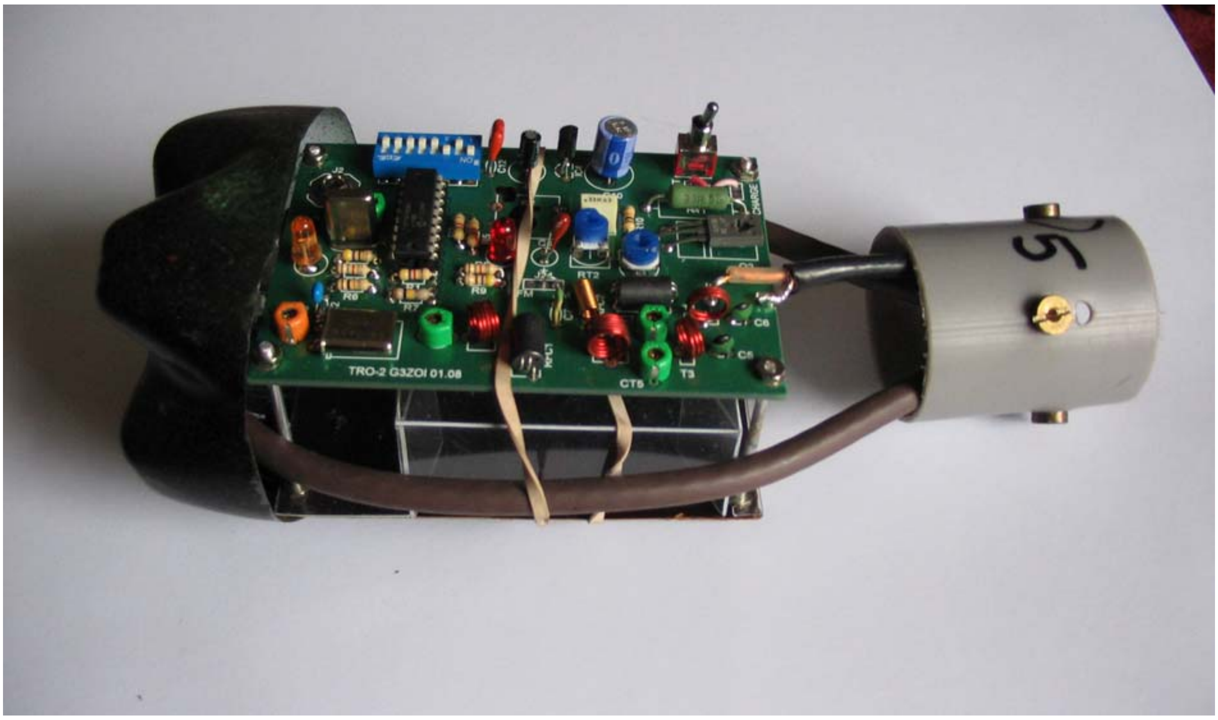
# OLDHAM AMATEUR RADIO CLUB

**G1ORC** **G4ORC**

# OLD HAMS NEWS

**The Journal of the Oldham Amateur Radio Club**

## April 2008




**RSGB Affiliated Society**

# ARDF MOVES ONWARD

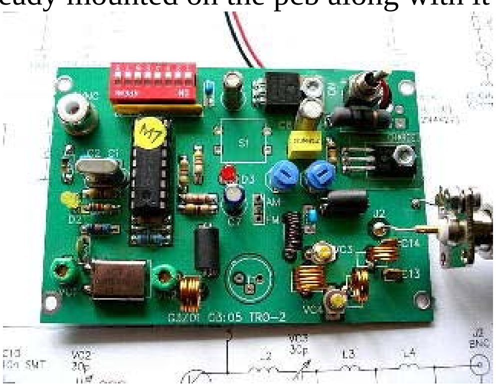
Our club's exploits and achievements in Amateur Radio Direction Finding (ARDF) are now becoming known nationally. The article that appeared in the January 2008 edition of Old Hams News was published in a brief form in Practical Wireless Magazine and in its complete form, taking up 3 full pages in the January 2008 edition of Radio User Magazine, thus giving valuable publicity for our club.

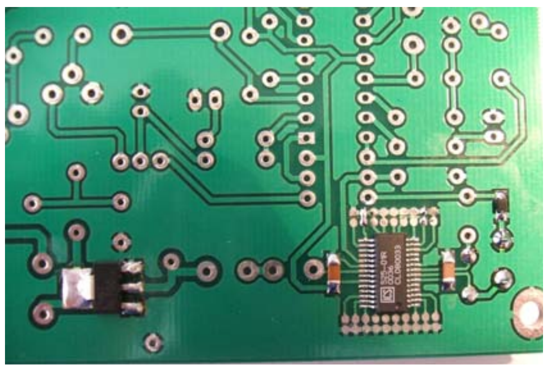
Since those few tentative steps were taken in Tandle Hill Park in October 2007 we organised and ran a national ARDF event in Lever Park, Bolton which attracted entrants from as far afield as Scotland and the South of England. Although this was regarded as a successful and notable event some contestants commented that the course was not as challenging as they would have liked. Also we were only able to arrange an 80 metre event where a full event would have included 2 metres as well.

The Lever Park event was organised in conjunction with the South East Lancashire Orienteering Club and now we have been requested to run another event by the Pennine Forest Orienteers. This is to take place on 26th April at Witton Country Park in Blackburn. At first our ARDF organisers, Phil Ellis M0GIE and Geoff Oliver G0BJR, were reluctant to be involved with it as there already was an event planned on the National ARDF Calendar for that weekend in Godalming. However when the Godalming event was cancelled due to lack of support the Witton Park event was adopted by the contestants instead.

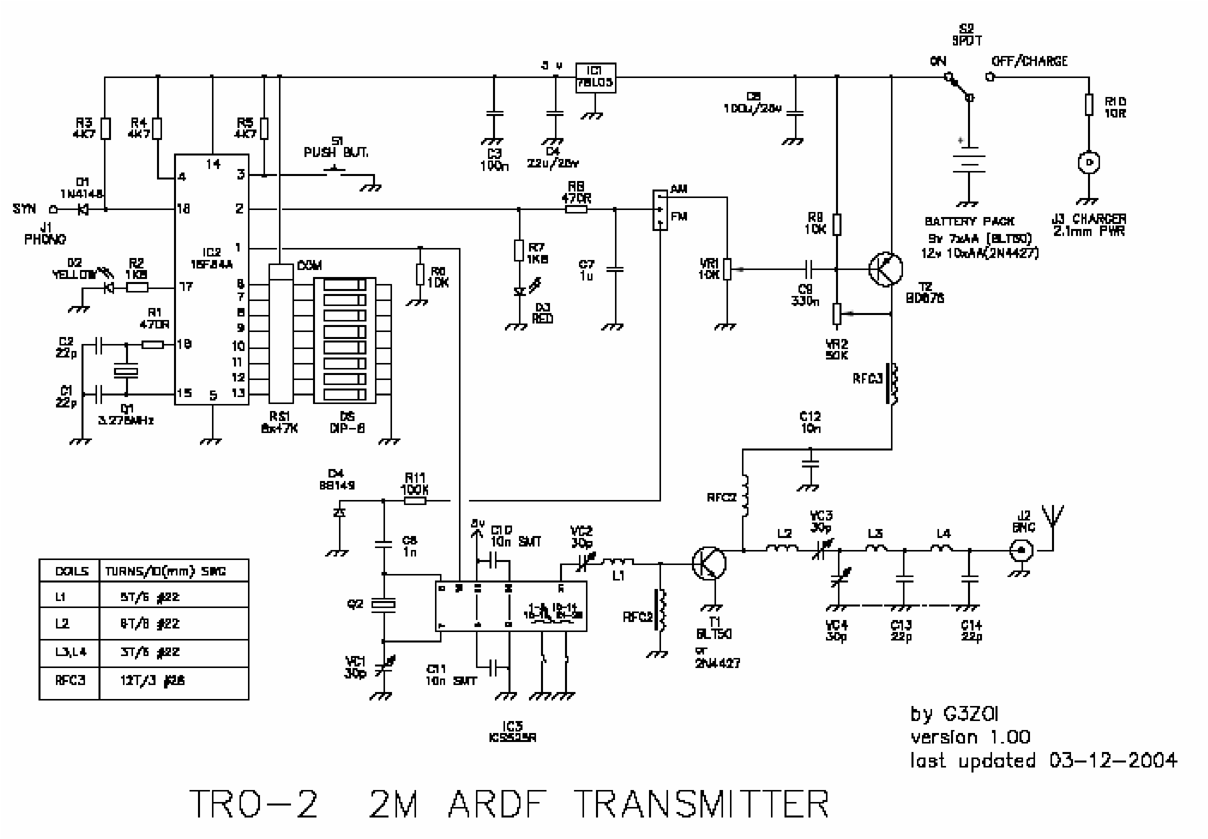
The Witton Park event will be a much more challenging prospect because, unlike Lever Park, the full area of the park will be available making it possible to increase the distance between the transmitters and include a height difference of over 300 feet. Also a full 80 metre and 2 metre event can be run because during the winter months Phil M0GIE has been beavering away building a set of six 2 metre transmitters with additional assistance from Geoff G0BJR and Alan Burgess G4GLV.

The 2 metre transmitter used is that described in the RSGB ARDF Handbook and is designed by Dave Dean G3ZOI. It was developed from designs by VE2EMM (RF strip), ON7YD (PIC timer) and DL3BBX (AM modulator). The TX specification gives 250-1000 mW output continuous carrier with AM, tone modulation. An ICS525R clock multiplier IC is used for the oscillator which radically reduces the component count compared with the normal multi-stage RF strip. This chip is a Surface Mount Device and when purchased from Dave Dean comes already mounted on the pcb along with it's associated SMD components.





The basic timing and event format of the ON7YD PIC software has been adopted and is much the same as the 80 metre version but the PIC outputs MCW tones instead of a keyed CW output, eliminating the need for a separate tone generator. The MCW square wave output, passes through a simple low pass filter, into a PNP, high gain Darlington pair transistor, to modulate the PA.

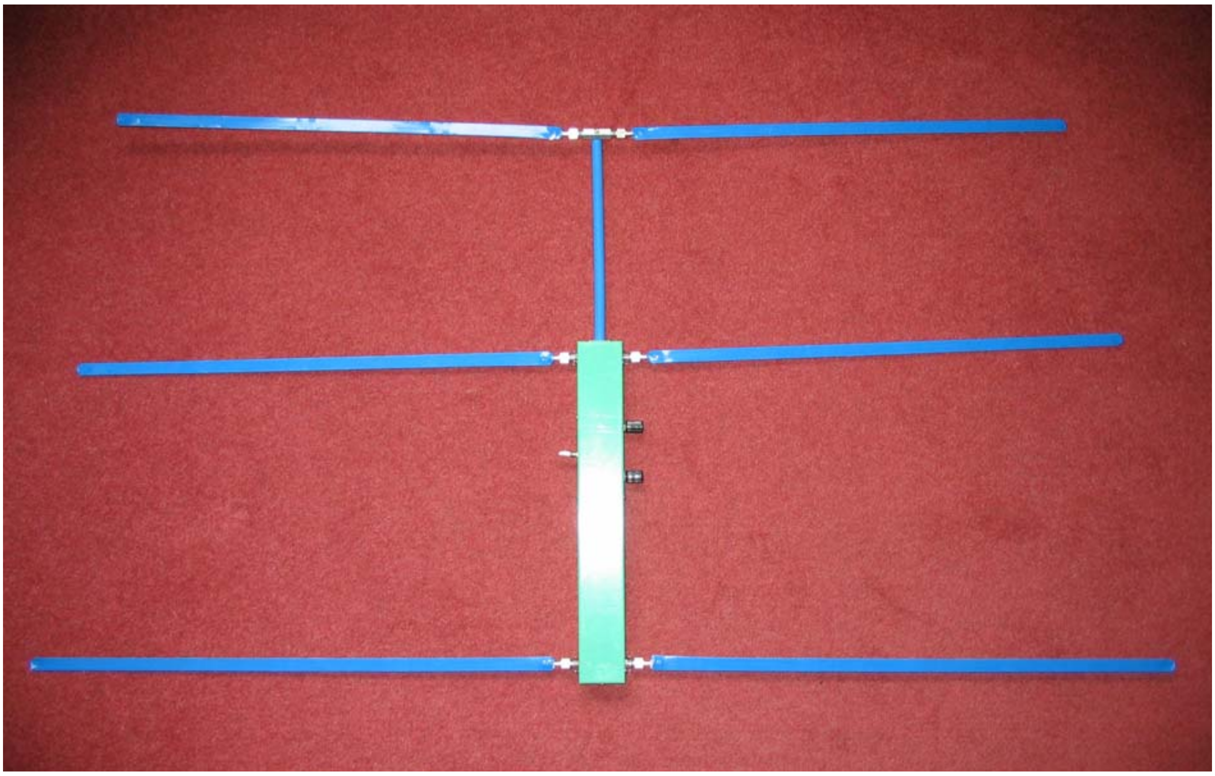



| COILS | TURNS/D(mm) SWG |
|---|---|
| L1 | 5T/5 #22 |
| L2 | 6T/8 #22 |
| L3,L4 | 3T/5 #22 |
| RFC3 | 12T/3 #28 |

The antenna is a Turnstile design resulting in a horizontally polarised output and the designer intended for the antenna to be connected to the transmitter with approximately 12 feet of 50 ohm coax. Unlike the 80 metre antenna which is just a 24 foot length of wire with a counterpoise which is fairly easy to conceal running down a tree trunk, this shape of antenna with a length of coax attached is not so easy to hoist into a tree and hide. However Phil has completely changed this by building the transmitter, battery pack and antenna as a single unit housed in a plastic drinks bottle which has been camouflage painted. The whole assembly can be hidden in the branches of a tree without too much effort.

(Please see front page photo of the assembly before it is placed inside the bottle). Initial tests of this design when in situ at Witton Park showed that the transmitter output could be heard very well some 5 kilometres (approx 3 miles) distant.

The receiver and antenna is a combined unit. The antenna is a typical 3 element Yagi design, however for portability it is made from very light materials. The antenna elements are made from precisely cut lengths of a steel measuring tape and the receiver itself (in the commercially built version) is constructed in such a way that it fits inside the Yagi's boom. It is equipped with tuning and volume controls and has a socket mounted at the rear end of the boom for either headphones or a small loudspeaker. In the homebrew version the receiver is housed in a small box mounted on the rear end of the boom.



In operation it is extremely directional and gives an accurate bearing of the transmitter. The front to back ratio is large enough for there to be no confusion when deciding whether or not the transmitter is ahead of, or behind you.

Let's hope this proves to be a challenging event for the competitors.

Photos by Geoff Oliver G0BJR and Alan Burgess G4GLV

# BACKPACKERS CONTEST SERIES

For many years the club participated in the RSGB backpackers contest. This year will be no different. In the past we have done the contest from Grains Bar, Moss Moor and various hills around Saddleworth moors. As time has gone on it has got harder and harder to find parking places that allow the space to put up the antenna system. Grains Bar had become a no go because the Council wardens who patrol the area thought that it was a public liability risk.

Saddleworth moors had become the same, the wardens had told use that we were spoiling the landscape and we could be a public liability risk. For this reason a few club members had started to look for a different location. It had to be close to the club for easy avalability to club members, transporting equipment and most importantly not used by anyone else.

After nearly two years of looking and trying different locations, some good and some poor. It was found that most good places had already been taken by other operators and it is not good manners to move in on someone else’s patch. A look closer to home found one place that showed potential - Tandle Hill Country Park.

2006 and club members that are involved in contesting decided to give the Backpackers contest a go from here. The park is 110 acres in size, has the largest area of beech trees in England and has panoramic views of the Pennines and the Manchester Plain.

Looking from the monument towards Manchester there are views on clear day all the way down the Pennines and over Cheshire, Derbyshire and Merseyside.

The park is used by the public for many different activities and this means that we still have be careful and it may take a few trips to get just the right location for use to operate from.

It is a big park with lots of opportunity.

At the monument you stand 165m above sea level and sit 8 miles away to the west of the Pennines. This should give us a better chance than up at Grains Bar where the station was situated in the middle of the hill with paths being blocked to the east and northeast. The park houses public toilets and a small café but the down side is that the café and toilets are on one side of the park and the monument is on the other side. Most of the public congregate around the café and it is for this reason that we have decided to operate from the monument. After unloading the car we carried the equipment across the hills and up to the monument. Just to the south the path drops down and there is a bit of land just off to the right that is perfect for operating the radio. The mast was quickly assembled with the 10 element Maspro put on top. Only 18 feet of pole was used, ideal for testing and seeing as this is the system that is used many times for contesting from different locations then it would give use a good comparison with other locations.

Over the next few months a couple of contests were done and the results checked against what had been achieved from Grains bar and Moss Moor. Even though the results were down on Moss Moor, which had been expected, the results based on Grains Bar showed promise. Some stations had reported a stronger signal and others were coming in better especially from the South East.

Running contests from Tandle Hill had proved a point. It could be used for contesting but only a serious contest entry could be done from Moss Moor. There is still the problem of carrying the equipment and being away from public conveniences. For this reason the site would be used for some of the backpackers in the future but they would be more the fun activities. In 2008 the wardens on Moss Moor have become more relaxed and the use of the car park on top is allowed. This has meant that we are back up at Moss Moor but listen to club announcements as to the exact location.

If you do decide to go into Tandles don't forget to take your hand held and see what you can work, the woods are good, the café can do a good brew and the views are good on a clear day.

Have fun and enjoy the park, Chris Cunliffe G7OOD.

# THE DIGITAL HAM RADIO REVOLUTION

## Introduction

Communication technologies that are specifically designed to improve "live" HF keyboard operations that were previously only theory or too complex to be practical can now be achieved.

Thanks to the generosity of radio hams with programming knowledge, and to the World Wide Web, new and powerful communications tools are available to all hams. The evolution and wide spread use of the Personal Computer with a digital sound card for DSP, is allowing us to use these tools to "push the envelope". The distinguishing features of live HF digital operation today are the use of lower power, compact or indoor antennas and courteous operating technique. This reverses the trend of several years ago...

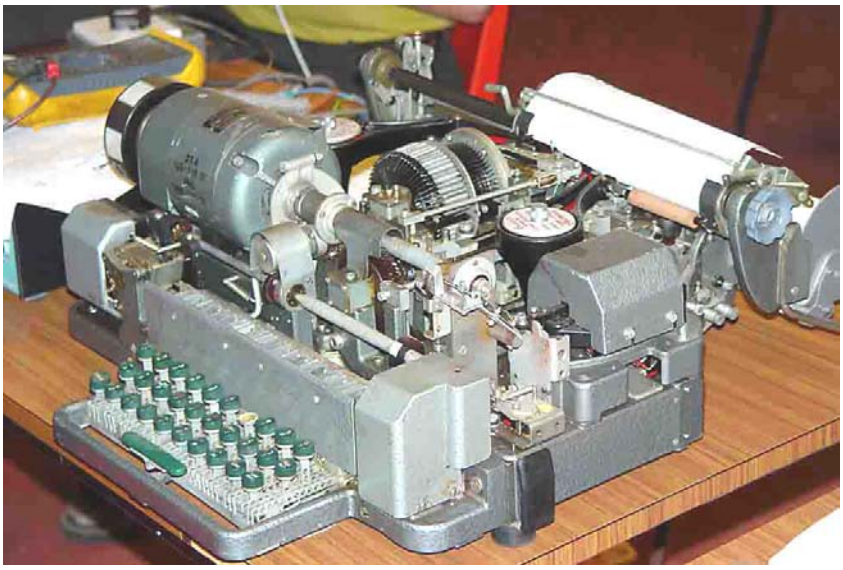
PSK31 mode led the way starting in 1997, and since then experimentation has shown that incremental improvements can be made. The popularity of a single mode, like PSK31, over other new modes seems to be driven at this time by how many freeware programs are available for the mode. It is possible that a more advanced mode like MFSK16 will emerge as a standard for HF band operation in the future. We can all participate in the revolution by trying out the other modes and judging their performance on all of the HF bands. Fortunately, the interface needed to operate these new PC sound card programs is the same for all the modes. The next challenge for the ham programmers out there is to create a single program that will incorporate modules for all the new sound card modes

Confusion over band space is the obvious down-side as new and old modes compete for band space. Crowding on a single band like 20 meters is partly to blame for this issue. Fortunately, the new modes, like MFSK16, are designed to improve performance for a wide range of operating conditions. This should allow for increased ham band usage to relieve crowding and extend contact opportunities as propagation changes to favour different bands. I don't know what is happening with the phone portion of the ham bands, but these are exciting times for us digital operators!

How has interest in HF Digital Ham Radio grown since the release of the first Windows program for PSK31 mode?

# RTTY

RTTY or Radio Teletype is a direct machine to machine communications mode using the Baudot (or Murray) code. This mode became popular with many amateurs when surplus TTY machines became available at a reasonable cost after World War II. These mechanical monsters provided a keyboard for Input and a paper roll for printed Output. They were also useful to help hold the house down in times of hurricane winds - they must weigh a ton. Video displays were still too exotic and expensive in those days. It was not until the mid 1970s that we began to see the Video Display come into more widespread use. (By the way, have you ever wondered why early Program Languages like BASIC use the command PRINT to display their output?)



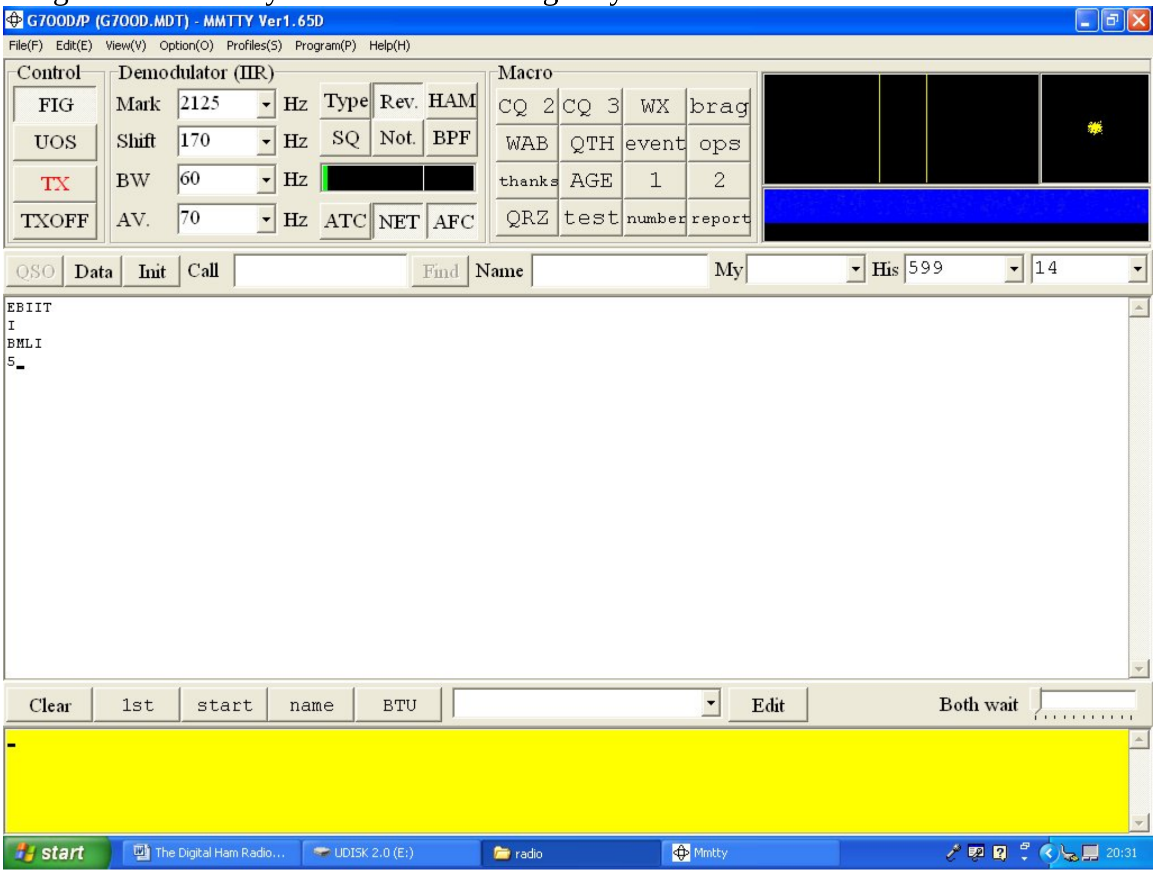
When transmitting Morse Code, the transmitter is switched on and off to make the dits and dahs. When sending Teletype however the transmitter runs continuously, sending either of two frequencies conventionally known as Mark and Space (a reference to paper tape reception of telegraphy). The early pioneers found on-off keying was not all that successful for Teletype signals because of interference from static. They experimented with FSK or Frequency Shift Keying and found it performed much better. With FSK, the transmitter is shifted up in frequency every time a Mark is to be sent, reverting to the lower frequency for a Space. The amount of the shift is usually 170 Hz for Amateur Radio use although many commercial Teletype signals use other shifts, notably 425 Hz and 850 Hz.

Many systems use AFSK or Audio Frequency Shift Keying. When this is sent, the transmitting station generates the Mark and Space audio tones and feeds them into the transmitter's microphone input. The result at the receiving end is that the same audio tones are heard and processed, whether the transmitting station used FSK or AFSK. When listening to a teletype signal off air, you will soon get to recognise the familiar warble of Mark and Space tones.

In the modern amateur shack the TTY machine is usually a Multi-mode controller or PC interface cable connected to an HF transceiver which the operator tunes so that the received audio is just the right pitch or audio frequency to trigger the demodulator's Mark and Space response.

If the transceiver is slightly off the correct frequency the tones vary and the text becomes garbled or even lost altogether. To help the other station tune the receiver correctly, a RTTY operator can send a string of alternate R and Y characters RYRYRYRYRY. This pattern is chosen as it produces the most frequent and almost symmetrical alternation of Mark and

Space tones, giving the receiving operator the best chance to tune the receiver before the "real" message starts. However, even if the signal is accurately tuned, the information can become garbled or completely lost due to interference, fading, or noise. Often, it is possible to make sense of the message even with parts missing, but RTTY is by NO means an error free mode! The new DSP based programs such as MMTTY, are able to decode RTTY signals with much greater sensitivity than the older analogue systems.




MMTTY Screen Shot

The Baudot code is a 5 bit code and those of you who are familiar with Binary Notation will know that the maximum number of values we can have with 5 bits is 32. That means that each unit of transmission, one keystroke if you like, can contain any one of 32 possible values. If you look up a table of Baudot codes you will see there are 32 values listed, one code for each letter of the alphabet plus a few other codes for other things such as a space and a Carriage Return. But, what if we want to send a number such as "9" or a question mark? These are not mentioned in that table because all 32 codes are already used.

The solution is rather similar to the Typewriter or Computer Keyboard where we have the Shift key to get various additional codes from the keyboard. Most keys will produce a different result if we hold down the Shift key as we type. Well, one of those original 32 codes is a special code known as FIGS (for Figures Shift). The convention is that when we want to send a number or some other special character such as a punctuation mark, we can do that by firstly transmitting a FIGS code.

Then instead of using that original table of 32 codes, we have a second table of codes to use, and that second table includes all ten numeric digits and various punctuation marks. Provided both sides of the conversation observe the convention, the sender can send a FIGS and start using the second table; the receiver will see the FIGS code and it too will interpret all data that follows from the second table.

With just 5 bits of data we then have almost 64 different codes we can send and receive. (I say almost because there is some duplication in the two tables, including a space and a Carriage Return but that is not important here). Even that many codes is not enough to handle all 26 letters of the alphabet in both UPPER and lower case, so RTTY systems always operate in upper case only.

If we wanted to type a big number (say "13579") we don't have to send FIGS before every digit. We send that code only once and the receiver then will take EVERYTHING we type from now as if it belongs in the second table. When we want to revert to the normal alphabetic table of codes we can send another special code, this one called LTRS (for Letters Shift). Then everything goes back to normal, using the original alphabetic table of codes. Normally we don't have to concern ourselves with these FIGS and LTRS codes. Our computing equipment will take care of those things for us. We just type away and rely on the system to generate and send those codes when necessary.

It is quite possible to lose bits here and there when receiving a RTTY signal, whether it be because of fading, interference, frequency drift, or whatever. One of the big problems with lost data is the possible loss of a FIGS or LTRS code! Say we had sent "13579" and then typed "HAPPY BIRTHDAY". Our equipment would have sent a LTRS code before the first "H" but what if the receiver did not copy the LTRS code we sent? Can you imagine what happens? As far as the receiver is concerned we are still sending numbers or other codes from the numeric table! So our "HAPPY BIRTHDAY" is going to come out looking something like "#-006 ?845#$-6". And EVERYTHING we type from then on is going to look just as strange until we happen to send another LTRS code later. It is for this reason that many systems include an option to "Un-shift on space". If you have a multi mode TNC capable of handling RTTY, you will probably have this option in your TNC. If that option is ON then your receiving system will imply a LTRS code every time it receives a space. So if you seem to be copying lots of funny numbers from a strong, well tuned signal, try setting that option ON.

We can overcome some of these problems by using ASCII instead of using the Baudot code. With ASCII we can have 128 different codes so we do not need the FIGS/LTRS codes. All Personal Computers use ASCII as their native "language" so it would be a reasonable thing to use. Although not part of the defined ASCII standard, it has become an almost de-facto standard in the computer world that an additional 128 characters are available, often called Extended ASCII. But, despite these benefits, Baudot continues to rule the airwaves for Amateur and Commercial Teletype transmissions.

Today, RTTY is still a popular mode especially on the HF bands, and the advent of the "Glass Terminal", firstly the Dumb Terminal and now the Personal Computer, has brought this mode to even more operators the world over. Many specialised RTTY systems were developed for the Amateur enthusiasts but have been superseded now by the Personal Computer with one of the Multi Mode TNCs or sound card DSP programs which handle RTTY and many other modes besides. The latest Computerised RTTY equipment generally allows us to use the mode better, quieter, more efficiently, using less power and occupying less space than the old TTY machines, but the limitations of the mode remain.

## BACKGROUND: THE PSK31 PHILOSOPHY

PSK31 is the result of my belief that the present batch of "data" modes have left a gap in amateur radio operating, the gap that was previously filled by AMTOR or even traditional RTTY, in which two or more operators chat to each other on an open channel. Modes such as packet radio, Pactor, and others, are highly complex, are unsuited to multiway conversations, and in particular, the long block lengths introduce an unacceptable delay in the processing of text such that even normal conversation is unpleasant and quick-break question/answer sessions are impossible. The move to automated unattended message forwarding has left a gap in the person-to-person communication field, and PSK31 is an attempt to remedy this situation with a simple but efficient code structure coupled with the narrowest possible bandwidth, and with only enough error-correction to match typical typing-error rates, and with no time-consuming synchronisation, changeover, and ARQ processes.

The 31 baud BPSK modulation system used in PSK31 was introduced by SP9VRC in his SLOWBPSK program written for the EVM. Instead of the traditional frequency-shift keying, the information is transmitted by patterns of polarity-reversals (sometimes called 180-degree phase shifts). This process can be thought of as equivalent to sending information by swapping-over the two wires to the antenna, although, of course, the keying is more usually done back in the audio input into the transceiver. A well-designed PSK system will give better results than the conventional FSK systems that amateurs have been using for years, and is potentially capable of operation in much narrower bandwidths than FSK. The 31 baud data rate was chosen so that the system will just handle hand-sent typed text easily.

There is a problem with PSK keying which doesn't show up with FSK, and that is the effect of key-clicks. We can get away with hard FSK keying at moderate baudrates without generating too much splatter, but polarity reversals are equivalent to simultaneous switching-off of one transmitter and switching-on of another one in antiphase: the result being keyclicks that are TWICE AS BAD as on-off keying, all other things being equal. So if we use computer logic to key a BPSK modulator such as an exclusive-or gate, at 31 baud, the emission would be extremely broad. In fact it would be about 3 times the baudrate wide at 10dB down, 5 times at 14dB down, 7 times at 17dB down, and so on (the squarewave Fourier series in fact)

The solution is to filter the output, or to shape the envelope amplitude of each bit which amounts to the same thing. In PSK31, a cosine shape is used. To see what this does to the waveform and the spectrum, consider transmitting a sequence of continuous polarity-reversals at 31 baud. With cosine shaping, the envelope ends up looking like full-wave rectified 31Hz AC. This not only looks like a two-tone test signal, it IS a two-tone test signal, and the spectrum consists of two pure tones at +/-15Hz from the centre, and no splatter. Like the two-tone and unlike FSK, however, if we pass this through a transmitter, we get intermodulation products if it is not linear, so we DO need to be careful not to overdrive the audio. However, even the worst linears will give third-order products of 25dB at +/-47Hz (3 times the baudrate wide) and fifth-order products of 35dB at +/-78Hz (5 times the baudrate wide), a considerable improvement over the hard-keying case. If we infinitely overdrive the linear, we are back to the same levels as the hard-keyed system.

There is a similar line of reasoning on the receive side. The equivalent to "hard-keying" on the receive side is a BPSK receiver which opens a gate at the start of a bit, collects and stores all the received signal and noise during the bit, and then "snaps" the gate shut at the end. This process gives rise to the receive-side equivalent of key-clicks, namely sidelobes on the receiver passband. So, although this "integrate-and-dump" method is 100% efficient in the task of sorting out signal from noise, it will only reject signals by 10dB at 3 times the baudrate wide and so on, the same spurious rejection figures that we got as spurious emission figures for the transmit side. The PSK31 receiver overcomes this by filtering the receive signal, or by what amounts to the same thing, shaping the envelope of the received bit. The shape is more complex than the cosine shape used in the transmitter: if we used a cosine in the receiver we end up with some signal from one received bit "spreading" into the next bit, an inevitable result of cascading two filters which are each already "spread" by one bit. The more complex shape in the receiver overcomes this by shaping 4 bits at a time and compensating for this intersymbol interference, but the end result is a passband that is at least 64dB down at +/-31Hz and beyond, and doesn't introduce any inter-symbol-interference when receiving a cosine-shaped transmission.

Note that the transmitter and receiver filters have to be "matched" to each other for the ISI performance to be right. Some systems like this use a pair of identical receive and transmit filters which are matched. If I did this and someone else came along wanting to improve the performance, they would have to get everyone else to change their transmit filters. I have therefore chosen to use the simple cosine shape for the transmitter and match that in the receiver. This leaves the way open for others to develop better receivers without new transmitters being incompatible with old. This is slightly different from the SP9VRC approach.

To summarise: PSK31 has been designed not only to give all the weak-signal-in-white-noise advantages that PSK has to offer, but to go further and optimise the performance in the presence of other signals, to reject them on receive and not to interfere with them on transmit. PSK31 is therefore ideally suited to HF use, and would not be expected to show any advantage over the hard-keyed integrate-and-dump method in areas where the only thing we are fighting is white noise and we don't need to worry about interference.

## THE QPSK MODE

In December 1997, PSK31 introduced the QPSK mode. In this mode, instead of just keying by phase reversals, that is, 180-degree phase-shift, an additional pair of 90 and 270 degree phase-shifts are possible. If you thought of BPSK as reversing the polarity of the signal, then QPSK can be thought of as two BPSK transmitters on the same frequency but 90 degrees out of phase with each other. By thinking of the receiver as being two BPSK demodulators at 90 degrees, we have two channels sharing the same frequency, but of course, with only half the transmitter power in each. We therefore have twice the bit-rate but at 3dB less signal-to-noise ratio. We could use this feature to transmit data at twice the speed with 3dB less noise margin.

The PSK31 philosophy is to stay at the speed needed to handle hand-keyed text, so why do we consider QPSK at all? The answer is that we can use the extra capacity to reduce the error-rate while keeping the bandwidth and the traffic speed the same. Note that because we have a 3dB SNR penalty with QPSK, any error-correction scheme we introduce has to be at least good enough to correct the extra errors which result from the 3dB SNR penalty, and preferably a lot more, or it will not be worth doing. By doing simulations in a computer, and tests on the bench with a noise generator, it has been found that when the bit error-rate is less than 1% with BPSK, it is much better than 1% with QPSK and error-reduction, but when the BER is worse than 1% on BPSK, the QPSK mode is actually worse than BPSK. Therefore, if we are dealing with radio paths where the signal is just simply very noisy, there is actually no advantage to QPSK at all!

However, all the tests we have done on the air show that QPSK with the chosen error-reduction scheme is better than BPSK, except where we have deliberately attenuated the signal to make it artificially weak. Typical radio circuits are far from being non-fading with white noise. Typical radio paths have errors in bursts rather than randomly spread, and error-reduction schemes can give useful benefits in this situation in a way that cannot be achieved by anything we can do in the linear part of the signal path. With the code used in PSK31, a 5:1 improvement is typical, but it does depend on the kind of path being used. For this reason it is worth keeping both modes available and remembering that there may be times when one mode works better than the other and others when the reverse will be the case. When comparing PSK31 with other modes, remember that the switch between "straight" and "error-corrected" modes in PSK31 is done with both the bandwidth and the data-rate remaining the same. In most other systems that can switch, either the bandwidth or the data rate changes when the system switches, and the figures for error-rate improvement can be misleading unless they are carefully compared.

The error-reduction code chosen is one of a type known as convolutional codes. The code systems used in the past have been block codes, where each character is a fixed-length code, and a fixed number of extra bits are added to make a longer block, and this longer block is capable of correcting errors within itself. These extended blocks are then transmitted as a serial bitstream. In a convolutional code, the characters are converted to a bitstream and then this bitstream is itself processed to add the error-reduction qualities. There is no relationship between the boundaries between characters and the error-reduction process. Since the channel errors are also not related in any way to the character boundaries, convolutional codes are better suited to serial links than block codes, which were originally designed for protecting errors in memory banks and similar structures.

It is not quite correct to refer to the convolutional code system as "error-correcting", since the raw data is not actually transmitted in it's original form and therefore it makes no sense to talk about it being corrupted by the link and corrected in the decoder. In PSK31, the raw data is transformed from binary (1 of 2) to quaternary (1 of 4) in such a way that there is a precisely known pattern in the sequence of quaternary symbols. In the code used in PSK31, the pattern of quaternary symbols is derived from a run of 5 consecutive data bits. For example, if we label the four phase-shifts as A, B, C, and D, and suppose that the transmitter sends continuous A's when the raw datastream is sending continuous 0's. Because the convolutional encoder works on a run of five bits, when the datastream sends ..000010000..., the transmitter actually sends ..AAAADCCBDAAAA..., that is, each binary bit to be transmitted results in a unique 5-symbol sequence, overlapping with the sequences from adjacent bits, in a predictable way which the receiver can use to estimate the correct sequence even in the presence of corruptions in parts of the sequence.

The decoder, known as a Viterbi decoder after the man who thought of it, is not really a decoder at all, but a whole bank of parallel encoders, each fed with one possible "guess" at the transmitted data sequence. The outputs of these parallel encoders are all compared with the received symbol-stream. Each time a new symbol is received, the encoders need to add an extra bit to their sequence guesses and consider that the new bit might be a 0 or a 1. This doubles the number of sequence guesses, but a clever technique allows half of all the guessed sequences to be discarded as being less likely than the other half, and this means that the number of guesses being tracked stays constant. After a large number of symbols have been received, the chances of a wrong guess at the first symbol tends to zero, so the decoder can be pretty sure that the first bit was right and it can be fed to the output. In practice this means that the decoder always outputs decoded data bits some time after they have been received. This delay in PSK31 is 20 bits (640mS) which is long enough to make sure that the decoder has done a good job, but not so long that it introduces an unacceptable delay in displaying the received text.

## INFORMATION CODING: VARICODE

This is a description of the variable-length coding used in the 31.25 baud BPSK system.

The normal asynchronous ASCII coding used on the original version of this system by SP9VRC, and indeed the asynchronous system used for transmission of RTTY for the last 50 years, uses one start-bit, a fixed number of data-bits, and one or more stop-bits. The start-bit is always the opposite polarity to that of the stop-bit. When no traffic is being sent the signal sits in stop polarity. This enables the receiver to start decoding as it receives the edge between the stop-signal and the start-bit.

One disadvantage of this process is that if, during a long run of traffic, an error occurs in either a stop-bit or a start-bit, the receiver will lose synchronisation, and may take some time to get back into sync, depending on the pattern of following characters: in some situations of repeated characters the receiver can even stay in a false sync. for as long as the repeated pattern persists.

Another disadvantage of this system arises when, as will be the case for normal amateur radio contacts, the traffic being sent consists of plain language. In all languages there are some characters which occur more often that others and there are some which may hardly ever be used. In morse code this is used to advantage by using short codes for the common letters and longer codes for less-common ones. In the asynchronous start-stop system all characters are necessarily the same length, and so the overall speed of transmission of plain-language is not as fast as a variable-length code would be.

The variable-length code used in the BPSK system overcomes both these disadvantages, and works in the following way.

1. All characters are separated from each other by two consecutive 0 bits.
2. No character contains more than one consecutive 0 bit.

It follows from this that all characters must begin and end with a 1.

With such a code, the receiver detects the end of one code and the beginning of the next by detecting the occurence of a 00 pattern, and since this pattern never occurs inside a character, the "loss of sync" problem that occurs with asynchronous systems can never occur. The 00 gap between characters is equivalent to the gap between letters in morse code in this respect, and in a similar way allows the possibility of a variable-length code system.

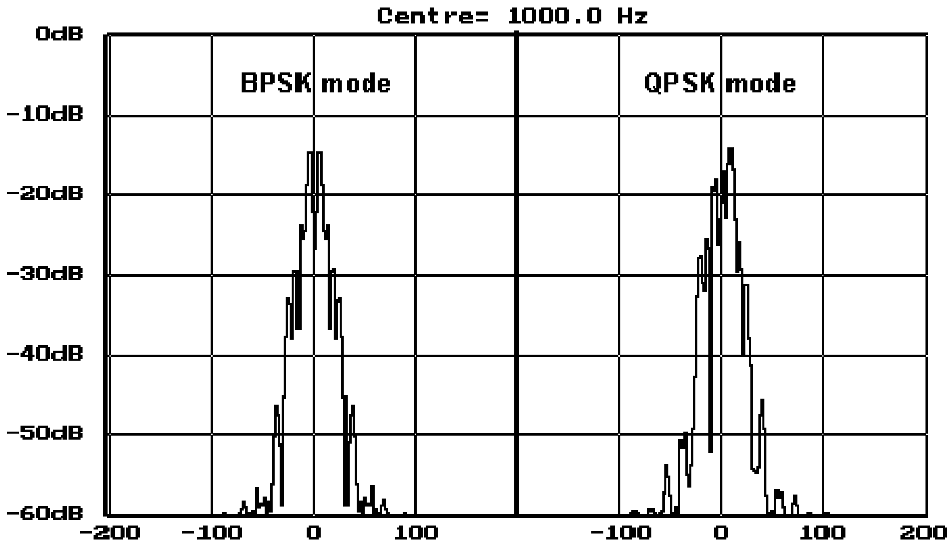
The variable-length coding used in the BPSK system was chosen by collecting a large volume of English language ASCII text files and analysing them to establish the occurrence-frequency of each of the 128 ASCII characters. Next a list was made of all the binary patterns that meet the above rules, namely that each pattern must start and end with a 1, and must not contain more than 1 zero in a row. This list was generated by computer, starting at the shortest. The list was stopped when 128 patterns had been found. Next the list of ASCII codes, in occurence-frequency order was matched to the list of binary patterns, in length order, so that the most frequently-occuring ASCII codes were matched to the shortest patterns, and that completed the variable-code alphabet. To finish the job, a simple calculation was made to predict the average number of bits in typical plain language text transmitted by this code, taking into account the 00 gap between characters. The result was between 6 and 7 bits per character. This compares very favourably with 9 bits per character for the asynchronous system.

The actual alphabet is shown below, shown in ASCII order starting with NUL and ending with DEL.

Note that the lower case letters have the shortest patterns and so are the fastest to transmit.

| | | | | | |
|---|---|---|---|---|---|
| NUL | 1010101011 | + | 111011111 | V | 110110101 |
| SOH | 1011011011 | , | 1110101 | X | 101011101 |
| STX | 1011101101 | - | 110101 | Y | 101110101 |
| ETX | 1101110111 | . | 1010111 | Z | 101111011 |
| EOT | 1011101011 | / | 110101111 | [ | 1010101101 |
| ENQ | 1101011111 | 0 | 10110111 | \ | 111110111 |
| ACK | 1011101111 | 1 | 10111101 | ] | 111101111 |
| BEL | 1011111101 | 2 | 11101101 | ^ | 111111011 |
| BS | 1011111111 | 3 | 11111111 | _ | 1010111111 |
| HT | 11101111 | 4 | 101110111 | . | 101101101 |
| LF | 11101 | 5 | 101011011 | / | 1011011111 |
| VT | 1101101111 | 6 | 101101011 | a | 1011 |
| FF | 1011011101 | 7 | 110101101 | b | 1011111 |
| CR | 11111 | 8 | 110101011 | c | 101111 |
| SO | 1101110101 | 9 | 110110111 | d | 101101 |
| SI | 1110101011 | : | 11110101 | e | 11 |
| DLE | 1011110111 | ; | 110111101 | f | 111101 |
| DC1 | 1011110101 | < | 111101101 | g | 1011011 |
| DC2 | 1110101101 | = | 1010101 | h | 101011 |
| DC3 | 1110101111 | > | 111010111 | i | 1101 |
| DC4 | 1101011011 | ? | 1010101111 | j | 111101011 |
| NAK | 1101101011 | @ | 1010111101 | k | 10111111 |
| SYN | 1101101101 | A | 1111101 | l | 11011 |
| ETB | 1101010111 | B | 11101011 | m | 111011 |
| CAN | 1101111011 | C | 10101101 | n | 1111 |
| EM | 1101111101 | D | 10110101 | o | 111 |
| SUB | 1110110111 | E | 1110111 | p | 111111 |
| ESC | 1101010101 | F | 11011011 | q | 110111111 |

| | | | | | |
|---|---|---|---|---|---|
| FS | 1101011101 | G | 11111101 | r | 10101 |
| GS | 1110111011 | H | 101010101 | s | 10111 |
| RS | 1011111011 | I | 1111111 | t | 101 |
| US | 1101111111 | J | 111111101 | u | 110111 |
| SP | 1 | K | 101111101 | v | 1111011 |
| ! | 111111111 | L | 11010111 | w | 1101011 |
| " | 101011111 | M | 10111011 | x | 11011111 |
| # | 111110101 | N | 11011101 | y | 1011101 |
| $ | 111011011 | O | 10101011 | z | 111010101 |
| % | 1011010101 | P | 11010101 | { | 1010110111 |
| & | 1010111011 | Q | 111011101 | \| | 110111011 |
| ' | 101111111 | R | 10101111 | } | 1010110101 |
| ( | 11111011 | S | 1101111 | ~ | 1011010111 |
| ) | 11110111 | T | 1101101 | DEL | 1110110101 |
| * | 101101111 | U | 101010111 | | |



The narrow bandwidth of both psk31 modes is shown above.

## CONTACT INFORMATION

The source code is freeware, provided it is used only for amateur purposes. If you have suggestions for improvements, or you find bugs, please report them back to Peter G3PLX and do not broadcast your own modifications or bug-fixes.

Peter Martinez G3PLX
High Blakebank Farm
Underbarrow
Kendal
Cumbria LA8 8BN
United Kingdom
peter.martinez@btinternet.com

## SOUND CARD DSP SETUP & OPERATING TIPS

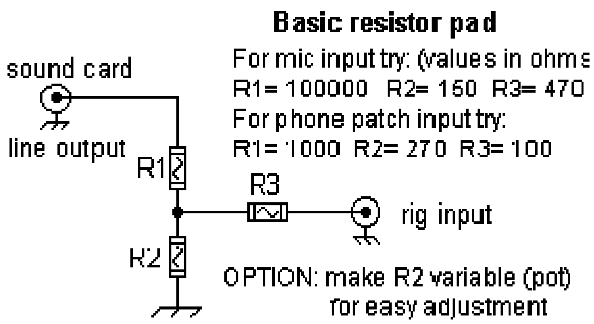
For most Ham Radio situations it will not be necessary to isolate with opto-couplers or transformers between the PC sound card and the radio. A simple interface between the computer and the radio can be constructed in a few hours time using junk box or Radio Shack parts. It is very important to use shielded cables and extra attenuation to the radio input. You can use small coax for the interface cable if you are careful to keep the shield on the "cold" side of the circuit. Make sure the shield is grounded to the radio chassis on any type cable you do use. Locate the appropriate connectors to fit your sound card and the radio jacks. A simple resistor network is used to attenuate the audio signal to the input of the MIC or AUX IN connector of the radio. If you are not handy with the soldering iron, several commercial interfaces are offered via the web, such as RigBlaster, to name one. The interface here applies to all the new sound card digi modes including SSTV!

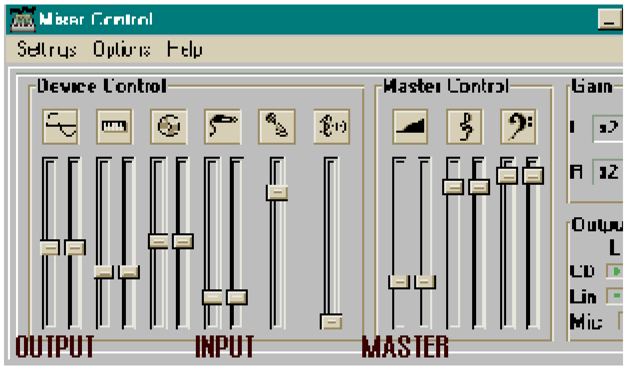
To build your own interface follow these instructions. Connect a shielded audio cable from the audio output of your SSB transceiver to the LINE IN jack of your sound card. Connect to either left, right or both of the stereo inputs. (Don't use the MIC input on the sound card.) It's best to use the auxiliary audio output on the transceiver which does not vary in level with speaker volume. This single cable is all that is needed to monitor digi signals with a program.

Make another shielded cable, that includes a simple resistor attenuation pad, between the sound card LINE OUT and the transceiver audio input. (See diagram below.) The pad will prevent high inter-modulation distortion (IMD) caused by overdriving the first audio amp stage of the transceiver. This resistor pad is not strictly necessary because you can adjust the sound card output to a very low level by using the Windows Master Sound Control Panel, however this will cause your PC system sound levels to be barely audible. (Ham programs like the MixW V2.0 will allow you to independantly adjust the audio out for each mode, but this is not found in other programs.) Using the MIC Gain control may not be adequate to prevent overdriving the first audio amplifier stage on some radios. Using a variable resistance pad will give you flexibility to balance your audio levels for use with other PC devices. Finally, use the sound card mixer control program (see example below) to make final adjustments of the audio levels.

Adjust the sound card audio INPUT level such that a nominal signal tuned in on the transceiver will be seen prominently in the program signal display window without overdriving the sound card input. Use the MASTER control to adjust the sound to the PC speakers (if you use them) and then re-adjust the other levels as needed. Note that other windows applications you run on your PC that provide sound card adjustment controls (such as WinAmp) may change your PSK31 settings! You may have to re-adjust these levels. Generally, once these levels have been set, you will not need to change them. Adjust the sound card audio OUTPUT level to the transceiver such that a small amount of ALC voltage is detected at the transceiver while transmitting into a proper antenna load (50 ohms) with the MIC Gain control at a nominal setting. After this adjustment, you can reduce the MIC gain control slightly to achieve full output power with no ALC deflection. Note that full output on a 100 watt transceiver would be about 100 watts if using a single tone (CW) from the sound card, but would be about 50 watts if using the double tones of a PSK31 carrier.

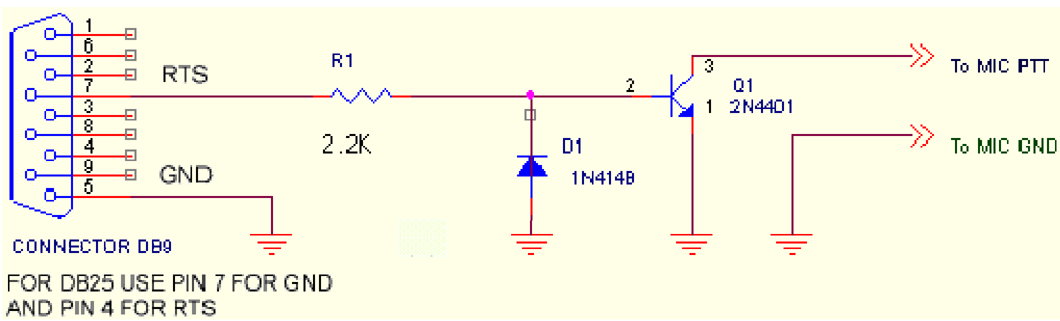
When you get your first PSK31 contact, ask the other station for an "IMD report" on your signal. (You must allow several seconds of "no text" for the reading to be made.) A properly adjusted audio input level to your transceiver will have an IMD at least -20 dB below the carrier and a very clean signal will be around -28 dB. It should be noted that a strongly received PSK31 signal (S9 or greater) can appear to have a high IMD level when viewed on the digi program display. This is often the fault of the receiver's front-end being over driven by the strong signal. Before issuing a bad IMD report to another station, you should add attenuation to the receiver and observe the reading under a nominal signal strength.




The most simple sound card to radio interface is with a single shielded audio cable between the transceiver AUX audio output and sound card input, and another between the sound card output and the AUX audio input on the back of the transceiver. Using a program like MixW, that supports CAT functions for most popular radios, a single RS232 cable is used to control the transceiver for TX and RX. For this setup, the audio output from the sound card must be adjusted very low to achieve a clean signal.

If you don't have VOX or CAT operation to act as your transceiver T/R switch, you can make a simple circuit that will switch your T/R automatically as you send and receive from the program. Use the DTR or RTS and GND pins from your computers unused serial comport to turn ON/OFF a simple transistor switch that operates the transceiver PTT input line. The configuration menu in your digi program will let you select the com port you will use.




The homemade interface described above need not be a complicated work of electronic design! A picture of a simple MIC attenuator wired with a variable resistor and a RS232 PTT switch DB9 connector that were both in use for several years is shown .

A tip for good PSK31 operation is to use the split VFO function of most modern transceivers. Set up the transceiver to allow transmission in the SSB mode and reception in the CW mode, so that the CW narrow IF filter can be used to eliminate QRM.

Another tip to help detect weak PSK31 signals in the presence of stronger signals is to turn OFF the receiver's AGC feature. This will allow the receiver to operate at full linearity and not reduce gain when a strong signal is present. Adding some attenuation to the front-end and using a narrow IF filter is advisable. Without receiver AGC your S-meter will not function and some strong signals may appear to be distorted, but with a sensitive waterfall display and a radio with good dynamic range, you will still be able to decode the weaker signals.

Operate at the power level needed to produce good printing. Operate PSK31 mode at half (or less) the rated power of the transceiver. Operate all digi modes at a level that produces no ALC voltage to the PA.

Multi-path QSB (flutter) will cause serious difficulty for PSK31 mode! Switch to MFSK16 or Feld Hell modes to eliminate this problem.

Use the other HF bands to find digi contacts when conditions permit. Try 7.0710 Mhz on 40 meters, 10.140 on 30 meters and 3581 on 80 meters in North America.

Use BPSK mode for calling and all normal PSK31 operation. Try QPSK mode if the print quality drops to 80% due to static or noisy band conditions.

Using lower case letters instead of all caps, will increase PSK31 speed and reduce TX time.

Space QSOs at least 100hz apart and help demonstrate the efficiency of these new low power narrow band digi modes.

*Have Fun!*

Finally a brief description of the various modes.

**TOR** is an acronym for Teleprinting Over Radio. It is traditionally used to describe the three popular "error free" operating modes, AMTOR, PACTOR and G-TOR. The main method for error correction is from a technique called ARQ (automatic repeat request) which is sent by the receiving station to verify any missed data. Since they share the same method of transmission (FSK), they can be economically provided together in one TNC modem and easily operated with any modern radio transceiver. TOR methods that do not use the ARQ hand-shake can be easily operated with readily available software programs for personal computers. For these less complex modes, the TNC (terminal node controller) is replaced by an on-board sound card or out-board audio device. These modes may use redundancy or "human processing" to achieve a level of error correction.

**AMTOR** is an FSK mode that has been fading into history. While a robust mode, it only has 5 bits (as did its predecessor RTTY) and can not transfer extended ASCII or any binary data. With a set operating rate of 100 baud, it does not effectively compete with the speed and error correction of more modern ARQ modes. The non-ARQ version of this mode is known as FEC, and known as SITOR-B by the Marine Information services.

**PACTOR** is an FSK mode and is a standard on modern TNCs. It is designed with a combination of packet and Amtor Techniques. It is the most popular ARQ digital mode on amateur HF today. This mode is a major advancement over AMTOR, with its 200 baud operating rate, Huffman compression technique and true binary data transfer capability.
**PACTOR-II** is a robust and powerful PSK mode which operates well under varying conditions. It uses strong logic, automatic frequency tracking; it is DSP based and as much as 8 times faster then Pactor. Both PACTOR and PACTOR-2 use the same protocol handshake, making the modes compatible.

**PACTOR-III** is a proprietary mode used for message and traffic handling over an HF radio circuit. Use of Pactor-III protocol is limited for US hams and some other countries due to the very wide bandwidth of the Pactor-III signal. Presently digital signals that occupy the bandwidth of PCT-III are restricted to a few sub bands: 28.120-28.189 MHz, 24.925-24.930 MHz, 21.090-21.100 MHz, 18.105-18.110 MHz, 14.0950-14.0995 MHz, 14.1005-14.112 MHz, 10.140-10.150 MHz, 7.100-7.105 MHz, or 3.620-3.635 MHz.

Only the embedded hardware (modem) from the German company that owns the rights to this mode, is capable of operating Pactor-III.

**G-TOR** (Golay -TOR) is an FSK mode that offers a fast transfer rate compared to Pactor. It incorporates a data inter-leaving system that assists in minimizing the effects of atmospheric noise and has the ability to fix garbled data. G-tor tries to perform all transmissions at 300 baud but drops to 200 baud if difficulties are encountered and finally to 100 baud. (The protocol that brought back those good photos of Saturn and Jupiter from the Voyager space shots was devised by M.Golay and now adapted for ham radio use.) G-tor is found in only one manufacture's TNC and is rarely used today.

**CLOVER** is a PSK mode which provides a full duplex simulation. It is well suited for HF operation (especially under good conditions), however, there are differences between CLOVER modems. The original modem was named CLOVER-I, the latest DSP based modem is named CLOVER-II. Clovers key characteristics are band-width efficiency with high error-corrected data rates. Clover adapts to conditions by constantly monitoring the received signal. Based on this monitoring, Clover determines the best modulation scheme to use.

**RTTY** or "Radio Teletype" is an FSK mode that has been in use longer than any other digital mode (except for morse code). RTTY is a very simple technique which uses a five-bit code to represent all the letters of the alphabet, the numbers, some punctuation and some control characters. At 45 baud (typically) each bit is 1/45.45 seconds long, or 22 ms and corresponds to a typing speed of 60 WPM. There is no error correction provided in RTTY; noise and interference can have a seriously detrimental effect. Despite it's relative disadvantages, RTTY is still popular with die-hard operators.

**PSK31** is the first new digital mode to find popularity on HF bands in many years. It combines the advantages of a simple variable length text code with a narrow bandwidth phase-shift keying (PSK) signal using DSP techniques. This mode is designed for "real time" keyboard operation and at a 31 baud rate is only fast enough to keep up with the typical amateur typist. PSK31 enjoys great popularity on the HF bands today and is presently the standard for live keyboard communications. Most of the ASCII characters are supported. A second version having four (quad) phase shifts (QPSK) is available that provides Forward Error Correction (FEC) at the cost of reduced Signal to Noise ratio.

**HF PACKET** radio is a FSK mode that is an adaption of the very popular Packet radio used on VHF FM ham radio. Although the HF version of Packet Radio has a much reduced bandwidth due to the noise levels associated with HF operation, it maintains the same protocols and ability to "node" many stations on one frequency. Even with the reduced bandwidth (300 baud rate), this mode is unreliable for general HF ham communications and is mainly used to pass routine traffic and data between areas where VHF repeaters maybe lacking.

**HELLSCHREIBER** is a method of sending and receiving text using facsimile technology. This mode has been around along time; the recent use of PC sound cards as DSP units has increased the interest in Hellschreiber. The single-tone version (Feld-Hell) is the method of choice for HF operation. It is an on-off keyed system with 122.5 dots/second, or about a 35 WPM text rate, with a narrow bandwidth (about 75 Hz). Text characters are "painted" on the screen, as apposed to being decoded and printed. A new "designer" flavor of this mode called FM HELL has some advantage for providing better quality print, at the expense of a greater duty cycle. As with other "fuzzy modes" it has the advantage of using the "human processor" for error correction.

**MT63** is a new DSP based mode for sending 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 100 WPM rate. The wide bandwidth (1Khz for the standard method) makes this mode less desirable on crowded ham bands such as 20 meters. A fast PC (166 Mhz or faster) is needed to use all functions of this mode.

**THROB** is yet another new DSP sound card mode that attempts to use Fast Fourier Transform technology (as used by waterfall displays) to decode a 5 tone signal. The THROB program is 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 but the author (G3PPT) has been improving his MFSK (Multiple Frequency Shift Keying) program. Check his web site for the latest developments.

**MFSK16** is an advancement to the THROB mode and encodes 16 tones. The PC sound card for DSP uses Fast Fourier Transform technology to decode the ASCII characters, and Constant Phase Frequency Shift Keying to send the coded signal. Continuous Forward Error Correction (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 (316 Hz) for this mode allows faster baud rates (typing is about 42 WPM) and greater immunity to multi path phase shift. This mode is becoming a standard for reliable keyboard to keyboard operation and is available in several popular programs.

**NOTES:**

Frequency-shift keying (FSK) shifts between two known states. Phase-shift keying (PSK) changes PHASE of a signal against some reference. FSK is sent by either shifting a carrier frequency (F1B) or modulating SSB with two shifting audio tones (AFSK). When sending PSK, a complex audio waveform is transmitted by SSB. Tracking is much more critical for PSK, thus requiring more frequency stability.

**DSP** (Digital Signal Processing) techniques use high speed processing to convert audio into digital coding, so that a program can manipulate the coded audio in ways not possible with traditional hardware filters. The 16 and 32 bit sound cards found in modern PCs provide this capability.

**FUZZY MODES** are those modes that allow the human eye/ear/brain to be used to its maximum potential. In order to do this, a number of rules are required, to ensure that any electronics or logic circuitry is not allowed to make decisions which may be less inspired than human decisions. Examples of potentially Fuzzy modes are Morse Code, HFFAX, SSTV and Hellschreiber. The rules are:

The transmissions must be uncoded. (The signal is sent as a real-time language.)
The receiver must not decide when data is present. (Untouched by any prior decisions.)
The receiver must not decide what data is present. (It must be presented as received

By Chris Cunliffe G7OOD

# BLAST FROM THE PAST

Former club member Derek Nicholls G0GTC, is now back permanently in UK (since April 2007) after making a move to Spain and resident at Dobcross. Derek has been in touch and says he hopes to pay us a visit some time soon.

In the meantime he recently received a letter from an old English friend in Spain, John Cubitt, an ex diplomat and wartime GCHQ veteran. He thought it might make interesting reading for OHN. He wrote:

*"I often have a look on 20 and 40 metres...and in the course of doing so have heard some unusual transmissions. And I was wondering if any of your ham friends could put me wise to a transmission, hopping about from one to another, of 25 different frequencies.*

*In all my experience, I have never heard "a group" like this one.*

*He sends 17 words a minute, auto five-letter groups, changing frequency sometimes nine times a day - even in the middle of a five-letter group - never sends a callsign or any procedure, sends the break sign not as BT but a separate "B" "T".*

*He sends messages of 100 five-letter groups, with a preamble like this: NR21 A30 19-45-07 1984 B T, and straight into high-grade five-letter cypher, then B T and on to NR 22 .*

*(The A is for August, with a date/time separated by a hyphen. But why in seconds before the 1984?)*

*In all my years of receiving messages in all languages, from all services and countries, I have never come across anything like it. I can usually pinpoint a transmission, down to country or service, but not this boy!*

*He is easy to pick up, because he never stops and to my knowledge goes on for over 14 hours a day, sending blind.*

*I call him "Pimpernel", as it is quite a game finding him between 7 and 10.5 MHz. Surely he is illegal, as you are supposed, in international law, to use a callsign.*

*Here are a few of his frequencies: 7400, 7500, 7550, 7620, 7700, 7800, 7880, 7995, 9140, 9300, 10640.*

*It is beautiful, accurate 17 words a minute morse - just what you need for more practice on the key!*

*Maybe you can help solve this mystery for me."*

Derek adds that John is now over 80 and still fit, active and a dedicated cw listener!

Best wishes to all at the club; hope to resume interest when settled here in Dobcross, where we moved in only recently.

# FORTHCOMING EVENTS

The following events are in the club diary for the next 3 months. Obviously these are subject to alteration, cancellation and augmentation so please keep an ear tuned to the announcements on club nights and an eye on our web site for the latest information.

Thursday 3rd April – Foundation Training Course begins.

Saturday 26th April – ARDF Event, at Witton Park, Blackburn.
All members are invited to either take part or support this event.

Thursday 8th May – Talk “An Introduction to Making a Home Video”
by Geoff Oliver G0BJR.

# SILENT KEY

It is with regret that we announce the passing of club member Graham Wood. Graham had not been a club member for very long and he was booked in to take our Foundation Training Course. We extend our sincere condolences to Graham’s family.

# OLD HAMS NEWS

The editor would like to thank all contributors to this edition of Old Hams News. The next edition is due to be issued in March 2008. Any contributions for this edition should be forwarded to the editor, Geoff Oliver G0BJR on or before Thursday 19$^{th}$ June 2008 to ensure inclusion.

Articles will be accepted on many formats, by email to "news@oarc.org.uk", by word of mouth, hand or type written notes, or as a .txt file on a CD ROM or 3½-inch floppy disc. Photographs, drawings, circuit diagrams and other graphics to enhance your article will also be most welcome.

An edited version of Old Hams News is available on the Internet at the following URL
www.oarc.org.uk
then click on the "Club Journal" hyperlink.

If you submit an article for inclusion in Old Hams News and you do not wish it to be included in the Internet edition you must state your wishes at the time of submission. Otherwise the editor reserves the right to include/exclude your article as he sees fit.

For reference the officers and committee members are listed below and will be happy to help with any enquiries you may have.

President – Geoff Oliver G0BJR
president@oarc.org.uk

Chairman – Ian Moth G8ZHC
chairman@oarc.org.uk

Secretary – Chris Cunliffe G7OOD
secretary@oarc.org.uk

Treasurer – Peter Rushton G7PMZ
treasurer@oarc.org.uk

Alan Burgess G4GLV
Phil Ellis M0GIE
Stuart Wilson G7MFK
Ian Firby G7VCG
Sue Burgess G0RKE

N.B. Old Hams News is the official journal of the Oldham Amateur Radio Club. It is distributed free of charge to all fully paid up members of the club. Articles appearing herein do not necessarily reflect the views of the editor, the Officers and Committee, or the membership of Oldham Amateur Radio Club. Whilst every effort is made to ensure the accuracy of articles included, the editor is not responsible for any inaccuracy that may occur.

Written, edited and produced by Geoff Oliver G0BJR