Sound Card Signal Considerations

Modern day sound cards are marvels of technology. Cards offered in the last few years provide 16-bit digital to analog converters, crystal controlled sampling rates and low noise. The economical Sound Blaster Live — Value Edition, employed here, is such an example as are many others. Our task is to put this technology to work to produce the best and most accurate signal that can be generated by the sound card.

It may come as a surprise to some, but the only signal generated by the sound card worthy of consideration is a sine wave. You may find some sound card generators that provide square waves or triangular waves, but these are very limiting. Here's why. A sound card has a sampling rate and therefore the software produces samples of the desired signal at discrete intervals in time. The signal samples are filtered (that is, put back in analog form) before being delivered to the output. Say that the sound card has a sampling rate of 48,000 samples per second and the desired output frequency is 5000 Hz. This works out to 9.6 samples per cycle, a problem for a square wave, but workable for a sine wave. Let's see why.

Figure 1 shows a 5000 Hz. sine wave and the resulting samples. Since there are 9.6 samples per period and 2π radians for one period of a sine wave, our samples are separated by about 0.655 radians. In software, the amplitude of the sine wave is calculated at each new sample based on its radian value. Since the samples start at zero radians and move horizontally at slices of 0.655 radians, we see that this process does not come out evenly, in radians, for one cycle. For a square wave this would be a big problem, since the samples may not line up properly, but for a sine wave we simply move to the next slice, calculate the amplitude, and the sine wave remains continuous and correct. This is how most computer programs calculate sine waves. But wait, there is a problem.

If we wish to calculate the samples for a sine wave of 1 second duration sampled at 48,000 samples per second, we need to provide 96,000 bytes of memory, since each sample requires 2 bytes. No problem yet. On the other hand if we wish to produce a sine wave lasting, say, 10 minutes, this would require 5.76 MB of memory. Programmers have sought a way around this storage problem by using looping. They will calculate, say 10 cycles of sine wave data and loop through the data over and over. Many programs found on the Internet work this way. But as we have seen from Figure 1, the sine wave data may not come out evenly. So if we loop through the data there will be a glitch at the end of the loop each time through the loop.

I pondered this problem and finally came to the conclusion that the only way to do it properly was to calculate chunks of sine wave data on the fly and continuously feed them to buffers for the sound card to output. In my program I calculate the data for the first buffer, output it, calculate data for the next buffer, output it and go back and repeat the process for the first buffer and so on. This works well. There are no glitches and the sine wave is generated continuously. The program took a few days to figure out in *Visual Basic*, but it was worth it since it is now a very accurate, continuous, sine wave generator with programmable frequency. This is especially important if we intend to multiply the frequency with a PLL.

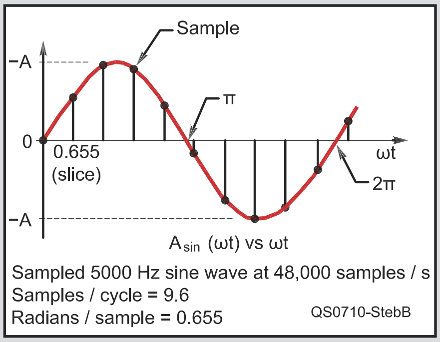

Figure B — Generation of a 5000 Hz sine wave from samples at 48,000 Hz.

A Unique SSB/CW Receiver

The astute reader has probably gathered that the above described method for checking signal generators is essentially the same technique used in a direct conversion (DC) receiver. The only difference is that instead of using a signal generator for input we can use an antenna. Not quite. It's true, if we connect an antenna to pin 1 of the mixer, we would pick up stations but mostly strong, local ones. To make a ham receiver perform better requires using more gain and selectivity up front. The simple circuit shown in Figure C provides a little bit of both and is intended to operate with synthesizer Range 1.

The output (from C5) connects directly to the mixer (pin 1). Be sure to disconnect the attenuator (R11 and R12 in Figure 5) noted above. The input is a simple antenna, in my case a wire about 25 feet along the floor. A 5 foot length of coax from the circuit to the wire is used to reduce pickup of noise from the computer, with the coax shield connected to ground at the circuit. Capacitor C1 and inductor L1 form a very sharp tunable peaking circuit to reduce strong local stations and noise. It needs to be adjusted carefully to peak the signal after changing frequencies. Inductor L1 should be in the range of 10 to 12 µH. I used 22 turns of wire on a toroid but even RF chokes would work okay here. The peaking circuit has a range of 3.2 MHz to 8.2 MHz via adjustment of C1, a broadcast variable from an old AM radio (35 to 365 pF). This is enough to tune through nearly all the stations and bands noted earlier in the article.

I tested the receiver with W1AW and the Morse code transmissions at 7047.5 kHz. They came in loud and clear on most days near dusk and had a pleasant tone. The tone was good for other CW stations too, even weak ones, but loud ones seemed slightly noisy, perhaps because of RF or audio pick up in my circuit which is not shielded. Building the circuit inside a metal box might help. I also received numerous SSB stations and nets on 80 and 40 meters with good audio quality. In the evening on 40 meters, AM shortwave broadcast stations invariably come pounding in here causing substantial interference but in many cases I could still copy ham operators. AM shortwave interference is hard to reject in a direct conversion (DC) receiver. A narrow bandwidth CW audio filter would help somewhat in this situation.

Since this is a bare bones receiver it doesn't have features like automatic gain control (AGC) and audio image rejection. In case you don't recall, images are copies of the signal found above and below the carrier. For example, when tuning W1AW Morse transmissions you can tune either above or below the carrier frequency noted above and hear the same code. This makes using a DC receiver a little strange if you're used to a modern superheterodyne with sideband selection. Nonetheless, this receiver does have features that set it apart from other low cost DC receivers. Its VFO does not drift around and there is a nice frequency readout telling you the exact frequency you are receiving.

This receiver will also pick up AM stations such as CHU (3330 kHz) and WWV (5000 kHz). For these stations you need to adjust the frequency to zero beat the carrier. I found my synthesizer to be right on the mark in these two cases.

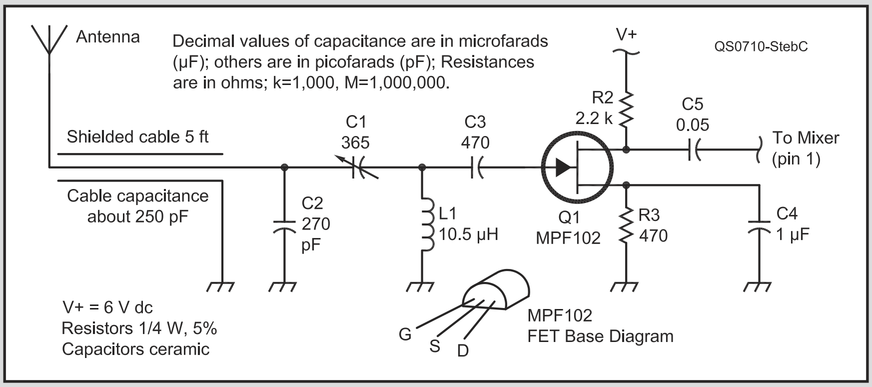

Figure C — Front end for SSB/CW direct conversion receiver.