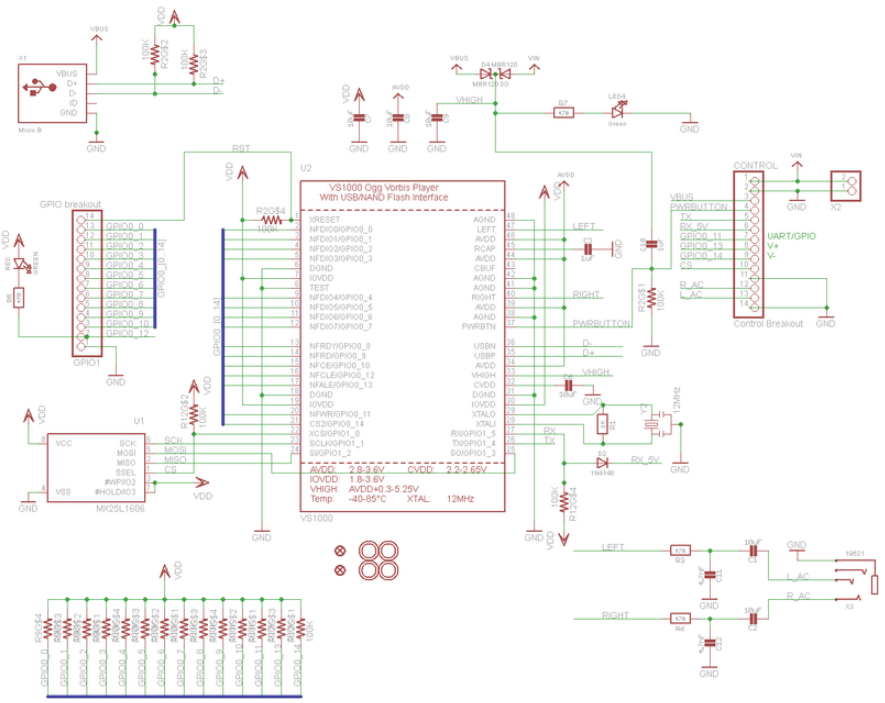
https://adafru.it/EYc

Schematic and Fabrication Print

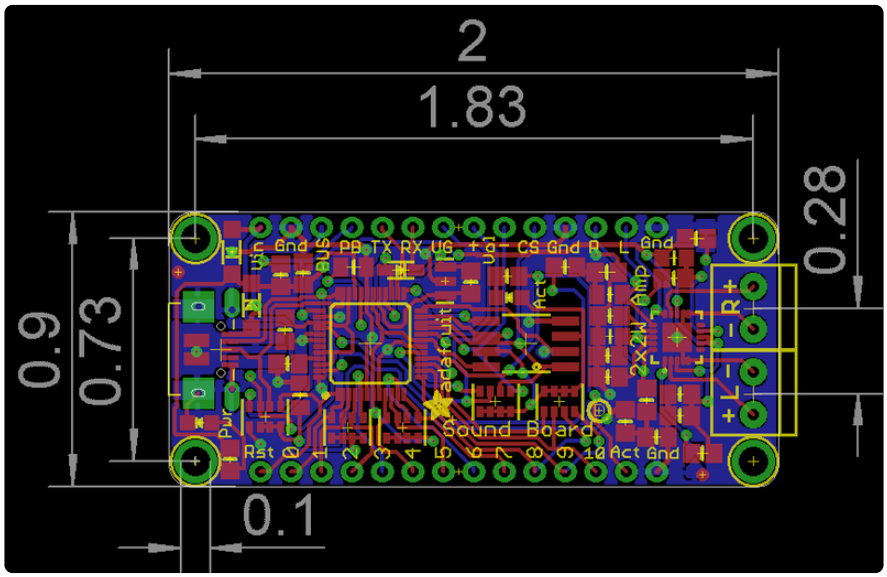
Dimensions in Inches

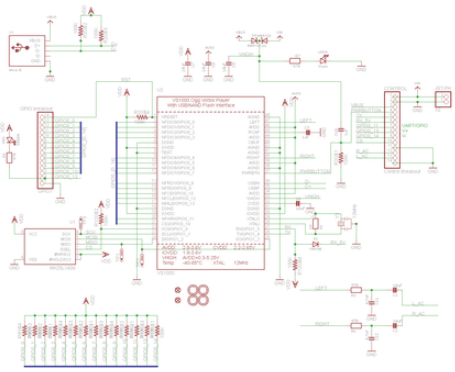
Headphone Out version

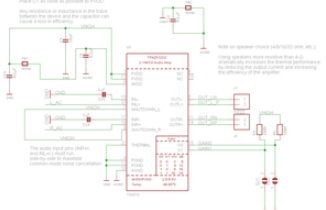
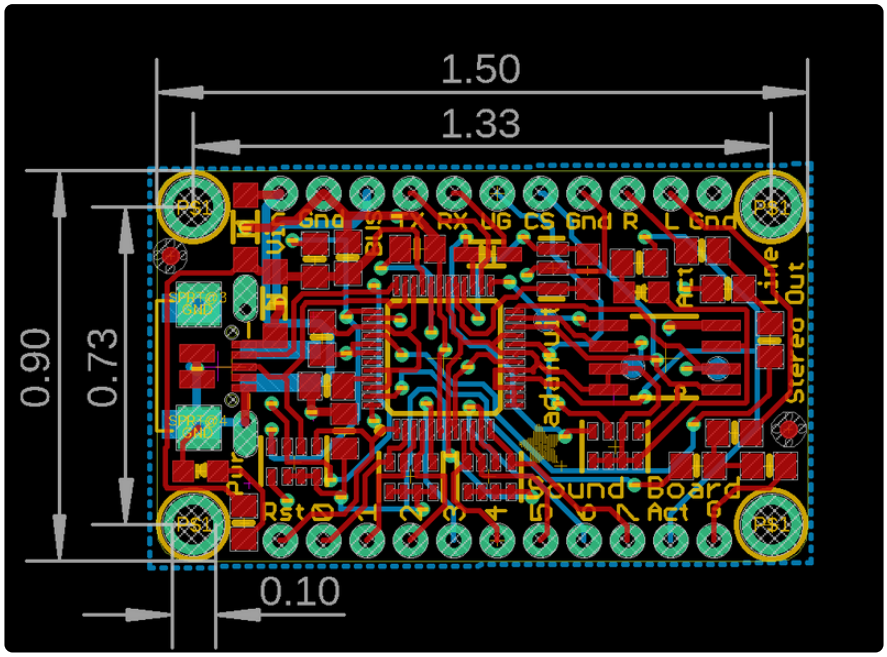
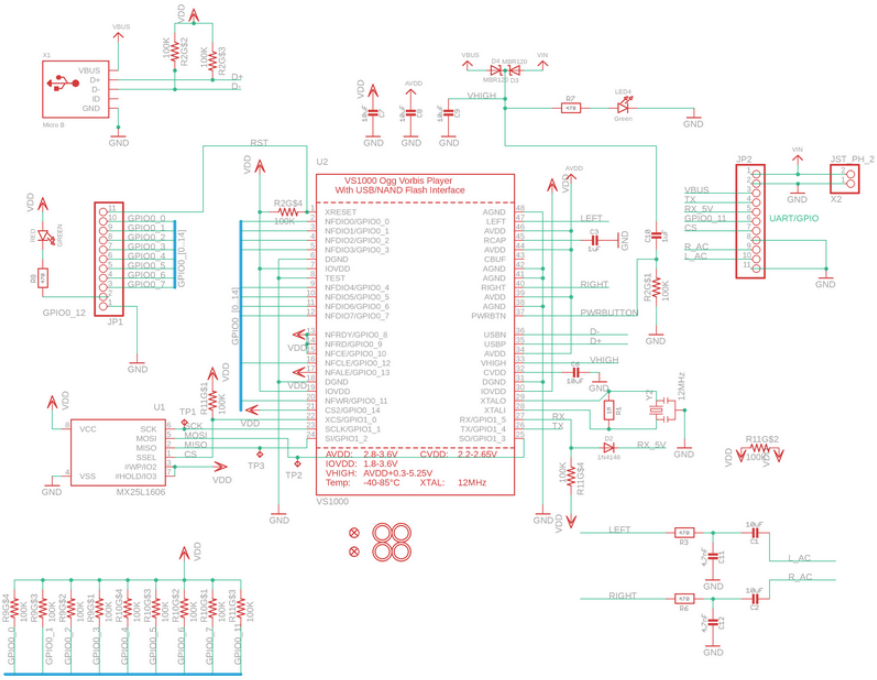




Amplifier Version







Mini Version