# Using a Sparkfun MP3 Trigger

The MP3 trigger is a board made by Sparkfun electronics that provides a way to play MP3 files from a Micro SD card via either one of 18 `TRIG` inputs on the board, or serial communication with the board.

The MP3 Trigger has a headphone output which can be connected to powered speakers, or to your headphones for testing.

Loading files

You must us a micro SDSC (up to 2GB) card, or a SDHC (up to 32GB) card formatted in FAT16 or FAT32. After inserting the card you must power cycle the MP3 Trigger so it detects the card, this can be done by switching the `USB <-> EXT` switch back and forth.

- **USB**: means power from the Arduino. This isn't done via a 'USB' connector, but instead by connecting the FTDI pins (see below). If you are connecting to the board using an arduino, use this option
- **EXT**: means power via the 2.1mm connector to the left of the switch. If you are only using trigger pins with no Arduino control, get an external power source and use this option.
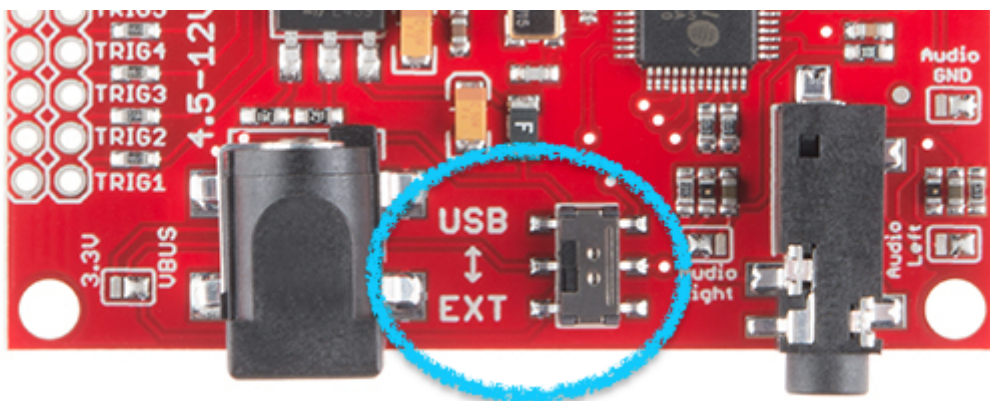
Play Files Using The Trigger Pins

File Naming

If you want to play tracks on the board using the trigger pins, files should be named 001.MP3 through 018.MP3, and the MP3 Trigger will match the file name to the value on the pressed `TRIG` pin.

If you have fewer than 18 files, you do not need to use all the pins -- just number as many files as you have, and trigger the pins accordingly. Hidden files are less likely to cause issues in this setup,

though if you are having problems it's worth checking the instructions below.

Wiring

As all the control takes place on the MP3 Trigger, you have the option of using an external power source, but you can also use an Arduino to supply a basic voltage over the FTDI pins (you just need to make sure you select the correct power input on the `USB <-> EXT` switch described above). If you do want to power the MP3 Trigger via Arduino, you need 2 wires:

1. Ground (GND on the on the MP3 Trigger connects to GND on the Arduino)
2. Power (VCC on the MP3 Trigger connects to 5V on the Arduino)
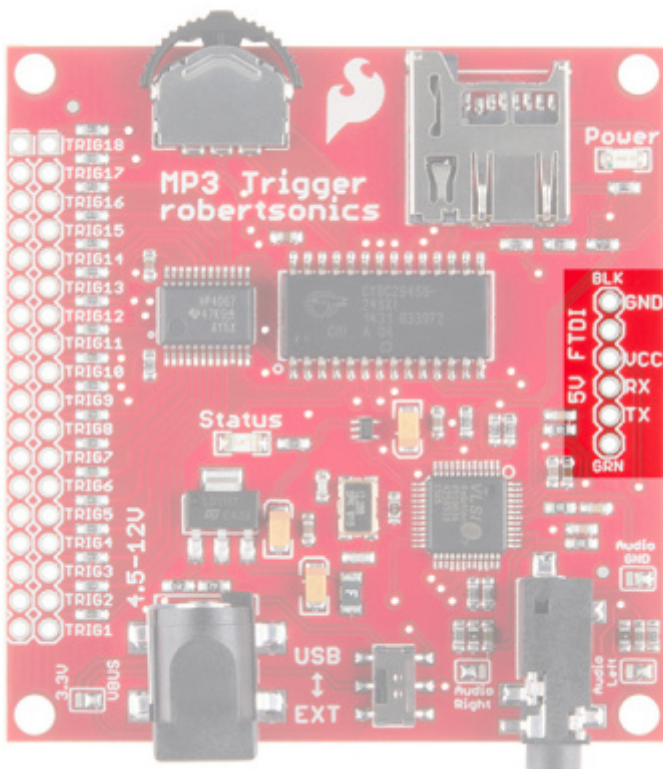
To test the trigger, take a jumper and try connecting together both pins of Trig1. The file with the name 001.mp3 should start to play. You can continue to test in this manner for all of your files.

If you would like to incorporate this triggering into a project, you can think about using switches to trigger this connection.

Play Files Using Serial Input

File naming

To control the MP3 Trigger using serial (as in the example below), you need to send it a track number to play. The command is sent in 2 parts: a 'p' which tells the board you want to play something, and then a number (start counting at 1). It looks like (in Leonardo code):

```
Serial1.print("p")
Serial1.write(1)
```

This will get the MP3 Trigger to play the 'first' track. Note that the order of tracks is *not* the alphabetical or numeric order, but instead the order that the files appear on the SD card. If you are having trouble getting tracks playing in the correct order, it's good to check this. This means that while you *can* name the files anything you like, it's good to have a clear order to allow you to keep track.

Important: If you are using a Mac to write the .mp3 files

Macs do a weird thing where every time you plug in a memory stick, it creates a bunch of hidden files. This shouldn't be an issue if you're playing tracks using the triggers (as those target files with particular names) but it causes a lot of issues if you're trying to play files using serial.

These files can't be removed in Finder, and instead need to be removed by opening up the volume using the terminal. You will need to repeat this step EVERY TIME you plug the SD card in (though you might only have to remove 1 or 2 files). Once you have finished copying your .mp3 files to the SD card, open up a mac terminal and type in the following:

```
$ cd /Volumes/<name-of-volume>  # this navigates to the correct volume
$ ls -a      # verify what's in the output of this
$ rm ._*.mp3    # this removes all of the thumbnails
$ rm -r .Spotlight-V100  # deletes the .Spotlight-V100 folder
$ rm -r .fseventsd    # deletes the .fseventsd folder
$ rm -r .Trashes     # deletes the .Trashes folder
$ ls -a      # this shows all the files you have left
```

> **Warning**
> Be *very* careful when you are running the `rm -r` command: this will recursively delete files and is NOT reversible. Only run it when you know you are in the correct folder, and make sure you know which folder you are deleting)

e.g. see below: I use `ls -a` multiple times to check what I still need to delete. At the end, all that's left is the .mp3s and the `.` and `..` shortcuts (these aren't file and can be ignored).

```
[acameron@MB-C1291 ~ % cd /Volumes/NO\ NAME
[acameron@MB-C1291 NO NAME % ls -a
.                ._1.mp3          ._4.mp3          2.mp3            5.mp3
..               ._2.mp3          ._5.mp3          3.mp3
.Spotlight-V100 ._3.mp3          1.mp3            4.mp3
[acameron@MB-C1291 NO NAME % rm ._*
[acameron@MB-C1291 NO NAME % ls -a
.                .Spotlight-V100 2.mp3            4.mp3
..               1.mp3           3.mp3            5.mp3
[acameron@MB-C1291 NO NAME % rm -r .Spotlight-V100
[acameron@MB-C1291 NO NAME % ls -a
.        ..      1.mp3    2.mp3    3.mp3    4.mp3    5.mp3
[acameron@MB-C1291 NO NAME % cd ~
acameron@MB-C1291 ~ %
```

Wiring

To connect the board to an Arduino, you need to connect the FTDI pins. This requires only 3 wires.
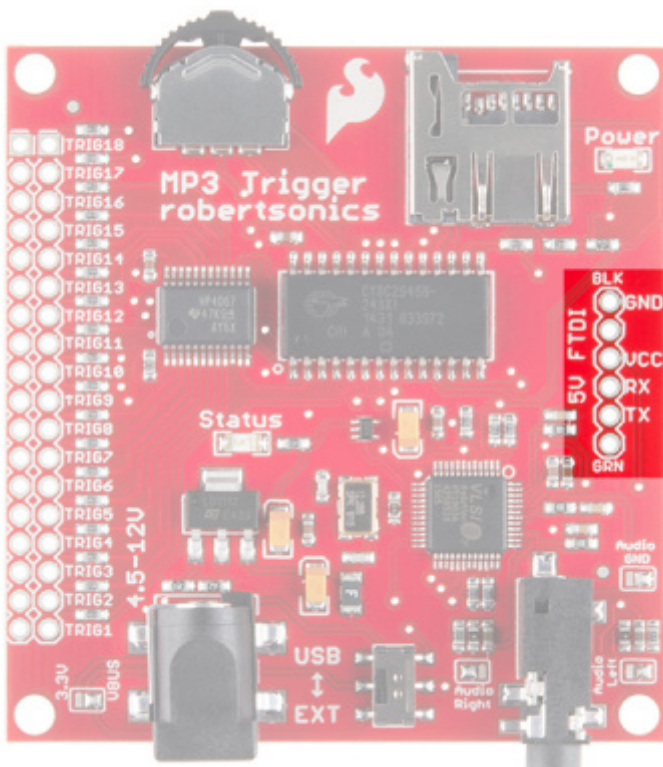
1. Ground (GND on the on the MP3 Trigger connects to GND on the Arduino)
2. Power (VCC on the MP3 Trigger connects to 5V on the Arduino)
3. Data (RX on the MP3 Trigger to TX on the Arduino)
4. *Optionally you can connect the TX on the MP3 Trigger back to the RX on the Arduino if you wish to get playback status information.*

Basic Serial Input Example

There are libraries available for the Sparkfun MP3 Trigger, but as it's very simple, it's easier to use `Serial.print` (or `Serial1.print` on Leonardo) to control it rather than a library. This Arduino code example will play 1.mp3 - 5.mp3 from the SD card with a 1 second delay between playing each.

- Sketch: Basic example - Arduino Uno

- Sketch: Basic example - Arduino Leonardo

> **Warning**
> Arduino Leonardo, unlike Arduino Uno has two serial ports. This tutorial contains code for both cases. The core difference is, if you want to use Arduino Leonardo you'll need to use `Serial1` which is the RX/TX pins on pins 0 and 1, rather than `Serial` which is the USB serial monitor.

Read more

Sample MP3 files

To help you get up and running quickly there are 5 example mp3's you can use with the basic example of Tom saying 1-5. If the files are getting played in a weird order, or some of them aren't playing, it's likely that you are having issues with hidden files. Please see the instructions above for more details.

Example MP3 Files

Convert your audio files to work with the MP3 trigger

Once you have the example files working, you probably want to add your own audio files. We discovered they work best with audio files in the MP3 format, at between 4800-9200kHz. Your files may already be another format, so we need to convert them for the trigger to read them properly.

There are a few programs with GUIs that do this, but this is also a great opportunity to learn about a command line program called `ffmpeg`, which is a very powerful program that converts media between formats.

The full instructions are here: https://www.ffmpeg.org/ffmpeg.html. If you don't have ffmpeg installed, you can usually install it according to the instructions here: https://ffmpeg.org/download.html. To check whether you have it installed, open a terminal and try typing `ffmpeg` -- it should print out a help page.

Ffmpeg takes an input file (preceded with '-i') and and output file (sometimes preceded with -o, but not always needed) and then you put in the settings you want it to use to do the conversion. So a simple line of code can change your MP4, ACC etc to our desired MP3 format:

```
ffmpeg -i filename.mp4 -b:a 92K filename.mp3
```

Resources

- [Sparkfun Hookup Guide](#) -> this has lots more details about other features on the board, and how file naming works

---

Revision #16
Created 10 July 2021 23:02:51 by Tom Lynch
Updated 29 April 2024 22:24:58 by agnes cameron