Vol 54 No 1, March 2020

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

AnalogDialogue

RAQ Issue 175: LTspice Audio WAV Files: Using Stereo and Encrypting Voice Messages

Simon Bramble, Senior Field Applications Engineer

Question:

Can you utilize stereo data and encrypt voice messages with LTspice audio WAV files?

Answer:

If music be the food of love, simulate on.

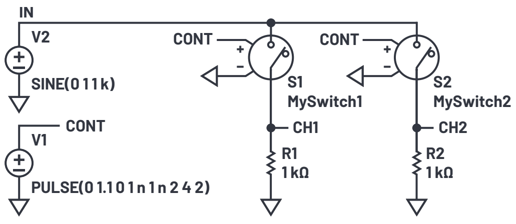
This RAQ explains how to use LTspice® audio WAV files for the syntax of stereo (and higher channel count).

LTspice can be used to generate WAV files as an output of a circuit simulation as well as import WAV files to excite a circuit simulation. It is well documented that mono WAV files can be used as an input in LTspice, and LTspice can be used to produce a WAV output. This article details how to use LTspice audio WAV files for the less well-known syntax of stereo (and higher channel count).

LTspice has many superpowers, but its handling of audio files is one of its more impressive talents. While it is fascinating to see a circuit come to life on the computer screen, creating a sound file that can be played outside of LTspice enables engineers to use another sense to evaluate simulations. Using mono LTspice audio WAV files is well documented. This article expands the discussion to stereo (or more channels), and shows how to export stereo data from, and import it to, LTspice via WAV files. It also illustrates a few tips and tricks with WAV files that will enable the reader to further utilize WAV files.

Generating a Stereo WAV File

Let's start by producing a stereo wave file from a mono signal. Figure 1 shows a circuit that generates a 1 V, 1 kHz sine wave and splits it into two channels, alternating the signal between them—the 1 kHz tone is switched in 2-second intervals between CH1 and CH2.



```
.tran 8
.wave "C:\export.wav" 16 44.1 k V(CH1)  V(CH2)
.model MySwitch1 SW(Ron = 10 m Roff = 1000 Meg Vt = 1)
.model MySwitch2 SW(Ron = 1000 Meg Roff = 10 M Vt = 1)
```

Figure 1. In this simulation, a 1 kHz sine wave is switched in two-second intervals between CH1 and CH2. The resulting two-channel signal is exported to an audio WAV file.

The command .wave **"C:\export.wav" 16 44.1k V(CH1) V(CH2)** digitizes each channel with 16-bit resolution, sampled at 44.1 kSPS, and stores the resultant audio in C:\export.wav. In the command above, each signal listed after the sample rate becomes its own channel in the WAV file. LTspice can store as many as 65,535 channels in a single LTspice audio WAV file—just append signals to the above command as desired.

By default, LTspice's .wave command saves the first listed channel as the *left* audio channel and the second as the *right* audio channel. In this case, when export.wav is played back through a media player, CH1 will be read as the left channel and CH2 will read as the right, regardless of the circuit node naming convention. Note that, by default, CH1 and CH2 are stored as chan 0 and chan 1, respectively, in the .wav file, which is important for reading the file discussed below.

This exported stereo audio file can be used to stimulate another circuit, shown in Figure 2, which uses the two channels from export.wav as signal inputs.

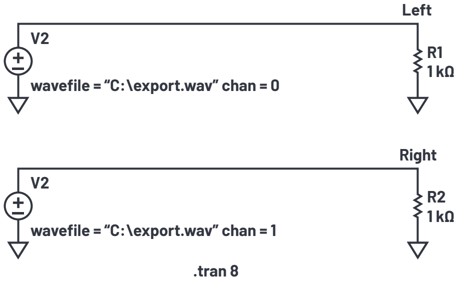
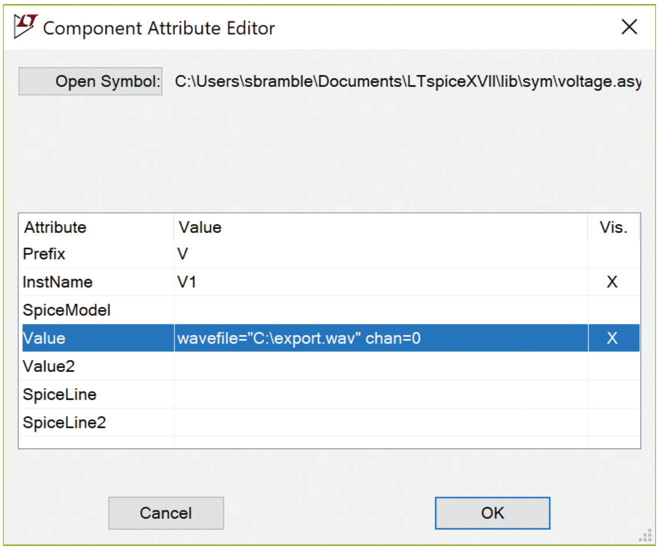

Left
V2
R1
1 kΩ
wavefile = "C:\export.wav" chan = 0
Right
V2
R2
1 kΩ
wavefile = "C:\export.wav" chan = 1
.tran 8

Figure 2. The two stereo channels from export.wav are used to stimulate two independent circuits.

The voltage sources, V1 and V2, are placed as usual, then the voltage signals from export.wav are assigned by holding down the CTRL key and right clicking over each voltage source, revealing the **Component Attribute Editor** as shown in Figure 3.



Component Attribute Editor

Open Symbol: C:\Users\sbramble\Documents\LTspiceXVII\lib\sym\voltage.asy

Attribute	Value	Vis.
Prefix	V	
InstName	V1	X
SpiceModel		
Value	wavefile="C:\export.wav" chan=0	X
Value2		
SpiceLine		
SpiceLine2		

Figure 3. Stereo signals from export.wav are used as inputs to the circuit of Figure 2. Here is the assignment for V1, with the value set to pull Channel 0 from export.wav.

As mentioned above, when the LTspice audio WAV file is first generated, as many as 65,535 channels can be digitized into one WAV file—simply append as many channels as desired onto the end of the .wave command. Remember, by default, LTspice names the first channel Channel 0, the next one is named Channel 1, and so on. In this case, export.wav (generated by the simulation in Figure 1), stores the voltage V(CH1) as Channel 0 and V(CH2) as Channel 1. To play back these channels using a voltage source, specify the .wav file and channel in the value line of the voltage source. In this case:

- To instruct V1 to replay Figure 1's V(CH1): **wavefile="C:\export.wav" chan=0**
- To instruct V2 to replay Figure 1's V(CH2): **wavefile="C:\export.wav" chan=1**

Audio Separation

Theoretically, playing export.wav through a media player should switch between playing the 1 kHz tone entirely through the left speaker (or headphone) for two seconds then through the right speaker for two seconds. Nevertheless, complete stereo separation is not guaranteed and this depends on the quality of the media player used during playback.

Playing export.wav through a laptop showed about 30% of the left channel appearing on the right channel when measured on an oscilloscope as shown in Figure 4.

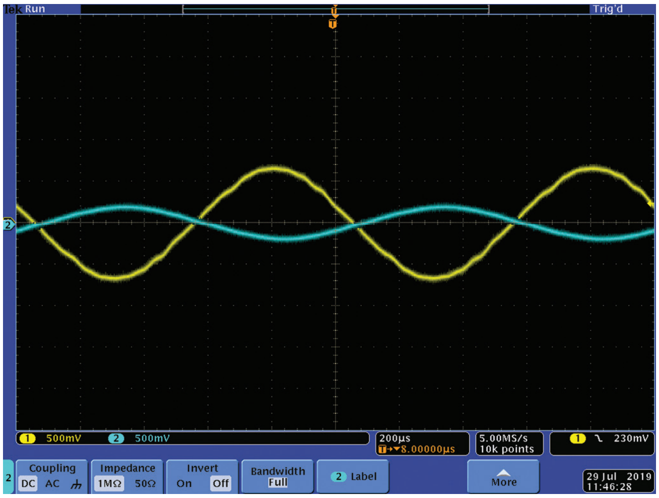


Figure 4. The left (yellow) channel shows about 30% feedthrough into the right (blue) channel when played on a laptop.

Playing the same file on a mobile phone handset (circa 2000) gave a more separated result, showing no perceivable crosstalk, but a slight amount of distortion at full volume as shown in Figure 5.

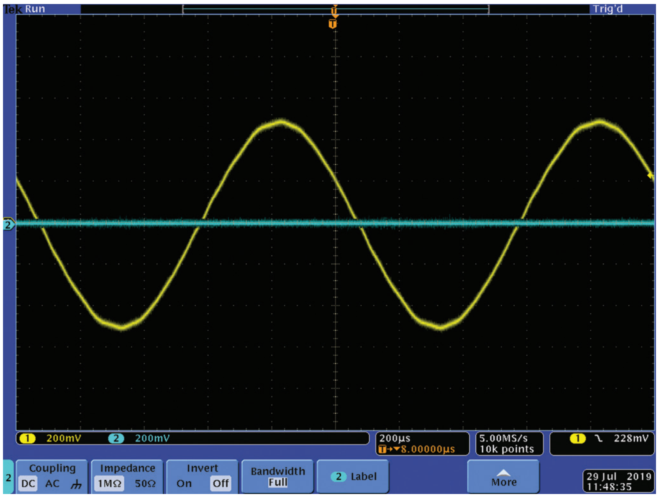


Figure 5. The mobile phone handset from the year 2000 showed no crosstalk, but distortion at full volume.

Repeating the experiment on a later generation 2018 handset showed no perceivable crosstalk, a full 1 V peak signal, and very little distortion as shown in Figure 6. Note the oscilloscope plots were taken with a 500 mV/div sensitivity.

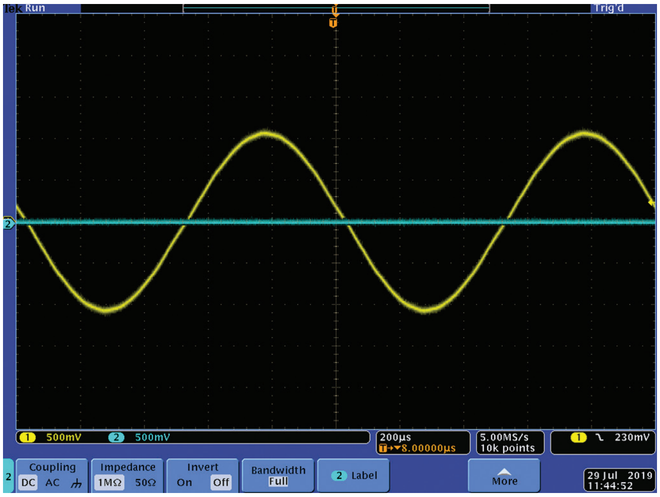


Figure 6. A later generation handset showed significantly better performance for crosstalk, distortion, and amplitude.

The same file was used across all three platforms, showing that LTspice is producing the WAV file with full separation, but resulting playback depends greatly on the quality of the player's audio stage.

Voice Encryption

The circuit in Figure 7 shows a basic method of voice encryption whereby an audio signal is encrypted using a random number sequence, then decrypted.

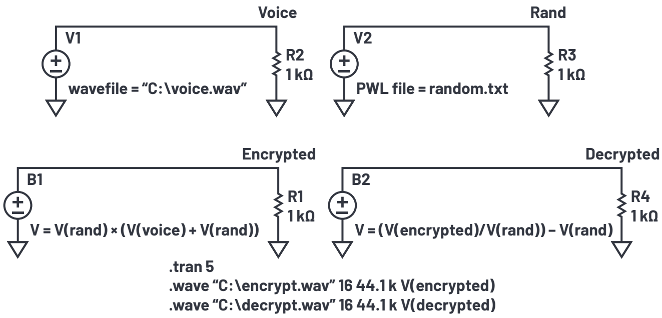
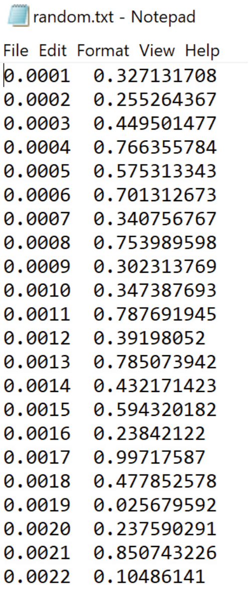

Voice
V1
wavefile = "C:\voice.wav"
R2
1 kΩ
Rand
V2
PWL file = random.txt
R3
1 kΩ
Encrypted
B1
V = V(rand) × (V(voice) + V(rand))
R1
1 kΩ
Decrypted
B2
V = (V(encrypted)/V(rand)) – V(rand)
R4
1 kΩ
.tran 5
.wave "C:\encrypt.wav" 16 44.1 k V(encrypted)
.wave "C:\decrypt.wav" 16 44.1 k V(decrypted)

Figure 7. Using a random voltage source to encrypt/decrypt an audio file.

The file voice.wav contains the original audio. An Excel spreadsheet is used to generate a random number sequence with a 100 µs change period. The results are copied into a text file called random.txt. An extract of random.txt is shown in Figure 8.



```
0.0001  0.327131708
0.0002  0.255264367
0.0003  0.449501477
0.0004  0.766355784
0.0005  0.575313343
0.0006  0.701312673
0.0007  0.340756767
0.0008  0.753989598
0.0009  0.302313769
0.0010  0.347387693
0.0011  0.787691945
0.0012  0.39198052
0.0013  0.785073942
0.0014  0.432171423
0.0015  0.594320182
0.0016  0.23842122
0.0017  0.99717587
0.0018  0.477852578
0.0019  0.025679592
0.0020  0.237590291
0.0021  0.850743226
0.0022  0.10486141
```

Figure 8. Random voltages generated using Excel and saved in a text file.

This file is used to generate a randomly changing voltage, V(RAND), using the piecewise linear (PWL) voltage source in LTspice.

V(RAND) is added to the voice signal using the behavioral voltage source B1. The output is then multiplied by V(RAND) and the result is sent to the encrypt.wav file. Listening to encrypt.wav showed that the original audio was barely perceivable.

Figure 9 shows the original voice, the encrypted voice, and the decrypted voice signals in the LTspice plot window.

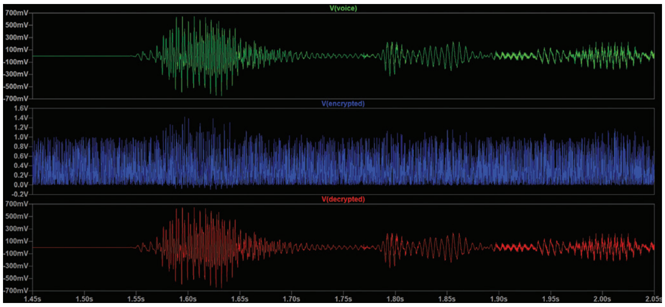
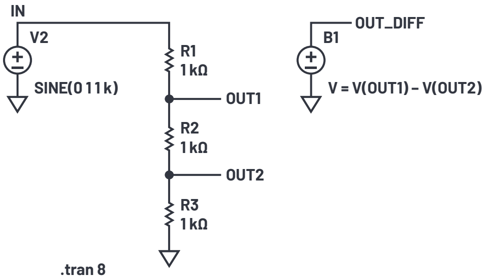


Figure 9. Output of original, encrypted, and decrypted voice signals.

A second behavioral voltage source was then used to decrypt the original audio signal and the result was sent to the file decrypt.wav.

Producing a WAV File from a Differential Voltage Source

The syntax of the .wave command does not allow for the digitization of differential voltages. However, this can easily be overcome by using a behavioral voltage source as shown in Figure 10.



```
.tran 8
wave "C:\export.wav" 16 44.1 k V(OUT_DIFF)
```

Figure 10. Creating a WAV file from a differential voltage.

The behavioral voltage source outputs a voltage equal to V(OUT1) – V(OUT2) and this can be used in the .wave command in the usual way as shown.

Indeed, the variables inside the behavioral voltage source's function can include any voltage or current in the circuit and these can be manipulated using any of the mathematic functions of LTspice. The net result can then be exported to an LTspice audio WAV file in the normal way.

LTspice is a powerful simulator, but its results do not have to be contained inside LTspice. Using the .wave command, LTspice can import, manipulate, and export audio files for playing on a media player.

About the Author

Simon Bramble graduated from Brunel University in London in 1991 with a degree in electrical engineering and electronics, specializing in analog electronics and power. He has spent his career in analog electronics and worked at Linear Technology (now part of Analog Devices). He can be reached at *simon.bramble@analog.com*.