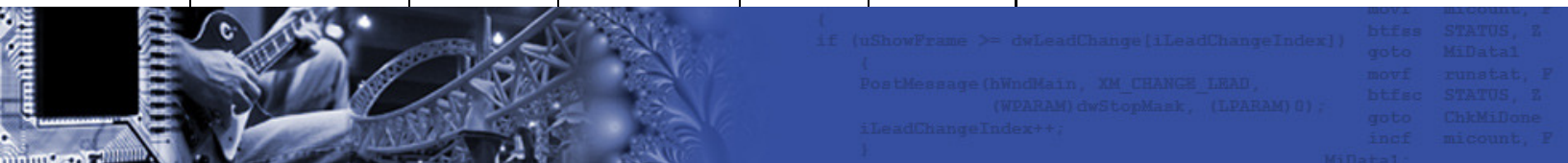


[Products](#)[Resources](#)[Music](#)[Projects](#)[Blog](#)[About](#)

## WAV Trigger

[➔ About](#)[➔ Buy](#)[➔ Downloads](#)[➔ Support](#)

## MP3 Trigger

[➔ About](#)[➔ Buy](#)[➔ Downloads](#)[➔ Support](#)

## SparkFun Electronics News Posts

- [New Tutorials To Get Your Learn On!](#) April 16, 2014

If you haven't been over to our education site in a while - well, you're missing out! Our Department of Education has been hard at work on new workshops, resources and more. We've also revamped our entire tutorial system to make it more user friendly and

## WAV Trigger Online User Guide

- [1. Powering the WAV Trigger](#)
- [2. The connectors](#)
- [3. Getting started](#)
- [4. Polyphony and why it's important to understand](#)
- [5. Serial communication and the apps](#)
- [6. The Init file and the Init File Maker utility app](#)
- [7. Global options](#)
- [8. Trigger options](#)
- [9. Serial control protocol](#)
- [10. Updating the firmware](#)

[Back to the top](#)

## Powering the WAV Trigger

The WAV Trigger is typically powered through the on-board 5.5×2.1mm 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.

easier to find the topic you are interested in. Today, we want to draw your attention to a few new tutori [...]

## **adafruit industries** blog

- [Charlie and The 3D Egg \(VIDEO\) #3DxArt #3DThursday #3DPrinting](#)

Charlie and The 3D Egg (VIDEO): This is the story of how one bored chick named Charlie learned how to 3D-print his own eggs using the new 3D printing capabilities in Photoshop CC; and how you could win your own exclusive egg (designed and printed by Charlie) by visiting our pop-up studio in East London [...]

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!

[Back to the top](#)

## Getting started

The WAV Trigger plays audio from uncompressed .wav files on a FAT16 or FAT32 formatted microSD card of any size. The .wav 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 .wav 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 16-bit, 44.1kHz .wav 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-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 “403\_Sneezing\_Elephant.wav” are both valid file names. The WAV Trigger currently recognizes “001...” through “999...” 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.

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.

[Back to the top](#)

## 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 8 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 4 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.

[Back to the top](#)

## **Serial communication and the apps**

The WAV Trigger has an asynchronous (UART) serial interface connector. It is a logic-level interface, not RS-232! If you plug an RS-232 level device into this connector you will likely damage something.

The connector is pin-compatible with the SparkFun FTDI Basic, which makes it simple to get your computer talking to the WAV Trigger through a USB port. The WAV Trigger apps all look for a USB serial device and will allow you to:

- Update the WAV Trigger firmware
- Test trigger options without having to write an Init file and cycle power
- Control the WAV Trigger functions using the serial port, and access all 999 tracks

The default serial baudrate is 57.6kbps, although this can be changed in the Init file.

[Back to the top](#)

## **The Init file and the Init File Maker utility app**

Like the MP3 Trigger, 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. However, the WAV Trigger also has a utility application that allows you to set these options using a graphical user interface, and 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 WT-InitMaker app makes this process much easier and less prone to error.

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 WT-InitMaker app to create an Init file.

[Back to the top](#)

## 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.

**Audio Sample Rate:** This sets the playback rate of the audio engine. The WAV Trigger assumes that all tracks are .wav files of this sample rate. For example, if all of your tracks are 22.05kHz .wav files, select 22,050Hz. The default is 44,100Hz

**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 -80dB 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.

[Back to the top](#)

## 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.

**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.

**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.

[Back to the top](#)

## 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 WT-InitMaker app.

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:**

**GET\_VERSION** Message Code = 0x01, Length = 5

Data = none

Response = VERSION\_STRING

Comments: Requests the WAV Trigger to transmit the VERSION\_STRING message

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

**GET\_SYS\_INFO** Message Code = 0x02, Length = 5

Data = none

Response = SYS\_INFO

Comments: Requests the WAV Trigger to transmit the SYS\_INFO message

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

**CONTROL\_TRACK** Message Code = 0x03, Length = 8

Data = *Track Control Code* (1 byte), Track Number (2 bytes)

Response = none

Comments: Sends a Track Control Code to a specific track number

Example: 0xf0, 0xaa, 0x08, 0x03, 0x01, 0x0a, 0x00, 0x55

**Track Control Codes:**

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

<b>STOP_ALL</b>	<p>Message Code = 0×04, Length = 5</p> <p>Data = none</p> <p>Response = none</p> <p>Comments: Commands the WAV Trigger to stop all tracks immediately</p> <p>Example: 0xf0, 0xaa, 0×05, 0×04, 0×55</p>
<b>VOLUME</b>	<p>Message Code = 0×05, Length = 7</p> <p>Data = Volume (2 bytes, signed int, -70dB to +10dB)</p> <p>Response = none</p> <p>Comments: Updates the output volume of the WAV Trigger with the specified gain in dB</p> <p>Example: 0xf0, 0xaa, 0×07, 0×05, 0×00, 0×00, 0×55</p>
<b>GET_STATUS</b>	<p>Message Code = 0×07, Length = 5</p> <p>Data = none</p> <p>Response = STATUS</p> <p>Comments: Requests the WAV Trigger to transmit the STATUS message</p> <p>Example: 0xf0, 0xaa, 0×05, 0×07, 0×55</p>
<b>TRACK_VOLUME</b> (Note 1)	<p>Message Code = 0×08, Length = 9</p> <p>Data = Track Number (2 bytes), Volume (2 bytes, signed int, -70dB to +10dB)</p> <p>Response = none</p> <p>Comments: Updates the volume of a track with the specified gain in dB</p> <p>Example: 0xf0, 0xaa, 0×09, 0×08, 0×01, 0×00, 0×00, 0×00, 0×55</p>
<b>AMP_POWER</b> (Note 1)	<p>Message Code = 0×09, Length = 6</p> <p>Data = Audio Amp State (1 byte: 0 = OFF, 1 = ON)</p> <p>Response = none</p> <p>Comments: Enables/disables the on-board audio amplifier</p>



Example: 0xf0, 0xaa, 0x06, 0x09, 0x01, 0x55

### Commands sent FROM the WAV Trigger:

**VERSION\_STR** Message Code = 0x81, Length = 25

Data = Firmware version string (char[20])

Comments: Data contains the WAV Trigger version – ASCII, not null-terminated

Example: 0xf0, 0xaa, 0x19, 0x81, ( char[20] ), 0x55

**SYS\_INFO** Message Code = 0x82, Length = 8

Data = Number of voices (1 byte), Number of tracks (2 bytes)

Comments: Data contains current config information

Example: 0xf0, 0xaa, 0x08, 0x82, 0x08, 0x7f, 0x00, 0x55

**STATUS** Message Code = 0x83, Length = variable

Data = 2 \* T, where T are the track numbers that are playing (LSB, MSB)

Comments: 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

Note 1: Implemented in Firmware Version 0.57 and above

[Back to the top](#)

## 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.

[Back to the top](#)

## Meta

- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.org](#)

## robertsonics

Los Angeles, CA

