**Software DSP Solutions for Weak Signal Communications**

**--by Roger Rehr, W3SZ**

This article was originally written in HTML. Some of the links will not be available if you are reading the printed version of it. If you are using the HTML version, the links below will take you to the appropriate part of the page for each subject. Color versions of the illustrations are available on the CD-ROM that was given out at the Conference. Figures are presented at the end of the article.



## I. DSP for Weak Signal Communications: A Brief Overview

DSP is an acronym for "Digital Signal Processing". Anytime an analog signal is converted to digital form and processed in some way, that is DSP. The use of DSP techniques in amateur radio is exploding, both in terms of its use in commercial transceivers and receivers, and in terms of homebrew hardware and software construction projects available to and undertaken by hams. The ARRL Handbook since at least its 2000 edition has had an excellent basic introductory chapter on DSP. ARRL publications such as QST and especially QEX have featured excellent articles on the subject during the past 2 years. A wealth of information is available on the Internet. A convenient starting point for exploration might be one of my web pages, with URL http://www.qsl.net/w3sz/start.htm . This article will discuss some of the DSP software that is available for enhancing the reception and decoding of RF signals. In this article I will discuss some of the software or software/hardware solutions that are now in use by hams who need to pull those really weak signals out of the mud, or to separate one weak signal from many other signals crowding around it.

I began using DSP techniques because of my interest in doing 144 MHz EME in a noisy RF environment. EME is truly "weak signal" communications. The "typical" round-trip path loss when the moon is at perigee (closest to the earth) is approximately 251.5 dB at 144 MHz. If you consider a system where maximum legal power is present at the antenna, the system starts with 31.76 dBW transmit power. If the antenna array has 19 dB gain, then the signal leaving the antenna will be 51 dBW. The signal arriving back from the moon at the receiving antenna will be on the order of -200 dBW. If the receiving antenna also has 19 dB gain, the signal arriving at the preamplifier on the mast will be -181 dBW. If the antenna has a noise temperature of 200 K, the preamplifier has a noise figure of 0.5 dB, the subsequent 144 to 28 MHz transverter a noise figure of 1 dB, and each has a gain of 20 dB then the receive system will have a noise floor of -

187 dBW if a bandwidth of 250 Hz is used. (As long as the 28 MHz IF is reasonably state of the art, its noise figure is irrelevant as it is divided by the product of the gains of the preamplifier and the transverter when figuring its equivalent noise temperature). Thus the receive system will detect the signal as (-181+187) or 6 dB above the noise. Throw in a couple of dB for cable loss, and you may be 2-4 dB above the noise. Compare this with a typical 1 watt HT at 10 kM with a 0 dB gain antenna, where the received signal would nevertheless be 60 dB OVER the noise (or roughly a million times stronger)! An astronaut using a 1 watt HT at 1000 kM from you, who would still be 20 dB above the noise! Thus EME IS truly weak signal work! [To convert dBW to dBm, just add 30; i.e. the noise floor of -187 dBW becomes -157 dBm]

The role of DSP techniques in EME and other weak signal work is of course to provide substantial improvement in signal reception and decoding (interpretation). There are two approaches to using DSP techniques to increase the success potential of 144 MHz EME efforts. The first and more obvious approach is to use DSP techniques to improve the human, aural detection of CW EME signals. There has been much work in this arena over many years. The second approach is to use DSP methods to provide for automated message detection and decoding of signals that may not even be audible with standard audio processing techniques. These methods have only very recently become widely available to amateur radio operators, and are exemplified by the modes PUA43 developed by Bob Larkin, W7PUA, and JT44, created by Joe Taylor, K1JT. Successfully using either approach for weak signal VHF/UHF/Microwave work requires considerable skill on the part of the operator. BOTH forms of communication have been accepted by the ARRL as meeting the requirements for their Awards Programs (Reference: Personal Communication to W3SZ, by email, Spring 2002). Thus which technique to use for weak signal communications is a matter of personal preference for each operator. Like other experienced EME operators, I have found that programs such as PUA43 and JT44, both examples of the computer decoding paradigm, could at times receive complete and accurate information when I could not hear the other station, and so at least under some circumstances, the human interface represents a weak link when compared with automated decoding by the computer.

When one is using DSP techniques to improve the accuracy of human decoding of the message, there are several features that we would like to have in our “ideal” DSP program.

Specifically, the ideal program should provide:

1. A waterfall display with adjustments possible for color gain, baseline level, visualized bandwidth, frequency bin size, and number of averages per displayed line. A waterfall display is basically a way of displaying the time course of signals that have been received by having one axis (usually the horizontal) represent frequency, the second axis (usually vertical) represent time, and then using color to display signal strength. A properly designed waterfall used in the correct way will allow one to visually detect signals that are considerably below the audible threshold. This is possible by virtue of both signal averaging and by the use of very narrow frequency bins, both of which increase signal-to-noise ratio. Signal averaging increases the signal-to-noise ratio by the square root of ‘n’, where ‘n’ is the number of signals averaged. This means that averaging two signals increases the signal-to-noise ratio by the square root of 2, or 1.414. Expressed in dB, this would be an improvement of 1.5 dB. Narrowing the bin frequency range increases the signal-to-noise ratio by ‘n’ where ‘n’ is the fractional bandwidth reduction.

For example, decreasing the bandwidth to ½ of its previous width doubles the signal-to-noise ratio, or increases it by 3 db, all other things being equal.

An example of an excellent waterfall display is shown in figure 1. This illustration is a screen grab from Linrad which displays here a 16 kHz portion of the 2 meter band as received by me during the ARRL 2001 EME Contest. You can see among the many birdies I have at my location, at least 12 vertical dashed lines; each one of these is an EME station's signal Although here it is reproduced in black and white, the display looks much better in color as can be seen on the CD-ROM or on my website.

2. A spectral display with the following parameters being adjustable: vertical gain, baseline level, visualized frequency range, frequency bin size, and number of averages per displayed spectrum. A spectrum is just the familiar plot of signal intensity vs frequency for a single point in time. A spectrum is shown just below the waterfall display in the Linrad image of figure 1.

3. DSP audio processing with
   a. variable bandwidth filtering with adjustable center frequency
   b. adjustable LMS (Least Mean Square) or other noise reduction algorithm
   c. binaural receive capability
   d. defeatable and adjustable AGC (if AGC is used)
   e. adjustable notch filtering designed so that it is useful when in CW mode.

The bandwidth filters that can be created with DSP have the advantages of (1) being immune to the problem of aging-induced changes in component values producing altered filter parameters with time, (2) being very flexible (i.e. easily altered by the user as requirements change), and (3) the fact that they can be designed to much more stringent specifications than is generally practical with analog components. They very much lend themselves to experimentation, as trying a different configuration often just involves just changing a parameter value in software and recompiling the program rather than substituting hardware components. To my ear the DSP filters I have used have had less ringing than analog hardware filters of the same selectivity.

LMS (Least Mean Square) noise reduction is a statistical method of using the signals received in the recent past to estimate the noise contribution in the total received signal, and then using those estimates to remove the noise so that only the desired signal remains. This technique achieves this by minimizing the mean squared error of the data set from the mean.

Binaural receiving methods delay the arrival of part or all of the signal going to one ear. This 'pseudo-stereo' sometimes makes the desired signal seem to pop out of the background. Linrad, DSP-Blaster, and the DSP-10 all have binaural receive capability available. Each processes a single incoming signal so that the desired signal, based on frequency and phase, seems to pop out of the background noise when the binaural mode is activated. Linrad goes further and actually permits the simultaneous reception of two separate receive channels, thus allowing reception of two signals simultaneously (for example vertical and horizontally polarized signals) and using computer/DSP techniques to linearly combine the signals using an orthonormal transformation, resulting in two new signals. One of these new signals has a polarization that matches the incoming wave, while the other is orthogonal and contains 'no signal at all'. The Linrad software allows one to just listen to the polarization-angle-matched component of the signal, resulting in an improvement in signal-to-noise ratio. In fact, with Linrad one can use this automatic polarization matching and also have binaural reception of the remaining polarization-angle-matched signal.

Digital notch filters can be made much sharper and deeper than analog notches. The ideal notch would, once assigned to an offending carrier, follow that carrier as it moved in frequency so that the carrier would stay suppressed if it drifted, or if the receiver was tuned or drifted. DSP-Blaster's Notch has such Automatic Frequency Control (AFC). For weak signal CW use, the notch filter must be able to be controlled so as not to eliminate the desired CW carrier. This can be done after a fashion with DSP-Blaster's notch, but not with the notch in DSP-10, so the DSP-10 notch is not useful in CW mode. Linrad will remove many spurs, by pointing and clicking on each of them with the mouse. But as a matter of practicality with Linrad, run with a 20 Hz filter (as I generally use it), there is only one signal in the audio pass band and no need for a notch filter.

When the final link in the receive chain is not human hearing and interpretation but computer analysis, the list of desired software characteristics boils down to three items: user friendliness, accuracy of the final result, and efficiency (speed) of achieving the correct solution.

## II. Available DSP Packages: A Brief Summary of Some First Generation Packages

Each of the DSP Solutions I will discuss takes an analog signal at a relatively low (audio or near audio) frequency, converts it to digital form using an Analog-to-Digital converter (ADC), and processes it digitally. The programs using the human ear and brain for final signal reception and decoding then convert the digital signal back to analog form using a Digital-to-Analog converter (DAC) and send it to speaker or headphones for human reception and processing. The systems using automated signal decoding may omit this final step. Programs that are primarily used with human signal detection as the final common pathway include an older MS-DOS-based program named DSP-Blaster; the Windows-based programs Spectran, ChromaSound, and GNASP1; and Linrad, which runs on Linux-based systems. DSP-10 can be used for either human or automated decoding and runs only in true DOS mode. An audio-only version permits use of DSP functions with human audio detection and analysis as the final stage. WSJT, which includes JT44 as one of two subprograms, is a Windows-based program designed for automated signal interpretation. In this article we will limit our discussion of WSJT to JT44. An additional DOS-based program, FFTDSP, offers no audio processing but will provide a waterfall display. It is not discussed further here.

DSP-Blaster works ONLY with ISA bus Creative Labs SoundBlaster soundcards. Linrad, Leif SM5BSZ's Linux PC-Based Receiver, can use a variety of soundcards. I use an M-Audio Delta44 for ADC and processing and a Soundblaster PCI64 for DAC and output with Leif's program. The Delta44 permits sampling rates of up to 96 kHz and thus with quadrature (I/Q) detection will permit bandwidths (and thus waterfall display widths) of up to nearly 96 kHz, a major advantage. The DSP-10 uses an Analog Devices EZ-Kit Lite for A/D and D/A conversions and processing. WSJT will work with most Windows-compatible soundcards.

DSP-Blaster was written by K6STI, Brian Beezley, who updated it to DSP-Blaster 2.56 in 2001. He is no longer providing updates. This program has good LMS noise reduction, narrow filters with adjustable bandwidth and center frequency, a beautiful notch filter, synthetic stereo, and much more. It requires a 486 or better with a math coprocessor, at least VGA video, a mouse, a

16 bit Creative Labs ISA sound card, and DOS 3.0 or later. It must run in DOS mode. You can run it from Windows 95 or 98, but not as a window; it must run in full screen mode. The program has three frequency-domain display modes: waterfall, single spectrum, and 3-D spectral history (stacked plots). Figure 2 shows a typical DSP-Blaster screen. There are two time domain modes: waveform and envelope. It has several processing functions that are useful for CW operation: a band pass filter, a peaking filter, LMS noise reduction, and a coherent band pass filter. The coherent band pass filter produces binaural (stereo) output. The latest version 2.56 was available for $75, or as an upgrade from DSP-Blaster 2.0, $50 (Cash, U.S. check, or money order, E-mail delivery), as of mid 2001 from:

Brian Beezley, K6STI
3532 Linda Vista Dr.
San Marcos, CA 92069
USA.

http://seti1.setileague.org/software/dspblast.htm gives some additional info. Brian's email address as of mid 2001 was k6sti@n2.net. I would recommend checking with him to see if the program is still available before sending him funds. More comments on DSP-Blaster are available at http://www.nitehawk.com/rasmit/dsp_soft.html . This is a very good program. Its main limitation is that it works only with genuine ISA bus Creative Labs Sound-Blaster Cards, and computers with ISA slots are becoming harder to find. It can be quite finicky in terms of not wanting to run unless your computer is set up 'just right' But when it works, its does a very good job and I find I can pull out the really weak EME signals with it very well, better than with any other program save Linrad.

ChromaSound (http://www.siliconpixels.com), by N7CXI and VE3EC provides graphic filter control and spectrum display over a bandwidth of 5000 Hz or so, along with DSP noise reduction but no waterfall. I don't think it does as good a job as Linrad, DSP-Blaster, DSP-10, or Spectran in terms of signal processing, and mention it here just for completeness. I don't use it at all as the other programs mentioned do a much better job for me.

GNASP1 provides selectable filtering and spectrum output but no waterfall, noise reduction, or notching. See http://members.tripod.com/~gniephaus/gnasp1/gnasp1.html . I've not used this at all recently as the filters didn't seem to be anything special and its lack of a noise reduction algorithm is a real negative here at W3SZ.

Spectran (by I2PHD and IK2CZL) replaces their older DOS-based program, Hamview. Spectran provides a waterfall display, a spectral display, mouse-adjustable filters, and an LMS type noise reduction algorithm. The filters and noise reduction algorithm in Spectran do not work as well for me as those in Linrad, DSP-Blaster and the DSP-10. The waterfall display is very good, but not as good as Linrad's. Links for Spectran are at http://www.qsl.net/padan/spectran.html . I don't use this because the filters as noted just aren't quite as good as I need and Linrad, the DSP-10, and DSP-Blaster work better for me.

The programs just discussed in some detail represent the simpler, "first generation" DSP programs that exist. The only one of these that I still use, and that just occasionally, is DSP-Blaster. Now we will move on to more sophisticated, current generation programs such as Linrad, DSP-Blaster, and JT44.

For a superb web page that has a link to an excellent DSP Resource page go to http://www.nitehawk.com/rasmit/ and click on the "DSP for Weak Signal" link at the top of the page. Or, go directly to http://www.nitehawk.com/rasmit/dsp50.html to get to DSP directly. If you do the latter you' ll miss a nice introductory page though.

## III. Linrad: Leif Asbrink SM5BSZ's Linux PC Radio for Intel Platforms

Leif Asbrink, SM5BSZ, has developed a superb weak signal receiver in software, which is named Linrad, short for "Linux Radio". This receiver is the ultimate DSP tool for optimizing the receive chain where the human is the final link. Here is what he has to say about Linrad, by way of introduction.

"Modern computers have the processing power to outperform conventional radios in receiving signals with poor S/N. Particularly when the poor S/N is due to interferences rather than to white (galactic) noise the computer can remove interference within the narrow bandwidth of the desired signal by use of the information about the interference source retrieved by use of larger bandwidths. The signal processing can be far more clever than what has been possible before. Each interference source can be treated as a signal and the DSP radio can receive AND SEPARATE a large number of signals simultaneously. The DSP radio package is under development with flexibility and generality as important aspects. The DSP-radio for LINUX is designed for all narrow band modulation methods for all frequency bands. To start with the following modes will be included: Weak signal CW (primarily EME), Normal CW, High speed CW (meteor scatter), SSB, FM". He goes on to say, "The system is designed for flexibility so it can be used for many different combinations of computers, A/D boards and analog radio circuitry. The platform is Linux and the package will typically operate with a 486 computer together with a conventional SSB receiver as the minimum configuration. The current high end operation is with a 4-channel 96 kHz A/D board and a Pentium III providing nearly 2 x 90 kHz of useful signal bandwidth in a direct conversion configuration (stereo for two antennas). When the Linux package is in full operation I will interface it to a modern radio A/D chip and digital data decimation chip. The component cost is very low and there will be an exciting improvement in dynamic range, bandwidth and flexibility. The LINUX PC-radio for Intel platforms will be continuously upgraded to show various aspects of digital radio processing and how they are implemented in the DSP package. The Linux PC-radio is not designed for VHF weak signal only. It is very flexible and designed to accommodate routines for all radio communication modes on all frequency bands. The program can run on a 486 to process 3 kHz bandwidth with almost any sound board. It can also run on a Pentium III with a 96 kHz board such as Digital River Delta44 [this is what I use; now called the M-Audio Delta44 -W3SZ] to produce spectra covering about 90 kHz bandwidth, using two mixers to provide a direct conversion receiver. (For EME it may be easiest to make a direct conversion receiver for a fixed frequency such as 10.7 MHz and put some converter in front of it). This is an ongoing project. The package will provide more than 30 kHz bandwidth with a standard audio board and should be very useful for 10 GHz EME and any other mode where a wide spectrum range has to be searched".

Leif started this project years ago with an MS-DOS PC radio (link http://ham.te.hik.se/~sm5bsz/pcdsp/pcdroot.htm) and has expanded the project and moved it to Linux for reasons of hardware portability (link http://ham.te.hik.se/homepage/sm5bsz/linuxdsp/linroot.htm). Details on how to install and get started using Linrad can be found on Leif's website at http://ham.te.hik.se/homepage/sm5bsz/linuxdsp/linrad.htm.

This is the current state of the art for weak signal work. The software receiver needs a baseband input of 0-xx kHz to achieve a waterfall/spectral bandwidth of xx kHz. Given this input, and using quadrature detection, a sampling rate of yy kHz will provide a waterfall/spectral bandwidth of up to approximately yy kHz (with yy less than or equal to xx, of course). If standard (non-quadrature) mixers are used, the maximal waterfall/spectral bandwidth will be approximately yy/2. With an M-Audio Delta44 soundcard for the input, operating at 96 kHz sampling rate, and with quadrature detection, the useable waterfall bandwidth available is about 90 kHz. There are several options available for use as a front-end to Linrad. First of all, it's a simple matter to homebrew a very good front-end, as Linrad does all of the hard work. Here I use a circuit with a couple of TUF-1 mixers from Minicircuits, a 10.7 MHz 1st IF and filter, with a transistor amp for the first IF amp and a low-noise OP amp (AD797) for the 2nd IF amp. This circuit is not quadrature, so I get just about 45 kHz of useful bandwidth when using it with the Delta 44. I use a computer-controlled 1st LO, so that I have wideband frequency coverage with the receiver; essentially from about 0 to 500 MHz (with appropriate filters at the input for each range, to prevent spurious responses). Figure 3 shows a diagram of the circuit for my homebrew receiver front-end.

I have two of these units operating simultaneously, one for the horizontally polarized antenna array elements and one for the vertically polarized elements. Linrad uses both of these to provide reception that is always at the correct receive polarization angle; if the incoming wave is at 37 degrees, that is how Linrad receives it. No more "lock-out" or signal degradation due to crossed-polarization. And Linrad does this automatically!

The second option is to use a kit that was introduced by Expanded Spectrum Systems at Dayton this year, called "The Time Machine" ( http://www.expandedspectrumsystems.com/prod2.html). The Time Machine is a quadrature mixer single-conversion HF receive chain that mixes the incoming signal down to baseband frequency (in this case, 0-96 kHz for me, with the output of The Time Machine connected to the input of the Delta44 sound card). I use two of these units, one for each receive polarization, and use a TUF-1 mixer and computer-controlled 1st LO before the input, to mix the 144 MHz signal down to 10.7 MHz, which is then fed into the Time Machine. Another approach would be to feed the output of a 144 MHz to 28 MHz transverter into the input of The Time Machine instead of using a homebrew front-end. Better weak signal performance will be obtained if the on-board LO is replaced by one with better phase-noise performance. Expanded Spectrum Systems is planning on offering an optional daughter board that will offer this improved performance. Until that's available, it is a simple matter to disconnect the output of the ICS-501 from the input of the 16V8B and to substitute one's own oscillator at 4 times the desired 2nd LO frequency.

The third option for a front-end is a little bit in the future; perhaps near the end of this year, perhaps early next year. Leif has designed a superb front-end for Linrad, consisting of several parts. It basically consists of separate units which provide conversion from 144 MHz to 70 MHz, from 70 MHz to 10.7 MHz, from 10.7 MHz to 2.5 MHz, and from 2.5 MHz to baseband. You can see his design for the latter two of these converters, complete with the circuit board masks on his website at http://ham.te.hik.se/homepage/sm5bsz/linuxdsp/optrx.htm. I believe that these lower frequency converters will be sold by Svenska Antennspecialisten AB, whose website is at http://www.antennspecialisten.se/. There is no information on their website yet about these units.

A block diagram of the functionality of Leif's SOFTWARE is helpful in understanding how it works. Figure 4 is a copy of Leif's block diagram, taken from his website.

The receiver input is at the top left of the diagram. Two input signal paths are shown, one for the horizontally polarized antenna elements and one for the vertically polarized elements. The FFT's are of course fast Fourier transforms, that take the signal from time domain to frequency domain, and the timf's are reverse transforms that take the signal from frequency domain back to time domain. The blue signal paths are in the time domain and the red signal paths are in the frequency domain. The green lines represent power vs. frequency displays and the black lines represent control signals. The diagram nicely explains the signal flow and processing. The first FFT is used to generate the wideband spectrum display. AGC functions are then performed and then the signals are subjected to reverse Fourier transformation that puts them back in the time domain. The noise blanker is then applied. The noise blanker is a novel, two stage circuit if the Linrad receiver has been calibrated for frequency and phase response using a pulser unit. (This procedure is not difficult, and is described in detail on Leif's website). The first stage blanker is called the 'clever' blanker. It models the noise and fits to each pulse a 'standard' pulse with amplitude, phase, fractional position, and polarization all calculated to match the actual noise pulse as closely as possible. The standard pulses are then subtracted from the signal waveform, reducing the noise pulses by approximately 30 dB. The 'dumb' second stage blanker then removes all data points for which the total power is above a given threshold. This reduces the noise by approximately 40 dB. The results achieved by this two-stage noise blanker are phenomenal! If the receiver has not been calibrated, then just the 'dumb' noise blanker is available, but even this does a very good job. After noise blanking is done, the second FFT is performed. This produces the waterfall display, and after the polarization control algorithm is applied, the high resolution display is generated. A second reverse FFT is performed. There is then the possibility for more noise blanking. A third FFT provides the baseband display and then another reverse FFT returns the signal to the time domain for final signal processing and audio is sent from the soundcard to either audio amplifier, speakers, or headphones. All of this is explained in very detailed fashion on Leif's website. In addition, Linrad's source code is there.

I have found that Linrad does an absolutely superb job of allowing me to hear the desired weak signal hidden in the midst of the all the noise and clutter present at my QTH. It does this better than any other receiving system I have ever tried. I generally use it with the filter set at 20-25 Hz. The best way to describe Linrad's operation and features is to discuss a series of screen grabs I made while using it as my primary receiver during the EME contests last year. I'll start with Figure 5, a screen grab taken with the receiver operating on 144 MHz, during an EME contact with KB8RQ. This is a newer version of Linrad than I used in the screen shot shown in Figure 1 as an example of a waterfall display, and I've arranged the screen differently. Leif's software lets you rearrange the screen almost anyway you want to.

Across the top of Figure 2 you see the frequency scale in Hz. At the time of the contest, I had Linrad set up to cover 22 kHz, which is a reasonable spread for 2 meter EME. With it set up like this, one can see everything that is going on in a 22 kHz slice of the band. Depending upon your soundcard, as discussed above, you can view up to a nearly 96 kHz-wide slice of the band at one time.

The small arrows near the left and right corners at the top of the screen allow adjustment of the frequency width and the center frequency of the waterfall and main spectrum displays (which track together in this regard),.

Below the frequency scale at the top of the screen is the waterfall display, showing signal intensity as a function of frequency horizontally, and as a function of time, vertically. Earlier times are nearer the bottom, most recent times at the top. Decimal minutes are displayed along the left vertical axis. The wide white horizontal bands extending from 23000 to 31000 Hz are my signal when I am transmitting. Of course, when I am transmitting, my transmitter overloads the receiver as you can see occurring periodically, resulting in dark horizontal bands across the waterfall display. You can also see a dashed vertical line at 27000 Hz; that is KB8RQ's signal on the waterfall. Lower on the frequency scale, at 21800 Hz or so is the dashed vertical line that represents W5UN's signal. Between the times 53.36 and 56.15 you can see a signal that starts at 27500 and then moves across the screen to lock in on KB8RQ's frequency at 27000. That is me 'spotting' the frequency to bring my transmitter onto KB8RQ's frequency.

Just below the waterfall on the screen is the real-time main spectrum display. Signal strength is the vertical axis and frequency is the horizontal axis, corresponding to the same locations on the waterfall and the frequency calibration at the top of the graph. The little up/down arrows at the bottom left and middle right of this display allow you to adjust the range and center point (baseline), respectively, of the spectrum amplitudes displayed, so that the signals are the right vertical size for best viewing, and centered as you wish on the display. It is much more difficult to pick out weak signals on this display than on the waterfall, and I don't use the spectral display very much. Leif notes that the main purpose of the main spectral display is to aid in noise blanker level adjustments and to show very strong signals that saturate the waterfall display.

Below this on the left are the boxes to set (by clicking on the box and then typing in the desired values): the number of FFT1 averages per displayed point of the spectrum, the number of FFT1 averages per line of the waterfall, the zero point of the waterfall display, the gain of the waterfall display, the number of averages per displayed point of the high resolution spectrum (this is the display with red horizontal lines) and the number of averages per displayed point of the baseband spectrum (the noise blanker / filter window; this is the display with green horizontal lines).

To the right of these parameter boxes is the adaptive polarization control. The software receiver can be set up to receive two channels of data. In my case, one is the signal from the vertical elements of my M2 array, and the other is the signal from the horizontal elements. By rotating the line with the mouse you can select any desired receive polarization angle. Or, you can leave this set to automatic or ' adaptive' mode and then the software constantly optimizes the polarization angle. Moving the line on the horizontal bar (green when you can see the colors) changes the polarization from linear to elliptical to circular. I usually leave the polarization control set to 'adapt' and let the computer do the work.

To the right of that little box is the high resolution display. To the right of that, on top, is the baseband (DSP filter) display, which will be discussed shortly. Below that immediately to the right of the high resolution display is the small coherence graph and signal amplitude box, and to the right of this below the baseband display is the automatic frequency control box.

Here is the important and, really, incredible part. By clicking with the mouse cursor at any point on the waterfall (or the main spectrum) you cause that portion of the spectrum to be placed in the high resolution spectrum box and DSP-processed. That is, that portion of the spectrum is DSP-filtered, noise-blanked, and converted to audio frequency so that it appears in your headphones or on your speakers. IT IS POINT AND CLICK RECEIVING!!! Because of the excellent DSP, this is an incredible experience. If you are not clicked on a signal, the receiver is silent. When you click on a peak, the signal pops into your headphones. To fine tune, you click on the peak in the high resolution spectrum, if need be, to touch up the tuning. You can see that KB8RQ's signal is nicely centered in a 20 Hz bandwidth filter by looking at the baseband window. The two tiny "A"s (yellow and blue when you can see the colors) at the bottom left of the high resolution window are for setting the 'dumb' and 'smart' digital noise blanker levels. You have your choice of none, automatic, or manual for setting these blanker settings. The tiny "o" at the right bottom of this display turns on the oscilloscope function that shows the real and imaginary time domain spectra for the various channels of data so you can really tell what the blankers are doing.

The line and 'hump' (yellow when you can see the colors) in the baseband display (which sits just to the right of the high resolution display) show the filter center frequency, bandwidth, and shape factor in graphical form. If you want a different filter bandwidth or shape factor, you just take the mouse over to the baseband display, and drag the filter curve wider or narrower, and the filter adjusts graphically. THIS REALLY WORKS!! There are several controls in the baseband window. As we just noted, by dragging the yellow lines with the mouse you can set the filter width and shape factor. There is a red horizontal bar at the left of the window that does not really show up with grayscale reproduction. This is the level or volume control. There is a red vertical bar at the right of the window that is the BFO control. You can vary the pitch of the received signal without taking it out of the filter pass band or moving it in the display. This is very nice. There are other controls at the bottom of the baseband display for setting coherent processing parameters, adjusting binaural receive parameters, and altering how the program handles the two signals in a dual polarity receive system.

How fast a machine do you need to run Linrad? It depends on the parameters. With my setup, Linrad says that my machine, a 1.4 GHz Pentium 4 is idling 92.4% of the time while it is running. In other words, only 7.6% of its processing power is being used by the program. Leif has a very nice page that discusses timing / computation intensity issues and gives some examples for various hardware combinations. It is on his website at http://ham.te.hik.se/homepage/sm5bsz/linuxdsp/fft1time/fft1time.htm .

If you don't have a receiver with a wide (20-90 kHz) IF bandwidth you can still experience the wonder of Leif's receiver. On his website he has lots of EME contest files that you can download. Once you have his receiver running on your computer you can put the names of these files in a text 'adfile' that you create in the Linrad directory. This file will direct Linrad to these data files and when you start Linrad you can type 'h' and the program will run just as if you were actually receiving this data via your own antennas. You can click on the various signals, and even play with the receive polarization control. It is truly amazing to do this! Leif's main Linrad radio page has links to his many useful pages of explanation, diagrams, screenshots, etc. His website is a real treasure trove.

Figure 6 is a screen grab of a spectrum with Dave, W5UN's signal as received at W3SZ via the EME route. I'll use it to explain a bit more about Linrad's operation:

On the waterfall display you can see Dave's signal appearing as a dotted line (because he is alternately transmitting CQ and listening) just to the right of an external signal near 27000 Hz. The very large white bands near the right side of the waterfall are my transmitted signal, which of course overloads the receiver, as you saw on the previous screen dump as well. On the waterfall display the external 'birdie' is just a hair's width below Dave's signal. Below the waterfall display, in the main spectrum display, you can see a green vertical cursor on Dave's signal. This serves to select it for processing for the high resolution spectrum display, DSP-filtering, noise blanking, and output for audio copy by the operator. The 'birdie' and Dave's signal run together on this lower resolution display. On the high resolution display, which is the tall, relatively large spectral display below the main spectrum display, there is excellent resolution of the signals; Dave's signal occupies the center of this display around 21672 Hz, and the mouse cursor '+' sits nearly exactly over the middle of Dave's signal. The green spectral peak is the "in phase" component of the received signal. The fact that there is no purple or "out of phase" peak indicates that the receiver has locked on to the polarity of W5UN nicely. The little blue "receive polarization" display to the left of the high resolution spectrum window shows that the received polarization angle is 83 degrees, and that Linrad is in the adaptive polarization mode, where it follows the polarization angle of the received signal automatically. The baseband display is below the main spectrum display to the right of the high resolution spectrum display. This window shows that the filter, shown in yellow and set to about 20 Hz bandwidth, is well centered on Dave's signal. You can see both the center of Dave's signal and the keying sidebands within the yellow outline of the filter band pass curve. The vertical magenta bar at the very left of this window is the level or volume control, and is adjusted by dragging the top of it with the mouse. Above it is a very bright red 'dot' (actually a short, horizontal line) that indicates the received signal level. It is 'pinned' at the top of the scale, commensurate with Dave's usual signal strength. In this same window, the vertical red bar on the extreme right of the display shows the BFO frequency. The BFO frequency is off scale with the settings used in this display. Note: all of the 'birdies' (and there are many) are external signals here, that I receive with each of my receivers. They are present only with the antenna connected, and come and go with time and with variation of the antenna heading and polarization angle. They are not a problem produced by Linrad or the homebrew receiver, although this combination does an excellent job of showing them due to its sensitivity! At the very bottom of the display to the right of the high resolution display, and below the filter window, are the coherence graph / signal amplitude window on the left, and the AFC window on the right. The coherence graph shows that Dave's signal has good phase coherence for automatic CW copying, and his signal amplitude as received here is 34.6 to 35.8 dB. This is relative amplitude, as I have not calibrated my system for absolute received signal strength. Both this window and the AFC box are discussed fully on Leif's website.

Finally, Figure 7 shows a Linrad screen of the W3CCX beacon on 13 centimeters, using the homebrew 144 MHz receiver and Linrad as the IF radio. I've changed the positions and sizes of the high- resolution, baseband, receive polarization, and AFC windows. This is just a matter of dragging the edges of the windows to change their positions. Note the frequency resolution on the waterfall and main spectral displays is now enhanced compared with the prior examples. The entire spectrum covers only 3 kHz. The frequency instability of the beacon signal can be easily seen. The receiver frequency is controlled by a Rb oscillator, and the variation seen represents the frequency instability of the beacon plus any Doppler shifting of the signal while en route to W3SZ. The high resolution window shows only a green spectral peak (which goes off scale at the top of this window) because there is only horizontally polarized input on this band at W3SZ. There is thus no signal in the orthogonal magenta-colored channel. The baseband display (which is below the high resolution display window) could be improved by contracting the vertical scale to cover just 0-30 dB rather than setting it to 0-60 dB as was done here. Again, this is just a point-and-click adjustment. This makes for an awesome receiver on the microwave bands. You can see (depending upon the filters you use in the receiver) up to 40 kHz or so of spectrum at a time if you are using a setup like mine, to spot those 'way off frequency' microwave signals!! If you use quadrature detection, you can expand the spectrum viewed at one time to a bit more than 90 kHz! In this case I went to the other extreme, to look at the frequency stability of the W3CCX beacon over a relatively long and obstructed path, during a period of suboptimal receive conditions.

**IV. DSP-10: Bob Larkin W7PUA's Innovative 144 MHz DSP Transceiver**

The superb DSP-10 hardware/software by Bob Larkin, W7PUA, really goes a step beyond the other solutions discussed here. It is more than a signal processor, when used as a hardware/software solution with Bob's hardware design. Then it is a weak signal PC-based 'software transceiver' for 144 MHz. It provides not only signal processing, and the traditional CW, SSB, and FM modes, but also entirely new digital modes of communication using long-term integration and FSK techniques. The DSP-10 is different from DSP-Blaster and some of the other products reviewed here in that it uses an Analog Devices EZ-Kit Lite to do the DSP processing rather than a sound card; it connects to a PC via a serial port and just makes use of the PC for control and display; the EZ-Kit Lite does all of the computational chores. It runs under true DOS mode only (NOT a DOS Window). Bob's website is at http://www.proaxis.com/~boblark/dsp10.htm .

The DSP-10 software can also be used without the RF hardware, as an audio DSP processor, using just the Analog Devices EZ-Kit Lite evaluation board (see below). Or, it can be used with an external homebrew front-end as the final IF of a homebrew receiver. I have done this at W3SZ with just a couple of TUF-1's and computer-controlled oscillators to make a very nice receiver that covers from below the AM Broadcast band to the 432 MHz band. Whether used with or without the DSP-10 RF hardware, it has an excellent waterfall display with adjustable gain, adjustable baseline, and adjustable signal averaging (see Figure 9). The visualized bandwidth can be set to 1200, 2400 or 4800 Hz. It has an excellent spectral display also with adjustable gain and baseline, and 1200, 2400 or 4800 Hz visualized bandwidth, as well as adjustable signal averaging. It also has DSP audio processing with 200, 300, and 450 Hz bandwidth filters at center frequency of 600 Hz, a 600 Hz bandwidth filter at center frequency 700 Hz, and a 300 Hz bandwidth filter at a center frequency of 800 Hz. In addition there is a sin(f)/f matched filter for 12 wpm, and a design-your-own filter too. It has adjustable LMS noise reduction, binaural receive capability, defeatable and adjustable AGC, and a notch filter. Unfortunately the notch filter is not manually adjustable, so it will very effectively eliminate the desired CW signals.

The software is free and can be downloaded from Bob Larkin's web site listed above. Bob's web site has a parts list that makes ordering the parts for the project a simple matter. Also, TAPR (http://www.tapr.org/) has periodically made kits available with all parts except for the

EZ-Kit Lite. Check their website for details on the next kit run. Unfortunately, the EZ-Kit Lite, originally Analog Devices Part ADDS-21XX-EZLITE, is now part number ADDS-2181-EZLITE, and the price has risen several fold. I purchased mine from Newark Electronics in 1999 and then again in early 2001, when the price was $94.70. As of 8/30/2002 the Newark price is $346.95 plus shipping.

Setting up the audio version of the DSP-10, without the RF hardware, is a piece of cake. You just plug the audio output of your rig into the input of the EZLITE, plug your headphones into the output of the EZLITE, connect a serial cable from your computer to the EZLITE, and fire up the software. Its just as easy if you are using your own external RF front-end. Building the complete DSP-10 computer-controlled 144 MHz transceiver as described in the September, October, and November 1999 issues of QST takes some time but it is not difficult, although it does make use of surface mount technology. I built a DSP-10 in early 2000 and have found it to be superb as a microwave IF. The ability to spot those weak signals that are up to 4 kHz off frequency at 2304 MHz and above is critical in the microwave contests, and this radio really does that superbly. I have also used the RF Hardware Version of the DSP-10 as a 144 MHz weak signal receiver and used it on 144 MHz EME. For this use its somewhat limited dynamic range is a shortcoming. It is at least in my experience, at an admittedly very noisy location, often overloaded during contests when there are strong local signals, and this has been at times a significant problem when trying to use it for weak signal work in such an environment. The ARRL lab rated its IP3 at -21 dBm. The limitation in dynamic range is I think probably a reflection of the limited dynamic range of the EZLITE; the EZLITE is based on the ADSP-2100 Series which includes 16 bit fixed point processors, and the Codec used in the EZLITE is the AD1847 which has probably just12-13 bits of usable dynamic range although it is a 16 bit device. The MAR-6 used as the first RF amplifier has an IP3 of +14 dBm and although it is followed by another MAR-6 before the first TUF-1 Mixer and the crystal filter, I don't think these components should be limiting in this regard.

Before describing the very original and unique automated detection and decoding features of the DSP-10 I'd like to share my experiences with it during the ARRL EME Contest 2000 when I used with a human (me) as the final step of the detection/decoding chain. The first night of the first leg of the contest I got my wish for making this a useful "acid -test": a noisy night with a lot of local activity, ideal conditions for comparing the RF version of the DSP-10 with my 'usual' EME station. This combination of a high noise level and high-powered local activity made some things very clear. The experimental system here for the contest included a 2 x 2 stack of M2 2MXP20's with separate receive lines for Horizontal and Vertical Polarity.There were dual ARR 144 MHz preamps on the tower, close to the power dividers (my cavity preamps were temporarily on the ground). The feed lines (LMR600UF to the bottom of the tower, and then 7/8 inch hardliner to the operating position) were connected to an EME grade relay in the shack so I could switch polarity with a flick of a switch, or monitor both H and V simultaneously on two different receivers. For the comparison I hooked a two way power divider after the relay to split the receive signal into two paths, one path going to the SSB Electronics LT2S Mk II / Elecraft K2 combination, and one going to the RF version of the DSP-10. As I have slightly more than 20 dB of preamp gain on the tower, and significantly less feedline loss than this, this signal splitting didn't produce any noticeable degradation in terms of minimal detectable signal. At the same time, the baseline signal level in the shack was low enough that I didn't need additional attenuation in the line to prevent unnecessary degradation of system dynamic range due to excessive preamplifier gain.

What I found was significantly different in some respects than what I had previously found for the Italian EME contest, as that was a rather quiet night. First of all, there are some 1.5 kW stations within a few miles of me, and the DSP-10 RF version folded up and died when these stations were transmitting at times when I was more of less in their line of sight to the moon and when the moon was low in the sky. In contrast, the K2/Transverter was really only bothered when these stations were very close in, frequency-wise, and then I could still work through their noise. Having the noise blanker on the K2 activated did increase the problems with local QRM, as is the case with any noise blanker I've ever tried, but I could still work with the NB on. In contrast, I just had to shut the DSP-10 down when the locals were transmitting. I should note that the only other operating conditions in which I've also experienced overload of the DSP-10 were when using it as the IF for 10 G experiments at very close range (30-50 meters), but that otherwise the DSP-10 has not had overload problems when used as an IF for microwave work here. Nor has it had such problems when used on 144 MHz with cavity preamps when the noise level is not excessive and high-powered locals are off the air. I should also note that when I use my FT-1000 MP in place of the Elecraft K2 I the receiver chain I experience similar dynamic range problems when both high-noise and high-signal-level conditions are present. And the FT-1000 MP is generally considered to be a fine receiver, if not in the same class as the K2, the Ten-Tec OMNI VI+, and the modified Drake R-4 (see September/October 2002 QEX for an excellent reference on receiver performance comparisons, by SP7HT).

Second, under these noisy conditions, the LMS noise reduction on the DSP-10 didn't take out the noise sufficiently. This is not surprising, as when the pulse noise is very bad here, NO LMS algorithm I've ever used is sufficient. It takes an excellent noise blanker such as the one in the K2 plus LMS (or just Linrad) to do the job (the FT1000 MP noise blanker is not up to the task, either)! During the periods where the pulse noise was absent, and the kilowatt-plus locals weren't transmitting, the DSP-10 was a joy to use. As I noted before, it is very comfortable to listen to and with the binaural implementation Bob Larkin has placed on it, it really pulls the signals out of the mud when the binaural mode is activated. During the quiet periods I preferred it to the K2/Transverter, but again it couldn't be used when the pulse noise or local QRM was present at substantial levels.

Third, I have a lot of birdies here at W3SZ. Frequently they were close in to the station I was trying to copy. Here the use of the K2/Transverter fed into the computer running DSP-Blaster was again superior, as I could either make the filter very narrow, or turn on the adaptive notch DSP-Blaster provides (its VERY sharp and narrow), or both, and eliminate the offending carrier. The filters in the DSP-10 weren't quite as good, and its notch can't be used on CW, as it automatically removes the desired signal.

Fourth, on the VERY weak stations, it was helpful to be able to play with the filter parameters on DSP-Blaster, and I found that for the really weak stations I could only copy them with the K2/Transverter/DSP-Blaster combination. The DSP-Blaster is more flexible in its filter selections than is the DSP-10, which was described more fully in this regard above. With the DSP-Blaster, the filter width is adjustable from 50 to 300 Hz in 25 Hz steps for the Coherent Bandpass Filter, and the center frequency is adjustable from 10 to 990 Hz in 10 Hz steps. For the standard bandpass filter the filter width is adjustable from 50-975 Hz, again in 25 Hz increments. In practice the lower limit of the center frequency for the DSP-Blaster filter chain is determined by the low-frequency roll-off of the filters in the receiver itself.

In terms of the waterfall displays, I ended up liking the spectrum display of the DSP-10 much more than that of DSP-Blaster.

To summarize the use of the DSP-10 for weak signal use with human psychoacoustic detection and decoding, it does a very good job for weak signal work except under conditions of high noise and strong signal interference. It should be excellent most anywhere for use as a microwave IF rig, and is very good on 144 MHz as a primary weak signal transceiver unless the environment is one of very high noise level or very strong local signals, or both. Unfortunately, this is my current environment much of the time.

What REALLY sets the DSP-10 hardware/software combination apart from the competition is the very innovative "new" modes that Bob Larkin designed into the software. These modes can be accomplished only when the unit is used to communicate with other DSP-10 units, and they place stringent requirements on frequency accuracy and stability, and correct timing. Frequency accuracy and stability must be on the order of 0.5-1 Hz AT THE OPERATING FREQUENCY (which may not be 144 MHz if the DSP-10 is used as an IF rig) when 2.3 Hz bin size is selected. Timing of the two stations must be within 50-100 milliseconds (30 milliseconds is preferred). These requirements pretty much mandate the use of GPS-controlled oscillators for frequency control, and GPS or other NIST-Standardized timekeeping at both ends of the attempted contact.

I have been successful using, for frequency control, either a Rb-controlled oscillator (Efratom FRS) or a GPS-disciplined oscillator (either Brooks Shera's homebrew unit or a surplus HP Z3801A). See K8CU's website for an excellent first look at the Z3801A, with many very helpful details (http://www.realhamradio.com/GPS_Frequency_Standard.htm). These units are available from Buy Legacy for $249.00 as of 8/30/2002. The units are listed on their website with many details at http://www.buylegacy.com/app/site/site.nl?site=ACCT9270&mode=items&category=55&it=A&id=1181. See Brooks' website at http://www.rt66.com/%7Eshera/index_fs.htm for details on his fine unit. I've had good results with either GPS or internet-controlled timekeeping. If, like me, you don't have network access on the DOS machine you use for DSP-10, getting the timing accuracy from the GPS or internet-controlled clock into the DSP-10 machine at the 30 millisecond level can be a bit tedious!

The "automated" weak signal modes available with the DSP -10 are named EME-2, PUA43, LHL-7, and LTI.

EME-2 is a mode for determining system performance and path conditions. A typical screen for this mode is shown in figure 9. In this mode the DSP-10 transmits a CW pulse for 2 seconds. There is a delay of 2.6 seconds from the START of the transmitted pulse to allow for pulse travel, followed by a 2 second receive period. (The actual pulse travel time will vary from about 2.4 seconds at perigee to about 2.7 seconds at apogee). The cycle then repeats (every 5 seconds) for as long as desired, and the DSP-10 averages the received signal to gain the 'square root of n' increase in signal-to-noise ratio obtained with n > 1. An averaged spectrum is displayed in yellow in the 'standard' DSP -10 spectrum display area. With this technique Bob W7PUA has detected well defined EME echoes using 5 watts of transmit power and a 4 x 12 element array! EME Doppler correction is automatically applied by the software. The software can introduce a random frequency shift with each cycle, to minimize the effects of any birdies that are present by 'smearing' them over many bins. A maximum range of spread of 200 Hz is a good starting point for this parameter. The software will automatically identify in CW periodically, to meet FCC requirements. For this mode the timing requirements are relaxed, since the same station is transmitting and receiving, but the timing should still be within about 20 seconds for 144 MHz work or 2 seconds for 1296 MHz work so that the Doppler correction calculations are adequate. Bin width is set to 2.3, 4.6 or 9.2 Hz by setting the Spectrum Analysis Width (SpecAnl Width) parameter to 1200, 2400, or 4800 Hz, respectively. The EME-2 mode will provide an estimate of the returned signal strength, and is thus an excellent way to initially characterize and then follow the performance of one's EME system over time.

The second "automated mode" is PUA43, so named because Bob W7PUA created it, and because there are 43 defined characters in the PUA43 code: the letters A-Z, the numerals 0-9, Space, Period, the comma, the forward slash, the pound sign, the question mark, and the Dollar sign. The pound sign is defined as "message received" and the Dollar sign is used to shift the meaning of the following character. This mode is the mode generally used for two-way low power level EME communications, and is the direct antecedent of JT44. With PUA43, message lengths of 14 or 28 symbols can be selected. The mode transmits for one minute out of each 2 minute transmit/receive cycle, and receives for other minute. Each symbol is sent for 2 seconds, so if a 14 character message length is selected, the message is repeated during the second half of the minute-long transmit period, giving an immediate 1.5 dB signal-to-noise advantage over the 28 character-length mode. At the end of each transmit minute a 4 second CW ID is inserted. In this mode each character is represented by one of 43 defined frequency bins. Within the received spectrum, every fourth receive bin is used for one of these 43 character tones, with guard bins adjacent to each tone bin used to allow for spectral broadening and frequency errors. Noise is estimated by measuring the level of the second-adjacent bins on either side of each tone bin. A pseudo-random frequency spread is again used to reduce the effects of birdies. The shift for each transmit period is taken from a lookup table that defines a value of the shift function for every minute of the day. Doppler corrections can again be automatically applied, and the pulse transit delay (2.6 seconds is again used) is applied automatically. The frequency accuracy of the transmit/receive system should be kept to within 0.5, 1.0, or 2.0 Hz AT THE OPERATING FREQUENCY depending upon the bin width, which is a function of the Spectrum Analysis Width, as noted above. The software uses a message estimation algorithm to estimate the received message based upon the previously received data. A black-colored character in a message estimation box means the estimate has a low confidence level for that character, a beige character indicates a medium level of confidence, and a white-colored character means a high confidence level has been achieved. It is possible to adjust the quality level required for achieving a particular result with this display, by using the "Quality Ratio" parameter. With this mode Bob W7PUA and his brother Beb W7SLB have completed a two-way EME contact on 144 MHz using single yagis and 150 watts transmitter power at each end. Ernie W7LHL and Larry W7SZ completed an EME contact on 10 G with TVRO dishes and 5-15 watts at each end with this mode. The details of these exploits are at http://www.proaxis.com/~boblark/wksig1.htm.

The third DSP-10 automated mode is LHL-7. This is essentially an FSK-based Morse code method with much greater tolerance for frequency errors than PUA43. IT can be used with either human or automatic decoding. Dots are sent at 750 Hz tone frequency, dashes at 900 Hz, end of character at 600 Hz, and 'times two" at 1050 Hz, 'times three" at 1200 Hz, 'times four" at 1350

Hz, and 'times five' at 1500 Hz. Thus 'e' would be 750 Hz, 600 Hz. The numeral '5' would be 750 Hz, 1500 Hz, 600 Hz. Each tone can be sent for either 2, 4, 10, 20, or 60 seconds. Automatic Morse ID can be sent periodically. Automatic Doppler correction can be applied. Timing requirements are relaxed slightly due to the potentially longer pulse lengths, but must still be within the range for adequate automatic Doppler correction. There is substantial flexibility in the parameter selections available for this mode, and they are spelled out very nicely in the file Readme20.txt, available on Bob W7PUA's website (listed above). This is the only DSP-10 mode that I have not used and tested.

The fourth and currently the last automatic mode available in the DSP-10 hardware/software package is the LTI, or Long Term Integration mode. This mode is able to detect signals in the range of -180 to -190 dBm. Such signals are on the order of 30-40 dB below audibility. This mode operates much like EME-2, with alternating transmit and receive periods, but the transmit and receive cycle lengths are adjustable variables, and either CW or a continuous tone can be transmitted, with automatic CW ID possible as well. Both frequency randomization and automatic Doppler correction for EME can be used if desired. There is no practical limit on the number of transmit/receive cycles or signal averages that can be used, as long as adequate frequency stability is maintained. For very weak signals the number of signal averages can be extended into the thousands. There is a 'noise blanker' function that will discard the data from a given period if the noise in the noise-measurement bins for that period exceeds the running average of the noise by 'x' dB. This mode is most useful for determining path performance between two stations, and for exploring the possibility of a link between two sites where conditions are expected to be at most marginal. It is an excellent experimental propagational tool.

The DSP can, for experimental purposes such as testing the robustness of the automated techniques, generate transmit signals with known amounts of Gaussian white noise added to the signal. Scrl-F6 puts one into the Setup menu for this feature.

### V. JT44: Joe Taylor K1JT's DSP Solution

Like its close relative PUA43, Joe Taylor's JT44 (a part of the WSJT software package), has been used successfully for EME contacts when the signals were too weak to be heard, when contacts by conventional means would have been impossible. The modes PUA43 and JT44 are very similar. Bob developed PUA43 for use with his superb DSP-10 transceiver, as described above, and Joe Taylor extended this type of mode for use with commercial gear by developing the JT44 mode. With these modes you do not audibly decode the other station's information; the software and computer hardware provide the decoding. If signal conditions are poor, you may not even hear the other station, but just see his signal on a time-averaged waterfall display. The skill with these modes is in setting up the hardware and software to do its job, and getting the radio on frequency. The programs then communicate by sending and receiving signals at precisely determined times and frequencies, and the information received appears on your computer screen.

As was described in the section above on PUA43 and the DSP-10, these modes are designed for very weak signal communication, and use relatively low data rates. They use narrow band filtering to divide the audio spectrum into many 'bins'. These bins are then assigned to different characters. Both programs use fixed message lengths to permit exact timing of messages. Bob

Larkin estimates that PUA43 may have up to a 20-25 dB advantage over human message decoding using the Morse code, and Joe Taylor estimates a 14 dB advantage for JT44. Both PUA43 and JT44 use the same character set described above for PUA43. They differ in that JT44 uses one additional tone frequency to provide frequency and time synchronization for the communicating stations. Whereas PUA43 achieves synchronization by means of specially designed hardware (the DSP-10) which allows for more efficient message encoding, this is at the expense of loss of applicability to a wider range of hardware. JT44 has in a very short time (I write this in August, 2002) become quite popular for 144 MHz EME work because of this wide applicability, and its effectiveness. Essentially the trade-off between PUA43 and JT44 is between greatest receive efficiency on the one hand and significantly relaxed frequency and time synchronization requirements on the other.

Each JT44 transmission lasts approximately 25.08 seconds, rather than 60 seconds as in PUA43. Each 25-second JT44 transmit period contains 135 intervals of data. 69 of these intervals contain only the synchronizing tone (1270.5 Hz), and the other 66 contain the encoded data. The synchronizing and data intervals are interleaved in a pseudo-random pattern such that the auto-correlation function of this pattern has a large spike at correct alignment of frequency and time, and very small values elsewhere. This property of the auto-correlation function is why JT44 stations can achieve alignment with each other without the need for special hardware. At the time I write this, JT44 will work with frequency errors of +/- 600 Hz or less, and clock errors of -2 to +4 seconds, according to K1JT. The asymmetric range is appropriate for EME because of the EME roundtrip delay. JT44 message lengths are fixed at 22 characters.

As with PUA43, sub-audible EME QSO's have been completed using JT44. In my first 24 hours of playing with 144 MHz JT44 EME, under terrible conditions where the only stations I could actually hear were W5UN and KB8RQ, I was able to complete QSO's with 5 smaller stations using JT44. My computer clock was controlled via one of the US government atomic clock Internet sites via the program AboutTime (Dimension 4 is another good one). The transceiver used was a Yaesu FT1000MP Mk V, with an SSB Electronics LT2S Mk II transverter. I was using about 800-1000 watts output to my 2 x 2 array.

Figure 10 is a screen shot of WSJT in JT44 mode obtained when I was receiving RU1AA calling CQ with JT44 on 144 MHz EME.

How do JT44 and PUA43 compare in performance? Bob Larkin has done a nice analysis of this which I will summarize here. Basically, when PUA43 is used in 14-character-message mode, it spends more than 3 times as long transmitting each character during each minute of transmission as does JT44. This translates to a 2.7 dB signal-to-noise advantage for PUA43. PUA43 bin widths are 2.3, 4.7, and 9.4 Hz while JT44 has a fixed bin width of 5.4 Hz. The three PUA43 bin widths have respective advantages relative to JT44 of 1.8 dB, 0.3 dB, and -1.2 dB. Thus the largest advantage PUA43 might be expected to have over JT44 would be on the order of 4.5 dB. This represents a power ratio of 2.8, and so it would take (2.8 x 2.8) = 8 times as long to achieve a given signal-to-noise ratio with JT44 compared with the time required with PUA43 when these parameters are used. This advantage for PUA43 is gained at the expense of many fewer stations having the capabilities of doing PUA43, as it requires the DSP-10 hardware. Additionally, JT44 can handle two character repeated messages such as 'ROROROROROR' as two rather than 22 character messages, and this would improve the JT44 signal-to-noise ratio by 5.2 dB. JT44 will also integrate the last four characters of a message, so that if these are identical an improvement of 3 dB is obtained. Thus under these circumstances JT-44 could outperform PUA43.

Joe Taylor's WSJT website is at http://pulsar.princeton.edu/~joe/K1JT/ . There are other JT44 resources on the web at http://www.qsl.net/wb5apd/JT44-eme.html and http://www.chris.org/cgi-bin/JT44talk and http://www.qsl.net/w8wn/hscw/papers/hot_news.html and http://www.qsl.net/w3sz/JT44.htm .

WSJT will soon include a test mode that is very similar to the DSP-10's EME-2 Mode.

## VI. Summary

For further information on DSP techniques for weak signal use refer to the links provided in this brief article, or refer to the links given on the web page "w3szstart.htm" included on the CD distributed in conjunction with this conference.

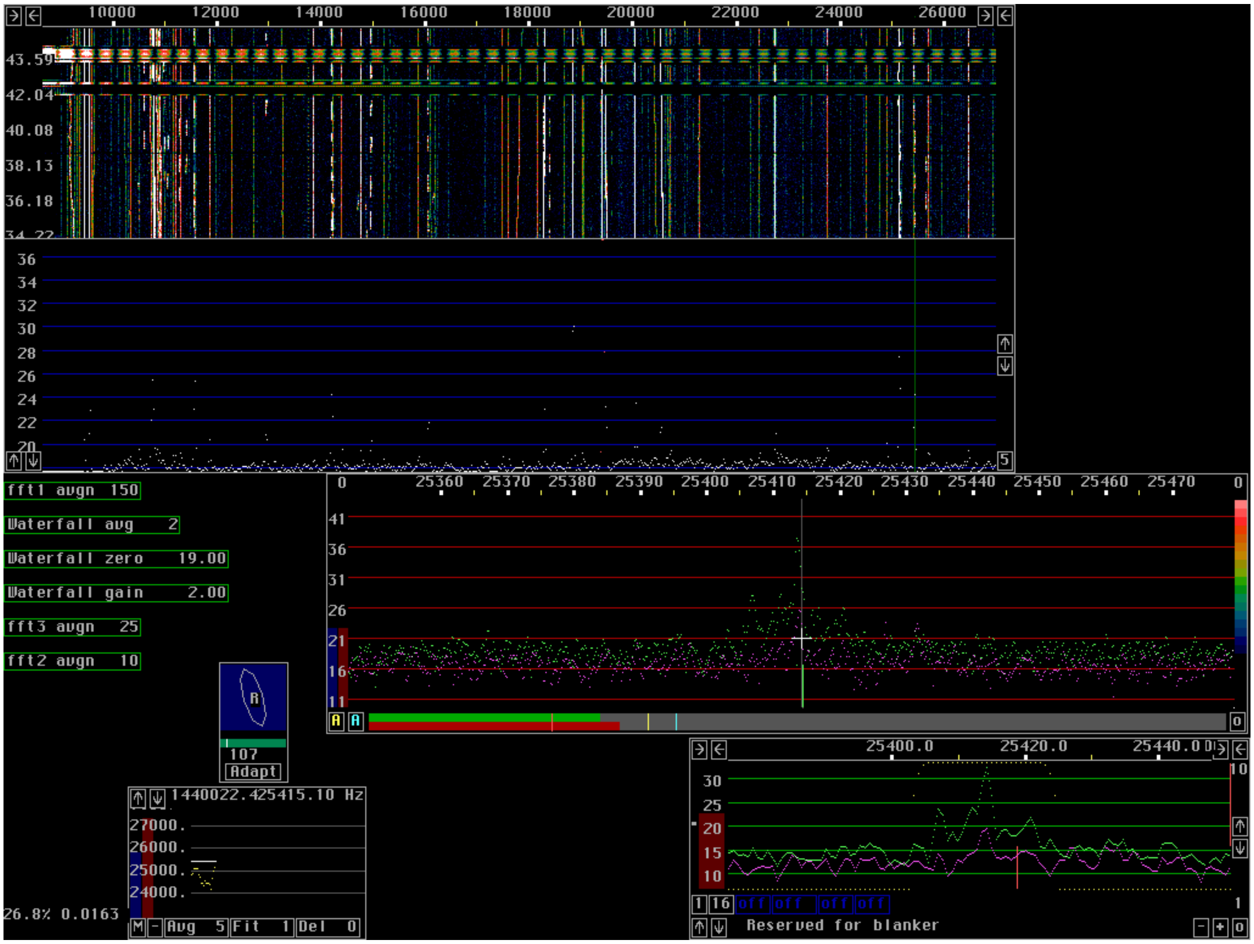


**Figure 1. Linrad screen showing a 16 kHz portion of the 2 meter band as received by me during the ARRL 2001 EME Contest. On the waterfall display at the top of the screen you can see among the many birdies I have at my location, at least 12 vertical dashed lines; each one of these is an EME station's signal as received by Linrad, SM5BSZ's software receiver at W3SZ. At 25413 Hz is G0RUZ, whom I was working when this screen shot was obtained. The broader horizontal bands across the waterfall represent my signal overloading the receiver when I transmitted.**

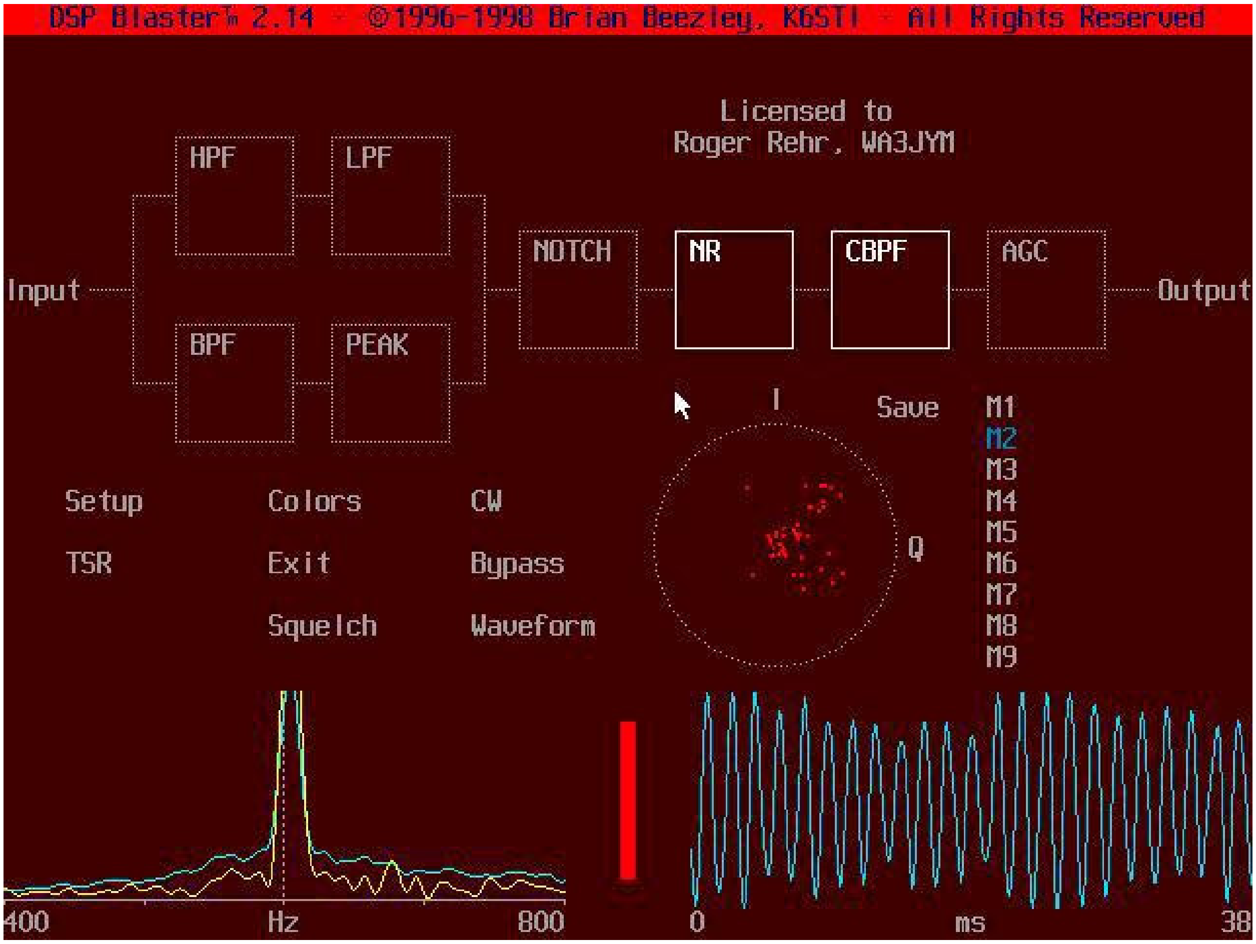


**Figure 2. The screen above didn' t do very well in the transition from bmp to jpg file type, but you can see the essentials of a DSP-Blaster screen captured as I was listening to F3VS off the moon. Across the top is the block diagram of the signal path in software. You put the mouse over a block and click to select it and adjust its parameters. Here I have the LMS Noise Reduction and the Coherent BandPass Filter activated. The circular display is of F3VS' s signal in binaural mode, the ' pseudo-stereo' mentioned in the text. The text in the center is to set certain parameters, activate TSR mode, etc. The designations M1-M9 are to save user-defined filter combinations for future use. On the bottom left is a spectral display, with yellow real-time and blue averaged. On the right is a time domain real-time display. The red vertical bar in the center is a level indicator, and indicates that F3VS has saturated the system.**

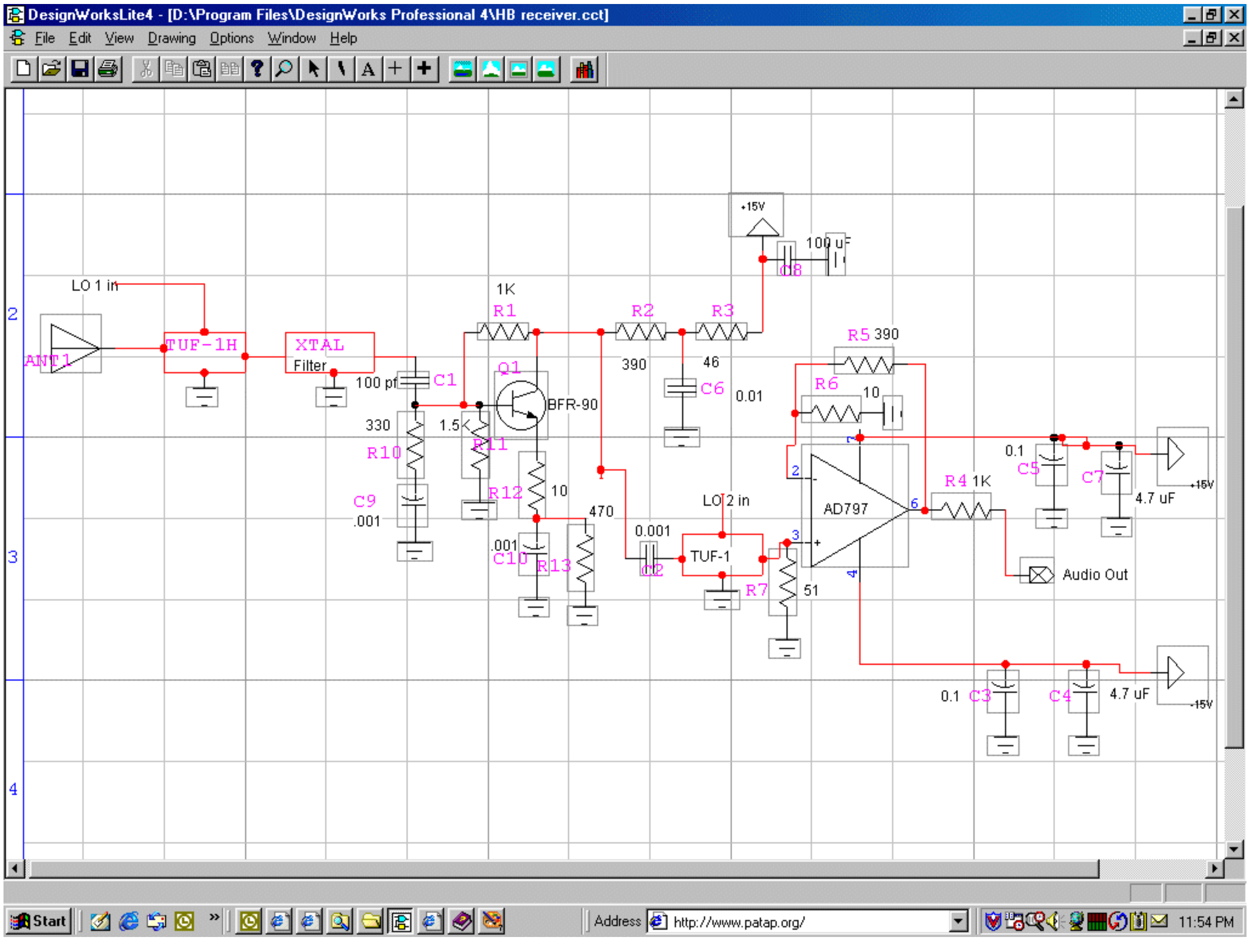

**Figure 3. This is a schematic of the front-end I use with the Linrad software receiver. It is a simple configuration, but works well for me. The combination of this front-end and Linrad outperforms the conventional receiver combinations I have tried.**

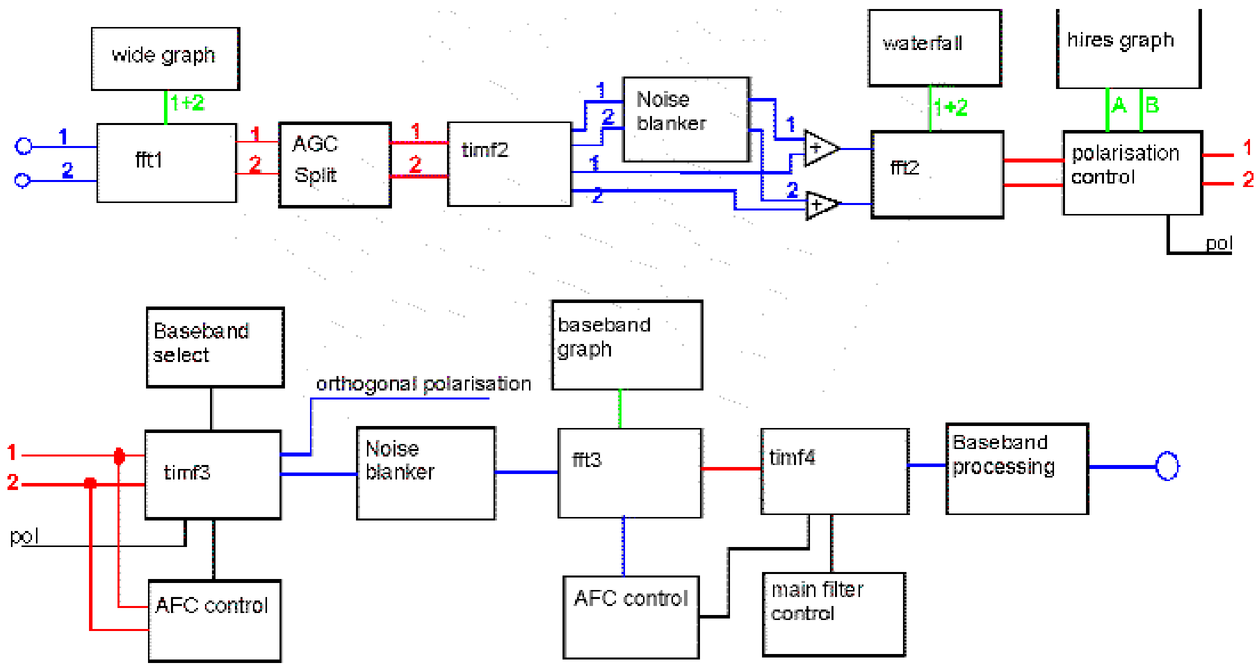


**Figure 4. This is Leif SM5BSZ's block diagram of the Linrad Linux PC Receiver. Blue signal paths are time domain signals and red signal paths are frequency domain signals. Green signal paths represent power vs. frequency displays, and black signal paths represent control pathways.**

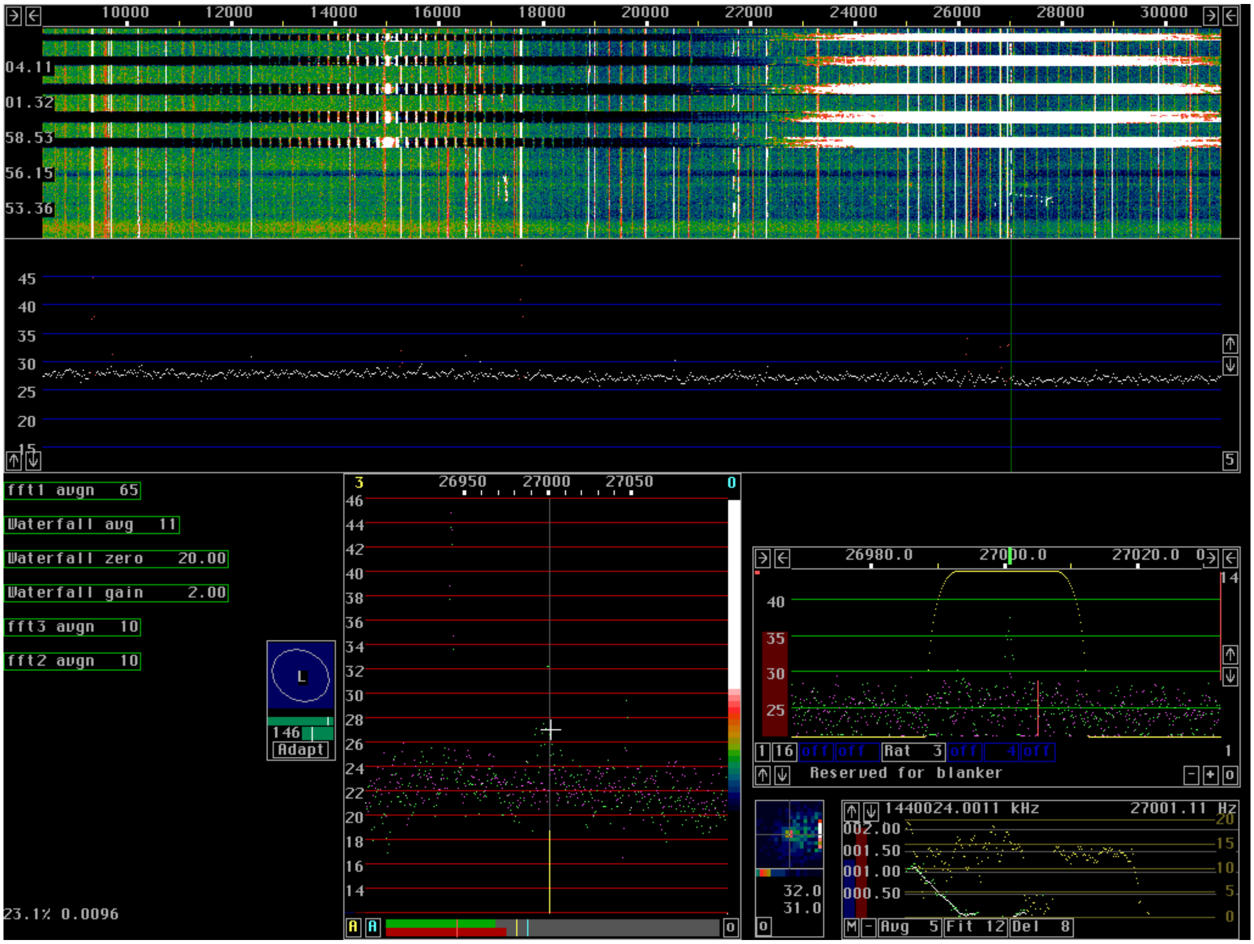


**Figure 5. Linrad screen taken during ARRL EME contest. KB8RQ is at 27000 Hz. You can see his signal as a series of vertical dashes beneath the green cursor on the upper full-width waterfall and main spectrum displays at the top of the screen, and above the vertical yellow cursor, aligned with the vertical grayish cursor on the high resolution display in the center of the screen below these displays. It is also visible centered within the yellow filter outline in the baseband display, to the right of the high resolution display bottom of the screen, above the AFC window. Further details of the display are given in the text.**

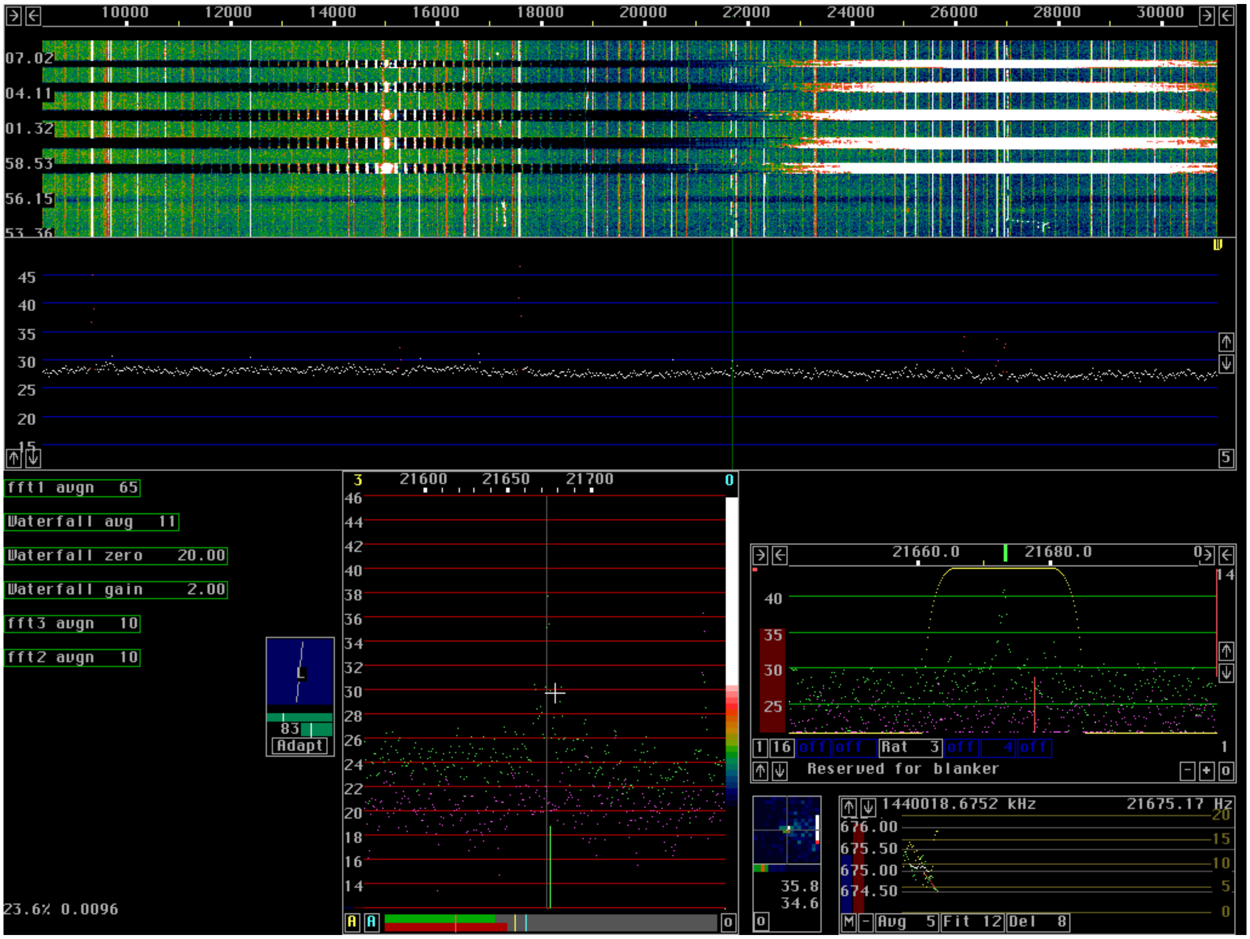

**Figure 6. Linrad screen receiving Dave W5UN's EME signal at 21672 Hz. You can see Dave's signal beneath the cursor in the waterfall, main spectrum, high resolution spectrum, and baseband spectrum displays. In the baseband spectrum display you can see his keying sidebands within the yellow filter curve. To the right of the high resolution display the polarity display shows that Linrad is automatically controlling the polarization angle of the received signals and that Dave's polarization as received is nearly vertical, at 83 degrees. Additional details are given in the text.**

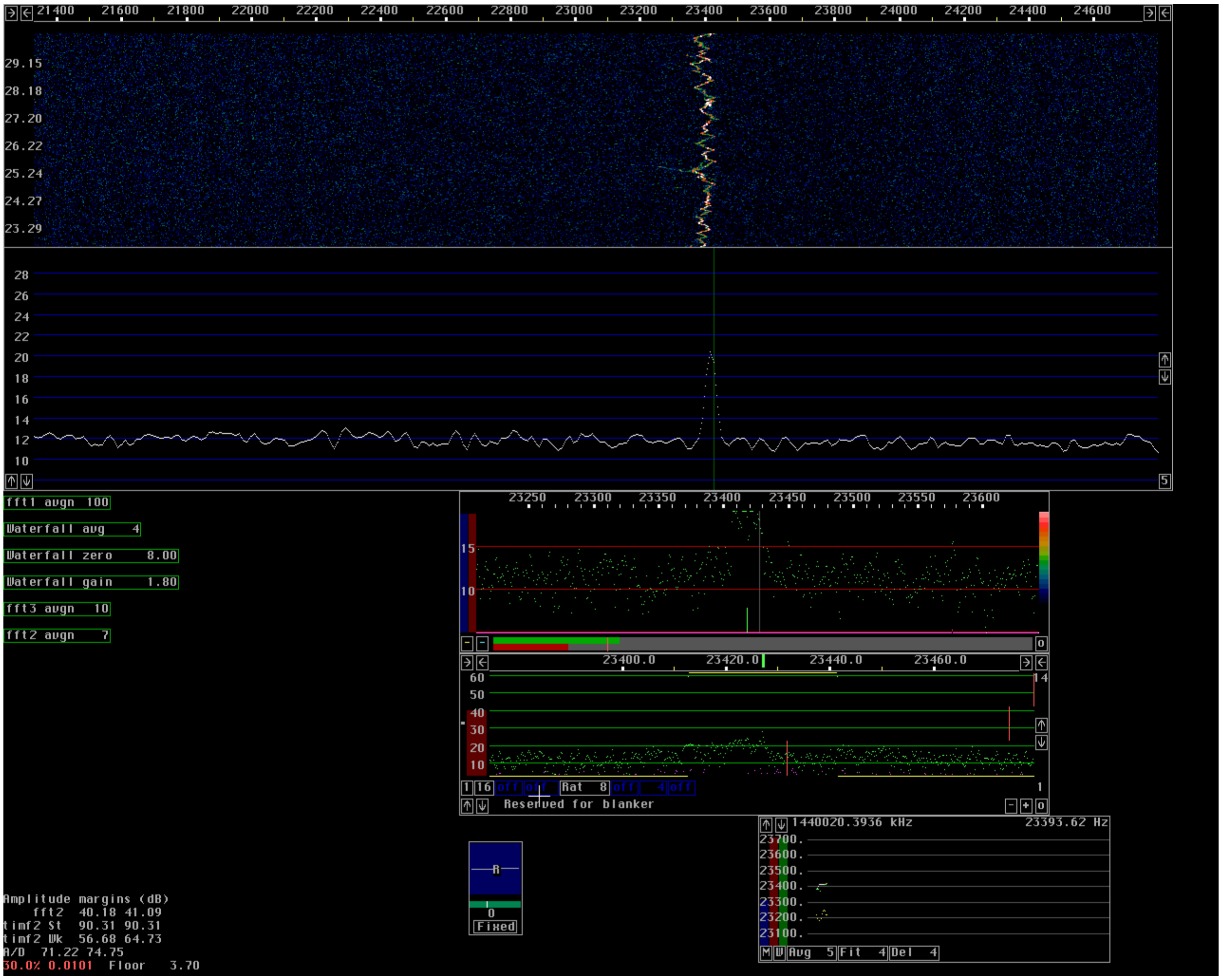


**Figure 7 shows a Linrad screen with the homebrew receiver and Linrad being used as the IF for a 2304 MHz receiver, monitoring the W3CCX 13cm beacon. The frequency instability of the beacon as received over this path is demonstrated nicely with the waterfall and main spectrum displays zoomed to show just 3 kHz of spectrum. There is a polarization angle of zero and only a green spectrum on the high resolution display because there is just one receive channel used on this band (horizontal polarization).**

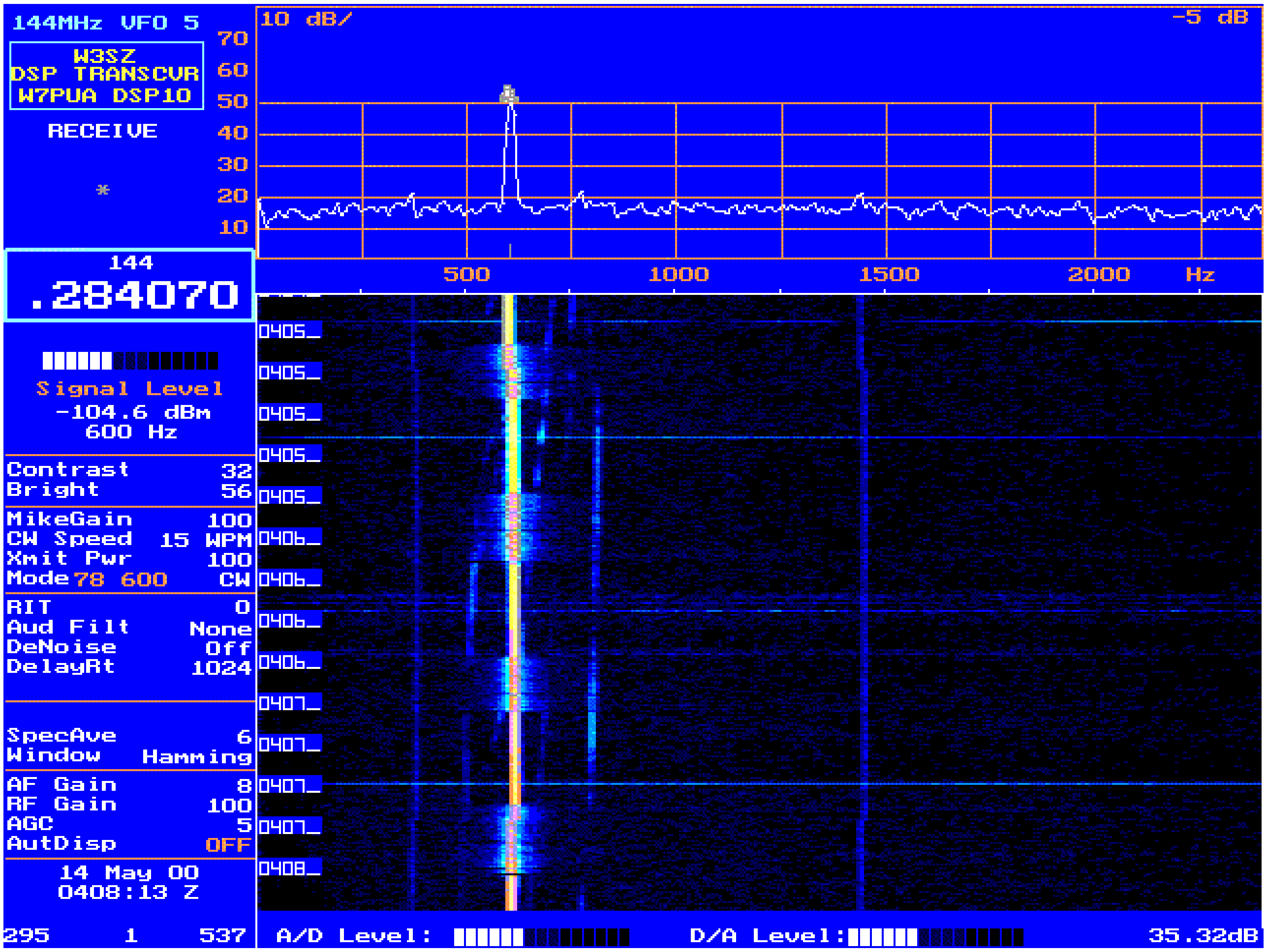


**Figure 8. DSP-10 screen showing the W3CCX 2 meter beacon. Across the top of the screen is the spectrum display. Labels on the spectrum include relative signal strength in dB (shown vertically on the left of the spectrum display), and frequency (shown horizontally along the bottom of the spectrum). Baseline level (-5 dB) is shown at the upper right corner of the display. The W3CCX signal peak is easily seen at 600 Hz. Below the spectrum display the waterfall shows a tracing of the beacon signal over time. To the left of the waterfall is the time axis label. Below the waterfall is a display showing the signal levels relative to saturation in the A/D and D/A converters. At the lower right corner of the display, adjacent to the D/A Level Meter, is the current average received power level. To the left of the spectrum display are displays of some DSP-10 operating parameters. We can see that we are in receive mode, that the VFO is set to 144.284070 MHz, and that the signal peak has a level of -104.6 dBm and is centered at 600 Hz. Next are indicators of the settings for display contrast and brightness, mic gain, CW speed, transmit power, and mode (CW). The mode display also tells us that the CW frequency offset is set to 600 Hz. Then we see that RIT, Audio Filter, and Denoiser are off, and that the binaural delay is set at 1024. Next we see that 6 signals are averaged for each spectrum display and waterfall line, and that a Hamming windowing function is being used. The values for audio gain, RF gain, AGC, and automatic display (essentially AGC for the spectrum display) are shown next. At the very bottom of the display on the left we see the date and time and a line containing 3 numbers showing the status of the serial port and the FFT scale factor.**

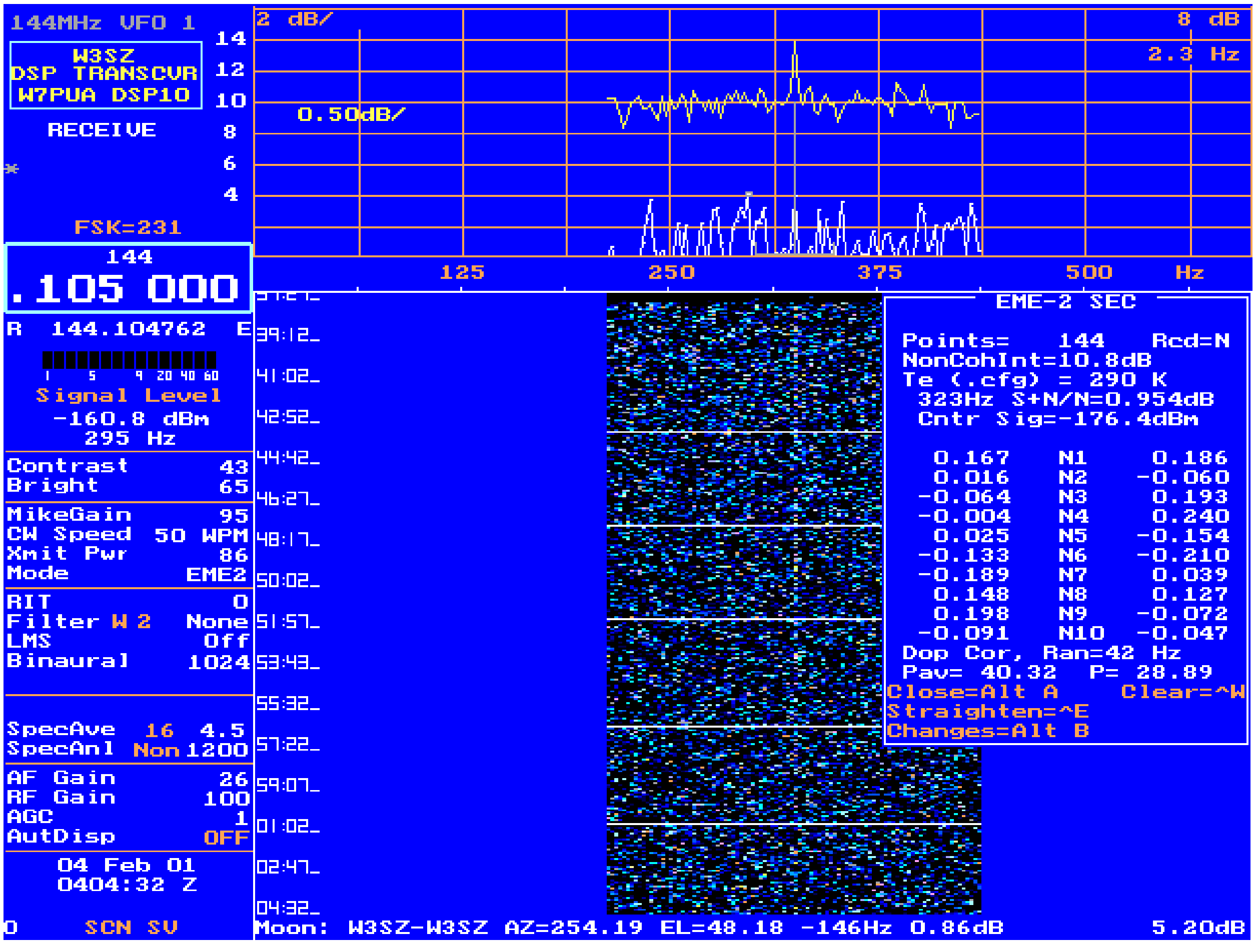


**Figure 9. This is a screen taken with DSP-10 running in mode EME-2. Transmit power is 150 watts. The yellow-line spectrum tracing at the top of the screen shows a clearly defined echo from this low power EME signal. No signal is apparent in the single-signal spectrum shown in white immediately below the averaged yellow-line spectrum. Labels on the frequency spectrum include relative signal strength in dB, vertically on the left of the spectrum and frequency horizontally along the bottom of the spectrum. Baseline level (8dB) and bin width in Hz (2.3) are shown at the upper right display corner. Below the spectrum display the waterfall shows intermittently a barely visible signal. To the left of the waterfall is the time axis label. To the right is the "Alt-A" parameter box. It shows that 144 points have been averaged and that this has improved signal-to-noise ratio by 10.8 dB. 290K is indicated to be the assumed noise temperature. The S+N/N ratio of the center bin is 0.954, and the bin is centered at 323 Hz. Center bin signal strength is estimated to be -176.4 dBm. Next the S+N/N ratios of the 10 bins above and below the center frequency bin are shown. We then see that EME Doppler Correction is turned on, as is random frequency spread, (43 Hz). To the left of the spectrum display we that we have set the VFO to 144.105.000 MHz, but that 144.104.762 is the actual received frequency due to the offsets associated with Doppler Correction and frequency stir. Setup parameters for the DSP-10 transmitter are then listed. These are labeled and self-explanatory. Below the waterfall the EME Doppler correction parameters are displayed.**

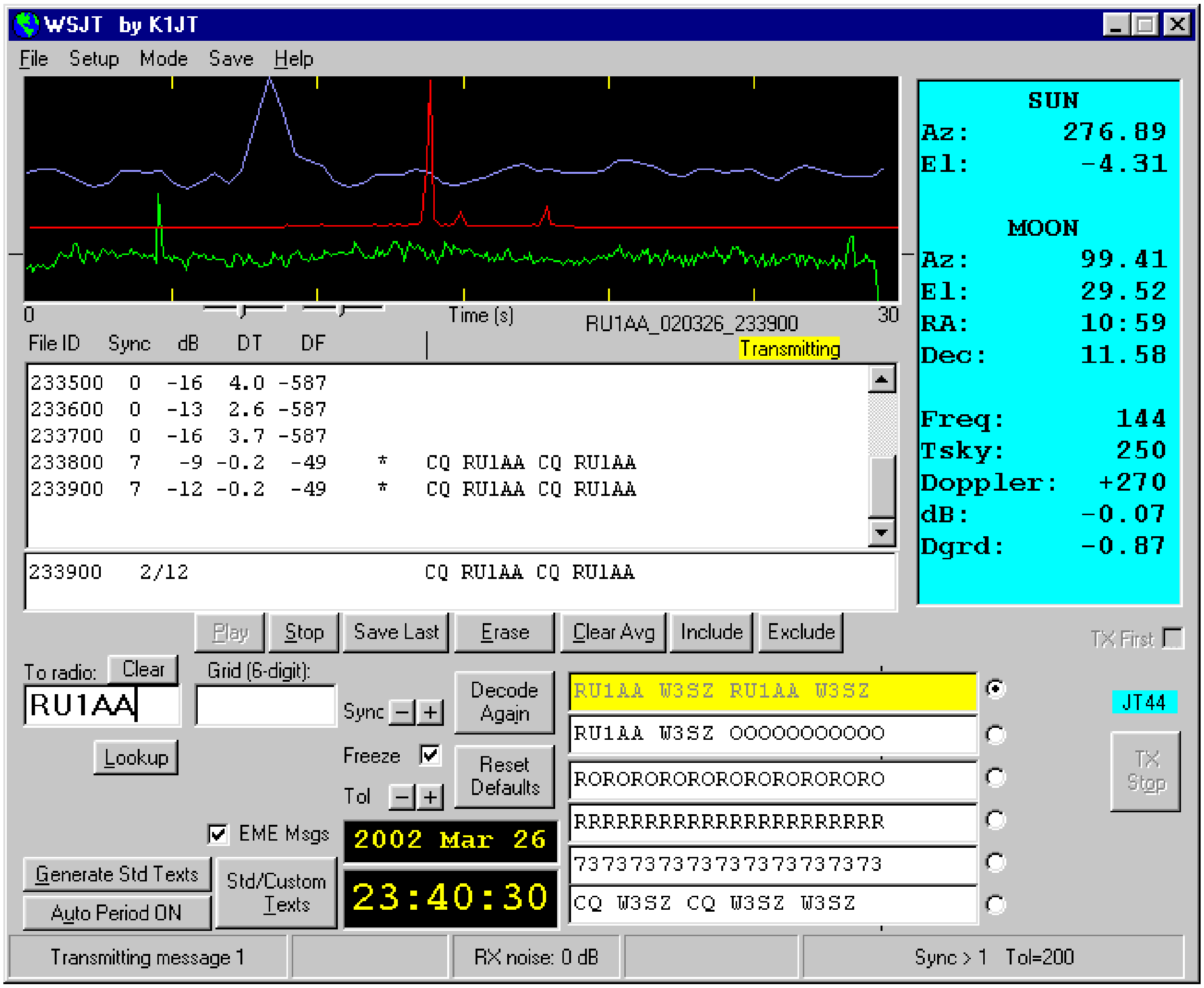


**Figure 10. JT44 screen of RU1AA being received on 144 MHz EME by W3SZ. You can see that there is immediately perfect copy of RU1AA by JT44. Unfortunately, RU1AA couldn't copy me. The display is very well designed. It consists of a waterfall graph at the upper left of the screen and a Sun/Moon/Sky data box at the upper right. Below this on the left is the decoded text box, and below this is the average text box. These boxes are where you see the perfect copy "CQ RU1AA CQ RU1AA". Below this are various function and parameter boxes. Download the WSJT User's Guide and Reference Manual from K1JT's website for further information (http://pulsar.princeton.edu/%7Ejoe/K1JT/).**