Low Frequency Transmission Modes

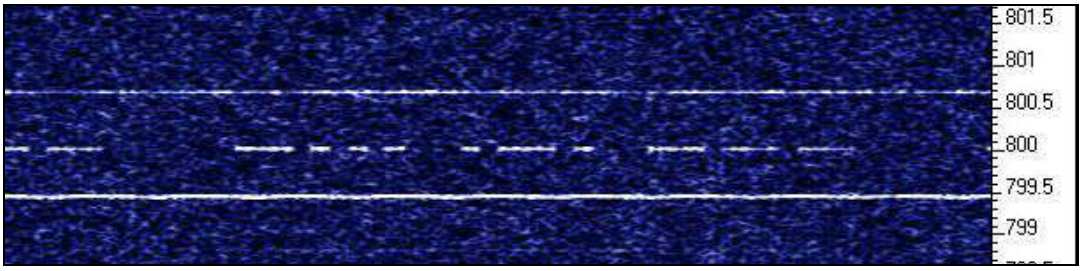
As with most radio communication, information may be transmitted at low frequencies by varying the amplitude, frequency and phase of a carrier wave. The limitations particular to this range include low signal levels due to under-sized transmitting setups, narrow frequency bands, and high noise levels. Approaches that work well at MF, HF or higher may be poorly suited for LF work. The 2200 meter band (135.7 - 137.8 kHz) is only 2.1 kHz wide, so voice work on SSB or AM is out. Some of the things that do work are listed below, along with comments and examples from actual receptions. Read on!

CW
DFCW
BPSK
WOLF
PSK
FSK
Graphical (Hell, etc.)
Chart of times to send a simple message

CW

You can't go wrong with plain old CW. It's simple to transmit, and simple to receive. It becomes the standard by which we can demonstrate the effectiveness (or lack of effectiveness) of the various other modes. As in the HF Amateur Radio bands, CW may be copied by ear from 5 words per minute (wpm) up to the limit of both operators' being able to pull it off. There is a very big "BUT" at LF, however: the lousy signal to noise ratio determines the highest speed you can copy, either by ear or by computer. You can improve the SNR by narrowing the bandwidth of your receiver, but there are practical limits as to how fast you can receive for a particular BW. With low radiated power and high noise levels working against you, you may not be happy with the distance to your radio horizon.

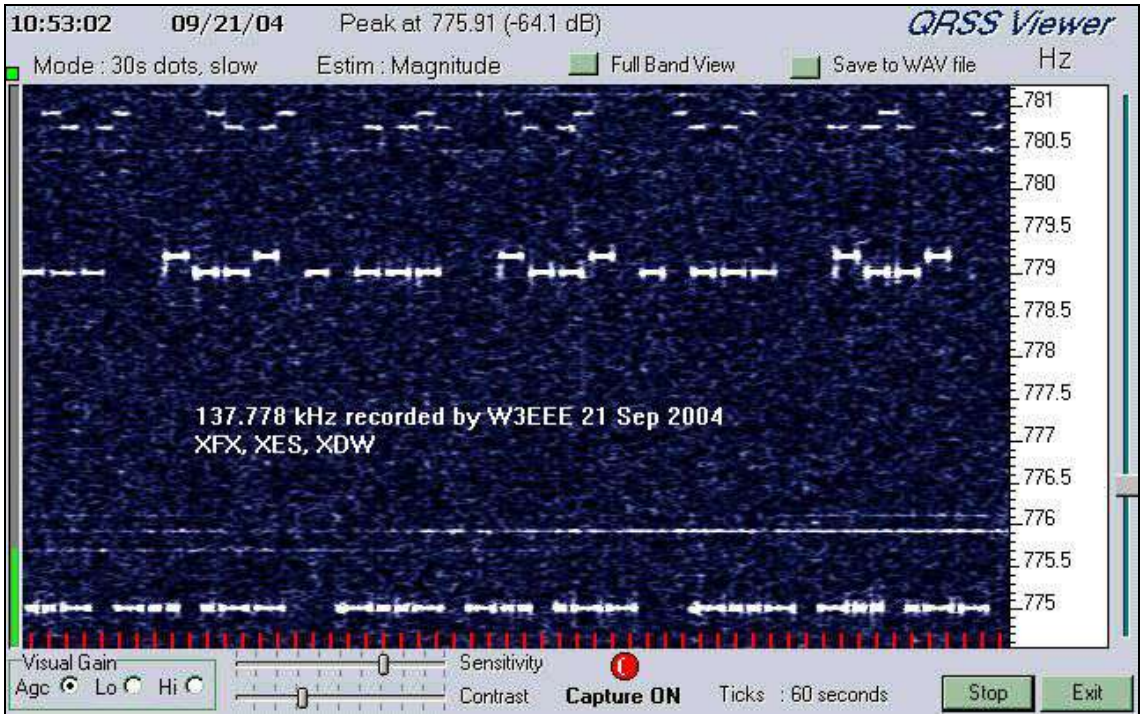
"QRS" is the shorthand for asking a CW operator to slow down. Interesting things begin to happen when you drop below 5 wpm or so. First, the receiving operator starts feeling sleepy. But assuming there is enough signal to copy, he may get better copy, as noise won't wipe out as many characters. If you continue to reduce the speed, the receiving guy will have trouble remembering if he's in the middle of a dash or dot, and things will fall apart. Computers to the rescue! It turns out that spectrum analysis software can be rigged to put nice dots and dashes on your screen, and the speed can be adjusted well below the limits of human concentration. You just stare at the screen, and see things print out slowly. And you quickly (or slowly) find that you can now visually copy slower signals that are buried in the noise and too weak to hear.



"QRSS" has become the way to refer to CW sent slowly enough that speeds are best expressed in seconds per dot. The fastest normal QRSS speed is 3 seconds/dot, and there have been some 240 seconds/dot transmissions! The

simplest program for QRSS is Argo by Alberto DiBene, I2PHD. The above Argo screen shot shows a 2 Hz slice of spectrum with the call sign of BRO, a part 15 beacon in MN, at a distance of 1100 miles. Each dot is 30 seconds long. An article on the effectiveness of QRSS in LF work may be found in the LWCA Library.

DFCW



"DFCW" stands for "Dual Frequency CW" and is a form of frequency and amplitude modulation. The International Morse Code is used, but dots and dashes are the same length. They are distinguished from each other by a small frequency shift which visually puts the characters on different lines on the screen. The screen shot above, shows two stations in DFCW mode, "XFX" near the top, and "XES" in the middle. In both cases, the dash characters are 0.2 Hz higher than the dots. The advantage to this mode, is that for a given dot speed, text may be sent more quickly. Actually, in the case of XES, which consists mostly of dots, the elements were set at 90 seconds, giving a very bold appearance. DFCW can be generated by "pulling" a crystal oscillator, but is usually done by switching between alternate frequencies in a synthesizer.

BPSK

```
AFRICAM log opened: Wed Dec  8 2004  at 02:20 Local Time
02:20:51  VD RAIN 45F VD RAIN 45F VD RAIN 45F VD RAx@ 45F VD RAIN 45F VD
02:29:03   RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45ChPD RAIN 45F VD R
02:37:19  AIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F Vh RAI
02:45:35  N 45F VD RAIN 45F VD rYINH3yF VD RAIN 45F VD RAIN 45F VD RAIN
02:53:51  45F VD RAIN 45R VD RAIN 45F VD RAIN 45F VD RAINpa5F VD RAIN 45
03:02:07  F VD RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F
03:10:23  GD RAIN 45F VD RAIN 45F VD RAIN 45F VD"RAII 45F VD RAIN 45F VD
03:18:39   RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F VD R
03:26:55  AIN 45F VD RAIy 45F VD RAIN 45F VD RAIN 45F VD RAIN 45F PD RAI
03:35:11  N 45F VD RuIN 45F VD RAIN 45F VD RAIC 45F VD RAIN 45F VD RAIN
```

"BPSK" stands for Binary Phase Shift Keying, and is a form of phase modulation. A constant frequency carrier is used, and its phase is flipped from 0 to 180 degrees to represent binary 1's and 0's. The normal ASCII character set is employed, with each character having one start bit, 7 data bits, one even parity bit, one stop bit, for 10 bits total. Unlike the Morse code, each character takes a fixed time to send. The visual output of BPSK is in text format, providing a clear record of reception. The repeating message shown above was received at a distance of 71 miles from Part 15 beacon VD, in Burlington, CT. The format was MS1000, where the "1000" indicates the number of milliseconds per bit. The most commonly used program for generating and receiving BPSK in LF work is AFRICAM, a DOS-based program by VE2IQ. Options for improved weak-signal performance include error correction and "GRAB," which allows repetitive text to be built up over a number of passes. The current version of AFRICAM also supports GPS synchronization, whereby the characters are sent out in step with 1 pps pulses from a GPS receiver. This allows the AFRICAM decoder to know when incoming bits will be received, and greatly speeds up the process of synchronizing to the signal.

WOLF

"WOLF" (Weak-signal Operation on Low Frequency) is the brainchild of Stewart Nelson, KK7KA. It is a specialized form of BPSK, with a data rate of MS100 (10 bits per second). Instead of sending individual characters, WOLF takes a 15-character message, heavily encodes it with error correction, and forms a 960 bit package which takes 96 seconds to send. The WOLF decoder analyzes each package and the ones that preceded it, up to a maximum of 27.2 minutes. The program prints a decoded output for each 96-second section, and the goal is to build up to at least one line of correct copy before the end is reached. Here is an example of a message transmitted by Part 15 1-watt beacon "TAG" in Massachusetts, and received by W4DEX in North Carolina: `C:\WOLF>wolf -q tag1700.wav -f 2000 -r 8000.48`

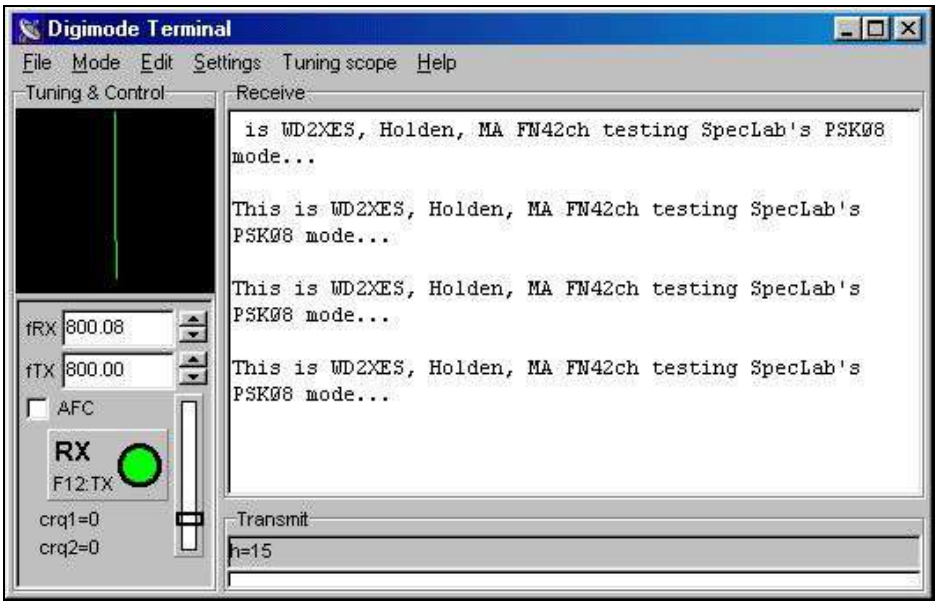
```
WOLF version 0.61
t: 24 f:-0.515 a: 0.1 dp:116.8 ci:12 cj:360 ULN*A*PRSP09AXW ?
t: 48 f: 0.449 a: 0.4 dp:114.4 ci:12 cj: 53 9BY*NB6*HO XWP9 ?
t: 96 f: 0.519 a:-0.2 dp:111.0 ci:12 cj: 93 72 OX9/*H???LKM ?
t: 192 f: 0.752 pm: 680 jm:496 q:-13.2 -9.5 M5Y.6PCFZ33G TV ?
t: 288 f: 0.752 pm: 898 jm:496 q:-12.0 -8.5 YDH6.8???RZR59* ?
t: 384 f:-0.381 pm: 938 jm:288 q:-10.1 -8.1 G27JOWQHB8DLDAW ?
t: 480 f: 0.010 pm: 1442 jm:691 q: -6.2 -9.0 63D1F4.*WBWISD6 ?
t: 576 f: 0.010 pm: 1669 jm:691 q: -5.1 -8.0 O4HB6NK94PBSM5D ?
t: 672 f: 0.010 pm: 2318 jm:691 q: -4.1 -5.3 NC TAG RO 768 -
t: 768 f: 0.010 pm: 3038 jm:691 q: -3.4 -4.7 NC TAG RO 768 -
t: 864 f: 0.010 pm: 3602 jm:691 q: -2.5 -4.4 NC TAG RO 768 -
t: 960 f: 0.010 pm: 4051 jm:691 q: -2.2 -3.7 NC TAG RO 768 -
t:1056 f: 0.010 pm: 4158 jm:691 q: -2.0 -3.5 NC TAG RO 768 -
t:1152 f: 0.010 pm: 4540 jm:691 q: -1.5 -3.1 NC TAG RO 768 -
t:1248 f: 0.010 pm: 4883 jm:691 q: -1.1 -2.7 NC TAG RO 768 -
t:1344 f: 0.010 pm: 5094 jm:691 q: -0.8 -2.6 NC TAG RO 768 -
t:1440 f: 0.010 pm: 5431 jm:691 q: -0.6 -2.0 NC TAG RO 768 -
t:1536 f: 0.010 pm: 5495 jm:691 q: -0.4 -1.8 NC TAG RO 768 -
t:1632 f: 0.010 pm: 5726 jm:691 q: -0.2 -1.4 NC TAG RO 768 -
```

In this case, correct copy was received after 672 seconds, and the signal continued to build from there. In its present

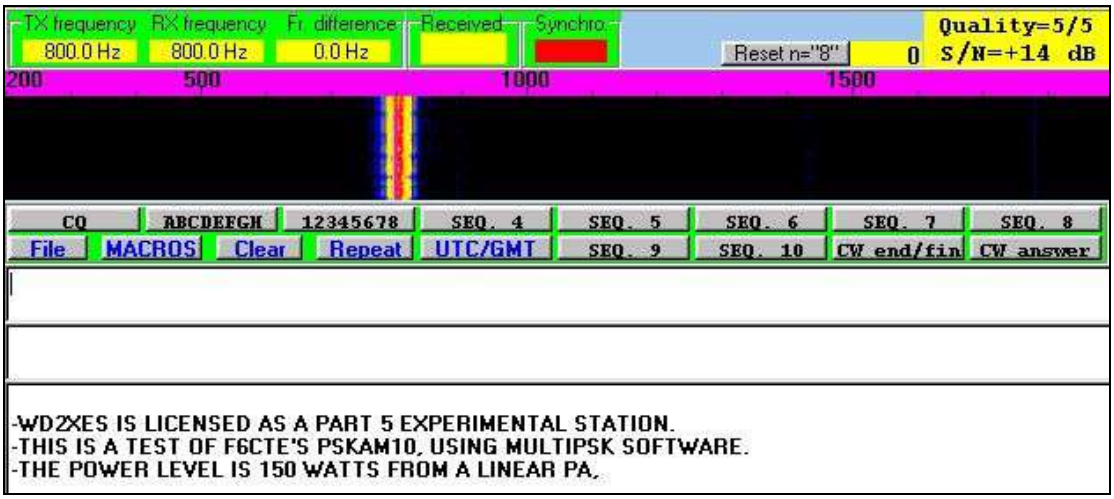
form, WOLF does not work in real-time. The receiving operator records a .WAV file and processes it with WOLF in a DOS box under Windows. The above example was taken from a several hour daytime QSO on 185.85 kHz done in early 2004. A report of that QSO is available. While the present WOLF program is a bit daunting, the mode is capable of producing text-based communication with signals that would be just copyable in QRSS with 60-second dots.

PSK

"PSK" is Phase Shift Keying, and uses the same 0/180 degree phase switching as BPSK. In Amateur Radio parlance, "PSK" has come to mean BPSK with a variable-length character set developed for PSK31 by G3PLX. PSK uses shorter codes for the more frequently used characters, and this speeds things up. There are a zillion programs that generate and receive PSK31, and that mode does work at LF, though the coverage isn't much better than you can do with regular speed CW. Most of those programs have a sound card interface to feed an SSB exciter for transmitting. That's the norm at HF, but not at LF, where on/off keyed, non-linear transmitters are prevalent.



One program that provides alternative keying is DL4YHF's Spectrum Laboratory, which has a rich collection of audio processing features. SpecLab's Digimode Terminal section offers PSK31, PSK08, PSK02 and PSK01, all based on the same character set, but with increasing bit lengths. The screen shot above shows the Digimode Terminal in the process of receiving a PSK08 signal. Part 15 stations running 1 watt don't need to worry much about occupied bandwidth, and the 0/180 degree keying for WOLF or SpecLab's PSK can be done with an XOR gate controlled by an audio detector connected to the PC sound card. For higher power operation, a shaped keying waveform is desirable, and can be done by driving an SSB exciter from the sound card, and using a linear amplifier.

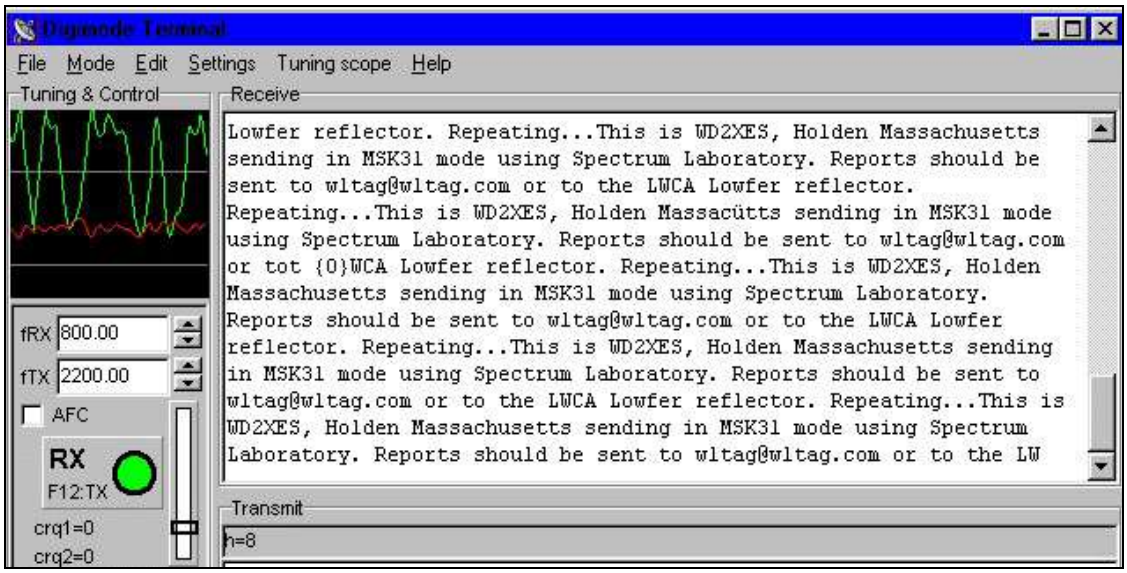


F6CTE's popular MultiPSK program contains a specialized mode of interest to LF experimenters. PSKAM10 is an MS100 (PSK character set) mode that has considerable error correction, but is still quick enough to support a 2-way contact. MultiPSK only supports a sound card interface.

FSK

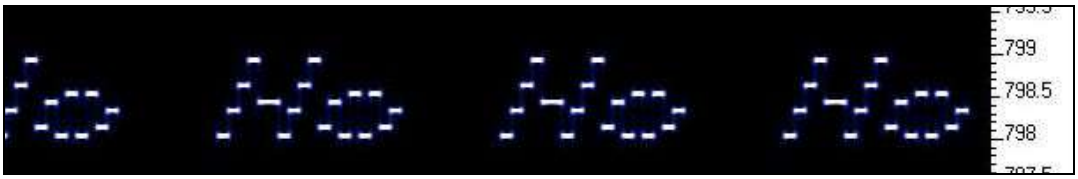
Frequency Shift Keying is common at HF for teletype (RTTY) communication, but the typical frequency shifts are much too wide to be useful at LF. Additionally, RTTY really isn't a weak signal mode, and would be useful only for local communication. The most common FSK modes among experimenters are DFCW (already mentioned above), JASON, and MSK.

Jason is another of I2PHD's creations. The transmitted frequency is varied in steps, with 16 possible values of frequency shift. The absolute frequency is not important; only the difference in frequency between successive tones. Two tone shifts are required for each of the 64 possible characters. JASON originally transmitted tones for 11.9 seconds, with all of the tones fitting into a 4 Hz bandwidth. Newer options allow 6 tone lengths in conjunction with three sets of tones. The slowest version was used for the screen shot above, which was taken by Hartmut Wolff in Germany while monitoring the 150 watt WD2XES signal (sending Happy New Year) from Massachusetts. In this case, the entire range of tones fit within a 0.5 Hz bandwidth, and the each tone was 95 seconds long. Slow, but very effective! Jason's transmitter interface may be done with a sound card, or by commanding a synthesizer to make the appropriate frequency jumps.



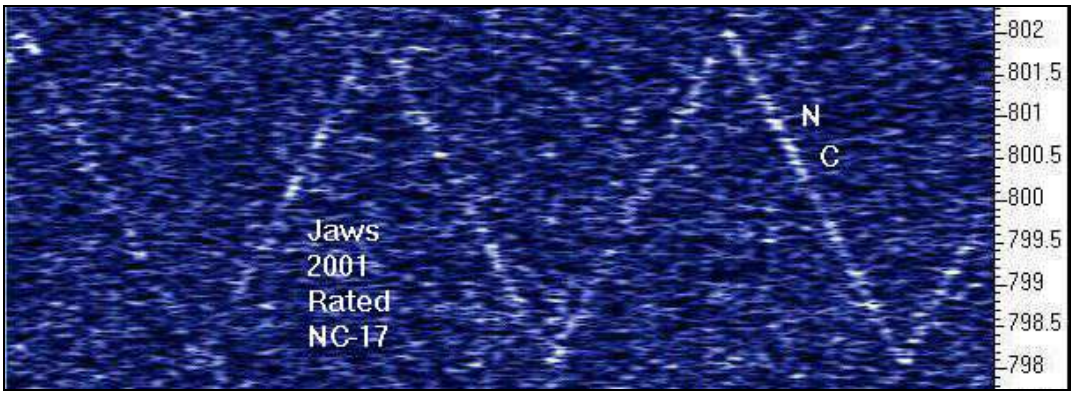
MSK (Minimum Shift Keying) is a specialized form of FSK, where the frequency shift is kept as small as possible. This is 0.5 X (bit rate). DL4YHF's Spectrum Laboratory has MSK08 and MSK02 modes in the DigiMode Terminal. They have proven to be useful for relatively local communication, and they do not require a linear power amplifier system. Bill de Carle, VE2IQ, has an MSK option in his AFRICAM program. At present, it requires GPS sync at both ends. The decoders in both programs are under development, so further improvements may be forthcoming.

Graphical (Hell, etc.)

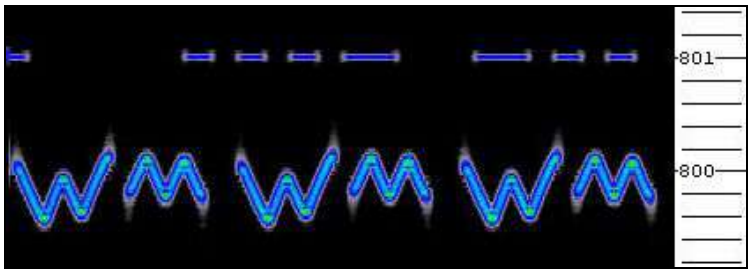


"Hell" (short for Hellschreiber) is a graphical mode that allows you to print characters on a spectral analysis screen normally used for receiving CW or QRSS signals. For a non-linear PA, fonts must be developed that only transmit one frequency at at time. At LF, versions of Hell are used that take up less bandwidth than at HF.

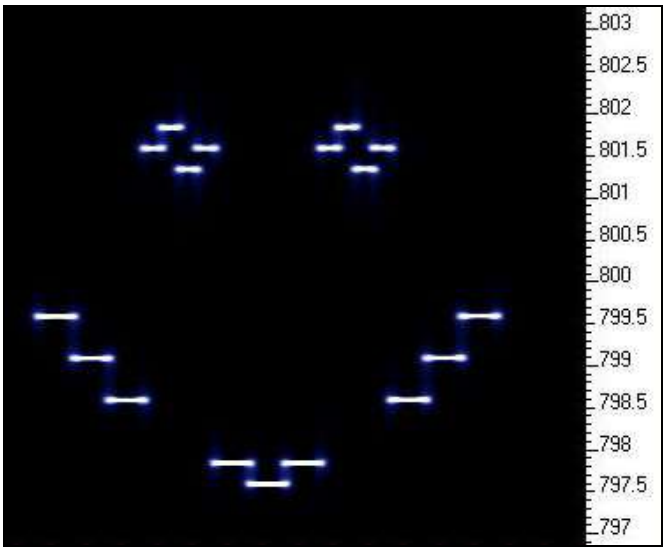
This much cooler form of Hell is "Chirped Hell," developed by DF6NM specifically for LF use, and is found in Spectrum Laboratory . The transmission above was taken from a screen grabber operated by DF6NM. A linear transmitter is required for Chirped Hell.



This triangle waveform actually consists of the letters "NC" sent in QRSS while the frequency is varied. Triangle and sine wave patterns have been used as attention-getters, as the eye seems to resolve them from the background noise, even with very weak signals.



Proving that purely analog means can be used to send graphical characters, Bill Ashlock's "Wm" Part 15 beacon uses the VCO voltage from NE555 timers to pull the transmitter crystal frequency.



And on Halloween, almost anything is possible.

Chart of times to send a simple message

It would be nice to know the implications of choosing a particular mode when your goal is exchanging information during a QSO (as opposed to repetitive beaconing). To give a sense of the time needed to send a simple message, I chose the phrase "WD2XES FN42CH ". Note the space after the "H", as it figures into the calculations. Here are my best estimates for the various combinations of modes and speeds:

Mode Timings				
Mode	**Program**	**Hrs**	**Mins**	**Secs**
PSK31	Spectrum Lab	00	00	04
CW (12wpm)	Any (or manual)	00	00	09
PSK10	MultiPSK	00	00	13
PSK08	Spectrum Lab	00	00	16
CW (6wpm)	Any (or manual)	00	00	20
JASON Fast Turbo	JASON	00	00	21
WOLF10 (Line 1)	DL4YHF WOLF	00	00	24
PSKAM10	MultiPSK	00	00	26
JASON Fast	JASON	00	00	42
WOLF10 (Line 2)	DL4YHF WOLF	00	00	48
PSK02	Spectrum Lab	01	01	04
WOLF10 (Line3)	DL4YHF WOLF	00	01	36
PSK01	Spectrum Lab	00	02	08
Chirped HELL (3sec)	Spectrum Lab	00	02	19
JASON Normal Turbo	JASON	00	02	47
Sequential MT Hell (0.5str/sec)	SpecLab	00	03	10
WOLF10 (Line 4)	DL4YHF WOLF	00	03	12
DFCW3	Several	00	04	03
Chirped HELL (10sec)	Spectrum Lab	00	04	40
WOLF10 (Line 5)	Spectrum Lab	00	04	48
JASON Normal	JASON	00	05	33
WOLF10 (Line 6)	DL4YHF WOLF	00	06	24
QRSS3	Several	00	07	21
WOLF10 (Line 7)	DL4YHF WOLF	00	08	00
WOLF10 (Line 8)	DL4YHF WOLF	00	09	36
WOLF10 (Line 9)	DL4YHF WOLF	00	11	12
WOLF10 (Line 10)	DL4YHF WOLF	00	12	48
DFCW10	Several	00	13	30
WOLF10 (Line 11)	DL4YHF WOLF	00	14	24
WOLF10 (Line 12)	DL4YHF WOLF	00	16	00
WOLF10 (Line 13)	DL4YHF WOLF	00	17	36
WOLF10 (Line 14)	DL4YHF WOLF	00	19	12
WOLF10 (Line 15)	DL4YHF WOLF	00	20	48
JASON Slow Turbo	JASON	00	22	13
QRSS10	Several	00	24	30

DFCW30	Several	00	40	30
JASON Slow	JASON	00	44	26
QRSS30	Several	01	13	30
DFCW60	Several	01	21	
DFCW90	Several	02	02	
QRSS60	Several	02	27	
QRSS90	Several	03	41	

Back to Home