

## AFP-FSK, a method of directly generating RF FSK signals from audio

With the advent of sound card generated FSK modes like FT-8 and DSP processing, these modes have become very popular. The one down side is they require a SSB transmitter, or at least a DSB transmitter to mix the audio tones to RF. For the homebrewer, SSB transmitters are a little complicated to build. DSB transmitters are less desirable due to the unwanted side band being also being transmitted and robbing you of power output, since it's split between the two side bands. Both methods suffer from poor efficiency due to the need for using linear RF amplifier stages.

A more efficient method would be to translate the audio FSK tones directly to RF. Then we can use more efficient Class C amplifiers. So, how can that be done? The first step is to determine what these audio frequencies are. The second step is to add those frequencies to the "base" frequency of the RF oscillator.

We are calling this AFP-FSK. (**A**udio **F**requency **P**rocessed - **F**requency **S**hift **K**eying)

There are two methods to determine the frequency of a signal.

The first, most common method, counts the number of cycles over a given amount of time. Hence the name "frequency counter" for the instrument most commonly used to measure frequency. How long the frequency is counted determines the amount of resolution. Count for 1 second to measure to 1 Hz. Count for 10 seconds to get 0.1 Hz resolution.

This is fine if you're in no rush to determine the frequency. But, if you are in a hurry, like our application, another approach is needed. The second method is to time a single cycle and then calculate the frequency from that time. This method works well with audio signals since the length of a single cycle is reasonably long.

This second method of measuring the time of a single cycle is used in the new QRPGuys FT8 V3 Digital Transceiver, using a program written and developed by Kazuhisa "Kazu" Terasaki (AG6NS) for the Atmel MEGA328 processor. The audio frequency measurement elements of it were first developed by Kazu while as a high school student in 1982. He designed a high speed method to improve cassette tape save/load of computer programs for Japanese PC (Z-80 based NEC PC-8001) from hardware based 600bps FSK to software based 7000bps FSK. He licensed this technique to a company in Japan that sold a product (CIF-7000) based on his hardware and software design. The implementation is straight forward, although the code gets a bit complicated and we won't go into those details. The source code is available for those who would like to dive into the nuts and bolts of the methods used.

Audio is applied to the Atmel MEGA328's analog comparator, which is biased near it's trigger point. Using one of the processor's time capture registers, the length of time that the comparator output remains high and low are separately timed in the interrupt routine. In the main loop, the signal's frequency is calculated, then added to the VFO base frequency value and sent to the Si5351 digital clock chip. Simple as that. The processor and data loading for the Si5351A/MS5351M is fast enough that audio up to 4 kHz can be timed and the clock chip registers updated. The processor's time capture register has 62.5ns resolution, meaning it can distinguish the tiny difference between 1500.00Hz and 1500.14Hz.

Since the output of the Si5351A/MS5351M is square waves, it's a simple matter to buffer the signal with a fast (74AC02) logic gate, which in turn drives a MOSFET power amplifier for good efficiency. Coupled with a sensitive Direct Conversion receiver, a simple and compact FT-8 transceiver is produced. And of course, besides FT-8, any sound card generated FSK mode can be transmitted. PSK31 is not supported, as strictly speaking it's not a FSK mode, but a very slow modem with on/off keying.

Steve Weber - (KD1JV)  
Kazuhisa "Kazu" Terasaki - (AG6NS)