PSK31 — A Different View

PSK31 is easier to visualize using pictures.

Thomas W. Brooks, KE1R

Like many hams, I have been excited by radios, electronics and computers for years. In the last few decades computers have become a major part of my life and are now integrated with my Amateur Radio station. Talk to a ham anywhere in the world and in seconds they may look up your call letters on **qrz.com**.[1] They can find your location, name and perhaps see your biography or picture, all with one click. You might even send them an e-mail after the contact because you found their e-mail address on the site. Now, through the use of computers with built-in sound cards, digital modes such as PSK31 can bring digital communication into real time.

PSK has both fascinated and mystified me. I wanted to know how an HF signal with a bandwidth of 31 Hz and power of 25 W can get around the world and be reliably read. In this article I will focus on decoding PSK31 waveforms to understand what they mean. You will see "railroad tracks" in the waterfall displays, but no trains. Let's have a look.

In the '80s and '90s Commodore and RadioShack computers became available to the ham community and computerized radioteletype (RTTY) contacts were first possible for a number of hams.[2] RTTY, the classic digital mode, is typically generated using audio frequency shift keying (AFSK) with tones of 2125 and 2295 Hz representing *mark* and *space* elements respectively [The terms *mark* and *space* go back to the earliest Morse communication using paper tape. The condition "mark" referred to a mark on the tape that occurred during the interval with current on the line. The term "space" referred to no current on the line and showed up as a space between the marks. — *Ed.*] Each RTTY character is formed with one or more start (space) pulses, five mark or space data elements and then one or more stop (mark) pulses.

PSK31, on the other hand, uses phase shift keying (PSK) and a variable length coding system.[3] PSK is an amazing digital communication tool that was first used by Pawel Jalocha, SP9VRC, and was called SLOWBPSK (Binary Phase Shift Keying).[4] Peter Martinez, G3PLX, further developed this concept and offered it to the Amateur Radio community in December 1998 (G3PLX was also the father of AMTOR).[5] PSK uses a *varicode* (variable length code elements) that uses variable length data encodings to represent characters. For efficiency, the most common varicode characters have the fewest zeros and ones. This is similar to Morse code, in which the most common characters have the smallest number of dots and dashes.

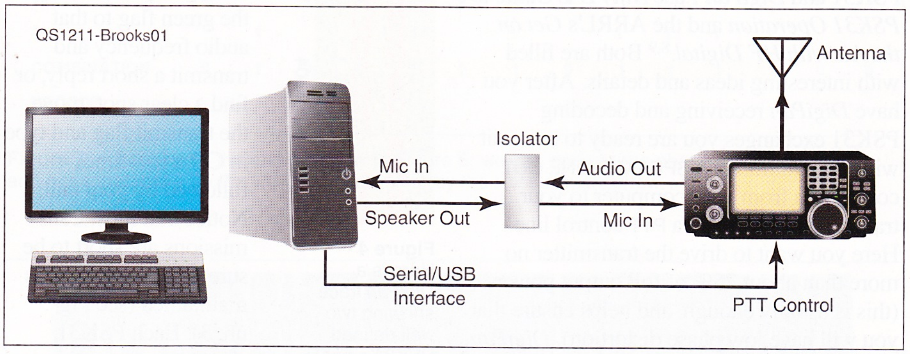

Figure 1 — PSK computer and radio setup at KE1R with audio isolation added to avoid ground loops.

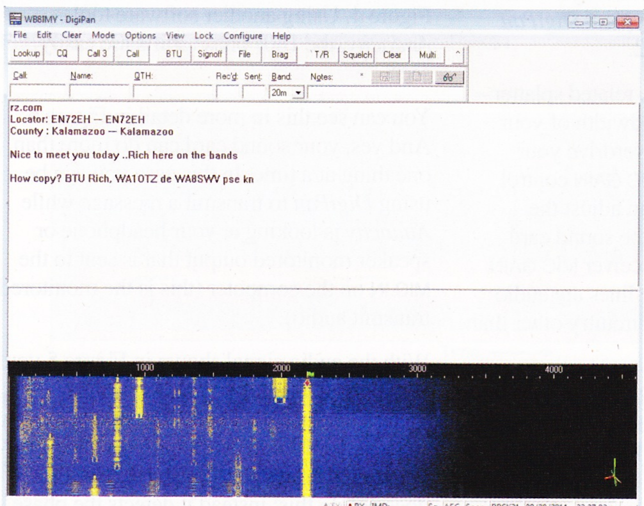

Figure 2 — What does PSK31 look like? Here is a typical *DigiPan* waterfall display screen.

QS1211-Brooks03

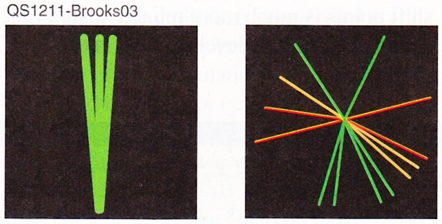

Figure 3 — This is the *DigiPan* tuning indicator, which can be turned on in the waterfall display. The display on the left indicates good tuning while the one the right indicates a signal that is out of tune or weak.

The PSK31 Setup

I use a dedicated computer as part of my ham radio station for logging, transceiver control, CW, RTTY and now for PSK31. With a sound card standard in most current computers and laptops, you can download one of several free software tools such as *DigiPan* for *Windows* or *CocoaModem* for OSX and start decoding PSK31.[6,7] The most common PSK31 frequency is 14.070 MHz (but other bands are also used). Adjust your sound card MIC IN control to start seeing displays of signals from other stations. Figure 1 shows my basic computer and radio

[1]Notes appear on page 40.

setup for PSK31. As you can see, if you have an HF station and a PC, you're almost there.

DigiPan Software

For PSK31 I like to use the popular *DigiPan* software. For good reference material on PSK31 and *DigiPan* I use *Nifty E-Z Guide to PSK31 Operation* and the ARRL's *Get on the Air with HF Digital*.[8,9] Both are filled with interesting ideas and details. After you have *DigiPan* receiving and decoding PSK31 exchanges you are ready to transmit with the addition of a SPEAKER/LINE OUT connection from your computer to your transceiver MIC IN and a PTT control line. Here you want to drive the transmitter no more than about 25% of full output power (this is usually enough, and helps ensure that you will have low phase distortion). *DigiPan* features an intermodulation distortion (IMD) monitor on received PSK signals so you can ask fellow hams to give you the IMD level of your transmission. A good target IMD level is –25 to –30 dB. A typical *DigiPan* screen is shown in Figure 2.

To avoid distortion and the related splatter (resulting from excess bandwidth of your signal), be careful to not overdrive your transceiver by using the MIC GAIN control of your transceiver. You can adjust the SPEAKER/LINE OUT from the sound card window or adjust the transceiver MIC GAIN. At my station the interface lines are audio cables with no additional circuitry other than a PTT foot pedal.

In the waterfall in Figure 2 there is one small green flag at the top near the 2000 Hz audio frequency marker. All the PSK31 signals shown in the waterfall are displayed as vertical traces or railroad tracks and are decoded by *DigiPan* simultaneously. In this case the transceiver is tuned to 14.070 MHz, USB (upper sideband). The top scale of the waterfall display shows 1000, 2000 and 3000 Hz. These numbers represent the audio carrier that is transmitted, just as if you whistled an audio tone of 1000 Hz, 2000 Hz or 3000 Hz into your microphone.

Figure 4 — Here is a *DigiPan* trace showing two well defined PSK31 signals as vertical traces.

Look for someone calling CQ, then move the green flag to that audio frequency and transmit a short reply, or find a clear spot, move the transmit flag and type in CQ a few times followed by your call. Notice that most transmissions are short to be sure that propagation is maintained (see Figure 3). Each PSK31 signal has two major frequency components that show up as the tracks that we see in Figure 4. Using another software tool, *Audacity*, the 180° phase shifts are easily seen.

You can see this in more detail in Figure 6. And yes, your sound card can do more than one thing at a time. In this case you can be using *DigiPan* to transmit a message while *Audacity* is looking at your headphone or speaker monitored output that is sent to the MIC IN on the computer (this is the monitored transmit audio).

With the audio signal shown in Figure 5 you could create a voltage detector set at an arbitrary level to detect the bits and use software to decode zeros and ones, just as you would do with a CW signal. But PSK31 does not do this; instead it detects the phase shift points where the voltage level falls to near zero, and decodes these into zeros and ones. This is done because detecting phase shift points is much more reliable than trying to detect amplitude levels that are more subject to noise. The phase shift points are at the minimum voltage level to reduce the overall signal bandwidth and eliminate possible splatter and key clicks. The PSK references describe this in detail.[3-9]

Regardless of the audio carrier frequency used, the phase shift points are at precise intervals that depend on the data pattern being transmitted. This pattern is a combination of zeros and ones.

PSK31 varicode uses a 00 between letters, while [illegible] is used as a SPACE character between words.[5] Any time there is a phase shift at a minimum signal level, it is a "0" data bit. At every clock bit with no phase shift it means a "1" character is transmitted. The sequence 00 never occurs in any character and is used only as a space between characters or words. Notice that both lower and upper case characters and a back space are available in this character set. All letter and control characters consist of no more than 10 binary bits. Table 1 provides a few examples of varicode.

How to Decode PSK31

Figure 7 shows the way PSK31 in decoded. The details are shown below.

Line 1: The symbol ▲ marks the phase shift

Table 1
Examples of Varicode Character Encoding

Character	Varicode
[illegible]	1 (space)
[illegible]	11
[illegible]	111
[illegible]	1111
[illegible]	101
[illegible]	1101
[illegible]	1110111
[illegible]	111010101
[illegible]	1010101101
[illegible]	1011111111
[illegible]	

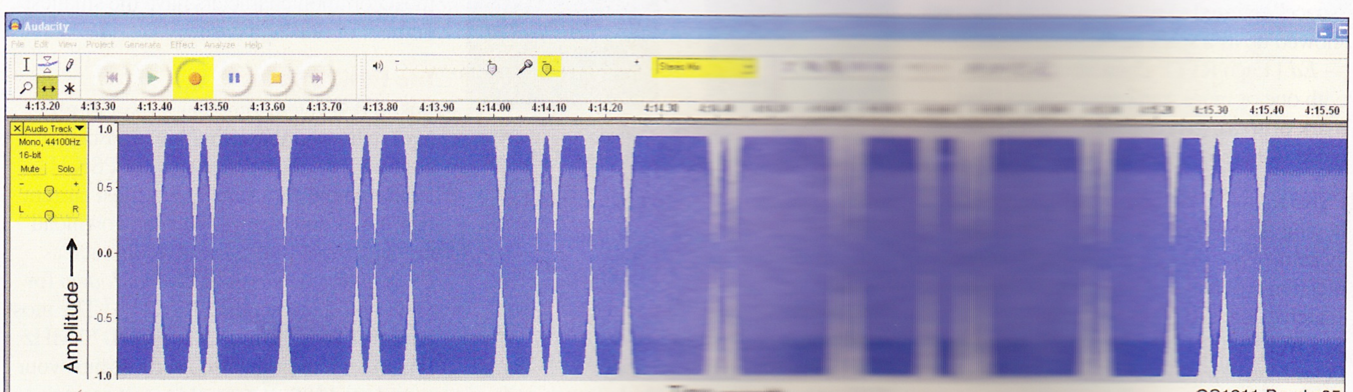

Figure 5 — Looking more closely at one of the pairs of *DigiPan* tracks in Figure 4 with the *Audacity* audio display tool you will see something like this display.[10] The 180° phase shifts appear at the minimum amplitude points on the envelope and between bursts of data.

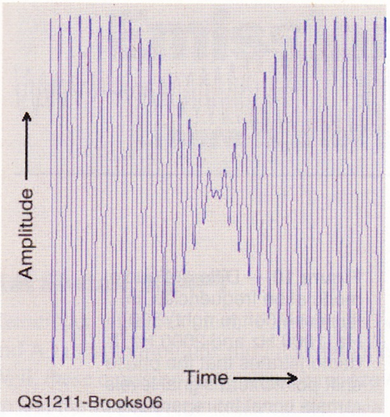

Figure 6 — If you look closely at this *Audacity* audio display you can see the 180° phase reversals at the near zero amplitude points of the signal.

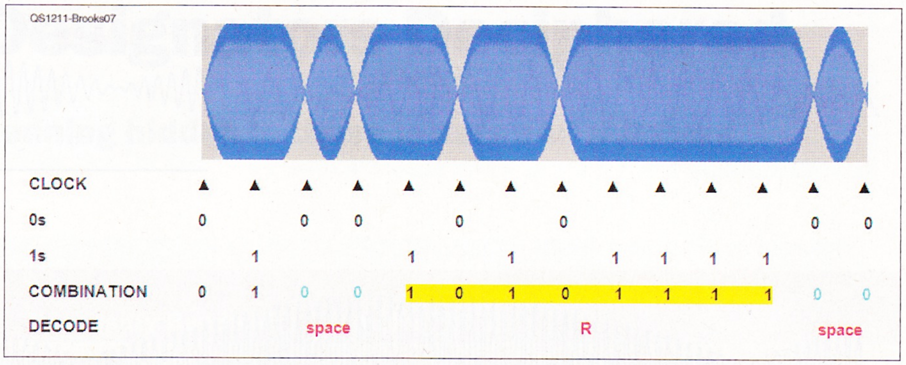

Figure 7 — Looking at the audio display in Figure 5, we can decode the phase shifts into zeros and ones that make up the varicode characters by the rules shown below.

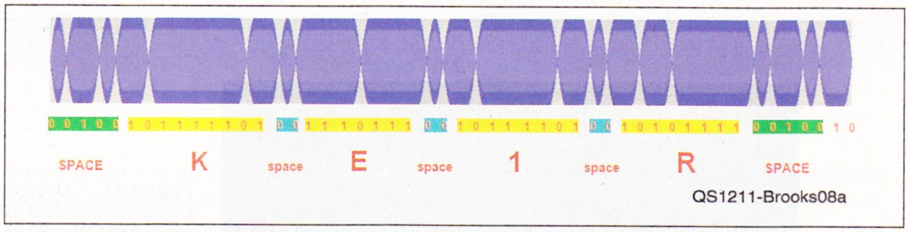

Figure 8 — Character encoding of author's call in PSK31.

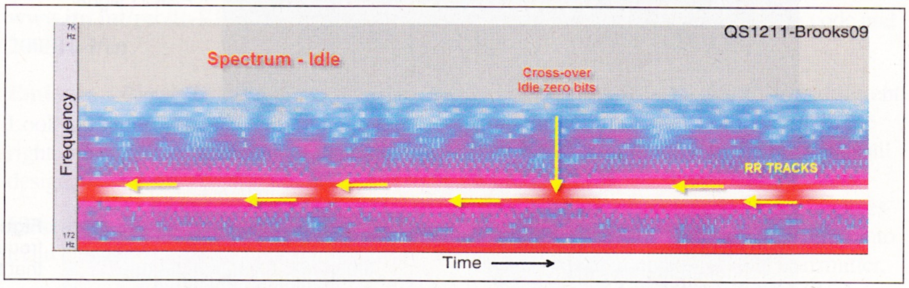

Figure 9 — Using the *Audacity* audio spectrum mode you can clearly see the railroad tracks (horizontal) and the crossties (vertical). The crossties are where the amplitude of the signal goes to zero, which line up perfectly with the string of zeros phase shifts. The railroad tracks are the ± 15 Hz side frequencies that occur regardless of the sub-audio modulation frequency.

or clock points with an all-zeros pattern being transmitted.

Line 2: At each phase-shift point (minimum signal amplitude), a 0 data bit is transmitted.

Line 3: The point at which a phase shift or clock pulse occurs, but the amplitude is at full value, indicates a one data bit.

Line 4: Look for 00 (letter space) or 00100 (word space) patterns (characters do not contain **00** together). The characters that you can decode will be found between 00s or 00100s.

Line 5: Here you can decode two letter spaces and the character R.

When you have learned how to decode a random PSK31 signal into zeros and ones you are ready to decode letters and words. Table 2 and Figure 8 provide two views of the decoding of my call.

I have now decoded the "da Vinci eCode" that I spent many hours looking at until it all came together. My curiosity was almost satisfied but not quite. A few mysteries remained. Figure 9 uses *Audacity* in audio spectrum mode to look at a PSK31 transmitter idle bit patterns, which is a string of zeros.

Table 2
Varicode Encoding of KE1R

Character	Varicode
Space (word)	0 0 1 0 0
Space (letter)	0 0
K	1 0 1 1 1 1 1 1 0 1
E	1 1 1 0 1 1 1
1	1 0 1 1 1 1 0 1
R	1 0 1 0 1 1 1 1

If we look at it using 500 Hz, 1000 Hz or 2000 Hz as the audio carrier we see the *Audacity* audio traces below. This is an example of the modulation audio frequency generated by *DigiPan* (you can use any other audio frequency in the transceiver passband) after setting the transceiver to SSB mode. Since we are using SSB, the carrier and lower sideband will be filtered out by the transmitter, leaving only the upper sideband. The phase shift points never move regardless of the audio modulation frequency used. This is an important point in understanding PSK31. The waveforms look almost exactly alike, and in fact could be a two tone test pattern used to test a sideband transmitter for linearity (see Figures 10 through 12).

The all-zeros pattern appears at a 31.25 Hz rate (phase shift to phase shift mark) regardless of the underlying audio modulation frequency. From amplitude modulation theory we know that the overall corresponding bit modulation frequency is 15.625 Hz (half the zeros' idle pattern). Hence we see the two rails at ±15.625 Hz with a bandwidth of 31.25 Hz. Now decoding and bandwidth questions have been solved, you have learned how to decode any PSK signal and you understand the railroad tracks. All pretty nifty and a great example of how computers and ham radio have become intertwined.

Why is the number 31.25 Hz critical? Typical sound cards use an 8 kHz sampling rate with the digital telephone standard using 8 bits per sample. Therefore,

$8 \text{ kHz}/2^8 = 8 \text{ kHz}/256 = 31.25 \text{ Hz}.$

This allows typing at up to 50 WPM, a little less than standard RTTY (60 WPM) but

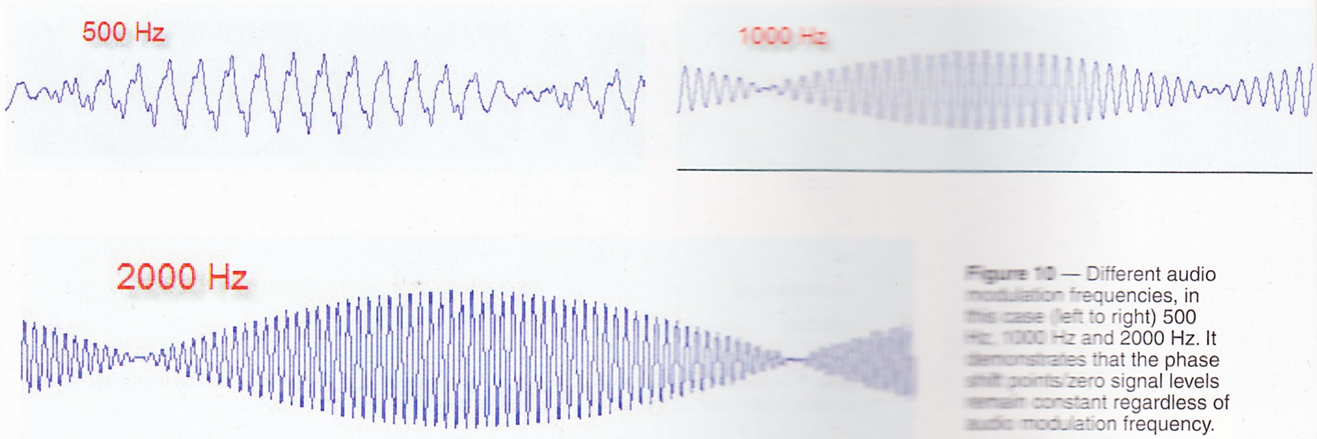

Figure 10 — Different audio modulation frequencies, in this case (left to right) 500 Hz, 1000 Hz and 2000 Hz. It demonstrates that the phase shift points/zero signal levels remain constant regardless of audio modulation frequency.

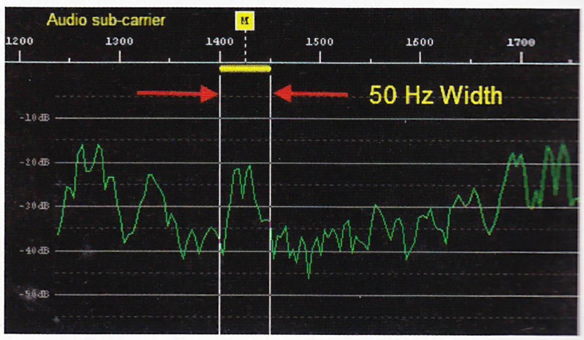

Figure 11 — Using the frequency spectrum scope from *Digital Master 780* part of *Ham Radio Deluxe*, available at [illegible], we clearly see the railroad tracks (the two peaks in the middle, 31.25 Hz apart) with this PSK signal.[7] This also shows the narrow 31.25 Hz bandwidth of a PSK31 signal.

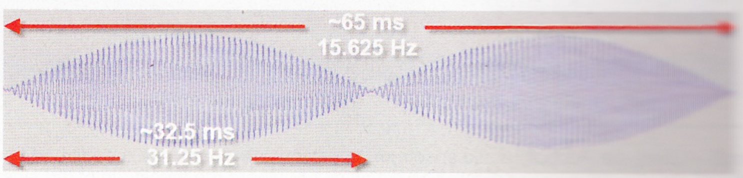

Figure 12 — Showing the two basic bit frequencies of 15.625 Hz and doubling that to 31.25 Hz.

more than adequate for most keyboard to keyboard contacts, fewer errors, less power and a narrower spectrum space. For those who are concerned about the lack of error correction capability of PSK31 you can look at QPSK or quadrature phase shift keying in which four different phase shifts are employed.[3, 11, 12] Want more speed? Try PSK63 with double the speed (~100 WPM) and double the bandwidth.[11] You will occasionally see these wider bandwidth railroad tracks while looking at PSK31 waterfall displays.

Thanks to my local friends Craig Deuby, NV5M, and Jim Garland, W8ZR, for their comments and support in the writing of this article. We hope you will join us for a PSK31 contact in the near future. With a ham radio transceiver, antenna and a few cables you should be quickly on the air using PSK31. Good luck!

Notes

[1]**www.qrz.com**.
[2]R. Cooke, G3LDI, *RTTY Basics*, [illegible] **gatearc.org/data/rtty**.
[3]Wikipedia article on PSK31, [illegible] **wiki/PSK31**.
[4]PSK31 home page, [illegible] SP9VRC, [illegible]
[5]PSK31, Peter Martinez, G3PLX, [illegible] **es/~jtpjatae/**[illegible]
[6]*DigiPan*, **www.digipan.net**.
[7]*CocoaModem*, [illegible]
[8]B Lafreniere, N6FN, [illegible] *Operations*. Available from your ARRL dealer or the ARRL Bookstore, ARRL order no. [illegible] Telephone 860-594-0355, or toll-free in the US 888-277-5289; [illegible] **pubsales@arrl.org**.
[9]S. Ford, WB8IMY, [illegible] Available from your ARRL dealer or the ARRL Bookstore, ARRL order no. [illegible] Telephone 860-594-0355, or toll-free in the US 888-277-5289; [illegible] **pubsales@arrl.org**.
[10]**audacity.sourceforge.net**.
[11]Wikipedia article about Phase Shift Keying, **en.wikipedia.org/wiki/Phase-shift_keying**.
[12]Wikipedia article about PSK63, [illegible] **org/wiki/PSK63**.

ARRL and QCWA Life Member Tom Brooks, KE1R, was first licensed in 1966. He worked for the FCC in San Francisco, then IBM in California, England, Virginia, Texas and Vermont. He holds BSEE and MSEE degrees, Professional Engineering licenses (Retired) from California and New Mexico, FCC General Class Radiotelephone License and FCC Second Class Radiotelegraph License with Ship Radar Endorsement. Tom has operated from New Zealand, Nepal and Siberia. Check QRZ.com for his latest address or use **ke1r@arrl.net**.

For updates to this article, see the *QST* Feedback page at www.arrl.org/feedback.

VOTE
Did you enjoy this article? Cast your vote at
www.arrl.org/cover-plaque-poll