THE WEST RAND AMATEUR RADIO CLUB

September 2002
Volume 3, Issue 2

ANODE

Inside this issue:

Special points of interest:

- Contact details on back page

Editor's Comments

This last week we have seen the onset of spring with the rain. Some have seen lightning damage their equipment in previous years and unplugged the antennae as a precaution. I hope you have....

The well attended Boot Sale went off without any rain dampening spirits.

The article below was inspired by the talk we had a the last meeting on PSK31. It was interesting to hear about a new bandwidth conservative mode of operation. This is really Amateur Radio communication using methods theorised 50+ years ago and now practical thanks to the availability of powerful personal computers. The software can be run on a 486 with a sound card.

It is possible .....

- To make Windows 95 look like the latest version, XP. Just by downloading a theme from an Internet site. You can also make the 'wallpaper' into a TellyTubbies scene!
- To use the previous years back page in the latest Anode. Moral: don't use last years template for this year!

FYI

The RSGB's GB2RS weekly news bulletin is

(Continued on page 3)

PSK31 Fundamentals

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

(Continued on page 2)

PSK31 Fundamentals

(Continued from page 1)
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 baud rates 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 key clicks 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 baud rate wide at 10dB down, 5 times at 14dB down, 7 times at 17dB down, and so on (the square wave 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

(Continued on page 5)

A reminder of the dangers of recycling code.

Check out this Item taken from Software Testing and Quality Engineering magazine, Volume 1, Issue 6 (November/ December 1999)

“Careless code recycling cause killer kangas -- Mutant Marsupials Take Up Arms Against Australian Air Force”

The reuse of some object-oriented code had caused tactical headaches for Australia's armed forces. As virtual reality simulators assume larger roles in helicopter combat training, programmers have gone to great lengths to increase the realism of their scenarios, including detailed landscapes and -- in the case of the Northern Territory's Operation Phoenix -- herds of kangaroos (since disturbed animals might well give away a helicopter's position).

The head of the Defence Science & Technology Organization's Land Operations/ Simulation division reportedly instructed developers to model the local marsupials' movements and reactions to helicopters. Being efficient programmers, they just reappropriated some code originally used to model infantry detachment reactions under the same stimuli, changed the mapped icon from a soldier to a kangaroo, and increased the figures' speed of movement.

Eager to demonstrate their flying skills for some visiting American pilots, the hotshot Aussies "buzzed" the virtual kangaroos in low flight during a simulation. The kangaroos scattered, as predicted, and the visiting Americans nodded appreciatively..... then did a double-take as the kangaroos reappeared from behind a hill and launched a barrage of Stinger missiles at the helpless helicopter. (Apparently the programmers had forgotten to remove THAT part of the infan-

(Continued on page 3)

Editors Comments

(Continued from page 1)

now sent out via a yahoo group mailing list. This will be helpful to those who like to find the news in their inbox rather than go looking for it on the web. Unlike other national radio societies, the RSGB refuse to provide this service for non-members.

The news will be posted over the weekend, after it has been released for non-members.

Subscribe at- http://groups.yahoo.com/group/GB2RSnews/

or simply send an empty email to

GB2RSnews-subscribe@yahoogroups.com

Other weekly amateur radio news provided via yahoo mailing lists which are worth reading can be found at-

http://groups.yahoo.com/group/arnewsline/

http://groups.yahoo.com/group/apcnews/

http://groups.yahoo.com/group/QNEWS-VK-TEXT/

The first is American, and the last two Australian, but they make interesting reading regardless of location.

A reminder of the dangers of recycling code.

(Continued from page 2)

try coding.)

The lesson?

Objects are defined with certain attributes, and any new object defined in terms of an old one inherits all the attributes. The embarrassed programmers had learned to be careful when reusing object-oriented code, and the Yanks left with a new-found respect for Australian wildlife.

Simulator supervisors report that pilots from that point onward have strictly avoided kangaroos, just as they were meant to.

K1TTT Technical Reference

Question:

Over the past several years, I have tried several different PC compatible computers in the shack in an attempt to use with my HF Station. It seems no matter with PC I have tried, and having tried toroid chokes, ac filters, etc. my PC (now a 486 /66) still QRM's the heck out of the ham hams with birdies and other annoying noise.

I know my PC is not a CLass B machine, just a clone. But over the years, I have had several clones, real IBM's, Compaqs, and AST's all with basically the same result.

Does anyone out there know of a manufacturer of a PC tower or desktop cabinet that has excellent RFI suppression and shielding? If so, would be very interested in finding this out.

Short of spending megabuck for a new class B machine, is there any other solution? My pc and my hf station are totally useless together. What does all you guys do?

A

It may not be your computer. At least not the computer box itself.

One handy tool for finding rf leakage is a 10' piece of RG58 as an antenna for the main RX, with 3" of the shield pulled back on the 'working end', with heat shrink over the exposed shield and center conductor. Locate a strong birdie with the case open, then close up the case and use this rf probe to find the leaky holes.

The first thing you need to do is to unplug everything but the power cord from the computer, and turn it on. Look for birdies on the bands. I'll bet that they are greatly reduced.

In short, the first step is to make the PC box itself quiet with nothing plugged into it but the power cord... If you have birdies, look inside the power supply - some have Corcom filters on the AC input, some don't. You may have to suppress the PS before anything else. I've seen the exact replacement RF filter power assemblies at the swap meets for $5. The common name is a "Corcom" - the name of the most common manufacturer. Once the power cord is quiet, a piece of copper screening (from the local hobby store) between the fan body and the metal box may be required. The back side of the faceplates for the empty drive bays may have to be covered with copper screening, just as any other non-shielded RF windows. Then hook up the keyboard. If the trash level in the RX goes up, work on what was just changed: the keyboard connection. It might take pulling RG-8 braid over the keyboard cable, and grounding it on each end. It might even take removing the keyboard connector from the motherboard, and replacing it with a DB-9 (like a serial plug) mounted on the cabinet with suppression networks between it's pins and the motherboard. Then cut the DIN plug off the keyboard cable and put the other half of the DB-9 set on it - using good quality metal hoods, of course. If you had to change the connectors, don't forget to cover the keyboard connector hole). Once you can't tell the RF difference between the PC by itself and the PC-keyboard combination, you can start on the monitor. Do the same thing here - it might take shielding the inside of the cabinet with grounded foil and insulating fish paper over that. One article I read years ago about a monochrome monitor ended up with replacing the video wire between the picture tube socket and the circuit board with RG-174. When the PC-kbd-monitor set is quiet, then plug in the printer. It might take a shielded cable here.

On mine, I ended up making one with shielded wire and metal hoods on good quality connectors. I also found that of the half-dozen "ground" pins on the DB-25 parallel socket on the I/O card, only 3 were actually tied to ground. A jumper on the card fixed that. Then test the system with both the printer power on & off - my friend's laser printer is quieter than his old dot-matrix (the laser has more metal in the case).

Years ago you could buy

(Continued on page 6)

PSK31 Fundamentals

(Continued from page 2)

products of 25dB at +/-47Hz (3 times the baud rate wide) and fifth-order products of 35dB at +/-78Hz (5 times the baud rate 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 side lobes 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 baud rate 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 inter-symbol 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 band-

(Continued on page 6)

PSK31 Fundamentals

(Continued from page 5)

width 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

(Continued on page 7)

K1TTT Technical Reference

(Continued from page 4)

"aquadag" metallic coating in a spray can - it makes adequate shielding if you can ground it. Shielding the inside of the printer case may be a bit difficult with the flip-top aver the print head area... As to modems and TNCs, the route of least effort is to go internal to the PC case. Unfortunately, external TNCs are more common than internal. However for RF & shielding purposes, TNCs and modems can be treated similarly. My first modem radiated badly - it was a Prometheus with a plastic case. My second (a Hayes 2400 external) was much better (it had a metal case but perhaps that was a coincidence?).

My current modem is internal - one less cable/antenna. I took the RJ-11 off the card and replaced it with a shielded RJ-11 with built in RF filters on the pins - AMP makes it. I changed it before I even installed the card in the computer - so I have no idea as to the performance before the mod.

Mice cables can be replaced with shielded wire - but again, before you do anything, try it with the mouse plugged in and unplugged - you might discover that there is no difference. RF chokes and bypass caps on the plug may be enough.

The above is a potentially a lot of work. But it is worth it. My 386 is audible on my scanner on 6m AM from 50 feet away (and my 2m ssb rig will hear it from further) with the case open, but with the case closed there are a couple of minor birdies. If I had shielded the back side of the

PSK31 Fundamentals

(Continued from page 6)

correcting errors within itself. These extended blocks are then transmitted as a serial bit-stream. 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 data-stream 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

(Continued on page 8)

PSK31 Fundamentals

(Continued from page 7)

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 occurrence 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 occurrence-frequency order was matched to the list of binary patterns, in length order, so that the most frequently-occurring 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.

NUL	1010101011
SOH	1011011011
STX	1011101101
ETX	1101110111
EOT	1011101011
ENQ	1101011111
ACK	1011101111
BEL	1011111101
BS	1011111111
HT	11101111
LF	11101
VT	1101101111
FF	1011011101
CR	11111
SO	1101110101
SI	1110101011
DLE	1011110111
DC1	1011110101
DC2	1110101101
DC3	1110101111
DC4	1101011011
NAK	1101101011
SYN	1101101101
ETB	1101010111
CAN	1101111011
EM	1101111101
SUB	1110110111
ESC	1101010101
FS	1101011101
GS	1110111011
RS	1011111011
US	1101111111
SP	1
!	111111111
"	101011111
#	111110101
$	111011011
%	1011010101
&	1010111011
'	101111111
(	11111011
)	11110111
*	101101111
+	111011111
,	1110101
-	110101
.	1010111
/	110101111

(Continued on page 9)

PSK31 Fundamentals

(Continued from page 8)

0 10110111
1 10111101
2 11101101
3 11111111
4 101110111
5 101011011
6 101101011
7 110101101
8 110101011
9 110110111
: 11110101
; 110111101
< 111101101
= 1010101
> 111010111
? 1010101111
@ 1010111101
A 1111101
B 11101011
C 10101101
D 10110101
E 1110111
F 11011011
G 11111101
H 101010101
I 1111111
J 111111101
K 101111101
L 11010111
M 10111011
N 11011101
O 10101011
P 11010101
Q 111011101
R 10101111
S 1101111
T 1101101
U 101010111
V 110110101
X 101011101
Y 101110101
Z 101111011
[1010101101
\ 111110111
] 111101111
^ 111111011
_ 1010111111
. 101101101
/ 1011011111
a 1011
b 1011111
c 101111
d 101101
e 11
f 111101
g 1011011
h 101011
i 1101
j 111101011
k 10111111
l 11011
m 111011
n 1111
o 111
p 111111
q 110111111
r 10101
s 10111
t 101
u 110111
v 1111011
w 1101011
x 11011111
y 1011101
z 111010101
{ 1010110111
| 110111011
} 1010110101
~ 1011010111
DEL 1110110101

Contact Information

The source code which you may have with this distribution 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 me 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

K1TTT Technical Reference

(Continued from page 6)

mini-panel that has the key lock, turbo switch and reset button those would have probably been eliminated. If I ever rebuild the system, I'm probably going to replace that plastic piece with a sheet metal piece with a metal (shielded) push button for the reset, leave out the turbo switch and lamp, and put mini-rf chokes on the disk activity leds.

After looking at what I have written, perhaps this would make a good QST article! "The care and starving of a birdie".

Mike Morris WA6ILQ | PO Box 1130 | Arcadia, CA. 91077 | All opinions must be my own since nobody pays me enough to be their mouthpiece...

David Robbins, K1TTT
k1ttt@arrl.net

The West Rand Amateur Radio Club
26.14122 South - 27.91870 East

P.O. Box 562
Roodepoort
1725

Phone: +27 11 726 6892
Email: john.brock@pixie.co.za

Bulletins (Sundays at …)
11h15 Start call in of stations
11h30 Main bulletin start

Frequencies
145,625 MHz (West Rand Repeater)
10,135 MHz (HF Relay)

Radio Amateurs do it with more frequency!

We are going on the Web
www.zs6wr.org.za

Please note this has been just been registered. Our site will be up in the new year.

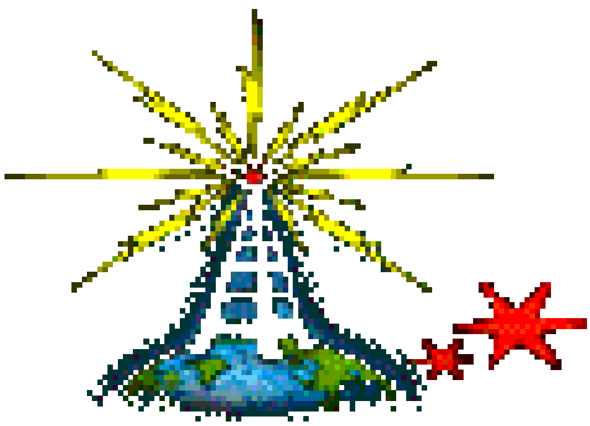
Chairman/Treasurer	Dave	ZR6AOC	475 0566 (H)	zr6aoc@mweb.co.za
Vice-Chairman/Events	Simon	ZR6SS	084 308 2665	ssnyman@securehome.co.za
Secretary	John	ZS6FJ	672 4359 (A/H)	
Technical	Phillip	ZS6PVT	083 267 3835	
Technical	Greg	ZR6JDD	083 289 2072	gjarrett@webb.co.za
Member	Craig	ZR6CRW	795 1550 (H)	craig.woods@absamail.co.za

West Rand members input - we need your input!

To make this the best ham radio magazine in South Africa we need your input. Please submit articles, comments, suggestions etc.

Please send plain text with no formatting to the email address below.

In November 2001, we published an Anode Compendium on CD. It has the issues from July 2000 until November this year. This included IE5.5 and the new Adobe reader. It is soon to be updated, check with the vice-chairman for details.



We need your input! Email us articles, comments and suggestions please.
john.brock@pixie.co.za