

For a thorough introduction and “how-to” for the WAV Trigger, please consider following the excellent [Hook-up Guide on the SparkFun website](#).

This document details the specifications and features of the latest version of the WAV Trigger and its firmware as of 1/14/2023. Some features described herein may not be supported in earlier firmware versions, so you are encouraged to update your board to the latest version.

## Powering the WAV Trigger

The WAV Trigger is typically powered through the on-board 5.5x2.1mm barrel jack. The allowable input voltage range is 6 to 12VDC, and the jack is center-positive. The resulting on-board regulated 3.3V and 5V supply voltages are available to power small external circuits and sensors via J8, pins 3 and 4.

You can also power the WAV Trigger by supplying a regulated 5V to J8, pin 4, provided you are not using the barrel connector. Keep in mind that the WAV Trigger can draw over 200mA if you enable the on-board audio amplifier and drive the speaker at high volume.

In addition, you can also power the WAV Trigger directly from an FTDI Basic 5V by installing SJ2 (solder jumper.) **IMPORTANT:** If you install SJ2 you should not connect the FTDI basic 5V while also powering the WAV Trigger from the barrel connector.

At 9VDC, the WAV Trigger draws about 80mA when idle and on average about 110mA when playing tracks. If you enable the on-board audio amp and drive the speaker at high volumes, the WAV Trigger can momentarily draw over 200mA. Audio power is not free!

## Getting Started

The WAV Trigger plays audio from uncompressed WAV files on a FAT16 or FAT32 formatted microSD card of any size up to 32GB.

Note: Cards larger than 32GB often come formatted as XFAT, which the WAV Trigger does not support. These cards can be used by reformatting as FAT32, but this will limit the usable size to 32GB.

The WAV file format currently supported is 16-bit, 44.1kHz, stereo, which is the same bit depth and sample rate as audio CDs. Audio in other formats (including MP3) can easily be converted using almost any audio editor, including the free, open-source Audacity. While the WAV Trigger file format is capable of "CD quality", keep in mind that your audio will only be as good as your

source material. Converting a 128kbps MP3 file to a 16-bit, 44.1kHz WAV file will not make it sound any better than 128kbps MP3. On the other hand, if your source material is from an audio CD, it will sound just as good as the CD coming out of the WAV Trigger.

To be recognized by the WAV Trigger, WAV files must be named starting with a 3 or 4-digit number, which becomes the "track" number. The remainder of the file name doesn't matter, and it can be as long as you wish. Both "001Track.wav" and "0403\_Sneezing\_Elephant.wav" are both valid file names. The WAV Trigger recognizes "0001..." through "4096..." as valid track numbers. For the remainder of this document, the word "track" will be used to refer to an appropriately named WAV file.

All tracks must be located in the root directory of the microSD card. **The card must be formatted either FAT16 or FAT32, and it should be formatted using 32k allocation size. Using a smaller allocation size may introduce clicks and pops in the audio playback.**

Any time the microSD card is removed and re-inserted, the WAV Trigger must be power cycled. This is because it reads the init file (see below) and indexes all the tracks only during a power-on reset.

The WAV Trigger has an on-board, momentary push-button that can be used to quickly test basic functionality. The default behavior for this button is to play the lowest number track found on the microSD card.

## Polyphony and why it's important to understand

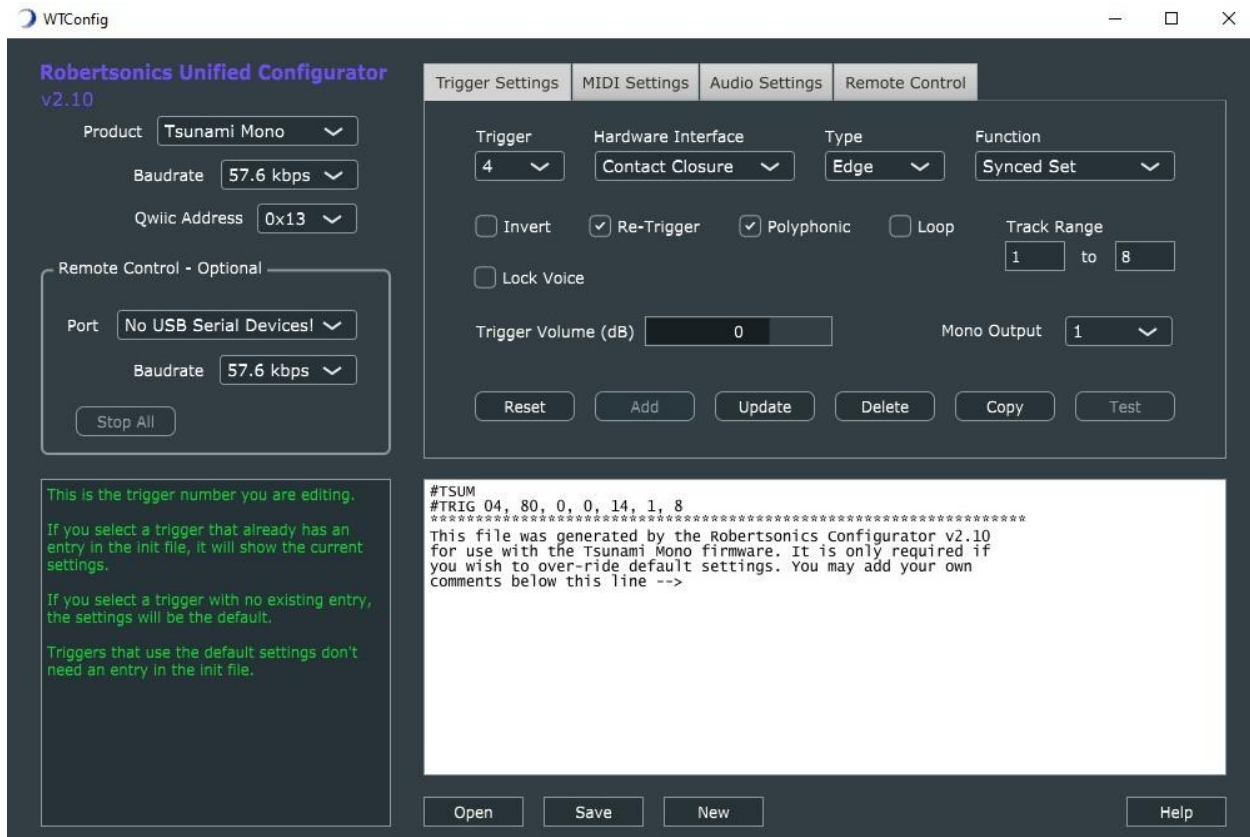
Unlike most other embedded audio players, the WAV Trigger is "polyphonic". This means that it can play more than one track at a time. It does this by maintaining multiple file pointers and digitally mixing the tracks to the WAV Trigger's stereo output. When operating in polyphonic mode, tracks can be started at any time, regardless of whatever tracks may be playing, including other copies of the same track (depending on the settings). The only limit is the number of "voices" - the number of separate tracks that can be played at any given moment. Currently, the WAV Trigger provides 14 low-latency stereo voices.

Polyphony can make things quite complex, and it's important to understand the various options so that you can achieve the results you want.

Polyphony also affects how you handle volume. If you mix many tracks together that are all full-scale volume, the result will be clipped and distorted. In this case, it won't help to lower the WAV Trigger gain setting since this only affects the final output stage - you'll just lower the volume of an already clipped mix. You must reduce the volume of the WAV files themselves so that the resulting addition of the tracks does not clip.

## The Init file and the Robertsonics Configurator Utility app

The WAV Trigger has a number of options that can be set through the use of an Init file - a text file that is placed on the microSD card along with the WAV files. Robertsonics provides a utility application that allows you to set these options using a graphical user interface, and that will then create the Init file for you. The Init file is still a text file and can be created and/or edited by hand, but the Configurator app makes this process much easier and less prone to error. (The same application is used to set the options of our multi-channel Tsunami player as well.)



To be recognized by the WAV Trigger, the Init file must be named "wavtrigr.ini" and be placed in the root directory of the microSD card.

An Init file is only required if you want to change a default behavior, and the only entries that are required are for options that are not default.

The following discussion of both global and trigger options assumes that you will be using the Configurator app to create an Init file.

## Global options

**Serial Baudrate:** This sets the baudrate for the async serial port. This baudrate will remain in effect until the next power cycle. The default is 57.6kbps.

**Initial Volume:** This gain will be applied to the final mix output of the WAV Trigger, until overridden by a volume command. The range is -70dB to +10dB, and the default value is 0dB (unity gain). This gain affects the final output mix of the WAV Trigger, not the individual tracks that make up the mix.

**Audio Amp Power:** This enables or disables the on-board mono audio amplifier. If you plan to use the speaker connection, check this option. If you plan to use the stereo line-level output, leave this option unchecked to save power. Important: The default setting is disabled.

## Trigger options

Trigger options are set by first selecting the trigger number in the WT-InitMaker app. If there's already an entry for that trigger in the Init file, the options will indicate the current values for that entry. If there is no entry for that trigger, the options will reset to default. Using the "Save" or "Update" buttons will create or update an entry in the Init file for that trigger. Once saved, trigger settings may be transferred to the WAV Trigger over the serial port (if it is connected) using the "Test" button. This allows you to easily test the effect of changing options without having to deal with writing the init file, transferring the microSD card and power cycling the WAV Trigger.

Not all trigger functions support all options. For example, the Next, Previous and Random functions are inherently edge triggered type functions, so the "Type" field defaults to "Edge" and is disabled for these functions.

**Hardware Interface:** The "Contact Closure" option enables an internal pull-up for the trigger input so that with nothing connected, the trigger input is "high". When combined with the non-inverted (default) option, a contact closure (button or switch) to ground will pull the trigger input low and activate the trigger. The "Active - 3.3V/5V" options disables the internal pull-up for the trigger, requiring either an external 3.3V or 5.0V signal to bring the input "high". This allows the trigger to be driven with the digital output of another micro controller. The invert option will control whether the trigger will be activated on a low or high signal level.

Please note that you should never select "Active - 3.3V/5V" option without having an active signal connected to the trigger. With nothing connected, the trigger input is floating and will produce false and random triggers. Similarly, you should never apply 5V to a trigger without first setting it to "Active - 3.3V/5V".

**Type:** This option affects only certain trigger functions. For the "Normal" trigger function, "Edge Triggered" starts the track when the trigger first goes active, and the track continues to play until stopped by some other event. "Level Triggered" starts the track when the trigger goes active, but the track plays only as long as the trigger remains active. When combined with the "Stop" function, "Level Triggered" will prevent one or more tracks from starting as long as it remains active, creating an "inhibit" function. Beginning with firmware v1.20, "Latched" will activate on a transition but remain active as if it was level type. Another trigger is required to stop any action that is started by a "Latched" type trigger.

**Function:** This option selects the function that the trigger will perform. Some of the functions require that a range of tracks be specified, and the values in the range can be from 1 to 999. The high value must be equal to or greater than the low value.

"Normal", the default function, directs the WAV Trigger to play the corresponding track number. The track range is not used. Combining with the "Level Triggered" type option will make the trigger act like a MIDI Note On, causing the track to play only as long as the trigger remains active.

"Next" plays the next track in the specified range, starting with the low track and wrapping back to that track after playing the high track in the range.

"Previous" plays the previous track in the specified range, starting with the highest track and wrapping back to that track after playing the lowest track in the range.

"Random" plays a random track from the specified range. The only rule imposed is that it will not play the same track twice in a row.

"Pause" will pause all the tracks in the specified range. These tracks will remain paused until they are resumed or restarted. Combining with the "Level Triggered" type option creates a momentary Pause feature; the specified tracks will resume when the trigger is released.

"Resume" will resume all the tracks in the specified range, provided that they are still "paused".

"Stop" will stop all the tracks in the specified range. Combining with the "Level Triggered" type option will create a "Stop and Inhibit" function, preventing all the tracks in the range from restarting as long as the trigger remains active. (Triggers that would have otherwise started an "inhibited" track are ignored, not queued.)

"Volume Up" or "Volume Dn" will re-purpose the trigger to be a volume control. Each activation of the trigger will increase or decrease the output stage volume of the WAV Trigger by 1dB.

"MIDI Bank Up" or MIDI Bank Dn" will re-purpose the trigger to increment or decrement the MIDI Bank number. Each MIDI Bank maps MIDI note numbers 1 through 128 to the next bank of 128 track numbers.

"Trigger Bank Up" or Trigger Bank Dn" will re-purpose the trigger to increment or decrement the Trigger Bank number. Each Trigger Bank maps the normal function triggers to the next bank of 16 track numbers.

**Invert:** This check box will invert the logic of the trigger. The default is active low. Checking this option makes the trigger active high.

**Re-Triggers:** This option enables certain trigger functions to work again immediately while the track(s) that they may have previously started is (are) still playing. For example, with the "Normal" trigger function, enabling this option allows the trigger restart the corresponding track even if it has not finished playing, while disabling this option prevents the trigger from working until the track has finished playing (or is stopped by some other event.) Similarly, for the "Next", "Previous" and "Random" functions, disabling this option prevents the trigger from working until whatever track was previously started from this trigger is finished.

Re-Triggers is enabled by default.

**Polyphonic:** This option specifies that the track played by the trigger function (for those functions that play tracks) should be mixed with any track(s) currently playing. When this option is disabled, activating the trigger will stop any and all tracks that are currently playing before the new track is started.

**Loop:** This option, when enabled, will cause the track to loop continuously regardless of the state of any trigger which may have started it. Please note that the WAV Trigger does not implement truly seamless looping - there will always be a small "gap" of silence at the loop point.

**Lock:** This option allows you to specify that the track will not be subject to the WAV Trigger's voice stealing algorithm. This allows you, for example, to use a trigger to start a backing track, and play MIDI notes on top of it without risk of the backing track being stopped because you played too many notes for the available number of voices.

## Serial Control Protocol

The bidirectional asynchronous serial port can be used to control and monitor the WAV Trigger's operation. The default baudrate is 57.6kbps, but this can be changed by placing the appropriate entry in the init file using the Configurator app.

All of these commands are implemented in the [WAV Trigger Serial Control Arduino Library](#), so you need not know these details if you plan to control your WAV Trigger with an Arduino compatible board. Furthermore, if you do need to implement your own serial control on a non-Arduino board and you are using C or C++, then the Arduino library source code can be used as a template.

All messages **to** the WAV Trigger start with the same 2-byte Start-of-Message (SOM) header, followed by a byte indicating the total number of bytes in the message. Next is a 1-byte message code in the range of 0x00 to 0x7f, followed by a number of bytes of data specific to the message. Immediately following the last data byte is an End-of-Message (EOM) byte.

All message **from** the WAV Trigger are formatted the same way. The only difference is that the message codes are in the range of 0x80 to 0xff.

16-bit data values such as track number and volume are sent "little-endian", that is with the LSB first and the MSB second.

Message format: ( SOM1, SOM2, length, message code, data \* n , EOM ), where SOM1=0xf0, SOM2=0xaa, EOM=0x55

Commands sent TO the WAV Trigger:

Command	Code	Length	Data Bytes(0)	Response
GET_VERSION	0x01	5	none	VERSION_STRING

Requests the WAV Trigger to transmit the VERSION\_STRING message

*Example: 0xf0, 0xaa, 0x05, 0x01, 0x55*

Command	Code	Length	Data Bytes(0)	Response
GET_SYS_INFO	0x02	5	none	SYS_INFO

Requests the WAV Trigger to transmit the SYS\_INFO message

*Example: 0xf0, 0xaa, 0x05, 0x02, 0x55*

Command	Code	Length	Data Bytes (1)	Response
SET_REPORTING	0x0e	6	Reporting State = 0/1	none

Enable/Disable asynchronous serial reporting of track status. See TRACK\_REPORT in the next section

*Example: 0xf0, 0xaa, 0x06, 0x0e, 0x01, 0x55* Enables reporting

Command	Code	Length	Data Bytes (0)	Response
GET_STATUS	0x07	5	none	STATUS

Requests the WAV Trigger to transmit the STATUS message

*Example: 0xf0, 0xaa, 0x05, 0x07, 0x55*



Command	Code	Length	Data Bytes (4)	Response
CONTROL_TRACK	0x0d	9	Control Code (see below)	none
			Track number LSB	
			Track number MSB	
			Lock Flag = 0/1	

Sends one of the following Track Control Codes to a specific track number:

PLAY\_SOLO = 0x00: Play track without polyphony, stops all other tracks

PLAY\_POLY = 0x01: Play track polyphonically

PAUSE = 0x02: Pause track

RESUME = 0x03: Resume track

STOP = 0x04: Stop track

LOOP\_ON = 0x05: Set the track loop flag

LOOP\_OFF = 0x06: Clear the track loop flag

LOAD = 0x07: Load and pause track

Lock Flag = True (1) prevents the track's voice from being stolen

*Example: 0xf0, 0xaa, 0x09, 0x0d, 0x01, 0x07, 0x00, 0x01, 0x55*

Plays track 7 polyphonically with voice locked

Command	Code	Length	Data Bytes(0)	Response
STOP_ALL	0x04	5	none	none

Commands the WAV Trigger to stop all tracks immediately

*Example: 0xf0, 0xaa, 0x05, 0x04, 0x55*

Command	Code	Length	Data Bytes(0)	Response
RESUME_ALL_SYNC	0x0b	5	none	none

Commands the WAV Trigger to resume all paused tracks in sync. Can be used with the CONTROL\_TRACK / LOAD command to start multiple tracks in sample sync.

*Example: 0xf0, 0xaa, 0x05, 0x0b, 0x55*

Command	Code	Length	Data Bytes (2)	Response
VOLUME	0x05	7	Gain LSB	none
			Gain MSB	

Updates the output volume of the WAV Trigger with the specified gain. Gain value is a signed 16-bit integer, -70dB to +10dB

*Example: 0xf0, 0xaa, 0x07, 0x05, 0xf6, 0xff, 0x55* Sets output gain to -10dB

Command	Code	Length	Data Bytes (2)	Response
SAMPLERATE	0x0c	7	Offset LSB	none
			Offset MSB	

Increases or decreases the WAV Trigger's playback speed for all tracks. Offset value is a signed 16-bit integer, -32767 to +32767.

*Example: 0xf0, 0xaa, 0x07, 0x0c, 0x5a, 0x1f, 0x55* Increases pitch 2 semitones.

Command	Code	Length	Data Bytes (1)	Response
SET_TRIGGER_BANK	0x0f	6	Bank (1 – 32)	none

Sets the trigger bank to the specified value.

*Example: 0xf0, 0xaa, 0x06, 0x0f, 0x08, 0x55* Set the trigger bank to 8

Command	Code	Length	Data Bytes (4)	Response
TRACK_VOLUME	0x08	9	Track number LSB	none
			Track number MSB	
			Gain LSB	
			Gain MSB	

Updates the gain of a specific track with the specified gain. Gain value is a signed 16-bit integer, -70dB to +10dB. The new gain becomes effective immediately.

*Example: 0xf0, 0xaa, 0x09, 0x08, 0x01, 0x00, 0xf6, 0xff 0x55* Set track 1 gain to -10dB

Command	Code	Length	Data Bytes (1)	Response
AMP_POWER	0x09	6	Off/On (0/1)	none

Turns On/Off the on-board audio amplifier.

*Example: 0xf0, 0xaa, 0x06, 0x09, 0x01, 0x55* Turns the audio amp On

Command	Code	Length	Data Bytes (7)	Response
TRACK_FADE	0x0a	12	Track number LSB	none
			Track number MSB	
			Target Gain LSB	
			Target Gain MSB	
			Milliseconds LSB	
			Milliseconds MSB	
			Stop Flag (0/1)	

Fade (up or down) a track to a target gain from the current level in the specified number of milliseconds, with an option to stop the track at the end of the fade.

Target gain is specified in dB and range is from -70 to +10. If Stop Flag = 1, the track will stop at the end of the fade. This is most useful for implementing a fade out, since the WAV Trigger will automatically stop the track and release the voice at the end of the fade out.

*Example: 0xf0, 0xaa, 0x0c, 0x0a, 0x02, 0x00, 0xc8, 0xff, 0xe8, 0x03, 0x01, 0x55*

Fade track 2 to -50dB in 1000ms and stop.

Commands sent FROM the WAV Trigger:

Command	Code	Length	Data Bytes (22)
VERSION_STRING	0x81	27	ASCII Version String

Sent in response to a GET\_VERSION command. Contains the current WAV Trigger firmware version string.

*Example: 0xf0, 0xaa, 0x1b, 0x81, "WAV Trigger v1.34 ", 0x55*

Command	Code	Length	Data Bytes (3)
SYS_INFO	0x82	8	Number of voices
			Number of Tracks LSB
			Number of Tracks MSB

Sent in response to a GET\_SYSINFO command. Contains the maximum number of voices in the current firmware and the total number of tracks found on the installed SD card.

*Example: 0xf0, 0xaa, 0x08, 0x82, 0x0e, 0xc4, 0x09, 0x55*

The WAV Trigger reports 14 Voices and 2500 Tracks found on SD card.

Command	Code	Length	Data Bytes (2 * No of Tracks)
STATUS	0x83	Variable	Track n Number LSB
			Track n Number MSB

Sent in response to a GET\_STATUS command. The data is a list of 2-byte track numbers that are currently playing. If there are no tracks playing, the number of data bytes will be 0.

*Example: 0xf0, 0xaa, 0x09, 0x83, 0x01, 0x00, 0x0e, 0x00, 0x55*

The WAV Trigger reports 14 tracks 1 and 15 are currently playing.

Command	Code	Length	Data Bytes (3)
TRACK_REPORT	0x84	9	Track Number LSB
			Track Number MSB
			Voice Number
			State (0/1)

When Reporting is set to On, this message is sent anytime a track starts or stops. State = 1 means Track started using Voice, State = 0, means Track stopped, voice released.

*Example: 0xf0, 0xaa, 0x09, 0x82, 0x0a, 0x00, 0x09, 0x01, 0x55*

Track 10 started using Voice 9

## MIDI implementation

The WAV Trigger can be programmed to treat data arriving on the serial control interface as MIDI. When you use the Configurator app to enable the “MIDI” serial baudrate option, the baudrate is automatically set to 31.250kbps. You’ll need a MIDI-to-TTL adaptor circuit to convert the MIDI 20mA current loop signal to the logic level required by the WAV Trigger’s serial port.

MIDI Note-On messages start tracks and Note-Off stop tracks, with optional velocity sensitivity. The WAV Trigger supports an internal MIDI “Bank” Number, 0 through 32. The default value is 0. When a MIDI Note-On message is received the corresponding track number is calculated as follows:

Track Number = (MIDI Bank Number \* 128) + MIDI Note Number.

When enabled, MIDI velocity determines the track volume, making the WAV Trigger velocity sensitive.

MIDI Controller number 7 (Volume) controls the WAV Trigger output volume.

MIDI Controller number 73 (Attack) controls the WAV Trigger note-on attack time (0 to 2000 msecs).

MIDI Controller number 72 (Release) controls the WAV Trigger note-off release time (0 to 2000 msecs).

The MIDI Bank Number can be incremented and decremented with triggers using the “MIDI Bank Up” and “MIDI Bank Dn” trigger functions.

The WAV Trigger recognizes MIDI Program Change messages, allowing you to change bank numbers using your controllers Program Change button or knob.

The WAV Trigger also supports proper use of MIDI Channel/Omni modes. When set to a specific MIDI Channel (via the init file), the WAV Trigger will only respond to the specified MIDI channel. When set to Omni, the WAV Trigger will respond to any MIDI Channel (MIDI Channel ignored.)

## Updating the Firmware

The WAV Trigger contains a serial bootloader, enabling firmware updates using the cross-platform WT-Flasher utility found on the product [download page](#). The serial connector on the WAV Trigger is pin-compatible with the SparkFun FTDI Basic, making it easy to get your computer connected. You may also use any USB-to-serial adaptor, so long as it is a TTL-level device, not RS-232. **Plugging RS-232 level signals into the WAV Trigger serial interface will damage the WAV Trigger!**

With the power to the WAV Trigger off, set the small switch near the power connector to the “load” position. Connect a USB serial interface to your computer, run the WT-Flasher app, then navigate to and select the desired firmware hex file. Connect the WAV Trigger to the USB serial adaptor and turn on the power to the WAV Trigger. Immediately press the WT-Flasher “Program” button. The app should establish communication with the bootloader and begin programming the new firmware.

When the programming is complete, turn off the power to the WAV Trigger and return the switch to the “Run” position. The next time you apply power to the WAV Trigger it should now be running the new firmware.

## Troubleshooting

If you have problems, here are a few things to check:

- 1) Make sure your WAV files are named properly (refer to the above) and are located in the root directory of the microSD card. When you first power on the WAV Trigger, it will locate and index the files and then blink 3 times to indicate that it found at least one WAV file. One long blink means that it either couldn't read the card or it didn't find any WAV files.
- 2) If there's no activity on the status LED whatsoever when you power up the WAV Trigger, check the “Load/Run” switch next to the power connector and make sure it's in the “Run” position.

3) Once you get the 3 blinks, pressing the on-board button will play the lowest numbered track on the microSD card. Confirm this by observing the status LED turning on to indicate a track is playing. You should hear the track through the stereo jack. If you're using the on-board audio amp and speaker connection, you won't hear anything unless you have enabled the amp using the init file (refer to the above.)

4) If you've gotten this far and tracks won't play, then it's most likely that your WAV files are not the correct format. **The WAV Trigger will only play WAV files formatted as 16-bit, stereo, 44.1kHz, and there can be no meta-data (non-audio data) in the file before the audio data.** It seems that many Mac audio applications insert a meta-data chunk prior to the audio, and this will prevent the track from playing. In such cases, simply importing the file and re-exporting with [Audacity](#) will likely fix the problem. See [this video](#) about converting files from MP3 to WAV Trigger compatible files, but the same process works with non-conforming WAV Files. You can always grab some of the (correctly formatted) WAV files from the [SparkFun Hookup Guide](#) to see if that's your problem.

5) If you're planning to use the on-board amplifier, [read this first](#). Next, please consider using a supply voltage of no more than 9V, preferably less. Playing loud audio with the on-board amp for extended periods can draw a lot of current and the lower the input voltage, the less warm the 5V regulator will get. This is especially important if you're encasing the WAV Trigger in a small project box. I personally use 6V wall mount adaptors like [this](#). Alternatively, you can power the WAV Trigger with regulated 5V (5Vin) and bypass the on-board regulator completely.