ARRL Periodicals Archive – Search Results

A membership benefit of ARRL and the ARRL Technical Information Service

ARRL Members: You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

Need a higher quality reprint or scan? Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Information Service for assistance. Photocopies are $3 for ARRL members, $5 for nonmembers. For members, TIS can send the photocopies immediately and include an invoice. Nonmembers must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

QST on CD-ROM: Annual CD-ROMs are available for recent publication years. For details and ordering information, visit www.arrl.org/qst.

Non-Members: Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

QST Issue: May 1999
Title: PSK31 - Has RTTY's Replacement Arrived?
Author: Steve Ford, WB8IMY

Click Here to Report a Problem with this File

By Steve Ford, WB8IMY

PSK31—Has RTTY's Replacement Arrived?

There's a new HF digital mode in town and the gossip is flying!

I love RTTY. We have a warm relationship that goes back to my new-ham days in the early '70s when I was pounding green keys and reading conversations off rolls of yellow teletype paper. (Personal computer? What's that?) I jumped on the AMTOR bandwagon in the '80s and dabbled in CLOVER, G-TOR and PACTOR II in the '90s. Still, like most HF digital enthusiasts, I kept coming back to RTTY.

RTTY didn't have the error-free copy of the handshaking modes, and you certainly could not swap binary files using its limited code. But for contesting, DXpeditions and just casual conversation, RTTY was hard to beat. Unlike the 'TOR modes, you didn't have to worry about setting up handshaking links; type-and-transmit was the order of the day for RTTY. Conversations flowed easily and round-table discussions (and nets) were possible. During digital contests and DX pileups, RTTY became a mode of rapid-fire contacts. It wasn't as fast as phone, of course, but you could maintain a decent "rate."

No doubt you've noticed that I've been speaking of RTTY in the past tense. No, RTTY hasn't been tossed onto the ash heap of ham history—at least not yet. Something, however, has arrived that has the potential to knock RTTY out of the spotlight. That "something" is *PSK31*.

Prometheus Unbound

PSK31 was the brainchild of Peter Martinez, G3PLX. If the call sign seems familiar, you might recall Peter as the father of AMTOR. Up until relatively recently, PSK31 was the favorite of a small cadre of experimenters who used DSP development kits to put the mode on the air. That was all well and good, but it kept PSK31 in the shadowy corners of our hobby where few knew it existed. Like Prometheus bringing fire to the mortals, however, Peter blew the doors wide open by creating a *Windows* version of PSK31 that did all of its DSP magic using ordinary 16-bit PC sound cards. (The gods haven't yet bound him to a rock and summoned an eagle to eat his liver, but that remains to be seen!)

Not content with creating PSK31 for *Windows*, Peter placed it on the Web for *free* distribution to the global ham community. Talk about blasphemy! This meant that any ordinary ham could download the software and become active on PSK31 almost immediately.

In an article that appeared in *RadCom*, the journal of the Radio Society of Great Britain, Peter explained why he developed PSK31. Simply put, he wanted to create a mode that was as easy to use as RTTY, yet much more robust in terms of weak-signal performance. Another criteria was bandwidth. The HF digital subbands are narrow and tend to become crowded in a hurry (particularly during contests). Peter wanted to design a mode that would do all of its tricks within a very narrow bandwidth.

So What is PSK31?

First, let's dissect the name. The "PSK" stands for Phase Shift Keying, the modulation method that is used to generate the signal; "31" is the bit rate. Technically speaking, the bit rate is really 31.25, but "PSK31.25" isn't nearly as catchy.

Think of Morse code for a moment. It is a simple binary code expressed by short signal pulses (*dits*) and longer signal pulses (*dahs*). By combining strings of dits and dahs, we can communicate the entire English alphabet along with numbers and punctuation. Morse uses gaps of specific lengths to separate individual characters and words. Even beginners quickly learn to recognize these gaps—they don't need special signals to tell them that one character or word has ended and another is about to begin.

When it comes to RTTY we're still dealing with binary data (dits and dahs, if you will), but instead of on/off keying, we send the information by shifting frequencies. This is known as Frequency Shift Keying or *FSK*. One frequency represents a *mark* (1) and another represents a *space* (0). If you put enough mark and space signals together in proper order according to the RTTY code, you can send letters, numbers and a limited amount of punctuation.

The RTTYcode shuffles various combinations of five bits to represent each character. For example, the letter A is expressed as 00011. To separate the individual characters RTTY must also add "start" and "stop" pulses.

For PSK31 Peter devised a new code that combines the best of RTTYand Morse. He christened his creation the *Varicode* because a varying number of bits are used for each character. Building on the example of Morse, Peter allocated the shorter codes to the letters that appeared most often in standard English text. The idea was to send the least number of bits possible during a given transmission. For example:

E is a very popular letter on the English alphabet hit parade, so it gets a Varicode of 11.

Z sees relatively little use, so its Varicode becomes 111010101.

As with RTTY, however, we still need a way to signal the gaps between characters. The Varicode does this by using "00" to rep-

The BARTG PSK31-40 Award

As part of its 40th anniversary celebrations, BARTG, the British Amateur Radio Teledata Group, has announced the debut of the BARTG PSK31-40 Award. The award is available to amateurs who can prove that they have worked 40 DXCC entities using only the PSK31 mode. No crossband or crossmode QSOs will be allowed, and no single-band endorsements are offered. Applicants should supply a list of verified QSLs that must indicate PSK31 as the mode used. The award fee is $10 or 30 IRCs (no checks) to: Nigel Roberts, G4KZZ, BARTG Awards Manager, 13 Rosemore Close, Hunmanby, North Yorkshire YO14 0NB, United Kingdom.

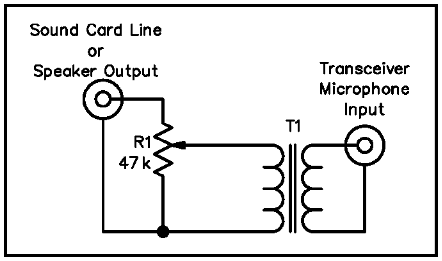


Figure 1—Use this attenuator circuit if you intend to feed the PSK31 transmit audio to your microphone jack. RadioShack part numbers are indicated below.

T1—1:1 isolation transformer (273-1374)
R1—47 kΩ trimmer potentiometer (271-283)

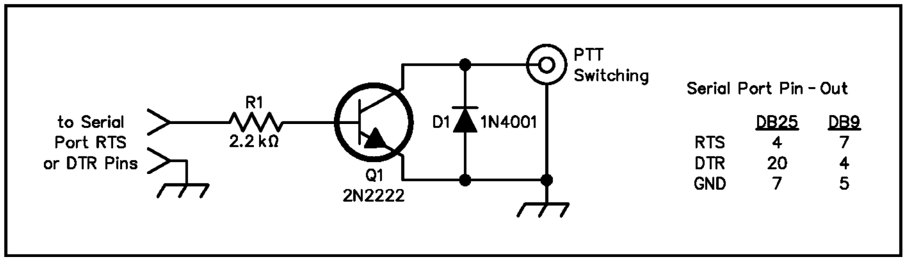


Figure 2—With this simple interface you can use a computer COM port to key your transceiver.

resent a gap. The Varicode is carefully structured so that two zeros never appear together in any of the combinations of 1s and 0s that make up the characters.

But how would the average ham generate a PSK31 signal and transmit Varicode over the airwaves? Peter's answer was to use the DSP capabilities of the common computer sound card to create an audio signal that shifted its phase 180° in sync with the 31.25 bit-per-second data stream. In Peter's scheme, a 0 bit in the data stream generates an audio phase shift, but a 1 does not. The technique of using phase shifts (and the lack thereof) to represent binary data is known as Binary Phase-Shift Keying, or *BPSK*. If you apply a BPSK audio signal to an SSB transceiver, you end up with BPSK modulated RF. (If you want the gory details, read the PSK31 software Help files.) At this data rate the resulting PSK31 RF signal is only 31.25 Hz wide, which is actually narrower than the average CW signal![1]

Concentrating your RF into a narrow bandwidth does wonders for reception, as any CW operator will tell you. But when you're trying to receive a BPSK-modulated signal it is easier to recognize the phase transitions—even when they are deep in the noise—if your computer knows when to expect them. To accomplish this, the receiving station must synchronize with the transmitting station. Once they are in sync, the software at the receiving station "knows" when to look for data in the receiver's audio output. Every PSK31 transmission begins with a short "idle" string of 0s. This allows the receive software to get into sync right away. Thanks to the structure of the Varicode, however, the phase transitions are also mathematically predictable, so much so that the PSK31 software can quickly synchronize itself when you tune in during the middle of a transmission, or after you momentarily lose the signal.

[1]It's interesting to note that the CW identifier, which is a part of the PSK31 software currently available, is limited to the same spectrum width as the PSK31 signal itself. This means that the CW generated by the PSK31 program is actually narrower than average CW. In other words, it is completely free of key clicks!

The combination of narrow bandwidth, an efficient DSP algorithm and synchronized sampling creates a mode that can be received at very low signal levels. PSK31 rivals the weak-signal performance of CW and it is a *vast* improvement over RTTY, as I discovered first hand.

Its terrific performance notwithstanding, PSK31 will not always provide 100% copy; it is as vulnerable to interference as any digital mode. And there are times, during a geomagnetic storm, for example, when ionospheric propagation will exhibit poor phase stability. (When you are trying to receive a narrow-bandwidth, phase-shifting signal, phase stability is very important.) This effect is often confined to the polar regions and it shows up as very rapid flutter, which is deadly to PSK31. The good news is that these events are usually short-lived.

From BPSK to QPSK

Many people urged Peter to add some form of error correction to PSK31, but he initially resisted the idea because most error-correction schemes rely on transmitting redundant data bits. Adding more bits while still maintaining the desired throughput increases the necessary data rate. If you double the BPSK data rate, the bandwidth doubles. As the bandwidth increases, the signal-to-noise ratio deteriorates and you get more errors. It's a sticky digital dilemma. How do you expand the information capacity of a BPSK channel without significantly increasing its bandwidth?

Peter finally found the answer by adding a second BPSK carrier at the transmitter with a 90° phase difference and a second demodulator at the receiver. Peter calls this quadrature polarity reversed keying, but it is better known as quaternary phase-shift keying or *QPSK*.

Splitting the transmitter power between two channels results in a 3-dB signal-to-noise penalty, but this is the same penalty you'd suffer if you doubled the bandwidth. Now that we have another channel to carry the redundant bits, we can use a *convolutional encoder* to generate one of four different phase shifts that correspond to patterns of five successive data bits. On the receiving end we have a *Viterbi* decoder playing a very sophisticated guessing game. Peter describes it best:

"The Viterbi is not so much a decoder as a whole family of encoders. Each one makes a different 'guess' at what the last five transmitted data bits might have been. There are 32 different patterns of five bits and thus 32 encoders. At each step the phase-shift value predicted by the bit-pattern guess from each encoder is compared with the actual received phase-shift value, and the 32 encoders are given 'marks out of 10' for accuracy. Just like in a knockout competition, the worst 16 are eliminated and the best 16 go on to the next round, taking their previous scores with them. Each surviving encoder then gives birth to 'children,' one guessing that the next transmitted bit will be a 0 and the other guessing that the next transmitted bit will be a 1. They all do their encoding to guess what the next phase shift will be, and are given marks out of 10 again that are added on to their earlier scores. The worst 16 encoders are killed off again and the cycle repeats.

"It's a bit like Darwin's theory of evolution, and eventually all the descendants of the encoders that made the right guesses earlier will be among the survivors and will all carry the same 'ancestral genes.' We therefore just keep a record of the family tree (the bit-guess sequence) of each survivor, and can trace back to find the transmitted bit stream, although we have to wait at least five generations (bit periods) before all survivors have the same great great grandmother (who guessed right five bits ago). The whole point is that because the scoring system is based on the running total, the decoder always gives the most accurate guess—*even if the received bit pattern is corrupted*. In other words, the Viterbi decoder corrects errors."—*Peter Martinez, G3PLX*

Operating PSK31 in the QPSK mode will give you 100% copy under most conditions, but there is a catch. Tuning is twice as critical with QPSK as it is with BPSK. You have to tune the received signal within an accuracy of less than 4 Hz for the Viterbi decoder to detect the phase shifts and do its job. Obviously, both stations must be using very stable transceivers.

Like most PSK31 operators, I tend to stick with BPSK, but while I was enjoying a conversation with Joe, K9BH, on 20 meters, he suggested that we give QPSK a try. After a brief pause, Joe called me in QPSK. With a gentle tweak of my transceiver's VFO and a click or two on the software AFC box, a yellow cross appeared in the tuning indicator and text began flowing across my screen. Despite considerable fading and noise, I didn't lose a single character of text during the 20-minute conversation that followed. Impressive!

What Do You Need to Get Started?

The first step is to jump onto the Web and download the latest version of PSK31 from **http://aintel.bi.ehu.es/psk31.html**. Make sure you download the latest version that's compatible with all sound card equipped PCs running *Windows* 95 and 98. Once you have it safely tucked away on your hard drive, install the software and read the Help files.

Assuming you have a reasonably stable HF SSB rig, you'll need to run two shielded audio cables between your transceiver and your computer. If your radio has an accessory jack that offers an audio line output, this is the preferred way to feed receive audio to your PC. Connect one shielded cable between the radio line output and the sound card's line *input*. If your radio does not have a line output, you'll have to use the external speaker jack.

For transmit audio, use another shielded cable and connect it between your sound card's speaker or line output jack and the accessory audio *input* of your transceiver. You can also opt to route the transmit audio to your microphone jack, but you'll need an attentuator similar to the one shown in Figure 1. If you use the accessory audio input, don't forget to disconnect your microphone before you go on the air. When you key the transceiver, the microphone may be "live," too!

And what about keying your transceiver? There are two approaches: Use one of your PC's COM ports and an interface like the one shown in Figure 2 to key your rig via the PTT line at your accessory jack. Or, simply switch on your transceiver's VOX and let it key the rig when it detects the transmit audio from the sound card.

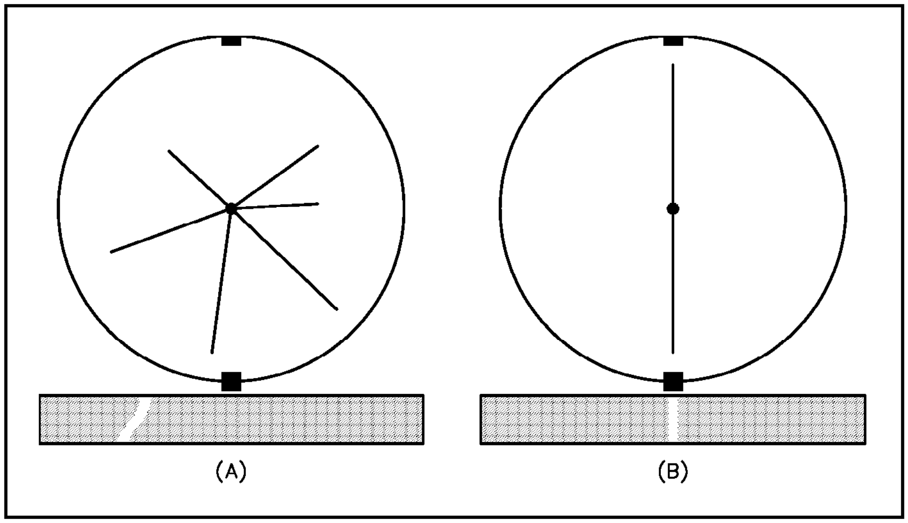


Figure 3—Tuning in a PSK31 BPSK signal. (In this diagram the tuning display has been enlarged for clarity.) As you tune across the signal, a white trace will appear (A). Just make a slight adjustment and the PSK31 signal is properly tuned (B).

Start the Software

You'll be delighted to discover that the PSK31 software is extraordinarily easy to use. As I've mentioned already, there are excellent Help files included, so I won't go into the software features in detail. We'll just hit the highlights and drop some useful tips along the way.

The first thing you must do is set the sample rate for your sound card. This isn't as complicated as it seems. The vast majority of sound cards support sample rates of 11025 Hz. All you have to do is type this number in the box that appears in the **SETUP** section. Leave the center frequency at 1000 Hz and add your call sign in the box below. In this section you can also choose which COM port you want to use for transceiver keying, or select "None" if you are using VOX.

Switch on your transceiver and adjust your on-screen-sound card volume controls. Make sure you know how to adjust the record sensitivity so as not to overdrive or underdrive the sound card input. You can usually do all this through the sound card mixer utility (this probably came with the sound card software, or may be part of *Windows*).

Listen First

Start by placing your radio in the upper sideband (USB) mode. That's the convention that's been adopted by most PSK31 users throughout the world.

PSK31 signals have a distinctive sound unlike any digital mode you've heard on the ham bands. You won't find PSK31 by listening for the *deedle-deedle* of a RTTY signal, and PSK31 doesn't "chirp" like the TOR modes. PSK31 signals *warble*—that's the best way I can describe them. They sound like high-pitched warbling carriers as you tune across them. (Until I learned to recognize the sound, I was tuning in computer-generated spurs and wondering why the text was nothing but garbage!)

Tuning in a PSK31 signal, like anything else, takes practice. If your rig has digital tuning (most modern radios do) select 1-Hz tuning steps or something close. If your radio will not tune in 1-Hz increments, you can use 10-Hz steps to get in the ballpark, then use your RIT control to zero in, or rely on the PSK31's automatic frequency control (AFC). You must tune slowly because PSK31 signals are narrow. If you become impatient, you'll sweep right past them! Begin your search on 20 meters, between about 14068 and 14080 kHz. That is where most of the PSK31 activity is concentrated, although you'll find PSK31 on other bands as well.

As you tune in a signal, watch the "waterfall" display immediately below the tuning circle. This display is analogous to an audio spectrum analyzer, continuously sweeping through a range of audio frequencies. Detected signals appear as white traces against the dark background, moving from top to bottom like water cascading over a fall. Your goal is to bring the white trace that represents a PSK31 signal to a point directly beneath the circle (see Figure 3).

As you center the signal in the waterfall display, you'll see that the flickering red bars in the tuning circle are suddenly switching to yellow and aligning themselves vertically. Keep tuning until the bars are as vertical as possible. By this time you should be seeing text on your screen. Congratulations!

You'll probably notice that the bars are starting to rotate slightly within the circle as you monitor the conversation. This is caused by drift—either the transmitting station is drifting or you are. No problem. Just click on the AFC box and the software will begin tracking the frequency changes.

Transmitting

Transmitting is actually easier than receiving, but before you begin, make sure you are not overdriving your radio by feeding too much audio from the sound card. Using your sound card volume controls, click on **TUNE** or tap the **F8** key and increase the card's output while watching your ALC indicator. Make this adjustment with the transceiver attached to a dummy load. If you must do it on the air, use the CW identifier and keep the transmissions short. When you see the ALC beginning to activate, stop (click on **TX OFF** or hit **F5**). That's all the audio you need.

You'll notice a long, narrow box just below the larger receive window. That's your outgoing text window. The instant you begin typing in this window, the software will switch your rig to the transmit mode. When you've finished (answering a CQ, for example), click on **TX OFF** or tap **F5** and you'll jump back to receive. After you've established contact, click on the **NET** box to allow the software to keep your transmit frequency on track with the receive frequency.

Sending a CQ is as easy as clicking on the **CQ** button or tapping **F7**. The software will take it from there and will return to the receive mode when it's finished. It's worth noting that you can also use the PSK31 software as a Morse CW keyboard. Just click on CW in the **MODE** section and start typing. (PSK31 does *not* decode CW, however.)

PSK31: A Testimonial

At the time this article went to press, I had been a PSK31 user for about two months. I've been running PSK31 using my ICOM IC-706 MkII transceiver and my end-fed long-wire antenna. To say the results have been impressive is an understatement!

My first PSK31 contact was with K8SRB on 40 meters. Stan was only running 25 W to a G5RV dipole antenna, but the text on my screen was virtually error free despite the high noise levels. We chatted for about 45 minutes and Stan passed along a wealth of PSK31 tips. For example, he demonstrated the need to keep your transmit audio at the proper level by momentarily overdriving his rig so I could hear the difference and see it on the waterfall display. (The splatter appeared on the display as vertical lines to the left and right of the center position.) Stan also clued me in on the importance of checking the **NET** box so that our transmit and receive frequencies would track each other. As Stan put it, "If we must drift, let's drift together!"

A few days later I had the spooky experience of being able to copy text from a signal I could not hear. I was tuning across the 20-meter digital subband when I saw a faint, ghostly trace on the waterfall display. I watched the display and tuned in the signal, but I could hear nothing recognizable in the noise. There may have been a warbling tone present, but I couldn't be sure. Suddenly the text began to print a CQ from WL7VO in Chicken, Alaska, near Fairbanks. I answered and was astonished to see his reply. The contact didn't last long because the band was dying fast, but it was captivating nonetheless. If we had been using RTTY a QSO would have been utterly impossible.

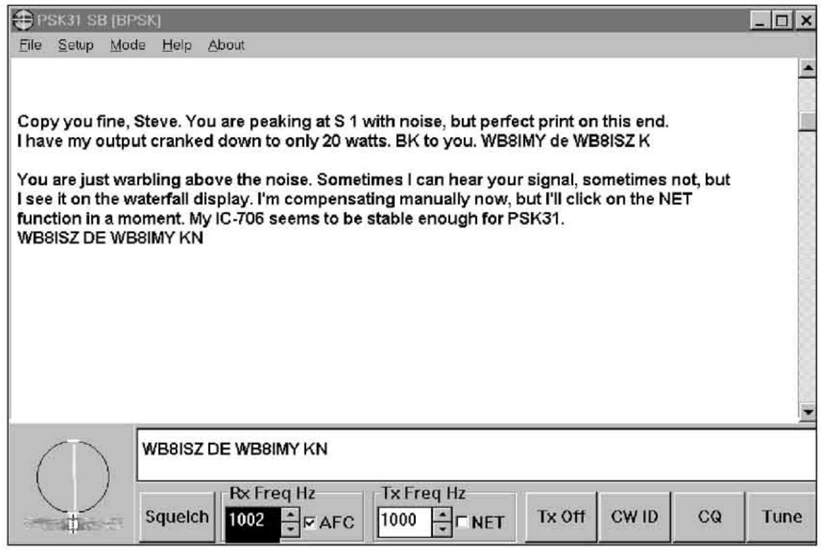
Since that time I've logged dozens of PSK31 contacts, including a fair amount of DX. The mode is catching on quickly in Europe, Australia and Japan. In fact, the activity level has risen to the point where you can find PSK31 signals on the air just about any time 20 meters is open. I've also been plumbing the depths of 80 and 160 meters, looking for PSK31 signals amidst the noise. The few contacts I've had on 160 have proven the power of PSK31. Despite Mother Nature's static crescendo I was able to copy readable text.

Is PSK31 the Heir to the HF Digital Throne?

The best answer is a strong "maybe." I'm not brave enough to stick my neck out and predict that PSK31 will overtake RTTY as the number one "live" HF digital mode, but it is off to a promising start. Even though Amateur Radio is a technological hobby, hams embrace change reluctantly. Will the graybeards of Baudot RTTY forsake their *deedle-deedles* for PSK31?

Well, this semi-graybeard is ready to make the switch. I love the weak-signal performance of PSK31. It seems far more suitable for HF than RTTY. I *will* predict that an upcoming DXpedition is going to travel with PSK31 software in one of its laptops. That much seems inevitable. And how long will it be before the first PSK31 contests appear?

In the meantime I'll be prowling the bands for those telltale warbles. Perhaps it is time for King RTTY to retire and yield to the heir apparent. As we enter the new millennium will PSK31 wear the crown? Time will tell!



Here is an actual PSK31 QSO in progress.

Is PSK31 Legal?

Some armchair lawyers have questioned the legality of PSK31 since its Varicode is not specifically mentioned as a "legal" digital code in Part 97. Some confusion is understandable, given the wording of 97.309(a). However, the FCC clarified the meaning of the rules in an Order released October 11, 1995 (December 1995 *QST*, p 84). The Order (DA 95-2106) reads in part: "This Order amends Section 97.309(a) of the Commission's Rules...to clarify that amateur stations may use any digital code that has its technical characteristics publicly documented. This action was initiated by a letter from the American Radio Relay League, Inc. (ARRL)."

The Order goes on to note that "The technical characteristics of CLOVER, G-TOR, and PACTOR have been documented publicly for use by amateur operators, and commercial products are readily available that facilitate the transmission and reception of communications incorporating these codes. Including CLOVER, G-TOR, and PACTOR in the rules will not conflict with our objective of preventing the use of codes or ciphers intended to obscure the meaning of the communication. We agree, therefore, that it would be helpful to the amateur service community for the rules to specifically authorize amateur stations to transmit messages and data using these and similar digital codes."

Given that PSK31 is in the public domain for amateur use, that software is readily and freely available, and that its emission characteristics clearly meet the standards of Section 97.307 for RTTY/data, there is little doubt that its use by FCC-licensed amateur stations is legal.

However, just to complete the documentation, in a letter to the FCC dated January 27, 1999, ARRL General Counsel Christopher D. Imlay, W3KD, documented the technical characteristics of PSK31 in a manner similar to how CLOVER, G-TOR, and PACTOR were previously documented. There is no need for PSK31 to be mentioned specifically in the rules, because CLOVER, G-TOR, and PACTOR are simply given as examples.—*David Sumner, K1ZZ*

Join the PSK31 E-mail Reflector!

Internet e-mail makes it easy to stay up to date with this rapidly developing mode. By joining the PSK31 reflector you'll receive the latest information, along with operating tips and other useful advice.

An e-mail reflector is like a group discussion. When mail is sent to the reflector, it is automatically redistributed for everyone else to read.

You can join the PSK31 reflector by sending an e-mail message to: **majordomo@aintel.bi.ehu.es**. In the subject field of your message enter your full name and call sign. In the message body simply type: **subscribe psk31**.

A special thanks to Peter Martinez, G3PLX, for his assistance in the creation of this article.
You can contact the author at 225 Main St, Newington, CT 06111; **sford@arrl.org**.

QST