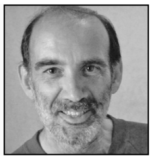
20 Digital Communications



Andy Talbot, G4JNT

First, a little history. Amateurs have been transmitting data, usually typed or printed text ,to each other for many decades. The first datacomms used mechanical teleprinters - a sort of early printer with a mechanical keyboard that generated a code for each letter made up of five bits. These bits were then transmitted serially using one of two tones to represent a 'one' or a 'zero'. Decoding was by purely mechanical means with a constant speed motor and cams, and much time was spent by the operators maintaining and oiling their machines. The use of teleprinters led to the term RTTY, for Radio TeleTYpe. RTTY is still in widespread use on the amateur bands today, although most now use computers for generating and receiving the signals

Once computers had become established in many shacks, but before the advent of the Internet, a widespread network of interconnected stations and nodes allowed operators to send messages to each other and to exchange files. The Packet Radio network allowed one station to 'connect' to another station by specifying the callsign, and the network was able to correctly route traffic through multiple nodes around the country and the world. In its heyday, the Packet network was a reasonably reliable messaging system, although in busy periods it could take tens of minutes or even hours for some messages to get through.

Once the Internet came along to do the same job a lot quicker and more robustly, the packet network mostly died, although a few nodes and stations can still be found. Packet radio still has a place on a more local and personalised level for emergency communications where it allows short uncorrupted messages to be routed over locally set up, ad hoc, radio networks when other infrastructure has failed

With the advent of the personal computer (PC), the whole amateur data communications field has grown to encompass a lot more than exchanging hand typed messages. What has made this possible is the rapid growth in what are usually referred to as ‘soundcard modes’. The computer's audio ports are interfaced to the radio so that the PC generates audio that is subsequently upconverted to RF, usually using the radio in SSB mode. On receive, audio is fed into the PC and after digitisation of the analogue waveform, the power of Digital Signal Processing is used to do all of the signal processing. transmit/receive switching is also usually placed under computer control with another interface line, allowing datamode operation to take place without having to make regular changes on the radio.

Details of computer interfacing, including a number of pitfalls for the unwary, are given in the Computers in the Shack chapter.

Before going on to look at some of the modern digital communications techniques in detail, we will need to take a look at some of the theory behind the transmission of data over a radio link.

DIGITAL COMMUNICATIONS PRINCIPLES

Modulation

Data consists of binary information, usually referred to as 'ones' and 'zeros’. To send these over a radio link we need to modulate the RF carrier in a way such that the receiver can differentiate between a 1 and a 0 being sent. The easiest and most obvious way is to switch a carrier on for a '1' and off for a '0' and is known as Amplitude Shift Keying or ASK. The modulation rate, or the rate at which the bits change, is usually referred to as the Symbol or Baud Rate. For simple modulations like ASK and FSK, Baud rate is equal to the Bit rate

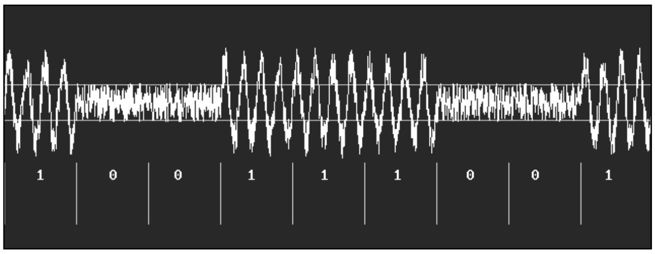


Fig 20.1: Amplitude Shift Keyed (ASK) carrier with added noise

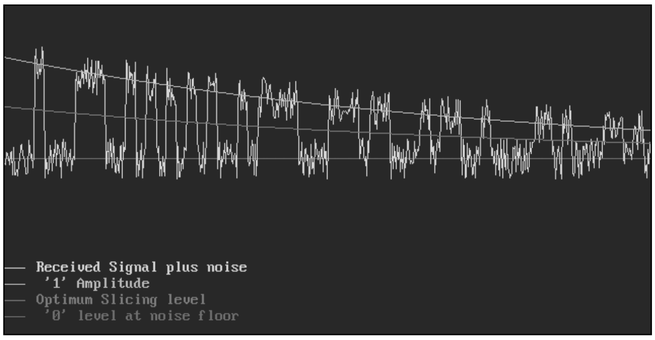


Fig 20.2: Fading amplitude shift keyed (ASK) signal

The problem with ASK is knowing where to make the decision for 0 or 1. Where the signal is varying in amplitude (fading) this 'slicing' or decision level, which is usually placed half way between zero and the maximum received amplitude, will have to move correspondingly. Automatic gain control (AGC) can be used to track the amplitude, but the AGC time constant will have to match the fading on the RF channel, and in addition we need to know the number of ones and zeros being transmitted as this will affect the average signal strength. A '0' will consist only of noise, which may itself have large random spikes on it that could mistakenly be measured as a '1'. Fig 20.1 shows a typical ASK signal corrupted with noise. The optimum slicing level for the best 0 / 1 decision in the presence of fading can be seen in Fig 20.2.

Frequency Shift Keying, or FSK, overcomes the amplitude threshold problem. The simplest route is to use two tones and switch them alternately - one tone for a '1', and the other tone for a '0'. Demodulation is then performed by looking at each tone separately and comparing the relative amplitudes of the signals, including added noise, to decide which element was sent. This particular technique is know as Frequency Exchange Keying, or wide shift Frequency Shift keying (W-FSK) and relies on the two tones being widely spaced in frequency. This is only the case if the data rate is significantly less than the difference between the two tones because the frequency spread from each tone being switched causes the sidebands from each to merge as data rate rises. RTTY with data at 45 or 50 bits per second (bit/s) and 170Hz shift is a typical example of this mode.

Wide shift FSK has a poor bandwidth efficiency, using significantly more width than is needed to send the data. This is not very socially acceptable in the congested amateur bands! The way to lower bandwidth is simply to use a lower frequency shift for a given data rate. This has the effect of merging the sidebands from the two tone frequencies into one continuous spectrum. It means that using two tuned circuits to separate out which tone was transmitted no longer works. By converting the varying audio frequency to a voltage it can be fed to a comparator to decide whether a One or Zero was sent. The Phase Locked Loop is the most popular means of doing this in hardware with other techniques used in DSP based demodulators. A PLL generally needs a higher signal to noise ratio to stay locked and to track the changing frequency reliably than a two tone demodulator would have been able to cope with.

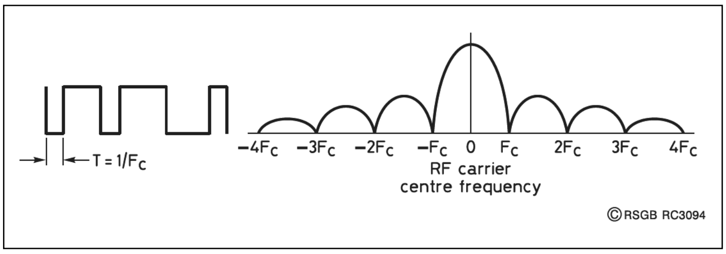


Fig 20.3: Keying sidebands of hard-switched PSK

Narrow shift FSK was employed with good results in the early days of telephone modems where originally 300bit/s was available using two tones with a shift of 200Hz. 1200bit/s with either 800 or 1000Hz shift followed shortly and this worked reliably enough over the 25 - 40dB S/N capability of the analogue telephone network in its day. The 1200bit/s standard, with its switching sidebands, takes up much of the available 3kHz audio bandwidth, and data rate could not be further raised by using FSK techniques. Simple low cost modem chips appeared to cater for these modes and the packet radio community adopted these telephone standards in the 1980s due to their simplicity of implementation.

Unfortunately, the greater S/N requirement for Narrow Shift FSK means that, for radio communication, high error rates frequently occur. If an FM link is used for transmission of the two audio tones, the inherently good quality of this mode when signals are strong allows Narrow Audio-FSK to work successfully for short range local contacts. The vast majority of the Packet network relies on this medium of 1000Hz shift audio carrying 1200bit/s data in a single FM channel, but it is hardly an efficient modulation scheme. On HF, attempts to use 200Hz shift for 300bit/s data were fraught with failure and lost data; HF packet forwarding was for some time limited to the 50bit/s or less of AMTOR or RTTY. A better way of modulating an RF carrier is needed if we are to improve on this.

FSK does, however, have its uses where frequency scattering is prevalent. This is often the case with narrowband amateur microwave links, and when using very narrow bandwidths at HF.

MSK, or Minimum Shift Keying is a special case of FSK where the shift is exactly half of the symbol rate. A Gaussian filter is often employed to shape the frequency transition and reduce the bandwidth of the signal, leading to the term GMSK. GMSK is used on the GSM Mobile phone network, and in the G3RUH 9600 Baud packet radio modem and some amateur satellite links. GMSK requires coherent decoding to work properly and a quite complex decoding algorithm if the best use is to be made of the mode.

Coherent Schemes

A solution to improving the bandwidth efficiency and S/N performance is Binary Phase Shift Keying, BPSK, often just referred to as PSK. In its simplest form the RF carrier is reversed in phase by 180 degrees for each '1' bit of data and zero degrees for a '0' bit. This is accomplished very simply by multiplying the RF carrier by 1 or -1 (for a '0') in a balanced modulator. It is obvious that at the receiver there will be an ambiguity as to which phase shift is a '1' and which corresponds to a '0'. One way around this is to transmit a preamble of a known pattern of '1's and '0's. If the preamble is received inverted then we know the whole demodulation process is upside down and needs to be swapped.

The other way is by differential coding. Instead of transmitting the absolute phase, a '1' is sent by changing the phase from one bit interval to the next, for a '0' the phase is not changed. Therefore a string of zeros will appear as an unchanging plain carrier, and a string of '1's as a repeated pattern of 0 / 180 degree phase changes.

The merits of coherent schemes can best be appreciated by looking at their analogue counterparts, comparing SSB voice with plain AM. SSB generally gives about 10dB advantage in noise over AM but at the expense of a more complicated receiver demodulator having to reinsert the carrier. PSK shows a similar advantage over FSK or ASK.

The spectrum of PSK depends on the symbol rate, and if the phase is hard switched then keying sidebands can extend to some considerable distance either side, as shown in **Fig 20.3**. To keep the bandwidth within reasonable limits, the amplitude of a PSK waveform is ramped at the transition point so the actual phase change occurs at zero amplitude as illustrated in **Fig 20.4**.

The rate of ramp is a trade off between bandwidth and S/N degradation due to loss of net energy in the entire symbol interval. PSK31 is an example at one extreme of this trade-off, where the entire symbol is shaped to a half sine-wave at the signalling rate. This leads to a very narrow bandwidth equal to the data rate, but sacrifices a couple of dB in S/N over a hard switched version with high levels of 'keyclicks'

Decoding PSK

PSK is more complicated to decode than is ASK or FSK with particular areas of complexity being those of carrier and bit clock regeneration, as well as controlling the spectrum of the signal. To demodulate a PSK signal, it is necessary first to generate a local oscillator - equivalent to the BFO in an SSB receiver - in order to mix with the incoming signal in order to recover the phase shift information. Unlike SSB, however, it has to be phase locked to the incoming waveform otherwise the recovered phase would slowly drift at the difference frequency, making resolution of the 0/180 degree phase shift impossible to achieve.

One way of doing this is to use a special phase locked loop (PLL) which can lock up to the signal when it is in either of its two phase states by squaring (frequency doubling) the incoming waveform which has the result of generating a constant phase continuous carrier at twice the input frequency. This can then be filtered out, divided by two and used to lock a PLL to recover the transmitted phase information. See **Fig 20.5** Another way is to

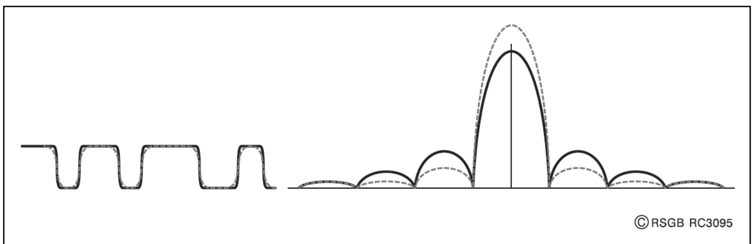


Fig 20.4: Reducing PSK bandwidth by reducing the amplitude at the phase transition point

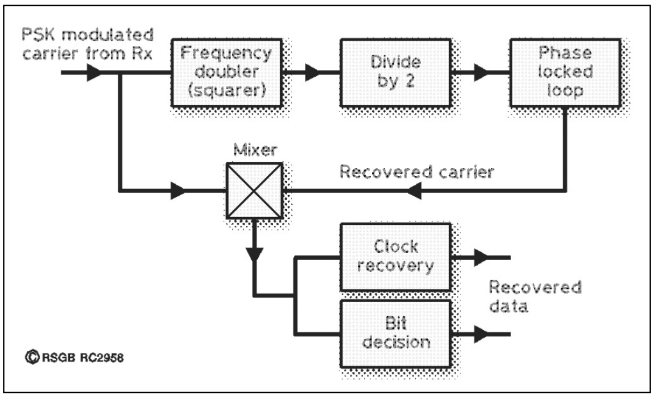


Fig 20.5: The first of two decoding schemes for PSK modulation. Suitable for non-differentially-coded signals. It uses a frequency-doubler to resolve the 0° / 180° ambiguity before regenerating the carrier via a phase-locked loop

use a variant of the third method SSB generator, called a Costas Loop, where a PLL works on the In-phase and Quadrature components of the signal and maintains lock, with the added bonus of demodulation as an inherent part of the carrier recovery. Costas loops are rather complex to build in hardware, but can be more easily programmed using DSP.

For differentially coded PSK demodulation of the binary data is slightly easier without having to resort to a PLL. Provided the incoming frequency is known with sufficient accuracy, by comparing the instantaneous signal with that received exactly one bit interval earlier using a mixer, the differentially modulated code can extracted (Fig 20.6) Provided the frequency error is low enough such that the phase drift during a bit interval is significantly less than the 180 degree data change, errors are minimised. To detect a 180 degree phase shift we need an error of less than 90 degrees. To achieve this phase shift during one bit interval the frequency needs to be set to within 90°/ 360° of the signalling rate. So, for a 300 bit/s DPSK waveform, the frequency has to be known to within 75Hz. For noisy signals, higher accuracy is needed and a figure of a tenth the bit rate is usually taken as a rule of thumb. The delay and comparison needed for DPSK has to be made at the bit transition point, where the phase may or may not have changed, so the data clock needs to be recovered for optimal decoding. In most cases a clock is also needed in order to be able to feed the recovered data into subsequent processor circuitry.

Clock regeneration is probably the most important part of a PSK demodulator, as the accurate timing needed for optimum demodulation of noisy signals is derived directly from this. The clock not only has to be locked to the frequency of the data bits,

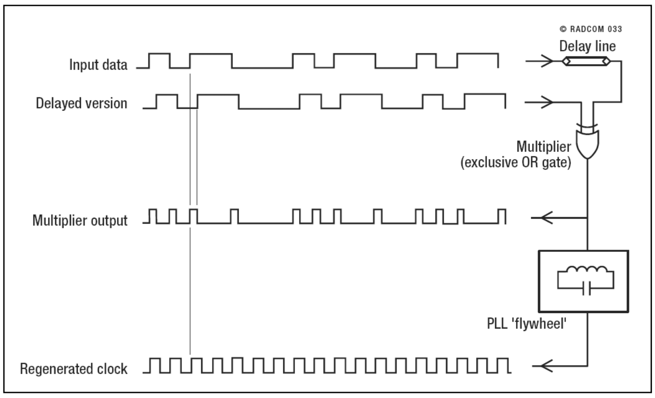


Fig 20.7: Clock regeneration from a random data stream

it has to be synchronous with the transitions so maximum use can be made of the full symbol period for making the vital 1/0 decision. Any clock jitter here degrades demodulation as the precise changeover point cannot be found.

One method is to use a similar technique as that for carrier recovery, a non linear operation such as a pulse generator triggered from transitions or zero crossings, followed by a narrow bandwidth Phase Locked Loop to extract just the clock signal from the jittery pulse output. Fig 20.7 shows the clock extraction. Other techniques that can be used are to examine the amplitude of filtered or band-limited PSK. The waveform will have a maximum at the middle of the symbol period and knowledge of the time at which the maximum occurs allows a loop to be locked. A simpler technique is just to look for any phase transition, assume this is at the correct point and synchronise a locally generated clock from there. Periodical comparisons can be made to keep the clock locked. Other more complex schemes are possible, including those that combine clock recovery with carrier regeneration, such as the Costas loop.

One variation on BPSK is Quadrature PSK, with four phases separated by 90 degrees as shown in Fig 20.8 This allows two bits to be sent at a time so Baud rate is now one half of the bit rate. QPSK gives better bandwidth efficiency than BPSK, but needs a 3dB higher S/N for reliable copy. This is a story that recurs repeatedly in the field of digital communications.

Combined Schemes

Many modern digital modulations combine or expand on these basic types. For example, Multiple Frequency Shift Keying, or MFSK, uses one of several tones for each interval. If, for example, eight tones were used then each tone could represent three bits at a time, binary '000' through to '111'. MFSK offers a lower bandwidth than FSK for a given data rate, and in DSP implementations is almost as straightforward to decode as binary FSK.

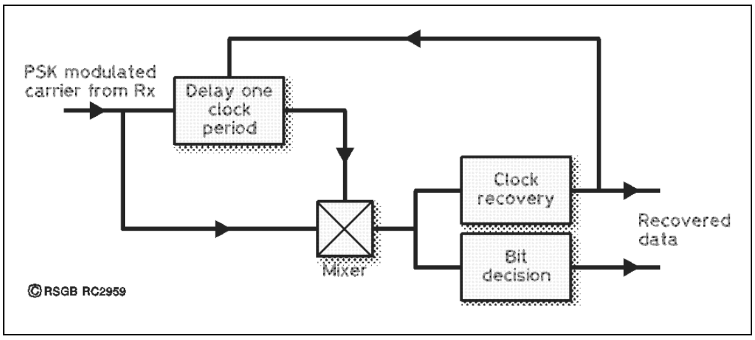


Fig 20.6: This scheme is used for differentially-coded modulation where the data is coded as a change from one bit to the next. Note the need for recovery of the bit clock in order to make the phase comparison accurately

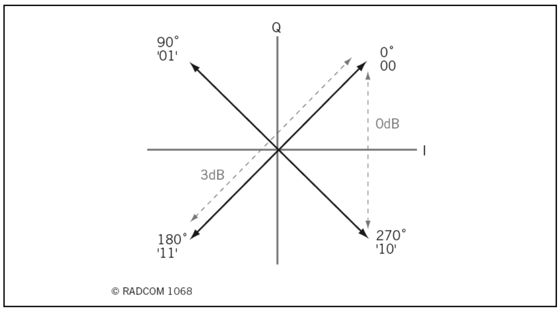


Fig 20.8: The four symbol states of a QPSK waveform showing the amplitude differences between states

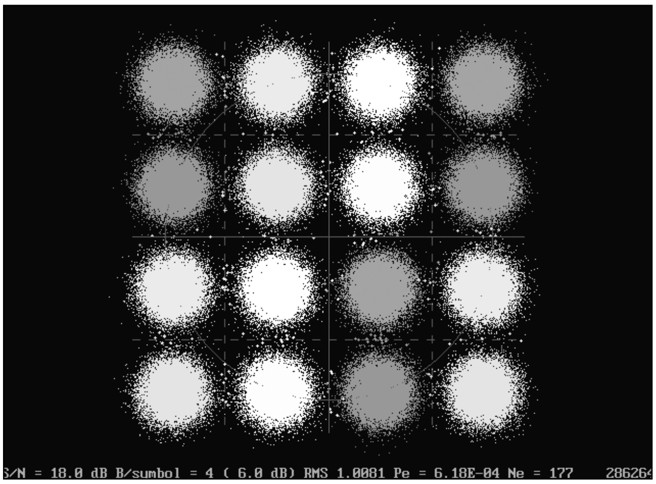


Fig 20.9: Simulation of a 16-QAM signal with 18dB S/N

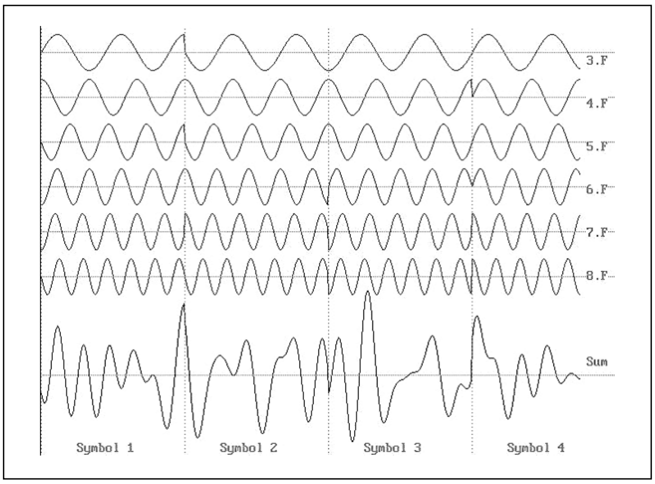


Fig 20.10: The problem of summing QPSK modulated carriers

In fact, a 32 tone MFSK signal, where each tone represents one of the 32 possible RTTY codes, was developed over half a century ago by the UK Foreign Office - where it was called Piccolo because it sounded a bit like the musical instrument.

Quadrature Amplitude Modulation is another combined scheme. A carrier is altered both in amplitude and phase to cram several bits into one symbol. 256 QAM, for example, has 256 states with each symbol carrying 8 bits of information. As each state is only separated from its adjacent one (in phase and amplitude space) by a small amount compared with the overall signal amplitude, a high Signal to Noise ratio is required to prevent a small amount of noise from corrupting one symbol into another. Fig 20.9 shows a simulation of a 16-QAM signal with 18dB S/N

And here lies the big trade-off for digital communications. Simple schemes like FSK and BPSK work on low S/N links but occupy large bandwidth. Making each modulation symbol carry more bits reduces the bandwidth requirement but increases the received S/N needed, and needs greater transmit power.

Parallel Tone Modulation

All the modulation types considered so far have been formed by modulating a single tone or a carrier, but there is now a whole family of digital modulation types that are built up from multiple simultaneous tones. One problem with single tone modulations is that when multipath is present - which is often the case on radio links - a receiver may pick up two or more copies of the transmitted signal with different time delays. If the amplitudes of each component are not too far apart, the resulting demodulated bits will overlap each other leading to errors. The overlapping bits cause intersymbol interference and especially on the HF bands multipath can be very bad, with delays of up to several milliseconds being typical. The overlap severely limits usable modulation rates to 200 Baud, or even lower. This limitation was the reason why, at HF, RTTY was so popular for so long; its 45 Baud signalling was robust enough to overcome multipath if signals were strong enough. At VHF/UHF, multipath delays of tens to hundreds of microseconds are typical.

The need for higher data rates over multipath-prone channels forced designers to come up with new schemes and rethink modulation completely. If a number of tones are transmitted simultaneously within a channel, these can each be modulated at a symbol rate, that is low compared with multipath delays. If each tone is modulated independently (with BPSK or QPSK for example), the data carried on each can be merged after demodulation, leading to a net data rate comparable with what might have been achieved in the same bandwidth with a single tone scheme. But now, as the symbol rate is much lower, multipath is not an issue. As an example, consider a 32 tone scheme with a spacing of 75Hz leading to a signal bandwidth of 2400Hz. This bandwidth could, ideally, support a data rate of up to 4800bit/s using BPSK, although for any typical implementation 2400bit/s would be more realistic. But, any multipath delays of more than a few tens of microseconds would damage the demodulation process. If the 32 tones were modulated at a data rate sufficient for the sidebands not to spread into the adjacent tones, which might be around 50bits/s each, then the net data rate becomes 32 * 50 = 1600bit/s which is not quite as good as a single tone scheme but much more multipath-resistant. Such parallel tone modulation types have been used for military communications for many years now.

DSP techniques allow this concept to be pushed further. By synchronising the data switching with tone spacing, it is possible to ensure that although the sidebands from each modulated carrier overlap they do so in a controlled way that prevents interference. The 32 tone example will now be modulated at 75 symbols/s, leading to a net data rate of 2400bit/s - exactly that possible with a single tone. Synchronising the tone spacing with symbol rate to avoid mutual interference leads to the name Orthogonal Frequency Division Multiplex OFDM - which is now one of the most widespread data modulations, being used for terrestrial and cable digital television transmission, DAB radio, Digital Shortwave Broadcasting (DRM) some military communications, 3G mobile phones and Wimax Internet. It is just beginning to make inroads to amateur radio.

A downside to OFDM is that a highly linear transmitter is required, so the resulting power efficiency is not very good as PAs have to be run considerably backed-off. Where many frequencies are sent in parallel, at certain times their amplitudes can add up coherently (voltages add) and a brief power output many times the mean level can be seen. Fig 20.10 illustrates this, where six QPSK modulated carriers are shown with their resultant voltage summation. The problem can be lessened by staggering the start phases in a pseudo-random manner to reduce linearity requirement.

TRAFFIC OR TYPE OF COMMUNICATION

Keyboard to Keyboard Modes

This is the simplest type of amateur data communications and one of the most 'friendly' in operation. RTTY and PSK31 are popular examples of Keyboard to Keyboard modes where everything that is typed on the sender's keyboard is transmitted directly over the air and printed or shown character by character at the receiving station. Transmitted codes will often include control characters such as backspace, and it can be quite intriguing to watch a distant typist correct a spelling mistake by backspacing and retyping.

Keyboard to Keyboard modes often do not include any error correction, so are prone to interference or bit errors which result in corrupted characters and garbled text. If interference is slight, such as just the occasional burst, the meaning of garbled text can often be inferred from the surrounding characters or words. Attempts have been made in some schemes to add in Forward Error Correction (FEC), but this nearly always leads to delays in timing, making quick break-in (or Tx/Rx changeover) time consuming, so FEC is rarely used.

File Transfer

In contrast to keyboard-keyboard modes, transferring data files requires that absolutely no errors are to be introduced, so a protocol has to used to check the validity of the received data. There are two ways of doing this, and both are often used together. Forward Error Correction adds redundancy to the transmitted data, so that if a few bits are corrupted on transmission, there is enough additional data to be able to regenerate the missing data.

Very strong FEC is needed for one way error free traffic, and in practice data integrity can never be guaranteed so handshake operation is usually employed. Here the transmitting station appends a checksum to short bursts of data. The receiver checks each burst with the checksum, and transmits an acknowledgement back to say either “send again” or “OK. Next please”. Protocols have to be developed to allow for lost bursts, and lost or corrupted acknowledgements, and are referred to as Automatic ReQuest (ARQ) protocols.

AMTOR (AMateur Teleprinter Over Radio), known as SITOR in the commercial world, adds simple error checking to teletype transmissions with an ARQ protocol for greatly reducing errors on short messages.

The latest generation of File Transfer modes may incorporate Internet Protocol (IP) type messages for compatibility with networking or Internet connection.

Visual Schemes - or Fuzzy Modes

These are a sort of half-way-house between data transmission and slow scan television. Signals are sent in a form that can be displayed on a PC screen and read by visual inspection. Frequency or amplitude (usually on/off) modulation can be used, and there is at least one case of a phase shift visual modes available in software

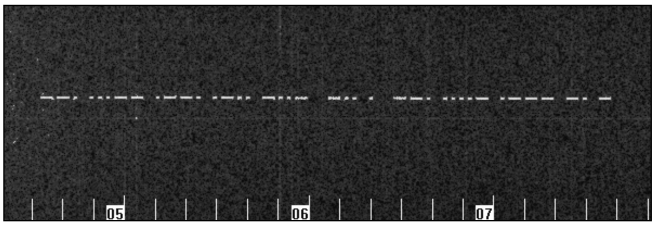


Fig 20.11: The first ever QRSS reception was on the old 73kHz band. The time axis shows that the transmission took three hours to complete

SlowCW or QRS(S) is the simplest of these visual schemes. A carrier is switched on or off using normal Morse code symbols, but slowed down to such an extent that a single dot interval can take anything from one second to minutes. Digital Signal Processing, uses the FFT routine to filter the signal to a narrow bandwidth commensurate with the dot interval and display the result on a waterfall plot. This type of plot shows frequency along one axis (usually the vertical one) with time is along the other. Signal intensity is shown by the colour or brightness of the display. The advantage of a waterfall plot is that a band of frequencies can be observed simultaneously, and several QRSS transmissions can be decoded at the same time

The first instance of QRSS being used on the amateur bands was in 1997 when G4JNT used it to transmit on 73kHz to G3PLX at a distance of 393km. The dot interval used for that transmission was 40 seconds, and the complete message took 3 hours to send. Fig 20.11 shows a plot of the received message at G3PLX.

DFCW, or Dual Frequency CW is an alternative to QRSS. Instead of 'dashes' being coded as a transmission three times longer than a 'dot', they are sent for the same duration as a dot, but on a slightly different frequency. Also, since each symbol is now of an equal duration, intersymbol gaps are redundant and can be removed. Sometimes, to aid readability, they are kept but made shorter. DFCW typically shortens messages by about 20 - 40 percent.

DFCWi (the I is for 'idle') carries the process a stage further and introduces a third tone for the intersymbol and, more importantly, the inter letter and word gaps. In noisy conditions a long series of symbols can get broken up, and give the appearance of multiple words. The third idle tone shows an operator if there really is a gap there, or if the signal has faded out, in which case the message contents can be inferred more easily than if DFCW

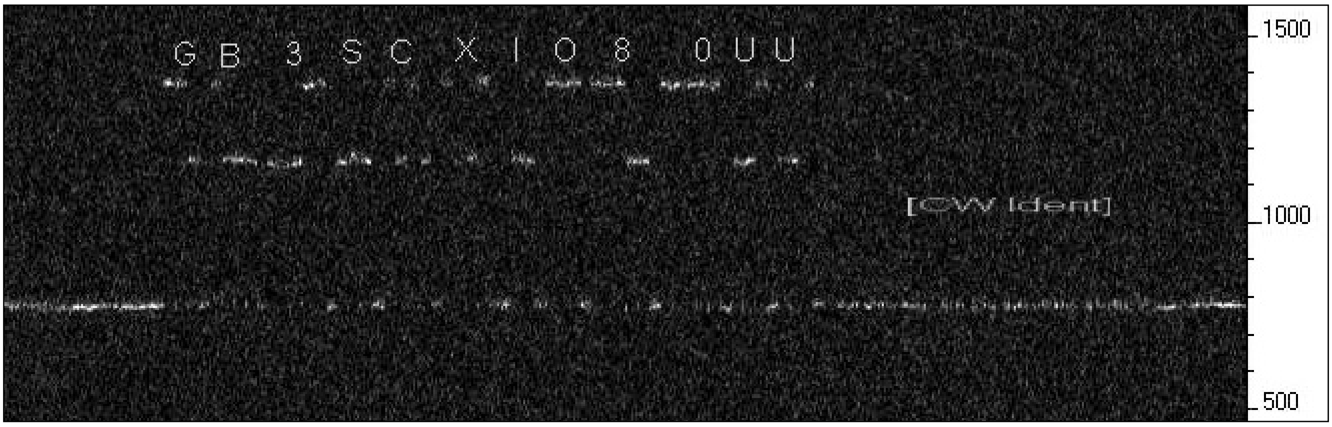


Fig 20.12: Weak signal DFCWi reception of the GB3SCX beacon on 10GHz (the callsign letters have been manually added to the picture for clarity). Dashes are on the top line, dots on the middle and the idle state on the bottom.

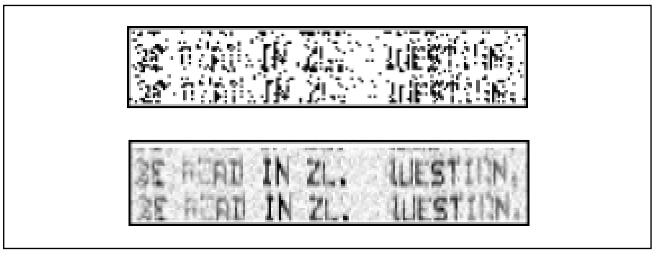


Fig 20.13: Hellschreiber in black and white (top) and greyscale (bottom)

were used. DFCWi transmissions often have a 100 percent duty cycle, although small intersymbol off periods can be inserted to aid readability. Fig 20.12 Shows the DFCWi received from the 10GHz beacon GB3SCX over a rain-scattered path.

Software for generating DFCW can be obtained from [1] and for DFCWi [2] More on using QRSS and DFCW can be found in the Morse chapter.

Hellschreiber is a visual scheme where individual letters are sent to be read off a screen. Traditional Hellschreiber (often abbreviated to just Hell) uses rapid on-off keying with a raster scanned display such as that shown in Fig 20.13. Later versions adopted frequency shifting so that the result can be read on a waterfall display. There are two types of FSK Hell. Sequential Multitone (SMT) Hell transmits the vertical components making up the pixels of each letter sequentially so only one tone is transmitted at a time. This allows a high efficiency non-linear transmitter to be employed, often the case on the LF bands, but means the letters will be shown with a slant to them. Suitable choice of font can make the result quite acceptable and in practice one of the best fonts for SMT-Hell is a simple 5x7 dot-matrix such as used on the earliest computer printers and screens.

Parallel tone MT Hell generates all the tones corresponding to each vertical element making up each letter simultaneously, so a linear transmitter is required. This gives a more pleasing and possibly easier to read display, but is generally constrained to HF where linear SSB transmitters are the norm. Software for generating and receiving all the Hell modes can be found at [3]

Modern Slow Scan TV is nowadays usually transmitted digitally, and is often included within the integrated data communications software packages. More details on SSTV can be seen in the chapter on visual modes.

MODERN DATA MODES

With the ready availability of powerful and fast DSP using PCs and soundcards, whole new families of advanced data modes have been designed. Some of these are just extensions of straightforward ASK / FSK or PSK with error correction, while others have been designed with specific target applications.

One aspect of any new datamode that has to be considered is that of matching it to the channel characteristics. This has to involve considering the bandwidth of the signal, the level and duration of multipath, any frequency scattering or Doppler effects that may be encountered, and if transmissions need to be in short bursts, or continuous. Linear transmitters as used at HF and for weak signal working at VHF/UHF allow waveforms with an amplitude component such as parallel tone signalling. This usually allows optimum bandwidth and S/N to be considered. Where non linear transmitters are in use, modulation is restricted to single tone constant amplitude (although on-off is usually possible). This might be the case at LF and also on the microwave bands where the transmitted signal is generated by multiplying up from a lower frequency source. A few examples are shown in Table 20.1.

ERROR CORRECTION & ITS IMPLICATIONS

Forward Error Correction can be strong or weak, or roughly equivalent to soft or hard. Strong FEC can expand the data to such an extent that six times as many bits are transmitted as actually make up the message; weak FEC may just double the number of bits sent. Error correction can be a complex mathematical process and many of the better FEC schemes can rarely be described without recourse to a thorough understanding of the maths behind the process. FEC always introduces a delay between transmission and decoding, and on the strongest schemes this delay can take up the complete duration of the message if data has been interleaved over its whole length

Soft error correction makes an attempt to correct as many bits as possible with limited redundancy, presenting the best it can do to turn what would otherwise be garbled text into something readable. Although some errors may still be present, hopefully these will have been reduced enough to make it possible to infer or guess the message contents. Hard error detection involves such a high degree of processing that the received message is either perfect or nothing. Occasionally a badly garbled message, or impulsive noise, can be decoded and a message presented as valid, but the contents are usually so ridiculous that it is quite clear a false decode has occurred.

LF 137kHz - 2MHz	Multipath quite low - if any usually due to a single ionospheric hop. Low bandwidths, due to narrow allocations, typically a few Hertz or tens of Hertz maximum. Often non linear transmitters are used for high power / efficiency. *ASK, FSK, (including MFSK) and PSK are usual*
HF Skywave 3 - 50MHz	Bad multipath - up to several milliseconds. Deep fading of several seconds. Bandwidths typically a few hundred Hertz to a 3kHz SSB channel width. ionospheric Doppler shifts of a few Hertz. *FSK, PSK, parallel tone are usual, with extended interleaving to cope with fades.*
V/UHF 144MHz - 2GHz	Severe multipath from multiple reflections, typically tens to hundreds of microseconds. Doppler shifts if communicating with moving vehicles. Bandwidths either SSB 3kHz for weak signals / DX work, or based on an FM channel of 12.5 / 25kHz for local working. Meteor Scatter requires short bursts of fast data. Moonbounce introduces scattering, and works with very weak signals over extended durations. *FSK/MFSK , PSK, MSK are all used*
Microwaves 3GHz and up	Massive multipath from moving objects leads to scattering. Very high levels of spectral spreading due to troposcatter and rain scatter Bandwidth not a major issue, but for weak signals, a 3kHz SSB channel is preferred. *FSK / MFSK with a shift matching the spread is the only practical modulation for badly scattered paths*

Table 20.1: Matching data modes to the radio frequency medium

Examples of both are seen in amateur service. Soft FEC can be seen in QPSK31 (see below) where the majority of errors due to impulsive type interference can be coped-with. Hard FEC with deep source coding is encountered in some of the WSJT modes for very weak signalling like EME.

Source coding is the technique of reducing the unwanted redundancy in transmitted data to lower the number of bits required for a given message. The varicode alphabet used in PSK31 and a few other amateur modes is an example, where fewer bits are used for the common letters in normal text, such as 'e' and 't', with corresponding more for 'z', 'q' and 'j'. More complex source coding can compress amateur callsigns to a few tens of bits making use of their known structure of letters/numbers only, as well as locators and other numeric information.

A downside to source coding it that it removes the possibility to make use of contextual correction or guesswork on the part of the receiving operator. So source coding tends to be used with high levels of usually strong error correction. This all leads to a big trade off; is it better to heavily source code then expand with strong FEC, or leave in the natural redundancy use soft or no FEC and let the operator make the best of the result? Examples of all these can be found in modern amateur data communications.

Soft Decision Decoding is often included as part of the FEC process. Traditionally a '1' or '0' is decided at the receiver by determining if the signal lies on one side or the other of a threshold, such as amplitude, frequency etc. But, in a fully DSP implemented modem, more information is available to the decoder than a single threshold, and the actual level of the signal and noise is known. A soft decision decoder can make use of this information, in conjunction with a particular type of signal encoding adding redundancy and interleaving known as convolutional coding, to make a best-guess at what was most likely to have been transmitted. Viterbi soft-decoding of convolutionally encoded signals can give one of the strongest FEC schemes for continuous data there is, and when combined with additional block error correction on the source data (for example Reed Solomon error correction) can lead to a completely all-or nothing solution. Combined strong error correction schemes are seen commercially for Digital TV broadcasting. Amateur radio applications can be found in the WSJT WSPR and JT65 modes.

READY-TO-GO SOUNDCARD BASED DATAMODES

Several software packages exist that allow nearly all of the datamodes currently in use to be implemented with a common soundcard interface. Many of them are free public domain software and can be downloaded from the Internet and installed.

Digipan	PSK31, QPSK31 - panoramic decoding. Download from [4]
MultiPSK	Most current datamodes, including panoramic decoding of PSK31 and RTTY. Additional facilities are offered after payment The user screen for *MultiPSK* can be seen in **Fig 20.14**. The software can be downloaded from [5]
FLDigi	Most current data modes [6]
TrueTTY	RTTY - Trial version expires unless a fee is paid
MMTTY	RTTY - Free. Download from [7]

Table 20.2: Examples of multimode data software

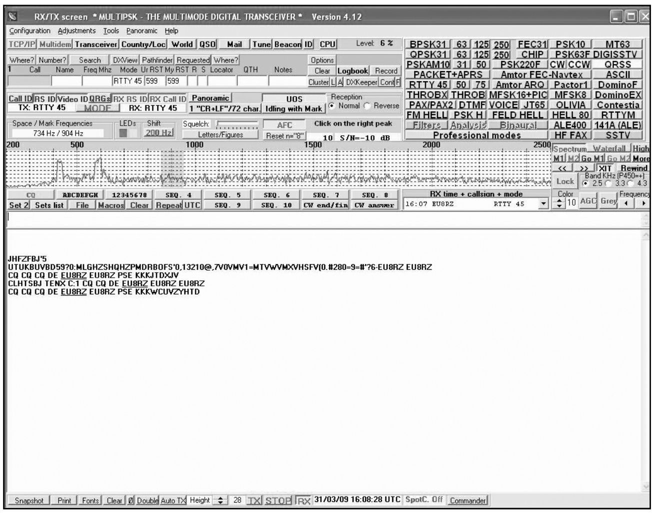


Fig 20.14: The comprehensive screen of multimode data program *MultiPSK*

Others offer a limited free trial period then ask for payment. Some offer additional facilities once payment has been sent. Some of the most popular packages are shown in **Table 20.2**. There are many others that are specific to particular modes, they are described in the relevant section.

RTTY

The oldest and most well established keyboard to keyboard mode it uses FSK with a shift of, usually, 170Hz in the amateur service. A five bit code with letters/numbers shift, allows only a restricted alphabet of upper case letters, numbers and punctuation. Most operation at HF is at a speed of 45.45 Baud (symbol period 22ms), although at VHF 50 Baud is often encountered. At UHF and above, higher shifts may be needed to cope with frequency scattered signals, and at 10GHz 850Hz shift 50 baud RTTY was used for a while on the GB3SCX beacon. The use of a letters/numbers shift makes the system quite error prone as if the shift character is lost; whole sentences can be garbled.

A number of software packages can be used for RTTY, including *MMTTY*, *MultiPSK* and *FlDigi*. MMTTY allows operator intervention to repair corruption to the letter/number shift and apply this to whole sections of garbled received text.

AMTOR

This is a derivation of RTTY that introduces error correction by ARQ. Signalling speed is increased to 100 Baud, and characters are combined into groups of three at a time for transmission. The five-bit code is expanded to seven bits to introduce error detection (but not correction) by only using those seven-bit codes that have exactly three ones (there are 35 of these out of the possible total of 128). On receive, if the number of bits in any character is not exactly three, an error is assumed and the receiving station requests a retransmission. The error correction / ARQ is not perfect, but does result in a mostly error free connection. AMTOR is a connected mode where one station has to connect to another to set up a link.

It is a handshaking mode and needs continuous, rapid receive/transmit switching. Its regular chirp-chirp can often be heard around 14100kHz where AMTOR mailboxes are still in use. The frequency shift is usually increased to 200Hz. An alternative FEC version is used for broadcasts and CQ calls and does not have to connect. Every character is sent duplicated and offset by 200ms between them, so the receiver has two attempts at receiving a valid version of each character. AMTOR used to

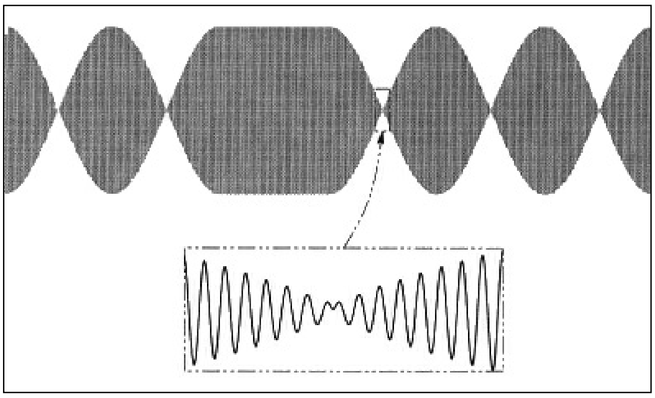


Fig 20.15: A typical BPSK symbol showing the amplitude reduction at crossover

require dedicated hardware, but now is available within the *MultiPSK* package.

PSK31

PSK31 was designed by Peter Martinez ,G3PLX ,as a keyboard-to-keyboard replacement for RTTY making use of modern DSP techniques. It uses Binary PSK at 31.25 baud with amplitude shaping at each phase transition to reduce signal bandwidth to the absolute minimum possible and results in a bandwidth of just 32Hz. A typical PSK31 symbol can be seen in Fig 20.15 showing the amplitude detail at the phase crossover.

The full set of 256 ASCII characters is available, but they have been re-coded into a variable length alphabet called Varicode. Here, the most popular characters (assuming lower case English language plain text) such as '[space]' 'e' and 't' have been allocated codes with a lower bit count than the less common letters like 'z' and 'x'.

The narrow bandwidth and inherently better signalling efficiency of coherent PSK over non-coherent RTTY means that PSK31 will work at far weaker signal strengths than RTTY can. In practice, it offers around 10 - 13dB improvement over 170Hz shift 45 baud RTTY. Also, something like five PSK31 stations can typically occupy the spectrum taken up by just one RTTY signal

An experimental version of PSK31 introduced FEC with convolutional coding on a QPSK waveform, produced by G3PLX at the same time. The FEC was aimed at improving the performance in the presence of interference bursts. The change to QPSK from BPSK introduced a 3dB S/N penalty so some of the advantages to the 'mild FEC' were lost. The narrowband signal can get corrupted when more than a few Hertz of signal scattering is present - such as on trans-polar paths.

Subsequently, other writers modified PSK31 to work at both lower and higher data rates. A quarter speed version, PSK08 was tried on 73 and 137kHz, and PSK62 and PSK125 have been designed for faster hand typed operation when conditions are good, or to cope with scattered signal paths.

Nearly all datamode software packages include PSK31 and QPSK31; many packages also include its later variants. Most popular is *Digipan* which provides a spectral display of up to 3kHz of spectrum and decodes all the PSK31 signal in this band up to a maximum of around 32 different stations. A simple point and click process allows the operator to reply to any one of the multiple decodes, and selects the transmit tone frequency appropriately. Fig 20.16 shows the *Digipan* screen in Panoramic Mode.

MultiPSK also offers this panoramic feature; Most other datamode packages just allow one signal at a time to be included

OLIVIA

This is a teletype protocol that transmits a stream of ASCII (7-bit) characters, sent in blocks of five at a time. Each block takes two seconds to transmit, giving an effective data rate of 2.5 character/second or equivalent to about 25bits/s. It is a Multi-FSK mode with the default mode being 32 tones within a 1000Hz audio bandwidth. The tones are spaced by 1000Hz / 32 = 31.25Hz and their amplitude is shaped to minimise the amount of energy sent outside the nominal bandwidth. The baud rate is 31.25 MFSK tones/second. To accommodate for different conditions and for the purpose of experimentation, the bandwidth and the baud rate can be changed.

Strong Forward Error Correction is built in by the use of Walsh functions so decoded text is of the all-or-nothing nature within the structure of the two-second five-character blocks.

THROB

THROB is a DSP sound card mode that uses Fast Fourier Transform technology to decode a five tone signal. The THROB program has been described as an attempt to push DSP into the area where other methods fail because of sensitivity or propagation difficulties and at the same time work at a reasonable speed. The text speed is slower than other modes and runs from a dedicated software package or as one of the modes within *MultiPSK* and *FlDigi* The name comes from the sound the signal makes - a slow throbbing tone

MFSK16 / MFSK8

MFSK16 is an advancement on the THROB mode and encodes 16 tones with Constant Phase Frequency Shift Keying to minimise bandwidth. Continuous FEC sends all data twice with an interleaving technique to reduce errors from impulse noise and static crashes. A new improved Varicode is used to increase the efficiency of sending extended ASCII characters, making it possible to transfer short data files between stations under fair to good conditions. The relatively wide bandwidth of 316Hz for this mode allows faster baud rates with a typing speed equivalent to about 42 WPM, and greater immunity to multipath interference. A second version called MFSK8 is available with a lower baud

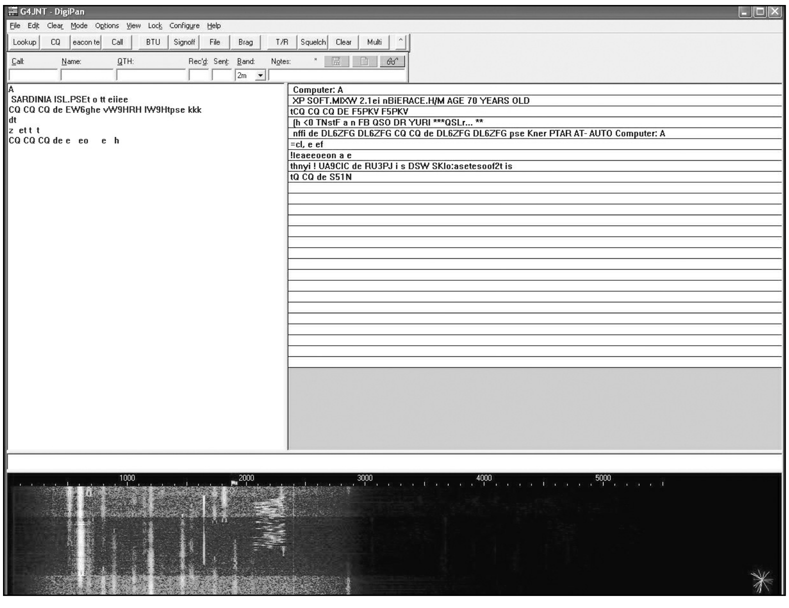


Fig 20.16: The *Digipan* screen

rate (8) but greater reliability for DXing when trans-polar interference is a major problem. Both versions are available in a package written by IZ8BLY, and also as options in *MultiPSK* and *FlDigi*.

MT63

This is a robust signalling scheme to keyboard text over paths that experience fading and interference from other signals. It is accomplished by a complex scheme to encode text in a matrix of 64 tones over time and frequency. This overkill method provides a 'cushion' of error correction at the receiving end while still providing a 100WPM rate. The wide bandwidth of 1kHz for the standard version makes this mode less desirable on crowded bands such as 14MHz. MT63 is available in *MultiPSK* and *FlDigi.*

Other datamodes can be found within some of the multi-purpose packages. In many cases these are variants of those described, operating at different speeds / bandwidths or with different error correction. Some are adaptations for specific purposes, like digital Slow Scan TV or image transfer. Such applications are specific to the software package employed, for instance *MultiPSK* .

ROS

A relatively new mode ROS is named after its designer José Ros. it is described as a "Digital Spread Spectrum Mode, but with a narrow total occupied bandwidth of about 2.2kHz". The first contact using ROS took place on 18 Feb 2010 from Vitoria in Spain to the University of Twente in the Netherlands, covering a distance of 1265Km on 7.065 MHz.

ROS can be downloaded free of charge from [8]. Some limited documentation is supplied as a PDF file. Not much technical information is provided about the internals of the mode but ROS is a 16-tone MFSK waveform arranged so the frequency is stepped pseudo-randomly over the occupied 2200Hz bandwidth to reduce the effects of interference.

At first launch two rates were supported, 16 Baud and 1 Baud to cope with different band conditions and signal strengths. The decoding software automatically synchronises to any symbol rate. Subsequently lower rate and bandwidth options were added, especially to cater for LF signalling. The latest version of the software now offers the option of I/Q stereo output mode for directly driving direct quadrature upconverters.

The relatively wide bandwidth and low data rate means that ROS offers weak signal advantages over many current established HF datamodes. In some cases signals that are so weak they don't show up on the waterfall display can be correctly decoded. The ROS User screen is shown in **Fig 20.17**.

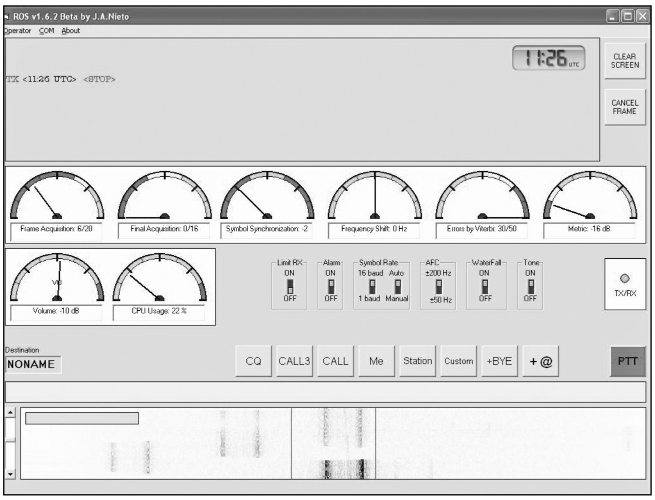


Fig 20.17: The ROS user screen

WIDEBAND DATA MODES

ROS is a typical example of modern HF digital communications practice in that it spreads the signalling energy well outside the necessary bandwidth needed for communication. Necessary bandwidth is equal to or higher than the symbol rate; imagine a continuous 101010 . . .. pattern being sent, the alternate high and low parts of the waveform, after filtering, each constitute a half cycle of a sine wave. As a rule of thumb necessary bandwidth is often taken as being equal to the symbol rate - for the standard ROS mode, it is 16 baud or 16Hz. For multi-symbol modes like MFSK or QPSK, several bits are sent for each symbol so the data rate is proportionately higher.

In environments like HF where multipath, scattering, Doppler shift and interference are prevalent the transmitted signal is usually expanded over a significantly wider bandwidth than necessary to minimise the damage done by the interference mechanisms. These can often be observed as the classic HF fading, often noticeable as a deep audio null drifting through an SSB or AM voice channel over a period of a few seconds. By spreading out the data out over the bandwidth occupied by a typical voice signal, 2 - 3kHz, the effects of the moving null are reduced.

There are many different ways of spreading the energy. Single tone modes like Stanag 4285 / MIL-STD -110A waveforms use an underlying 2400 baud symbol rate modulated with 8 PSK (3 bits per symbol) then use massive redundancy, error correction and interleaving to encode a lower data rate robustly. Other waveforms adopt a lower symbol rate, but hop a single tone over the band in different ways.

ROS uses a pseudo random spreading code with an underlying simple 16 FSK modulation. Contrast with K1JT's JT65 code in *WSJT*, where 65 tone slots are directly used. Reed Solomon error correction is directly applied to the tones to be able to cope with several being lost in QRM. Modern schemes such as OFDM transmit thousands of tones simultaneously, each one modulated at a low rate with the total being the combination of all individual data streams summed. All these techniques appear on the amateur bands, at some times.

Social Issues

Commercial and military users usually allocate their frequencies in 3kHz blocks corresponding to a single voice channel. With a guaranteed clear channel, all the advantages of the relatively wide band can be used for robust low data rate communications, often with the overall data throughput being adjusted automatically in response to band conditions. Stanag 4285 automatically adjusts over the range 4800 down to 75 Bits/second as propagation alters. On a crowded amateur band, with many different and incompatible modes each vying for space, a plethora of different wideband energy spreading modes doesn't work. If users try to stay in the segments allocated for data modes then narrowband modes like PSK31, RTTY and MultiPSK have to fight for their patch with the wideband modes obliterating several of the narrow ones.

Ideally, experimenters wanting to use the full 3kHz bandwidth for their data should really use the SSB part of the band but problems can occur here as well - not least SSB operators objecting to data transmission in 'their' segments and the all-too-frequent heavy occupancy of this part of the band during contests.

So, while wideband datamodes have their place in the experimental scheme of things in amateur radio, and offer very clear

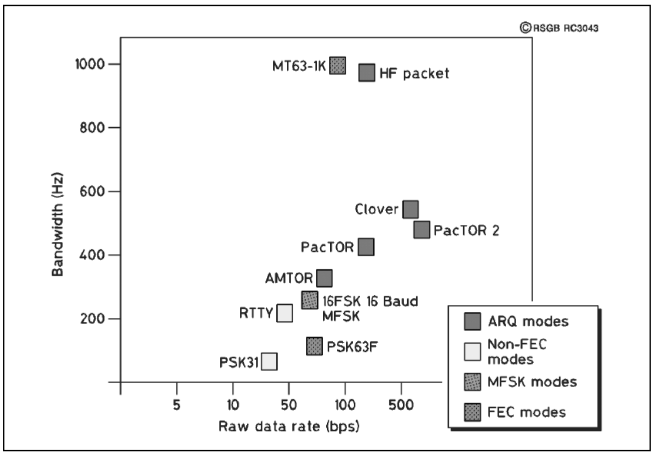


Fig 20.18: Comparison of the performance and speed of various data modes [BARTG]

advantages in communications reliability over their narrowband cousins, their place on the narrow and mostly de-regulated amateur bands really needs to be considered carefully. And should a new wideband mode be suddenly released for free download, knowing many operators will immediately want to try it out without fully appreciating the implications for interference.

SUMMARY

A summary of performance and speed of many of the amateur data modes was printed in the BARTG magazine, and is reproduced in Fig 20.18.

LOW BANDWIDTH SLOW DATAMODES FOR LF/MF

Operation on the LF and MF bands (137 and 500kHz) has generated a few datamodes customised to the requirement for narrow bandwidth extremely weak operation. LF propagation is usually stable with little ionospheric scattering. These are specialised, and usually need their own dedicated software for operation rather than being included as options within the multi-mode software suites.

WOLF

WOLF was written by Stewart Nelson, KK7KA, with a Windows graphical user interface by Wolfgang Büscher, DL4YHF. Wolf stands for Weak-signal Operation on Low Frequency, and can operate over a wide range of signal levels.

The original WOLF implementation by KK7KA sends ten symbols per second using BPSK at 10bits/s. Five of these symbols carry data bits, the other five bits contain a pseudo-random sequence for synchronisation. In the WOLF GUI, this original WOLF speed is called 'normal' speed. A few slower and faster variants were subsequently added. The slower variants may help with extremely weak signals and the WOLF GUI [9] offers:

- Normal speed: 10 symbols per second . One frame = 96 seconds; copy may be possible after 24 seconds.
- Half speed: 5 symbols per second. One frame = 192 seconds = 3' 12".
- Quarter speed: 2.5 symbols per second. One frame = 384 seconds = 6' 24".
- Double speed: 20 symbols per second ("Fast"). One frame = 48 seconds.
- Fourfold speed: 40 symbols per second ("Turbo"). One frame = 24 seconds

The example shown in Fig 20.19 is the first trans-Atlantic signal received on the now-defunct 73kHz band. It illustrates how a weak signal is built up after many iterations:

Using 10 Bit/second BPSK with no amplitude shaping of the waveform, WOLF is a rather 'antisocial' modulation to use on narrow LF bands and its use has fallen off, being replaced with more modern soundcard modes.

Jason

This is a keyboard-to-keyboard communication program, tailored for very low S/N ratios using only 4Hz of band. The coding scheme is based on the ideas about Incremental Frequency Keying, initially proposed by Steve Olney, VK2ZTO, where the information is coded in the absolute value of the difference between two frequencies sent sequentially. This has the advantage of not needing a precise initial tuning and a tuning error of a few Hertz is perfectly acceptable. Another characteristic is that as the frequencies are sent one at a time there is no need for a linear amplifier.

The frequency deltas can assume one of 16 different values. After sending one tone, the next is shifted by the appropriate amount, up or down depending on the setting of the USB/LSB switch. With 16 deltas there are 17 tones, and any overflow causes a wraparound.

With 16 possible deltas, each baud (change in frequency) encodes four bits (a nibble) at a time which are not enough for a reasonable alphabet, so each character takes two nibbles to send. But now a problem arises: how to get character synchronisation? In other words, which is the high-order and which is the low-order nibble?

The high order bit of each nibble is used to encode this info meaning the high-order nibble is of the form '1xxx'b, while '0xxx'b is the low-order one with xxx standing for the actual information transmitted. So now there are 6 bits available to encode our alphabet, enough for a shortened 64 character alphabet of ASCII code goes from 0x20 (the blank) to 0x5F allowing the

Signal received by John Andrews, W1TAG, on 19 March 2001 at Worcester, Massachusetts:

```
C:\wolf>wolf -f 799.892 -r 8001.95 -t 0.02 -q bmu2.wav -s 1200
WOLF version 0.51
t:  24 f:-0.020 a: 1.0 dp: 99.4 ci:11 cj:391 94R.7A8A4???UWC ?
t:  48 f:-0.029 a:-1.5 dp:103.7 ci: 5 cj: 12 6A.S69BGZ//LK8B ?
t:  96 f: 0.019 a:-1.4 dp: 98.7 ci: 4 cj:207 JFWWUL???N*E .Y ?
t: 192 f: 0.029 pm: 118 jm:119              LQ5HQ*2G569R2MW ?
t: 288 f:-0.029 pm: 150 jm: 52              /S SYS7FZSV XXZ -
t: 384 f:-0.029 pm: 175 jm: 52              ??????AWI2N 20Y ?
t: 480 f:-0.029 pm: 195 jm: 52              /KR/F 2U4X8ZSMT -
t: 576 f:-0.029 pm: 198 jm: 52              VMEVXXDPY2J4RTJ ?
t: 672 f:-0.029 pm: 205 jm: 52              .L5DKCA9GEDS.AX -
t: 768 f:-0.029 pm: 226 jm: 52              3J0FRV/XG6S7D7X ?
t: 864 f:-0.029 pm: 238 jm: 52              RJSWH TQU4IJ6NS ?
t: 960 f:-0.029 pm: 256 jm: 52              PE41I9K3DDS9ZHM -
t:1056 f:-0.010 pm: 321 jm:211              H96XY075JF4B YU ?
t:1152 f:-0.010 pm: 357 jm:211              Q*CW RQAU447BNT ?
t:1248 f:-0.010 pm: 447 jm:211              6D 6FLYJTCVY0.N ?
t:1344 f:-0.010 pm: 472 jm:211              Q*C*GBR88N0/5*T ?
t:1440 f:-0.010 pm: 482 jm:211              CQ M0BMU M0BMU  -
t:1536 f:-0.010 pm: 499 jm:211              CQ M0BMU M0BMU  -
```

Fig 20.19: Decoded text from the first WOLF transatlantic transmission, showing how the text is extracted by coherently integrating 18 seconds blocks of received data

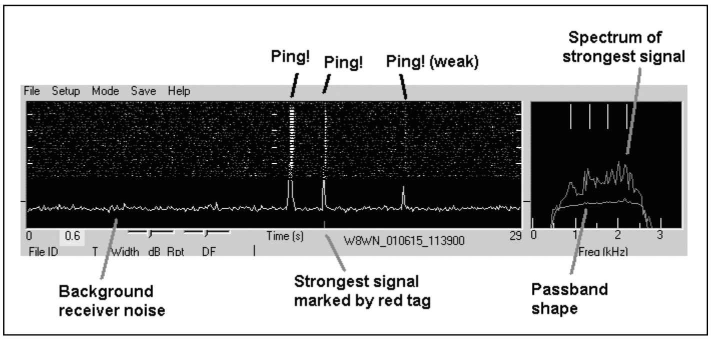


Fig 20.20: FSK441 in action

FSK441

This is a wide fast shift modulation designed for exploiting the short duration strong returns from meteor reflections. FSK441 uses four-tone frequency shift keying at 441 baud, the frequencies of the audio tones are 882, 1323, 1764, and 2205 Hz. Each encoded character uses three tone intervals and takes 3/441 seconds or around 2.3 ms for transmission. FSK441 accommodates an alphabet of 43 characters, the same ones used in the PUA43 system developed by Robert Larkin, W7PUA. No error correction is used, and text can be anything typed in.

The four possible 'single-tone' character codes, namely 000, 111, 222, and 333, are reserved for special use as shorthand messages. When sent repeatedly, these reserved characters generate pure single-frequency carriers. Their pings are easily recognised by the human ear and also by appropriate software. The present definition of the shorthand messages is respectively "R26", "R27", "RRR", and "73" for the four tones - messages that are frequently encountered in amateur meteor scatter communications.

Timing is based around slots of fixed length transmit/receive periods which have to be agreed beforehand by both parties, and are then specified in the setup screen. Fig 20.20 shows the FSK441 user screen during operation.

The tone frequencies and data rates are derived from the standardised soundcard sampling rate of 11025Hz. 441 Baud = 11025 / 25 and the tone frequencies are respectively two, three, four and five times this.

transmission of all upper case letters, the ten digits, and practically all the punctuation symbols normally used.

A dedicated software package, Jason [10] is needed for transmission and reception.

WSJT MODES

Joe Taylor, K1JT, has introduced a whole suite of customised digital communications modes for use across the entire amateur radio spectrum. The name derives from 'Weak Signals by k1JT' They are not keyboard-to-keyboard modes and rely on a strict protocol being set up with regard to timing and information exchanged. Most of them include:

- Source coding and compression
- Massive Forward Error Correction
- A strong synchronisation scheme
- High levels of data redundancy
- A modulation suited to the path in use such as narrow-band LF, badly scattered EME reflections or short duration meteor pings.
- A decoder that searches in time and frequency to allow for tuning / Doppler offsets and incorrectly set clocks
- A timing protocol that relies on UTC being known to within a few seconds, and a QSO exchange based on time slots.

Source coding first involves reducing the components of a QSO to its basic form including some or all of callsign, locator, report, transmit power and handshake signals such as OK and fail. By making use of the redundancy and limited character sets in callsigns and locators a total QSO 'over' can be compressed into a few tens or a hundred bits.

The reduction of the source data considerably shortens the message which now allows it to go along two different concepts for subsequent processing. One is to transmit it with little correction at a fast data rate over good quality channels. This is the solution seen in the FSK441 Meteor Scatter communications waveform as meteor pings are usually of good quality and high signal strength, but of length typically of a few hundreds of milliseconds.

Alternatively, the short source data can be expanded in a controlled way by interleaving it repeatedly, adding error correction and synchronisation bits to make a very robust weak signal mode capable of surviving bad QRM. This, in different forms, can be seen in JT65, and WSPR. As all the WSJT modes are designed for weak and scattered signal paths, they all make use of the various types of frequency modulation rather than coherent modulation like PSK. But, all are deliberately designed so that timing and frequency shift / spacing are synchronous and related directly to the soundcard sampling rates

All the WSJT modes are available as options within one package [11] although WSPR has its own dedicated software tool for beaconing.

JT65

This s a weak signal mode operating in alternate one minute time slots, starting on the minute boundary. Each transmission takes about 52 seconds. Massive source coding is used, and a message consists only of Callsign, four digit locator (for example IO90) and a few acknowledgement and reporting codes customised for EME operation. The compressed source data is expanded six times and all are interleaved, modified by adding error correction bits and spread out to form 64 separate symbols in a one of 63 code.

The signalling makes use of sequential MFSK in 65 individual tone slots. 63 of the tones are used in the 1 of 63 coding, one is left empty and the lowest tone frequency is used for synchronisation. There are a total of 128 time slots within a 47.5 second window during which one of the tones is always present. Timing derives from 11025Hz /4096, or approximately 0.32 seconds each. Half of the slots are used by the one-of-63 MFSK code and the other half, which are interspersed in a pseudo-random manner over the message duration, are allocated to the single lowest frequency tone used for synchronisation. This strong synchronisation code allows the decoding software to search over time to

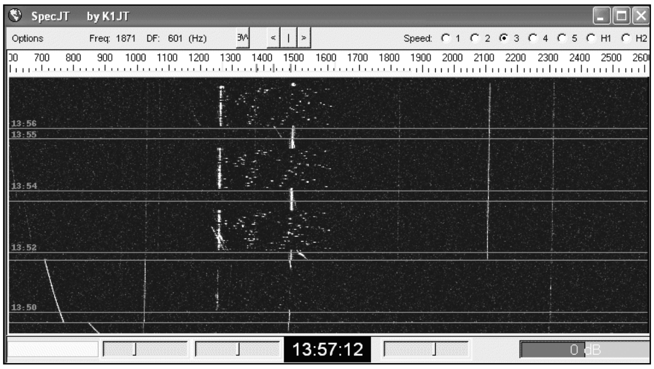


Fig 20.21: Spectrum of JT65B transmission

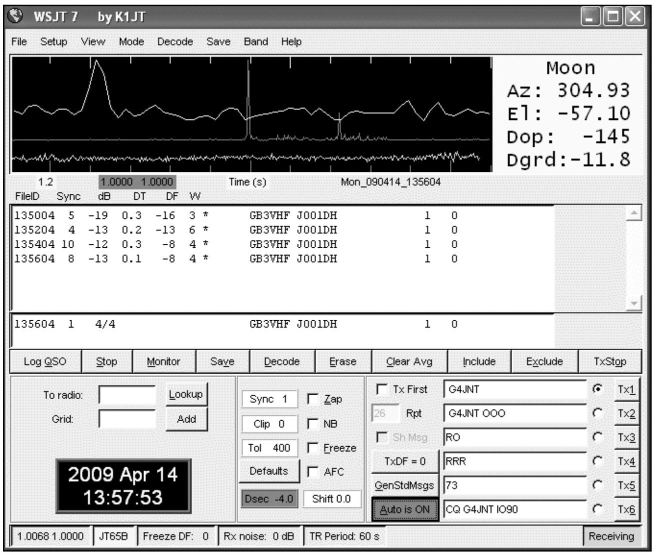


Fig 20.22: The JT65 screen in operation

find the vector, even when the EME delay and computer clocks may introduce errors adding up to several symbol durations.

Three variants of JT65 are in use, depending on the frequency band and expected spreading. The 65 tones can have three different spacings and signal bandwidths:

JT65A is designed for HF and 50MHz terrestrial operation with the narrowest tone spacing of 11025 / 3096 = 2.69Hz, giving a total signal width of 175Hz.

JT65B is targeted at 50 - 144MHz terrestrial and EME, with a tone spacing twice that of the A variant for a signal bandwidth of 350Hz

JT65C doubles tone spacing, again, for use at UHF with 700Hz bandwidth.

Fig 20.21 shows the spectrum of a JT65B received off-air from the GB3VHF 144MHz beacon, with the lowest synchronisation tone forming a clearly visible pattern.

Operation runs in alternate one-minute slots of transmit and receive, the use of even and odd minutes has to be agreed beforehand and is always specified as part of the EME setup scheduling.

The decoder stores the entire received data and searches in time and frequency for the sync vector. Once found, it knows the tuning and timing error and can then do a best estimate of each symbol using a sophisticated soft-decision process developed by K1JT. With a symbol time of 0.3s, and tone spacing of 2.7Hz the noise bandwidth of the decoder is of the order of 3Hz. This directly gives a 10dB S/N advantage over the (typical) 30 - 50Hz bandwidth needed for aural copy of CW. When normalised to the standard 2.5kHz bandwidth, reliable JT65 decoding can be made at S/N ratios of 26dB, equivalent to +3dB S/N in the 3Hz actual signal bandwidth. The reason that reliable decoding of an FSK signal can be made in a S/N so low is due solely to the massive encoding redundancy and error correction coupled with soft decision decoding.

The final message cannot be directly equated to a total effective-number of bits due to the use of a 63 symbol set. But, had a 1 of 64 symbol scheme had been used instead of 63, this could be said to represent 6 bits as $2^6 = 64$. In that case, the total effective number of bits could be said to be $64^{64} = 4096$. With the synchronisation vector adding another 64. This is a considerable expansion over the few tens of bits making up the compressed source data, and explains the extreme robustness of this protocol. **Fig 20.22** Shows the JT65 screen during operation.

WSPR

WSPR or Weak Signal Propagation Reporter is the latest of the WSJT modes. Unlike the others it has been designed primarily for beacon type operation (although the latest WSJT software does offer a 'QSO mode). The source data is compressed to callsign, 4 digit locator and transmit power specified to one of twenty levels ranging from -30 to +30dBm. The total again occupying a few tens of bits.

This is expanded to a one-of-four tone MFSK signal, with 162 symbols spread over a 110.6 second interval. Effective total number of bits is therefore 324. Unlike JT65, the synchronisation code does not form a separate tone, but is interleaved with the data, again in a pseudo random manner.

The 110s transmission fits into a two minute time slot (there is just time for a CW ident at the end for when licence conditions require this). The WSPR software is designed for simultaneous transmission and reception of other signals, so to manage this transmit slots are allocated in a pseudo random manner as a fixed percentage of the total. The percentage can be defined by the user as 33%, 25%, 20% etc of the total. By randomising transmit periods in this way, the likelihood of repeated clashes between two stations so that they would never hear each other is minimised.

The tone spacing is 1.46Hz, derived from the later 12000Hz soundcard sampling rate / 8192. The symbol duration is the reciprocal of this, or 682.7ms. This very low spacing and bandwidth, just 6Hz wide in total, means that WSPR primarily finds use at LF and HF. Every amateur band up to 144MHz now has a designated WSPR spot (200Hz wide) and on 500kHz, 7 and 10MHz in particular stations can always be found participating in 'WSPRing' as it is known.

As WSPR was designed for signal reporting, some facilities were built into the software to give additional benefits. The WSPR software monitors a 200Hz wide chunk of spectrum corresponding to tone frequencies from 1400 to 1600Hz and can decode every WSPR signal within it, searching over time as well to allow for errors in PC clock setting. The operating screen showing signals received on the 7MHz band can be seen in **Fig 20.23** The S/N is automatically measured in the decoding process and reported along with the message contents.

Now as signal power, location and S/N are known there is enough information to determine path details. The software allows successful decodes to be automatically uploaded to a central WSPR database [12] where every 'hit' is logged and a

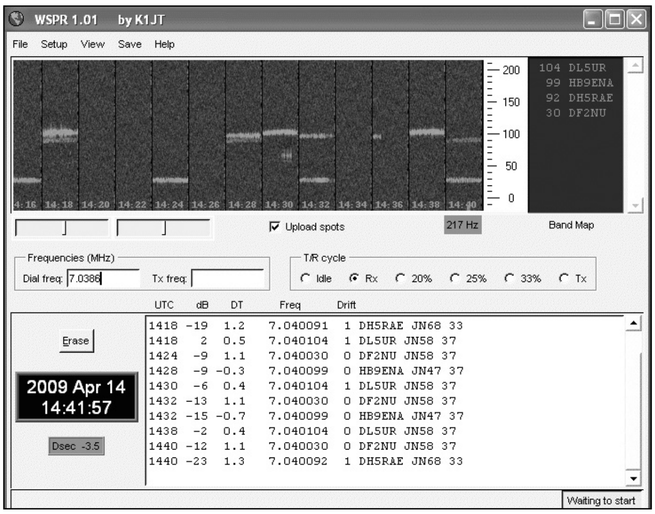


Fig 20.23: The WSPR screen

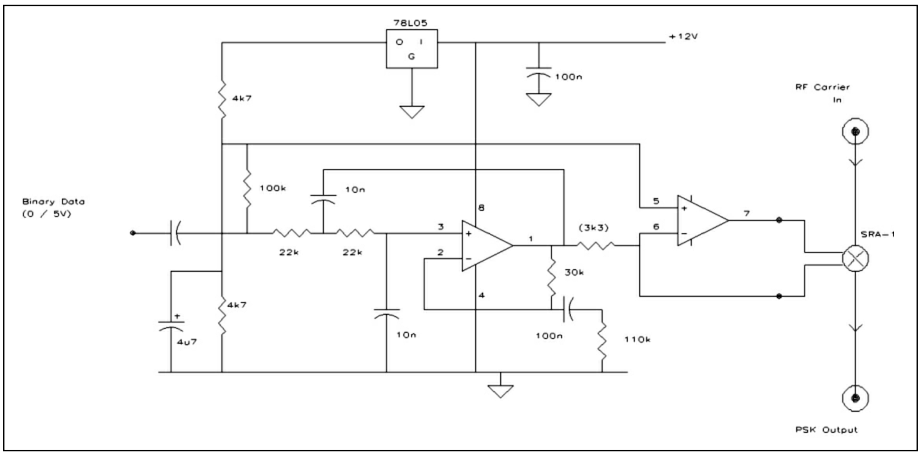


Fig 20.24: Direct PSK generation at 400 Baud

complete propagation map between all participating stations can be generated.

SPECIALIST MODES

A few datamodes exist that derive from the days before PCs and soundcards were ubiquitous. They are based around dedicated hardware and use proprietary, or at least customised, protocols. Although little used by amateurs now, they are often is use by commercial organisations.

Pactor is packet radio over Tor. The AMTOR signalling protocol forms the base modulation and a packet radio error correction/detection layer is overlaid.

Clover is a proprietary four tone MFSK system designed by the Hal corporation who supply the dedicated DSP modems. Hal modems can also be used for RTTY and Pactor.

DIRECTLY GENERATING DATAMODE WAVEFORMS

Instead of using a soundcard plus upconverter for transmission, some datamodes lend themselves to direct generation at RF using suitable hardware.

Any simple FSK mode can be generated from a voltage tuneable oscillator, although it will be necessary to set the shift correctly. This simple scheme is directly applicable to DFCW / DFCWi where the keying lines can drive a varicap across the crystal determining the output frequency.

Binary PSK can be generated at low level with a double balanced mixer, by feeding the data as a bi-directional current into the IF port as shown in **Fig 20.24**.

The diagram shows an optimised filter for a 400 Baud signal as used for the telemetry on some amateur satellites. A mixer with an IF port response to DC is needed; one where both IF port connections are floating from ground is ideal as no negative power rail is then required for generating a bi-directional current. For use up to UHF frequencies the SRA-1 or SBL-1 type are perfectly adequate.

For more complex schemes a Direct Digital Synthesiser can be programmed from a microcontroller such as a PIC, in real time. Devices like the AD9852 [13] offer frequency, amplitude and phase programmability so for the lower frequencies where no further frequency multiplication is needed, this technique can be used to generate some quite advanced modulations. For details of direct PSK31 generation using a PIC and AD9852 DDS see [14].

Beacon GB3VHF uses the same DDS chip followed by a X2 multiplier to generate JT65B on 144MHz, the VHF beacon cluster at GB3RAL directly generates JT65 on 40/50/60/70MHz coherently phase locked to a master GPS locked reference.

DATAMODES ON BEACONS

An experimental beacon GB3SSS used first on 1.9MHz then on 3.6MHz generated a multi purpose beacon transmission containing CW ident, Stepped power and PSK31 all under precise GPS timed control.

For microwave beacons where frequency multiplication is in use, DDS devices followed by multipliers are not the preferred route due to excessive spurious outputs. Several beacons of the GB3SC# cluster use a DDS as part of a phase locked loop with a crystal source forming the variable element. By reprogramming the DDS in real time, slow frequency shift type datamodes can be generated, provided the PLL is fast enough to follow the keying.

For microwave beacons where more complex modulations are needed, an upconverter approach can be used with the DDS generating at VHF which is then upconverted to the final microwave frequency. G6GXK adopted this route for the GB3XGH 10GHz beacon.

Several amateur beacons now include a data mode as part of their transmission sequence giving users several advantages over normal CW or aural reception:

- Ability to receive and correctly decode several decibels below what the human ear can detect
- Automatic detection and logging
- Accurate Signal to Noise measurement as part of the decoding process
- The possibility of sending real time data from remote sites.

Callsign	Locator	Frequency MHz	Modulation types
GB3RAL	JO01EH	5.290	CW, Power steps, timed carrier, PSK31
GB3RAL	IO91IN	40.000 50.000 60.000 70.050	CW, JT65B, (four VHF freqs, phase locked)
G4JNT/P	IO80UU59	70.031	PSK31 Telemetry
GB3VHF	JO01EH	144.4300	CW, JT65B, phase reversals
GB3CSB	IO75XX57	1296.9850 2320.9850 3400.9850 (5.76 and 10GHz planned)	FSK-CW, JT4G
GB3SCS	IO80UU59	2320.9050	CW, JT4G
GB3SCX	IO80UU59	10368.9050	CW, JT4G
GB3SCF	IO80UU59	3400.9050	CW, RTTY
GB3SEE	IO91VG	10368.8500 24048.9600	FSK-CW, JT4G FSK-CW, JT4G
GB3CAM	IO92WI	24048.8700	FSK-CW, JT4G

Table 20.3: UK Beacons carrying Datamodes / MGM

A summary of the UK beacons carrying data modes is shown in **Table 20.3**. At HF there are not many possibilities for different modes because the allowed bandwidth is so limited. PSK31 or WSPR are probably the only really suitable options that give a low bandwidth and can be decoded by anyone using several free software packages.

At VHF, frequency stability issues are less of a problem and a stability of a few Hertz can be assumed, taking propagation induced anomalies and receiver instability into account. The JT65B mode, part of the WSJT suite with its 65 tones spaced by 5Hz, was selected for the first 2m beacon to adopt new data-modes. It was chosen for consistency since JT65B is already a de-facto standard for 2m EME and weak signal terrestrial comms.

On the microwave bands, with frequency scatter and receiver drift more significant, the wider spaced JT4G was selected for several beacons as its four tones spaced by 315Hz can punch-though chronic rain scatter and tuning drift over the 48 second duration of its signal.

So far, data has been added to beacons on a pretty-much *ad-hoc* basis with beacon designers doing their own thing, choosing tones and spacings almost arbitrarily with little standardisation.

For example, JT4G on the Bell Hill beacons, GB3SCS and GB3SCX, use the lowest of the four tones as the nominated reference (10368.905000MHz in the case of GB3SCX) meaning listeners have to set an SSB receiver to 10368.9042MHz to correctly decode it. A total of four tones are used.

Taking a different approach, GM6BIG, on the GB3CSB beacon carefully defines the centre of the tones as being the reference point where carrier and the mark tone for the FSK CW lies. CW space, and the four JT4 tones mean that beacon transmits six different tones.

PACKET RADIO

The use of packet radio has declined in recent years, but it is still used for specialist purposes such as APRS (Automatic Packet Reporting System) and DXCluster.

Much more information on this data mode can be found in a three-page article featuring on the CD-ROM attached to this book.

REFERENCES

[1] DFCW Software. www.qsl.net/on7yd/136khz.htm
[2] DFCWi software. www.g4jnt.com/DFCW.htm
[3] Hellschreiber modes: www.k3pgp.org/software.htm
[4] Digipan. www.digipan.net/
[5] MultiPSK. http://f6cte.free.fr/index_anglais.htm
[6] FlDigi. www.w1hkj.com/
[7] MMTty http://mmhamsoft.amateur-radio.ca/mmtty/
[8] ROS Software: http://rosmodem.wordpress.com/
[9] DL4YHF's WOLF GUI. http://www.qsl.net/dl4yhf/wolf/wolf_gui_manual.html
[10] I2PHD Jason. www.weaksignals.com
[11] K1JT WSJT and WSPR. http://physics.princeton.edu/pulsar/K1JT/
[12] WSPR Site and Report database. http://wsprnet.org/drupal/
[13] Analog Devices AD9852 DDS. www.analog.com/en/rfif-components/direct-digital-synthesis-dds/ad9852/products/product.html
[14] Direct PSK31 generation. www.g4jnt.com/beacons.htm

About the Author

***Andy Talbot** BSc, G4JNT, is a professional electronics engineer currently working in the field of radio communications and signal processing. He has been a radio amateur for 29 years, with his main interests being at the two opposite ends of the spectrum - microwaves and the LF bands. He has always been a home constructor, designing and building equipment from scratch right from the early days. Working on the new 73 and 137kHz bands started off his interest in DSP, and in conjunction with G3PLX pioneered the first use of ultra- narrowband techniques and low data-rate signalling on the amateur bands. Later, in conjunction with G4GUO, he took part in the first amateur use of digital voice at HF. He has written many articles for various journals, and writes the 'Data' column for the RSGB's* RadCom *magazine.*