

Programming an AWG Arbitrary Waveform Generator

Introduction

Function Generators typically produce Sine, Square, Triangle, Saw Tooth and DC signals. Many modern instruments include an AWG or Arbitrary Waveform Generator signal as an additional selection. This allows you to construct your own custom waveform to meet your testing needs. Typical examples might be a dual/multi tone audio signal to test intermodulation distortion in an audio amplifier or a complex modulated signal to test an RF demodulator.

In order to use the AWG, you have to create your waveform using a special editor or a math program. Essentially you are creating the digital samples that paint the actual waveform that you want. The quality of the waveform is limited by the hardware sample limitation of the instrument.

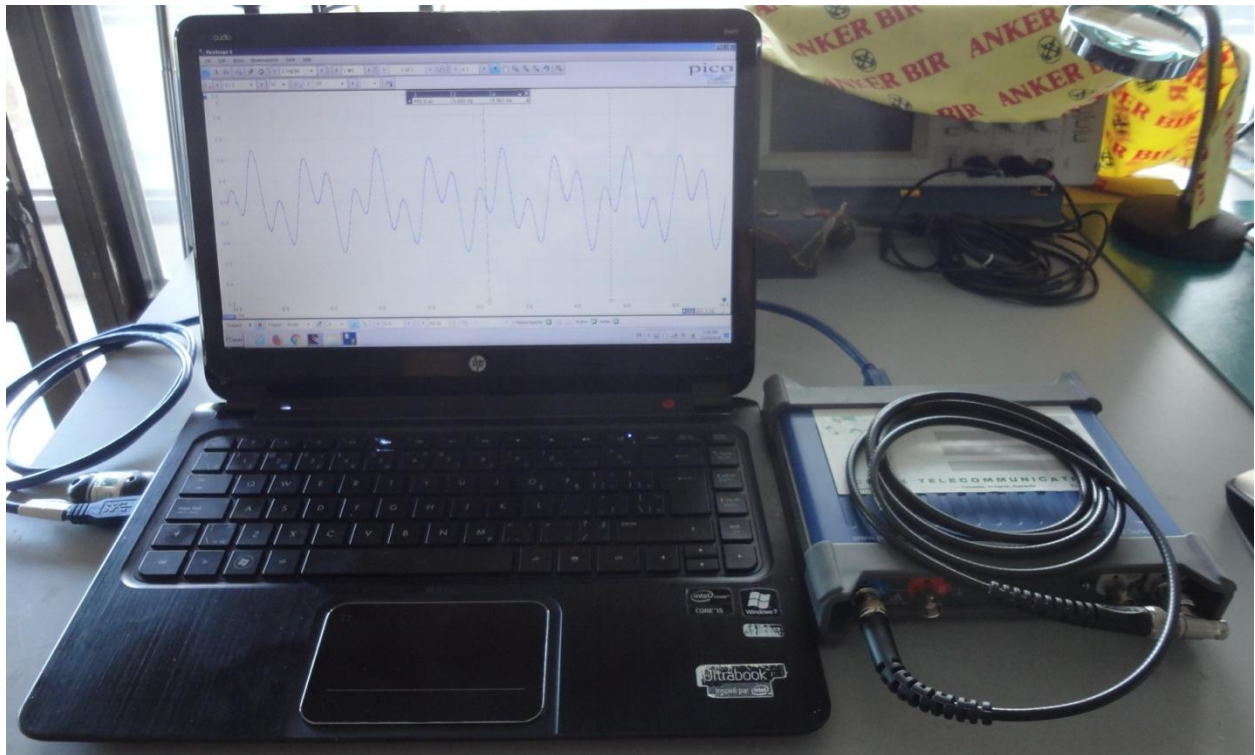


Figure 1 HP Laptop & PicoScope 3205 Scope/Spec/Function Generator

AWG Editor

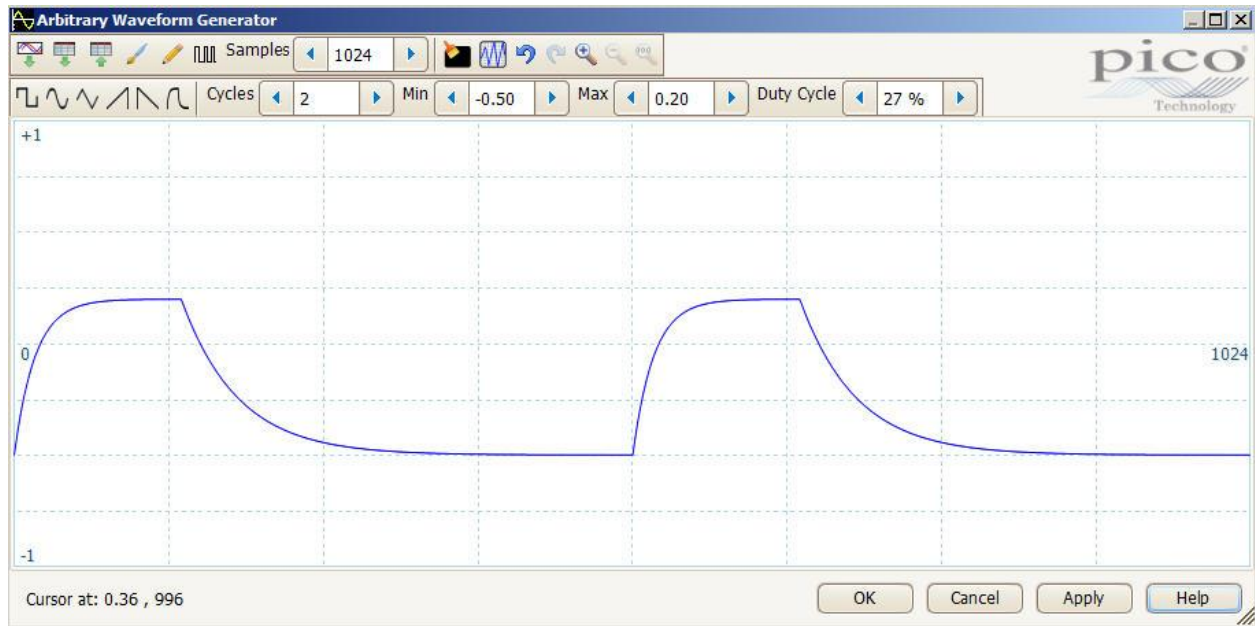


Figure 2 AWG Editor (PicoScope 3205D)

The AWG editor allows you to customize various waveforms, draw them, record them from a scope and modify them, or import CSV text files from a math program that was used to construct them. A CSV file is a text file consisting of a series of floating point numbers separated by commas, representing the signal sample values taken at a particular sampling rate F_s and lying between a minimum voltage of -1.0V and +1.0V.

AWG Output

Figure 3 shows a Scilab program to generate a waveform consisting of two sine waves at 400Hz, 1KHz and with F_s sampling frequency 44.10KHz. The resulting samples are written to a column using the function `csvWrite`. The common period for 400Hz = 2.5msec & 1000Hz = 1msec, is 5msec. Samples are collected for this period. Then the CSV file is read into the AWG, and the samples are written and repeated every 5msec or 200Hz. Figure 4 shows the AWG output taken over 2 periods or 10msecs. Figure 5 shows the AWG output spectrum clearly showing the two tones of 400Hz and 1KHz.

```
pico_awg.sce X
1 //PicoScope AWG-Dual-Sine-Wave
2 //Normalized-output-to-'avg.csv'
3 //J.Clark-Sept-22nd_2018
4
5 Fs1=400;
6 Fs2=1000;
7 Fs=44100;
8 Ts=1/Fs;
9 t=0:Ts:.005;
10 y1=sin(2*pi*Fs1*t);
11 y2=sin(2*pi*Fs2*t);
12 y=y1+y2;
13 y=y/(max(abs(y)));
14 y=y';
15 csvWrite(y,"avg.csv")
16 plot(t,y);
17
```

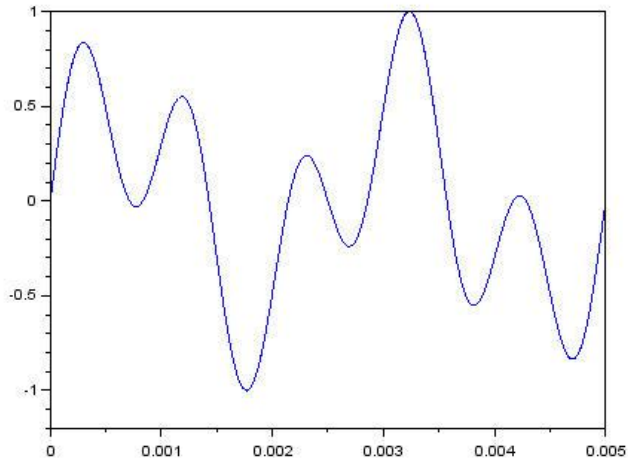


Figure 3 Scilab Program Dual Sine Generator to CSV File

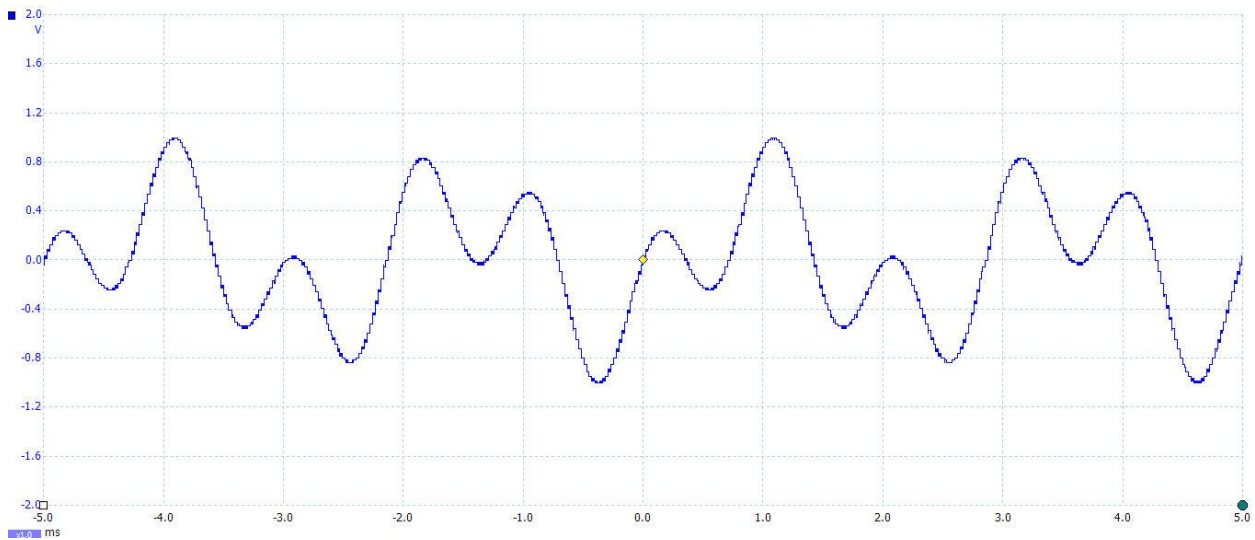


Figure 4 AWG Output Shown over 10msec (2 periods)

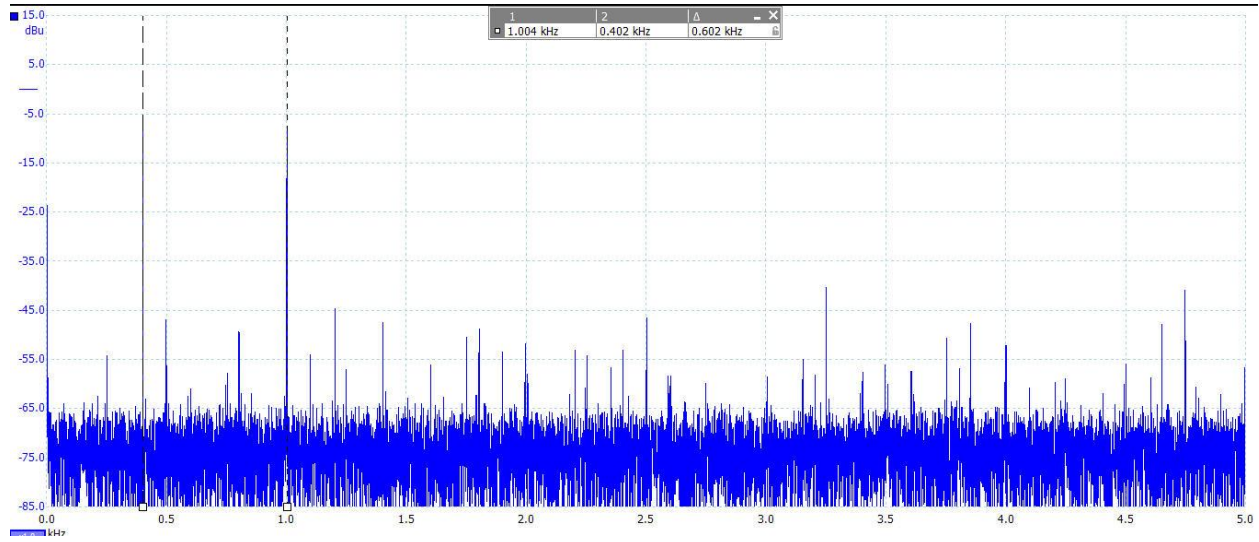


Figure 5 AWG Output Spectrum Showing 400Hz & 1KHz

J. Clark/Sept 22nd_2018