RMS Conversion

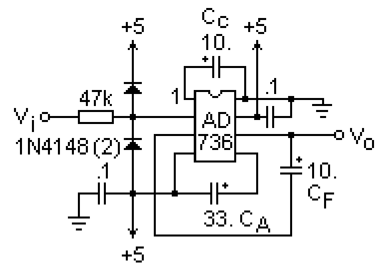
The average level of a varying signal can be expressed in terms of its peak value V_p, the full-wave rectified average value V_{av}, or the *rms* value V_{rms}, which is the square root of the average value of the square of the signal. For a signal of determinate waveform, all these values are proportional. The ratio V_p/V_{rms} is called the *crest factor*, and the ration V_{rms}/V_{av} is called the *form factor*. For a sine wave, these ratios are 1.414 and 1.11, respectively, for a triangle wave 1.732 and 1.15, and for a symmetrical square wave, both are unity. It is easy to verify these figures for yourself, an exercise that will give a better comprehension of what they mean.

If the waveform is not periodic, but still can be described statistically, these factors may be constant on the average, so the various measures can be related to one another. With Gaussian noise, for example, the crest factor is 3.0, a good rule of thumb for determining the energy content of a noise signal observed on an oscilloscope. In the general case, however, none of the measures are reliably related to one another and depend on the actual waveform. In this case, only the rms value is a good statistic, with its square proportional to the average power.

The usual way a DMM determines the rms value of an AC signal is to full-wave rectify and filter the signal, then calibrate the readout so that this average value is multiplied by 1.11. This works fine on sinusoidal signals, and also on signals that are not too far from sinusoidal. Even for a triangle wave, the result is only 4% too low, but for a square wave it is 11% too high. Good enough for government work, obviously, but not for careful work with strange waveforms or noise. Here we need to find the "true" rms value, rather than one calculated from the average value.

Analog Devices supplies a chip that will do just this, the **AD736**. There are more costly premium chips as well, but the **AD736** is completely adequate for the job. It uses a bipolar power supply ranging from a low of +2.8,-3.2 V to ±16.5 V. It works well on ±5 V, which I used. If you do not have a suitable supply, then use ±12 V. Bypass both supplies with 0.1μF capacitors to ground to prevent noise from entering here. There are two inputs to the input op-amp of the device, pin 1 (low impedance, inverting) and pin 2 (high impedance, noninverting). If you use pin 1, send the signal through a coupling capacitor, perhaps 10μF, and ground pin 2. If you use pin 2, pin 1 can be left open, or, better, grounded through a 10μF capacitor. Protect the high-impedance input with a pair of diode clamps and a 47k resistor, as shown in the circuit. This prevents signals outside of the power supply range from damaging the chip. Any signal diode can be used, though the common 1N4148 is shown.

A complete circuit is shown at the right. Inside, the chip contains a full-wave rectifier. If you do not connect the capacitor C_A, the output will be averaged by filter capacitor C_F to give V_{av}, which you can multiply by any constant you wish using an op-amp. There is also an RMS conversion unit, and if you supply an averaging capacitor C_A, the output will be proportional to V_{rms}. Then, C_F, which is optional, serves to



RMS-to-DC Converter

smooth the output. The values of the capacitors determine the averaging times and bandwidths, as is obvious. The values shown here are good for general service. C_A can range up to 150μF, and C_F up to 100μF, if desired. The **AD736** will handle crest factors up to 5.

I made measurements on the output of my function generator, which supplied square, triangle and sine waves. I adjusted the average (DC) value of the output to 0, although the **AD736** did not seem to be sensitive to the DC level and did not include it in the rms value (I do not fully understand why not). I presumed that the peak values of each of the three waveforms was the same, a rather coarse assumption, and calibrated the output with the square wave. Adjusting the average value to 0 with a DMM, I found outputs of 2.435 V, 1.476 V and 1.691 V for square, triangle and sine waves, respectively, using a DMM on the DCV function. This gave crest factors of 1.440 (ideally 1.414) and 1.650 (ideally 1.732), which shows that the converter is working. It seems to work better for values less than 1.0 V. The frequency was 1 kHz. If you apply a 2 V peak-to-peak sine wave, the output should be about 0.707 V. It would be interesting to observe the effect of varying the frequency. If the low-impedance input is used, the data sheets promise a rather wide bandwidth, over 100 kHz.