A Forum for Communications Experimenters

WØPCE updates his Fully Automated DDS Sweep Generator Measurement System from the Nov/Dec 2008 issue of **QEX**. This revision updates the circuit and layout, and describes how to perform precision measurements.

HF/50MHz ALL MODE TRANSCEIVER TS-590

Nothing But Performance

The TS-590S

Kenwood has essentially redefined HF performance with the TS-590S compact HF transceiver. The TS-590S RX section sports IMD (intermodulation distortion) characteristics that are on par with those "top of the line" transceivers, not to mention having the best dynamic range in its class when handling unwanted, adjacent off-frequency signals.*

- **HF-50MHz 100W**
- **Digital IF Filters**
- **Built-in Antenna Tuner**

- **Advanced DSP from the IF stage forward**
- **500Hz and 2.7KHz roofing filters included**
- **Heavy duty TX section**

- **2 Color LCD**

QEX

In order to ensure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.

September/October 2012

About the Cover

Dr. Sam Green, WØPCE, describes an update to his Fully Automated DDS Sweep Generator Measurement System from the Nov/Dec 2008 issue of *QEX*. In this revision, he updates the circuit and layout, and describes how to perform precision measurements. New control software automates the measurement process and saves the results, so they can be plotted by a spreadsheet program like Microsoft *Excel*.

In This Issue

Features

Index of Advertisers

Larry Wolfgang, WR1B

lwolfgang@arrl.org

Empirical Outlook

N1BKE

By now most of you have heard about the tragic death of our colleague and *QST* Managing Editor, Joel Kleinman, N1BKE, as a result of a house fire on August 18. Much has been written about the role Joel played in the production of *QST* every month. Any time you have worked closely with someone for 32 years, any kind of parting will have an impact. When someone retires, there is an opportunity to wish them well, and to reminisce with them about "old times." You might even cross paths with them from time to time in the future. You are sure to miss a friend that you worked with on a daily basis. Several times during my time at ARRL HQ we have had staff members pass away because of an illness, and that is much harder. Often there is no opportunity to say good bye, and we won't cross paths with them again on this Earth.

The tragic circumstances of Joel's death have affected everyone at HQ in a much deeper way. We won't soon get over his sudden death, but we will remember his dedication and passion for *QST* and ARRL.

When I began my career as a technical editor at ARRL I had a lot to learn about English grammar and that strange beast we call "*QST* Style." In his position as editorial supervisor at that time, Joel had to deal with my lack of knowledge about many of the nuances of editing and style. I was concerned about making sure the copy was technically correct, but I had much to learn about language. Eventually I probably started to improve, and we often made a game of me trying to turn in a "perfect" manuscript, and Joel having to find something wrong. His job was much easier than mine in that regard!

Over the years, Joel became my direct supervisor a couple of times. Joel was an extremely talented editor, and I continued to try to learn from him. Several times we also occupied neighboring office spaces, including for the past 6 years or so. We always exchanged a cheerful "Good morning" and "Good night" even if we didn't always see each other come out of our offices during the day. Now, suddenly, that won't happen again. I'll miss it for a long time, I'm sure.

And now on a brighter note…

Pushed and Prodded to Try New Things

I have always enjoyed building electronics projects. Whether it was my Knight Kit T-60 Novice transmitter, a Heathkit HW-5400 transceiver or Elecraft K2 for Product Review, or even just a simple station accessory, the joy of building a project and using it on the air always gets me excited. Lately, I haven't found much time to build, though. A few years ago I entered the realm of surface mount components when I built a NUE-PSK modem. That was a big challenge, but the sense of accomplishment when I managed to fix a problem with the way I soldered a multi-pin IC (with leads on all four sides) onto the circuit board was huge.

I have several other projects waiting to be built, and all of them involve at least some SMD work. I even bought a tube of solder paste and have decided I will try that and a heat embossing gun from my wife's craft room for at least part of a project. The ability to see those tiny parts, especially the ones with many pins along all four sides, is still the biggest challenge.

Even with a magnifying lens, it can be difficult to be sure the pins are all properly aligned with the circuit board pads. I've previously talked about using my digital camera to take a photo, and transferring it to my computer for enlarged display on the monitor as one way to see more clearly. A *QST* article from a year or so ago pointed to a high definition web cam as a way to display a live image on the monitor. I've started to explore that, but have not put solder paste to board yet.

Now, I've been asked to give an introduction to surface mount construction talk at the upcoming ARRL/TAPR Digital Communications Conference. So, it's time to take some action, collect some new photos, and update the talk I gave at DCC in 2009. I have a few weeks to accomplish that, so there is a certain urgency to prepare for DCC. I hope to see many of you in Atlanta, even if it isn't to take in my Intro Session.

Mark Bauman, P.E., KB7GF

1910 Sunflower Court, College Place, WA 99324: **kb7gf@arrl.net**

Introducing the Shared Apex Loop Array

Here is a wideband receiving antenna that delivers good things in a small package.

Imagine a wire directional receiving antenna that provides solid front-to-back and front-to-side low elevation angle response over a continuous frequency range of 1 to at least 14 MHz and that is about 30 feet long. Sounds too good to be true? Other receiving antennas like Beverages take up more real estate to achieve similar receiving patterns. Other terminated antennas like the K9AY loop are limited by an intrinsic cardioid response with the associated poor front-to-side ratio and limited frequency coverage. The antenna itself is easy to construct; it uses two identical loops with a passive ferrite coupling. Additionally, the shape of the received pattern and sensitivity as well as the backward elevation null can be adjusted by choosing an appropriate coupling location in concert with an appropriate delay line length.

Before you start thinking that we are violating some law of physics, the Shared Apex Loop Array described here is a physical reality, and I have had fun working on it the last few years. Physics, though, can be a cruel science, and there is a catch: The forward gain of the antenna is a function of the frequency, reaching a maximum as the distance between the loop feed points approach one-quarter wavelength and relentlessly diminishes as the frequency is lowered. Fortunately, this problem can be managed to some degree by using a suitable low-noise amplifier to overcome the negative forward gain.

Interestingly, the shape of the antenna pattern actually *improves* as the frequency is lowered, maintaining its highly desirable directional characteristics. The nature of the antenna makes it excel in situations where there is adequate signal strength, but the desired signal must compete with undesired signals on the same frequency that are arriving from other directions. A common example is on 80 meters during the spring,
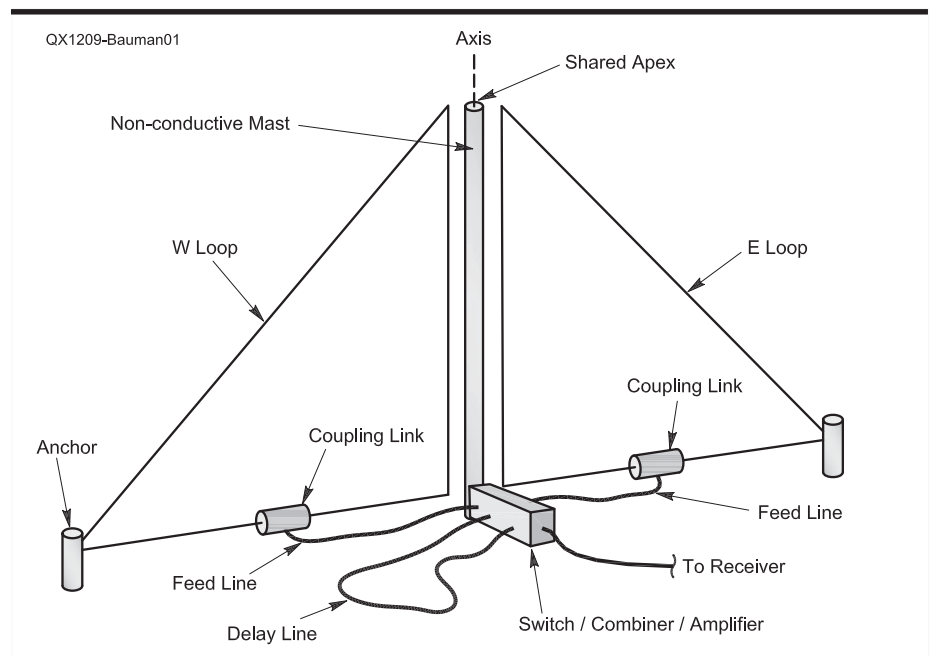
Figure 1 — This drawing shows a two element shared apex loop antenna.

summer, and fall when convective noise dominates. The antenna can be pointed toward the desired signals and (hopefully) away from the convective noise. It is also very useful as a spotting antenna for rapidly locating the direction of a signal. The combination of interference fighting and small size make this an ideal receiving antenna for rag-chewers and contesters alike. DXers will find a lot to like, but will want to increase the size of the loops to overcome the forward gain limitation so they can scoop up the really weak signals. It is also very effective at eliminating local noise sources — provided of course that they are coming from directions that are different from the desired signals.

The Shared Apex Loop antenna com-

bines the virtues of fractional wavelength magnetic loops with their inherent bi-directionality and true-time-delay end fire arrays, where a broad frequency response is achieved by combining signals from two identical antennas in time-delayed relation. The basic concept of receiving signals on two identical antennas and subtractively combining has been in the public domain for quite some time and is described in detail in US Patent 3,396,398 awarded to J. H. Dunlavy, Jr. in 1968.

The Shared Apex Loop antenna can be constructed in a simple ground-mounted form, where two identical loops are positioned in a common vertical plane about an axis as illustrated in Figure 1. Each loop is formed in the shape of a right triangle, and

QX1209-Bauman02
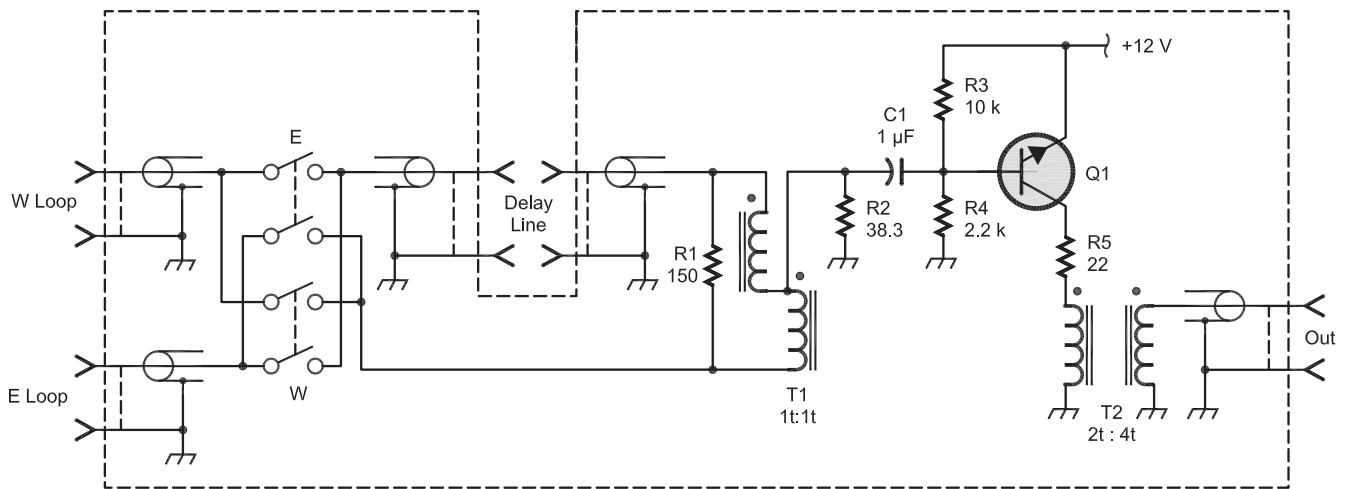
Figure 2 — Here is the schematic diagram of a simple switch/combiner/amplifier. Note that the delay line is connected outside of the waterproof circuit box, as depicted in Figure 1. No control circuitry for the direction switches is shown here.

each is positioned in mirrored relation about an axis sharing an apex at the top. A reasonable size for the loops is fifteen feet wide, fourteen feet high for a loop perimeter of a little less than 50 feet. A non-conductive mast is aligned along the axis and serves to support and separate the vertical leg of each loop. In practice, a separation of at least one inch is appropriate. Each loop is held in tension by an anchor at a height at least six inches from the ground.

Signals captured by each loop are transferred to a feed line using a coupling link provided by a set of ferrite cores forming a current transformer.[1] The feed lines are connected to the coupling link so that the signals from each loop are opposite in phase. The location of each coupler relative to the axis is designated as the feed point, and is an important parameter that will be discussed later in this article.

The feed lines that connect to each coupling link must be identical in length and character. In my experiments, I have used both coax and balanced lines for this task. My current preference is for balanced lines because they weigh less and are easier to manage, but they require a balun to convert to single-ended signals before running through the delay line.

Each of the feed lines connects to a switch/combiner/amplifier module and is shown schematically in Figure 2. Here the signals are routed or switched either directly, or via a delay line to a combiner, where they are subsequently amplified. To select the East direction, the E switch is closed, routing

signals from the West loop through the delay line to a combiner, while signals from the East loop are sent directly to the combiner without being delayed. To select the West direction, close the W switch to transfer signals from the West loop directly to the combiner, while signals from the East loop transit through the delay line and then to the combiner.

The combiner must provide two isolated ports over the desired frequency range, to ensure that the impedance from one loop does not significantly impact the impedance of the other loop. In my testing, I have tried both active and passive combiners, but it is hard to argue with the simplicity of the Magic-Tee combiner represented in Figure 2 by the combination of R1 and T1.

Further, the input impedance of the combiner must match the characteristic impedance of the delay line over the operating frequency range, to provide consistent time delays. This requirement hoists a burden on the input impedance of an amplifier (which is connected to the output of the combiner) as well as an isolation requirement to ensure that amplifier loading does not alter the input impedance. The circuit shown in Figure 2 meets the input impedance requirement by using R2 as a termination resistor and meets the isolation requirement by employing an emitter follower amplifier. The output of the amplifier connects to a transmission line that delivers signals to a receiver. While fully operational using loop sizes discussed in the article, the simplicity of the circuit in Figure 2 betrays us, as it is quite noisy and

enjoyable only with larger loops that deliver enough signal to overcome the low gain and amplifier noise.

The time difference between signals arriving at the combiner, as provided by the delay line, is an important parameter, predicting the behavior of the antenna. In principle, the time difference is selected so that the signals arriving from a direction opposite the favored direction and induced in the loop that is delayed so that the signals induced in the forward loop when combined provide a significant signal cancellation. This method enables wide bandwidth operation at the expense of forward gain when the spacing of the coupling links remains less than about one quarter wavelength.

So, how does the antenna perform? First, let's take a look at the results of a *4NEC2* model using the dimensions discussed earlier. Then, we will vary the feed point location and combined time difference to show how they affect the antenna receiving patterns.

To model the antenna, I first defined a number of symbols, as shown in Figure 3. The first five symbols define the size and location of the loops relative to each other. Next, the frequency is specified and associated with the symbol "freq." The "delay" symbol is used along with the frequency to derive a phase relation that is used to drive the model sources. The feed point location is indicated by the symbol "Tap" and is relative to the number of modeling segments, which, for this model, is 37. In this example, the feed point is approximately equal to the width of
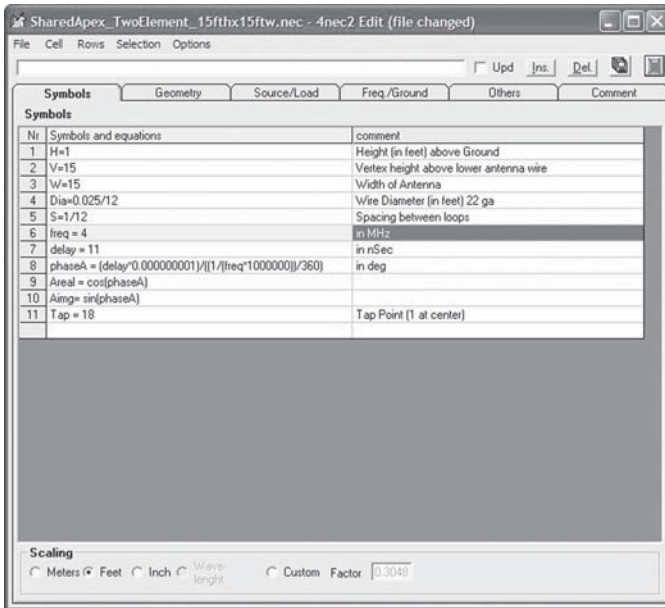
Figure 3 — This screen capture shows the Symbol Definition tab of the *4NEC2* modeling program.
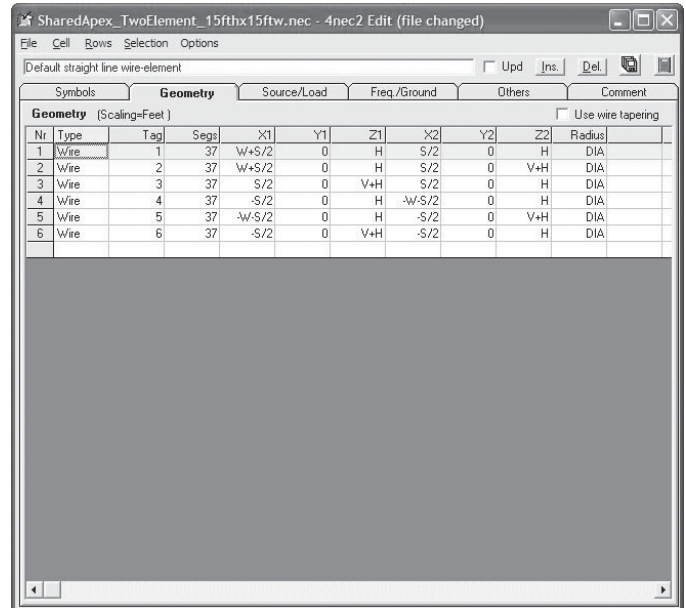
Figure 4 — Here is the Geometry tab of the *4NEC2* modeling program.
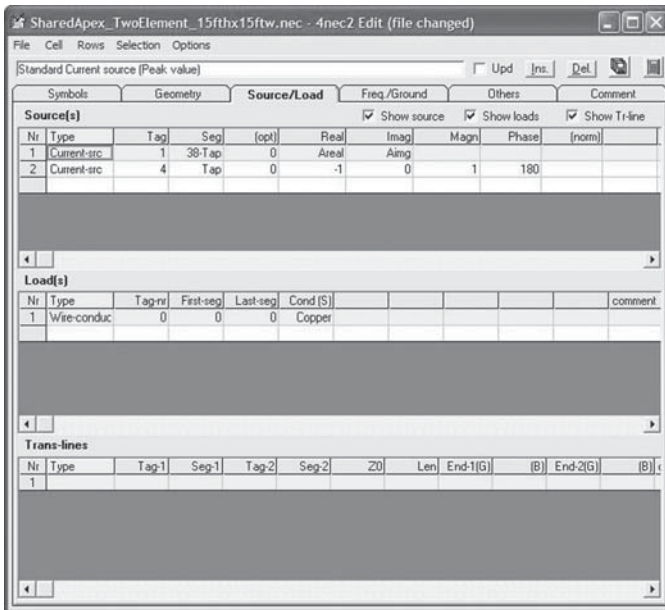
Figure 5 — A screen capture of the Source tab of the *4NEC2* modeling program.
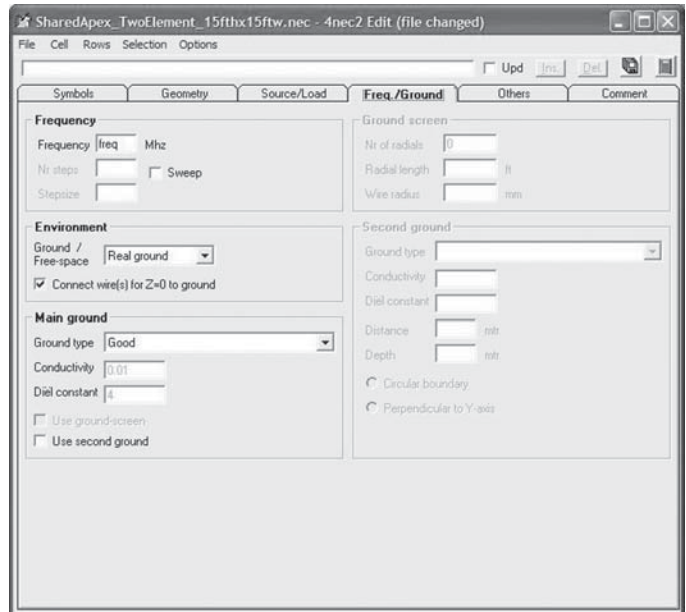
Figure 6 — This is the Frequency tab of the *4NEC2* modeling program.

the antenna multiplied by the Tap and divided by the number of modeling segments.

Figures 4, 5 and 6 show the model geometry, sources, frequency and ground condition entry sheets. The geometry of the antenna is defined using parameters specified earlier in the text. Selecting "current sources" seemed to be the best match for the sources, since the loops are magnetic, and coupled using current transformers. The real and imaginary components entered in Figure 5 are automatically derived from the parameters defined in the "symbols" screen or tab (Figure 3). The frequency is provided by the symbol

"freq" and its value is inserted as shown in Figure 6. Good ground conditions were selected for the model for all of the plots shown.

One interesting aspect of modeling includes comparing the output of the model to the physical reality and on-the-air results. The NEC2 engine is speculated to have some short-comings related to modeling small and intermediate-sized magnetic loops. From my experience so far, I would agree that there are limitations. For example, when modeling a triangularly shaped single, electrically small loop fed with a current source, the

"Total Gain" plot in the horizontal plane is omnidirectional. This is contrary to experience when the loop is properly balanced. The "Vertical Gain" plot shows the expected bidirectional response, however.

When modeling the Shared Apex Loop, there are differences between the "Total Gain" horizontal plot shown in Figure 7 and the "Vertical Gain" horizontal plot shown in Figure 8. From on-the-air testing, I would say that the "Total Gain" results are overly pessimistic, with real-world operation often exceeding their predictions, while the real-world operation approaches the "Vertical

Figure 7 — Here is an example of the "Total Gain" plot provided by the model.

QX1209-Bauman07
0°
0 dB
Total Gain: −17 dBi
Elevation: 35
Frequency: 4 MHz

Figure 8 — This graph shows the "Vertical Gain" plot provided by the model.

QX1209-Bauman08
Vert Gain: −17 dBi
Elevation: 35
Frequency: 4 MHz

Hor Gain: −1.8 dBi
Frquency: 18 MHz

——— SA2el_18MHz out
– – – SA2el_14MHz out
········ SA2el_8MHz out
–·–· SA2el_2MHz out
··–··– SA2el_4MHz out

QX1209-Bauman09

Figure 9 — Here is the horizontal response at 2, 4, 8, 14 and 18 MHz.

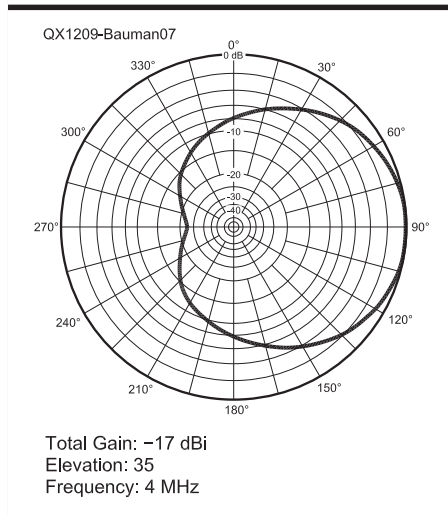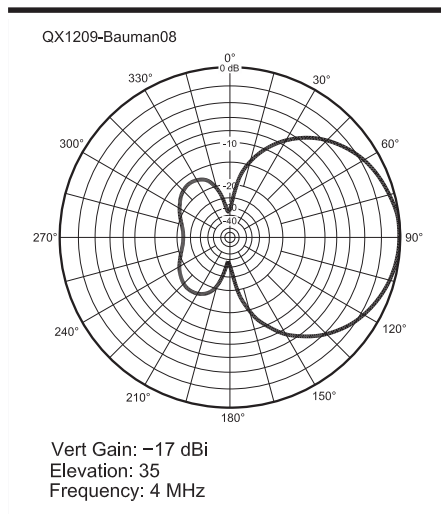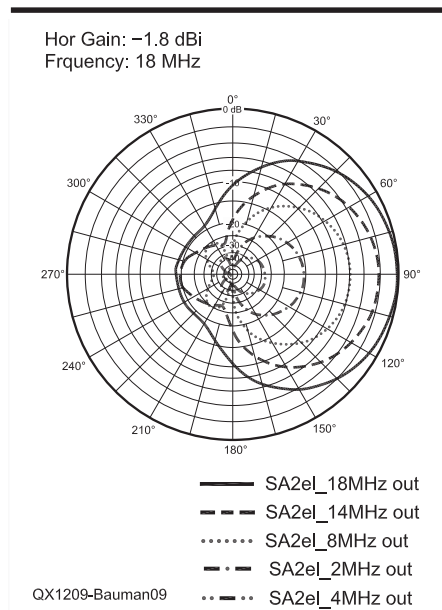Gain" plots especially for signals arriving from the elevations represented by the plots and across the frequency range. For this reason, we will focus on the vertical-gain plots for this article.

The horizontal response provided by the *4NEC2* model is shown in Figure 9 for several frequencies using a combined time difference of 8 nanoseconds (ns). The innermost pattern is the horizontal response at 2 MHz; the next pattern is at 4 MHz and then moving progressively outward, the next patterns are at 8 MHz, 14 MHz and the outermost pattern is at 18 MHz. Notice that the relative shape is largely preserved for the 2, 4, and 8 MHz runs. The 14 and 18 MHz patterns maintain directivity but lose the pristine pattern seen for the lower frequencies. The relative size of each pattern shows the overall gain of the array at the respective frequencies. The decrease in forward gain between 2 MHz and 18 MHz is nearly 30 dB!

Referring to Figure 10, the forward gain, front-to-side ratio, and front-to-back ratio for the array and the forward gain of an individual loop are each plotted from 1 to 28 MHz. The loop gain and array gain both peak at 20 MHz, at about unity. Below this frequency, the array gain diverges from the loop gain as the frequency is lowered. The phase angle difference between the signals as they are combined is responsible for this divergence. The graph shows that at 1 MHz, the loop gain is –25 dBi, while the array gain is –45 dBi. The phase difference at 1 MHz is (180° – 3°) or 177°, which is only 3° from total signal cancellation. That accounts for the 20 dB reduction.

The front-to-back ratio manages to stay above 20 dB for frequencies between 1 and 9 MHz, and between 11 and 14 MHz. The Front-to-Side ratio stays above 20 dB for fre-

QX1209-Bauman10

——— Font-to-Side   ·········· Loop Gain
– – – Front-to-Back   –·–·– Array Gain
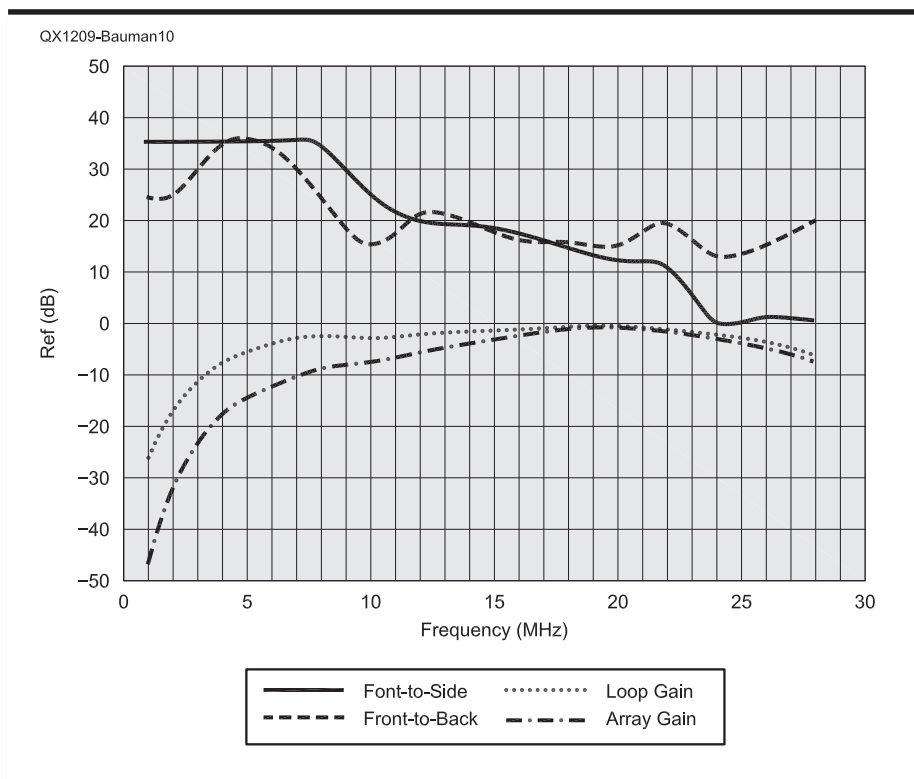
Figure 10 — This graph is a comparison of the gain, front-to-back ratio and front-to-side ratio of the antenna.

quencies between 1 and 14 MHz.

An intriguing aspect of the Shared Apex Loop Array is the relationship between the feed point location for a given loop and the pattern that it produces. For example, referring to Figures 11A-H, a feed point of 67 inches from the center post and a combined time difference of 3 ns yields the hori-

zontal pattern shown in Figure 11A. Moving each feed point out 10 inches to a location of 77 inches and increasing the combined time difference to 6.5 ns provides a pattern with improved front-to-back and front-to-side ratio as shown in the pattern of Figure 11B. A further improvement in front-to-side ratio is achieved by moving each feed
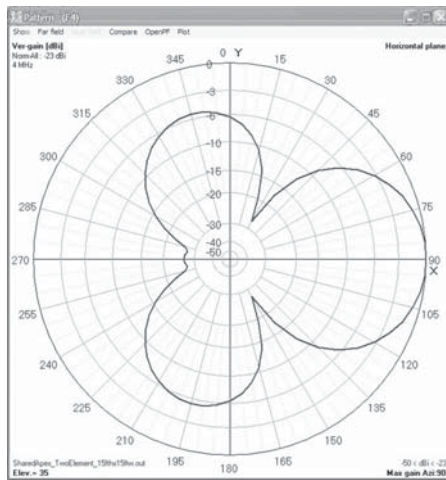
(A)
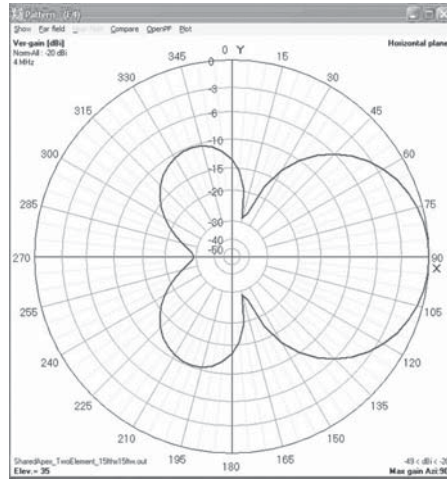
(B)
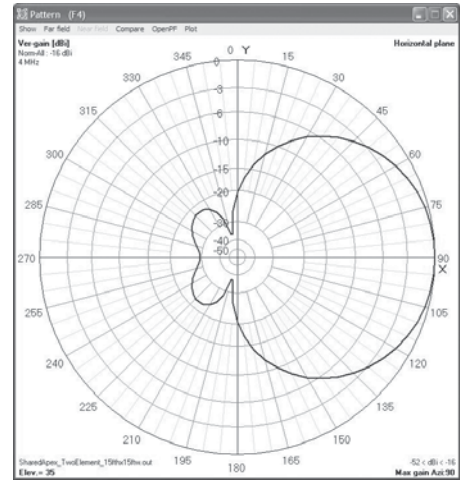
(C)
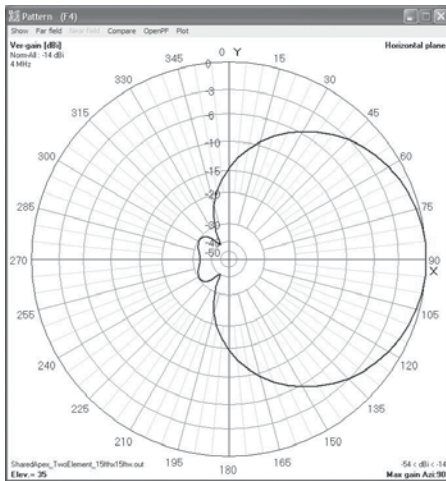
(D)
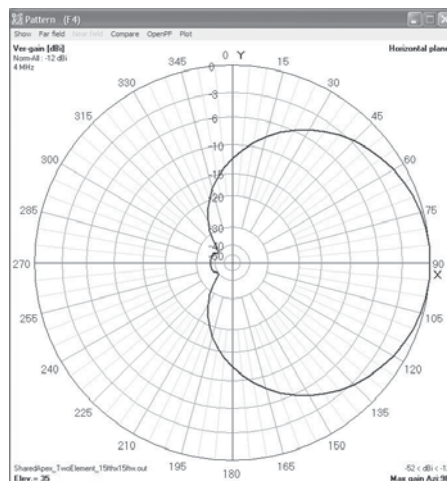
(E)
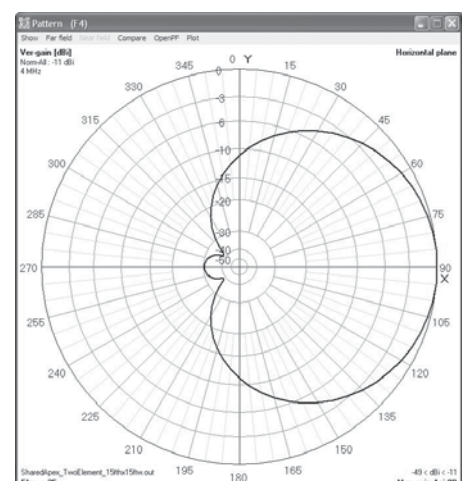
(F)
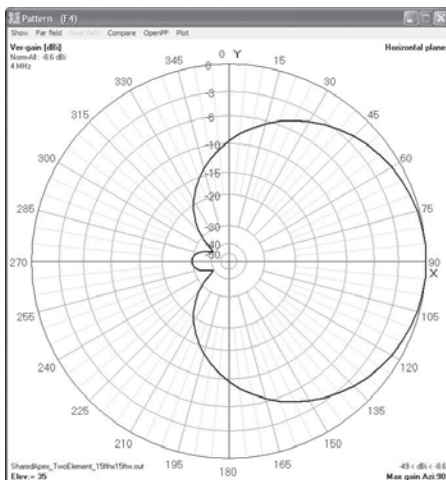
(G)
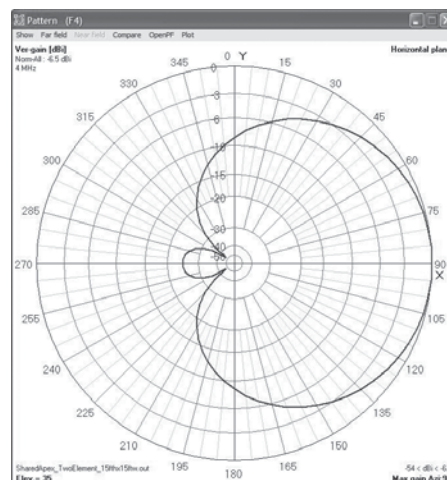
(H)

Figure 11 — These plots show the horizontal response for various feed points at 4 MHz. Part A is for a feed point of 67 inches from the center post and a combined time difference of 3 ns. Part B has the feed point at 77 inches with a delay of 6.5 ns. For Part C, the feed point moves out to 86 inches, with a delay of 11 ns. Part D has the feed point at 96 inches, with a delay of 14 ns. Part E represents a feed point distance of 105 inches and a combined time difference of 17 ns. For Parts F and G this trend continues, until we reach a feed point near the end of the loop at 162 inches, with a delay of 40 ns and a nearly cardioid pattern in Figure 11H.

point to 86 inches and providing a combined time difference of 11 ns as shown in Figure 11C. Adding another 10 inches, so that each feed point is at a location of 96 inches and providing a combined time difference of 14 ns improves the front-to-back ratio, and

increases the forward gain at the expense of a slightly degraded front-to-side ratio as shown in Figure 11D.

The pattern shown in Figure 11E represents a feed point distance of 105 inches and a combined time difference of 17 ns,

and improves the forward gain at a further expense of the front-to-side ratio. This trend continues as the feed point distance is increased in Figures 11F and 11G until we reach near the end of the loop at 162 inches, with a delay of 40 ns and a nearly cardioid
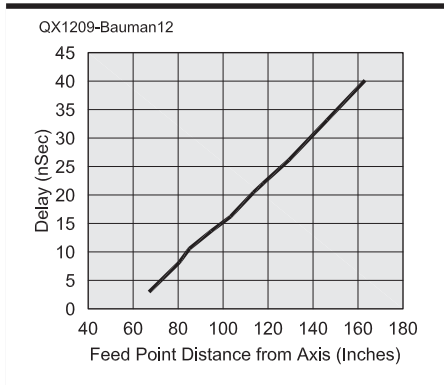
Figure 12 — This graph shows the delay as a function of feed point distance at 4 MHz.
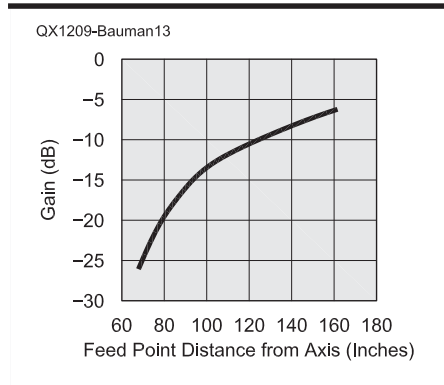
Figure 13 — Here we see the gain as a function of feed point distance at 4 MHz.
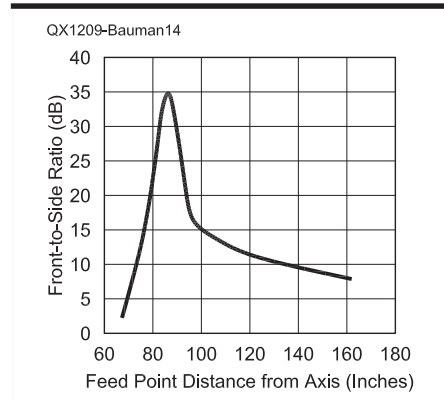
Figure 14 — This plot shows the front-to-side ratio as a function of feed point distance at 4 MHz.

Figure 15 — Here is the beamwidth as a function of feed point distance at 4 MHz.

pattern in Figure 11H. All of these patterns can be achieved simply by changing the combined time difference and moving the feed point. It should also be noted that the maximum frequency for the array is a function of the time delay (t_{diff}). As a rule of thumb, the maximum effective frequency can be approximated by the following relation: $f_{max} \approx (1 / (5 \times (t_{diff}))$.

In Figure 12, a relationship between the feed point distance and time difference for the array described in this article is plotted. It is evident from the graph that the combined time difference must increase as the feed point distance is increased. For this array, the combined time difference can be approximated by the relationship: $t_{diff} \approx [(\text{distance} - 60) / (2.7)]$ where distance is in inches and t_{diff} is in nanoseconds.

As noted earlier, the forward gain of the array can be increased by moving the feed point in an outward manner from the axis and this relationship is shown in Figure 13. At 4 MHz, a difference of nearly 20 dB is realized by moving the feed point less than 100 inches.

As shown earlier, there is a trade-off though between forward gain and front-to-side ratio. This trade-off is evident by a careful inspection of Figures 13 and 14. For this array, the peak front-to-side ratio of 35 dB occurs at a distance of nearly 85 inches.

The beamwidth for the array as a function of the feed point distance is shown in Figure 15, and is least when the feed point distance is the shortest. At the feed point distance of 85 inches, the 3 dB beamwidth is nearly 80°.

In a receiving antenna, the vertical null off the back of the antenna is of special significance. Interestingly, the angle of the null is a function of the time difference, wherein the vertical null angle increases relative to the horizon as the time difference is decreased. To illustrate this influence, consider the vertical patterns presented in Figure 16A-H for

selected time difference values for a feed point of 85 inches. Here, the lowest null angle of –90° (0° relative to the horizon) is achieved with a delay of 11 ns, as shown in Figure 16H. By decreasing the delay by 1 ns to 10 ns, the null angle changes to –70° (20° above the horizon) as shown in Figure 16G. Higher elevation null angles can be selected for improving rejection of high angle interference off of the back of the antenna by selecting a delay of 7 ns and shown in Figure 16D. Even higher null angles can be realized as shown in Figure 16A using a 4 ns delay line. At this setting, however, the low angle front-to-back ratio begins to suffer. This relationship between the delay time difference and backward null angle is summarized graphically in Figure 17.

At present, my preferred receiving antenna uses two Shared Apex arrays, each smaller than that described in this article and positioned at right angles to one another and sharing a single mast. Adding the second array modifies the response somewhat, so the delay line and feed points are adjusted according to an updated model that accounts for these additions. I've also built a remote switching unit and controller that allows the array to be remotely switched in four primary directions (along each of the loops), and four hybrid directions (by connecting loops together) to provide eight total directions.

When scanning the bands, it is common to realize 15 to 25 dB front-to-back ratios and 10 to 20 dB front-to-side ratios. Deeper nulls are also observed, but are not as common. The array is great for pinpointing and reducing local interference. The apparent sensitivity of the antenna described is largely bounded at frequencies below 5 MHz by the noise figure of the amplifier used in the switch/combiner/amplifier. Using a terminated cascode amplifier and the antenna as described, I can easily hear WBBM Chicago on 780 kHz in the winter, which is 2000 miles to the east of my location. At noon, I can hear
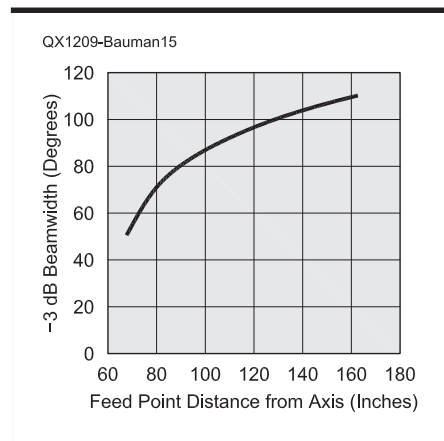
KEX in Portland on 1190 kHz, which is 240 miles to the west.

On 160 meters, east coast stations are easily heard during the evenings from my SE Washington state location, and I can occasionally hear JAs in the morning. On 80 meters, VKs and JAs are common in the morning, and occasionally I can hear stations from Europe and South Africa. Daytime regional nets are easily heard in the winter. Long path signals in the morning are very difficult to hear on 80 meters, although AIR from Chennai, India is easily heard over the long path on 4920 kHz during the early winter months. The front-to-back and front-to-side ratios between 500 kHz and 5 MHz are often greater than 20 dB.

Frequencies between 5 MHz and 18 MHz provide good performance also, although the front-to-back and front-to-side ratios are somewhat less at the upper end of the frequency range for sky wave signals. Local interference signals show sharp front-to-side and front-to-back ratios over the entire operational bandwidth.

Some parts of the antenna are novel, and I have filed a patent on these. I encourage ama-
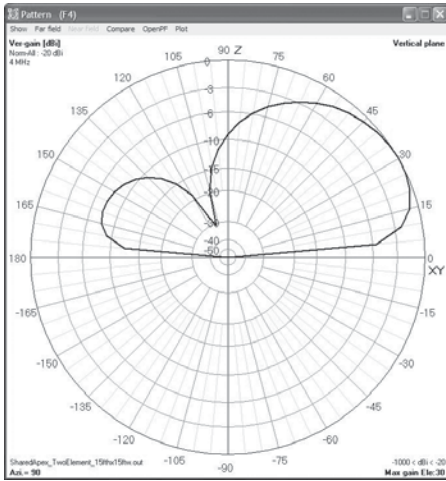
(A)
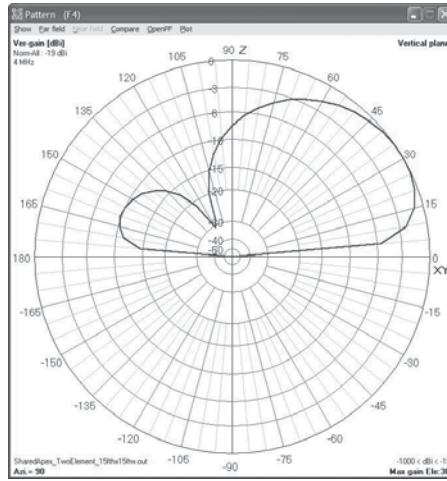
(B)
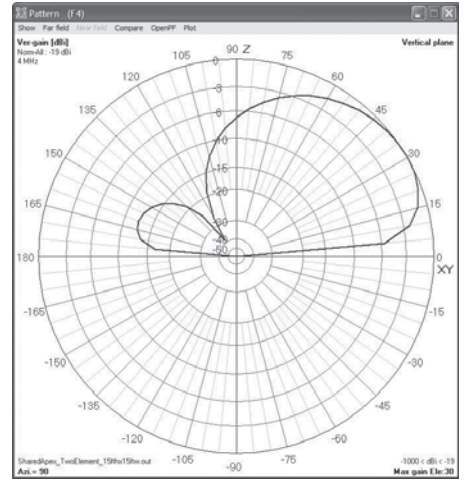
(C)
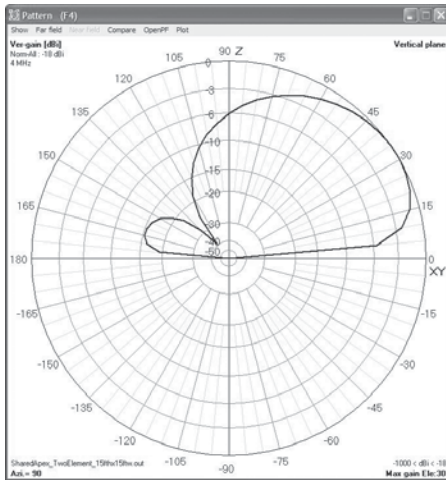
(D)
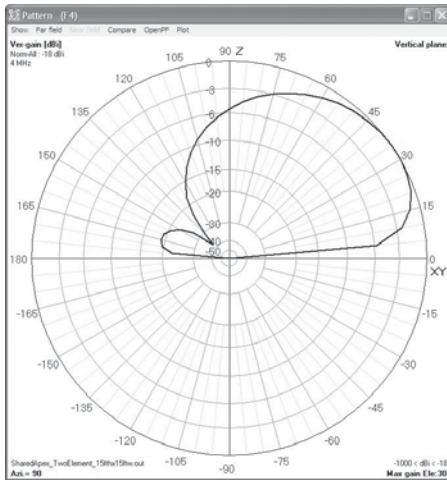
(E)
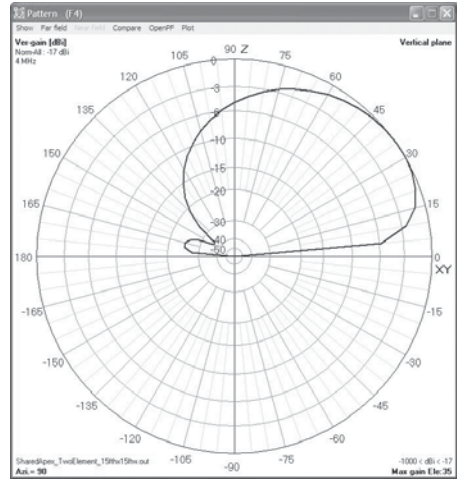
(F)
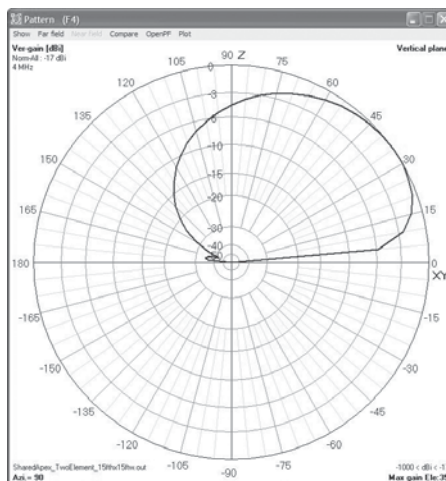
(G)
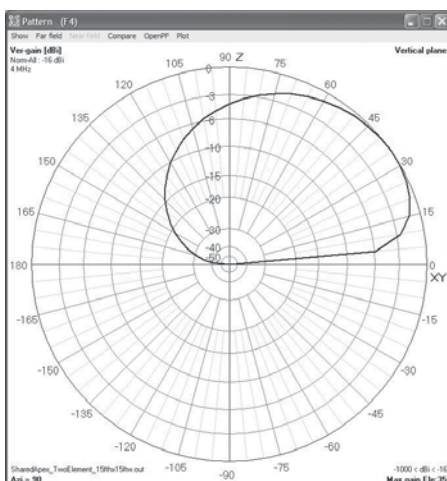
(H)

Figure 16 — Here we see the vertical response for various combined time difference values. The feed point distance is set to 85 inches for all cases. In Part A, the delay is 4 ns. For Part B, the delay is 5 ns and in Part C the delay is 6 ns. Part D has a delay of 7 ns and Part E uses a delay of 8 ns. The delay for Part F is 9 ns. Part G has a delay of 10 ns and Part H has a delay of 11 ns.

teurs to experiment with this array, however, and will provide the *4NEC2* model to anyone who is interested. There is nothing magic about the shape of the loop, so its aspect ratio and size can be adjusted to meet individual needs. A smaller version would provide less forward gain, but a wider frequency range; conversely, a larger version would provide more forward gain but a lower frequency range. The array can be mounted at other heights, although the vertical take-off angle does increase with height. It is important, though, that the supporting structure be non-conductive. The antenna does not require an RF ground, although a safety / lightning ground is always a good idea.
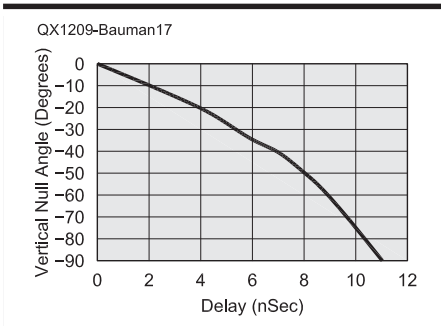
Figure 17 — Vertical null angle as a function of delay with the feed point at 85 inches at 4 MHz.

QX1209-Bauman17

Acknowledgements:

I would like to thank the engineers at the HCJB technology center for their encouragement during a crucial portion of the design phase of this array and for providing an initial NEC model to correlate my observations. In addition, I would like to thank Arie Voors for providing the *4NEC2* modeling software used to model the array and present these results.

Mark Bauman, KB7GF, has been an Amateur Radio operator since 1978.He is an Amateur Extra class license, and is an ARRL member. Mark is a licensed professional electrical engineer in the state of Washington and is a registered patent agent. He works for Nelson Irrigation Corporation, and is a small business owner. He lives with his wife and their four children in College Place, Washington.

Notes

QEX

Robye Lahlum, W1MK

45 Brookview Rd, Boxford, MA 01921; **w1mk@arrl.net**

F_a: Measurement and an Application to Receive Antenna Design

Does the mean level of noise at your location limit the effectiveness of your receive antenna? This study may help answer that question.

Receive antennas for 160 and 80 meters normally trade off absolute gain for directivity. The logic behind this is that since both signal and noise levels are high, the gain of the receive antenna is not as important as the signal to noise ratio, which is often improved more by directivity than by the absolute gain.

The classic Beverage antenna does exactly this; it trades off absolute gain for directivity. More extreme examples of this tradeoff are the K9AY loop and the Waller Flag receive antennas, with gains of –28 dBi and –50 dBi respectively, but with good to excellent patterns.[1, 2] Depending on the amount of man-made noise at the receive location, antennas with gains in the –30 dBi to –50 dBi range maybe too low for optimum receiving. In general, locations with low levels of man-made noise may be limited by the minimum gain of the receive antenna, whereas locations with high man-made noise levels are not limited by these low receive antenna gains.

For decades, the International Telecommunication Union (ITU) has made man-made noise measurements and published reports for various locations (ITU Recommendation P.372-7).[3] ITU-R P.372-7 has established excess noise values, "F_a," for four types of environments from "Quiet Rural" to "Business/Industrial."

The F_a values from ITU-R P.372-7 for various frequencies and environments are shown in Figure 1. In the listed references, the F_a results are often labeled F_a in that they are derived from the median values of the measured man-made noise taken at many locations over time using short verti-

[1]Notes appear on page 13.
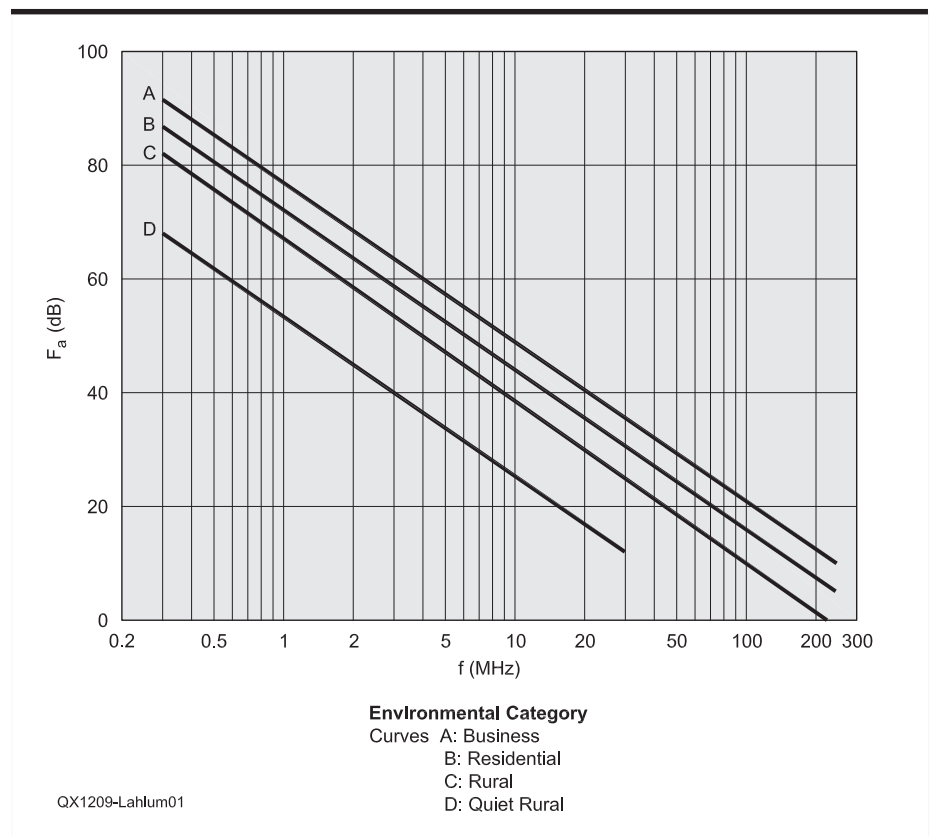
Figure 1 — This graph shows the man-made noise measured by the ITU and reported in ITU-R P.372-7 for various radio receiving environments.

cal lossless grounded monopole antennas. It is nothing more than a number at a specific frequency that gives the expected noise level above thermal noise for a lossless small vertical antenna.

The F_a values shown in Figure 1 can be approximated by a linear expression:

$$F_a = c - d \log f \qquad \text{[Eq 1]}$$

where:

f is the frequency in MHz

c and d are values given in Table 1, from ITU-R P.372-7.

Table 1

Values of the Constants c and d

Environmental Category	c	d
Business (curve A)	76.8	27.7
Suburban (curve B)	72.5	27.7
Rural (curve C)	67.2	27.7
Quiet rural (curve D)	53.6	28.6

Then the received noise is given by:

$$\text{Noise} = -174 + F_a \text{ dBm/Hz} \qquad [\text{Eq 2}]$$

The gain of the small lossless vertical antenna used in the ITU-R P.372-7 report is –1.25 dBi.[4, 5] The noise from a similar small vertical antenna but with a gain of G dBi will be:

$$\text{Noise} = -174 + F_a + G + 1.25 \text{ dBm/Hz} \qquad [\text{Eq 3}]$$

The F_a values published in ITU-R P.372-7 for 160 and 80 meters are given in Table 2.

Until recently, most amateurs have not had easy access to equipment that allows them to make moderately accurate measurements of F_a for their specific location. The advent of the RFSPACE SDR-IQ software defined radio, the Elecraft K3/P3 radio, and other similar receivers, now allows a relatively easy measure of F_a at a specific location and frequency. I have recently done measurements of F_a for 80 and 160 meters at my location north of Boston using an Elecraft K3/P3 radio.

My location, approximately 25 miles north of Boston, would be viewed in ITU-R P.372-7 as a Rural or Quiet Rural environment because it meets the ITU definition of a Rural environment with the density of housing being less than one per 2 hectare(approximately 5 acres). Residential or Suburban area is defined by ITU as an area used predominantly for single and multiple family dwellings with a density of at least five per hectare (2.47 acres, or approximately 1 house per ½ acre) with no large or busy highways. Business areas are defined as any area where the predominant usage throughout the area is for business, such as stores and offices, industrial parks or shopping centers.

The antenna I used for the F_a measurement was a vertical antenna used for receiving similar to one described in *ON4UN's Low Band DXing*.[6] The receive vertical used is a relatively short vertical of 44 feet in height with four 20 foot top loaded guys, and is thus less than a ¼ wavelength in height on both 160 and 80 meters. The antenna impedance is electrically built out to 75 Ω real using a series inductor and series resistor for receiving use on 160 meters. In order to approximate the gain of this antenna I used *EZNEC* to calculate the real part of the impedance
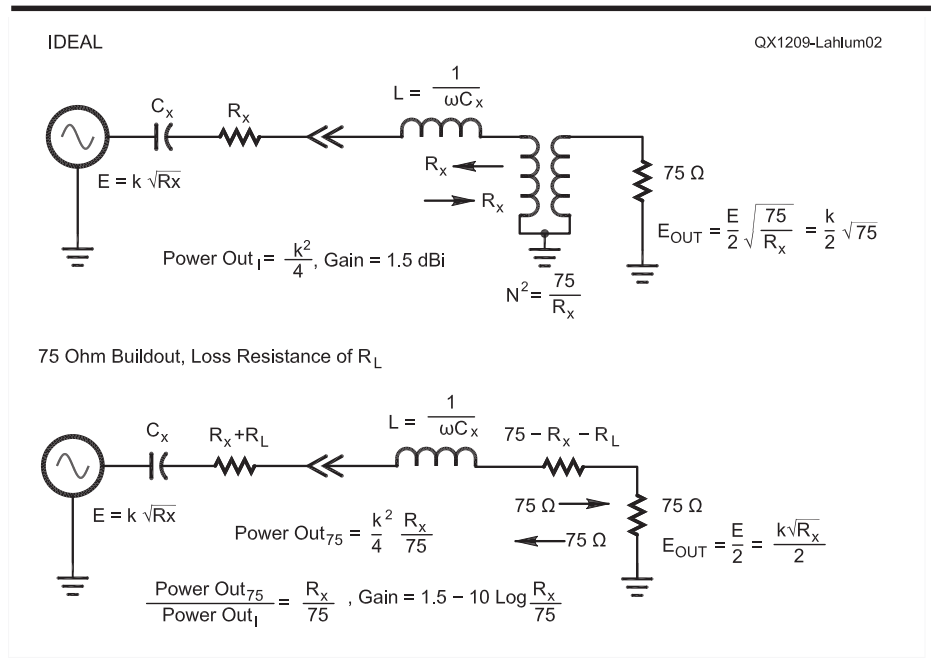
Figure 2 — Here are diagrams of the idealized receive antenna and the actual receive antenna that I used for my measurements.

Table 2

Excess Noise Values Measured by ITU

	Fa, 1.8 MHz	Noise/Hz	Fa, 3.5 MHz	Noise/Hz
Quiet Rural	46 dB	–127 dBm/Hz	38 dB	–136 dBm/Hz
Rural	60 dB	–114 dBm/Hz	52 dB	–122 dBm/Hz
Suburban	65 dB	–109 dBm/Hz	57 dB	–117 dBm/Hz
Business	70 dB	–104 dBm/Hz	62 dB	–112 dBm/Hz

and the ideal lossless gain of the antenna, assuming a perfect ground.[7] I then measured and adjusted the impedance to obtain the 75 Ω real input impedance and estimated the gain per the model shown in Figure 2.

Gain = 1.5 – 10 Log (Measured real part/calculated ideal real part).

At 160 meters, the calculated ideal real part is 4.5 Ω (ideal perfect ground return)

Ideal gain = 1.5 dBi (ideal perfect ground return)

Measured Input Impedance = 75 Ω real, 0 Ω imaginary at 1.825 MHz

Gain = 1.5 –10 log (75/4.5) –2 dBi = –11.7 dBi (2 dB is the feed line loss between the physical antenna terminals and the K3/P3 receiver.)

Noise measured at midday in 9 Hz bandwidth (as measured with the K3/P3 Receiver) = –136 dBm / 9 Hz

I wanted to measure at midday in order to minimize any contribution to the measured man-made noise by propagated noise. I also tried to make my measurement on days

where there were no thunderstorms within a few hundred miles. This approach was used on multiple different days in order to insure that the measurement results were consistent. I have four of these same receive vertical antennas, which are normally used as part of a receiving array, and in order to have a high confidence level in the measurements, all four antennas were measured (with the unused three open circuited) on multiple days.

Noise referenced to a –1.25 dBi antenna = –136 dBm / 9 Hz + 11.7 dBi –1.25 dBi = –125.55 dBm / 9Hz

Noise per 1 Hz = –125.55 dBm / 9 Hz –10 log 9 = –135.1 dBm/Hz

Thus:

F_a = –135.1 dBm/Hz – (–174 dBm) = 38.9 dB

This measurement of 39 dB for F_a on 160 meters is approximately 7 dB lower than the F_a value for a Quiet Rural area published in ITU-R P.372-7. I would estimate the measurement error at ± 4 dB due to a less

Table 3

Receive Antenna Minimum Gain for Various Noise Environments

	1.8 MHz Minimum Gain	3.5 MHz Minimum Gain
Quiet Rural	–39 dBi	–29 dBi
Rural	–51 dBi	–43 dBi
Suburban	–57 dBi	–48 dBi
Business	–61 dBi	–53 dBi

than perfect receiver and an uncertainty in the antenna gain estimation. The net result, however, is that the expected man-made measured noise was lower than predicted by ITU-R P.372-7. I would expect most active amateurs to have somewhat lower F_a values than predicted by ITU-R P.372-7, because in general we work to improve our man-made noise environment, first by improving the noise from our own homes, and then furthermore by working with the local utilities and our neighbors to try to achieve as low a man-made noise environment as possible. The measurements in ITU-R P.373-7 were not done after the man-made noise was improved. The sites were merely selected and the measurements were made with no man-made noise improvements being attempted.

For this measurement I used an antenna with a gain of approximately –11.7 dBi. The resulting noise that I measured was well above the noise floor of the K3/P3 receiver (which has a noise figure with the preamp on of approximately 10 dB). The measured noise was –136 dBm / 9 Hz, and the noise floor of the K3/P3 receiver was measured as –154 dBm / 9 Hz (equivalent to a 9.5 dB Noise Figure), thus the man-made noise was approximately 18 dB above the receiver noise floor. If I had used a receive antenna with a very low gain, like the K9AY Loop with a gain of –28 dBi, the received man-made noise would be close to the noise floor of the K3/P3 receiver.

For the received noise from the antenna to be greater than or equal to the noise floor of the receiver:

F_a + Gain +1.25 dBi = –174 dBm + NF

Gain = NF – F_a – 1.25 dBi

For my location:

Gain should be greater than 10 dB – 39 dB – 1.25 dBi = –30.25 dBi.

This would indicate that I am not a good candidate for a receive antenna that had a gain less than approximately – 30 dBi, if I wanted to receive down to the level of my receiver noise floor. If my man-made noise F_a values where higher, however, such as the F_a for a suburban environment of 64 dB, then gains as low as –55 dBi would be acceptable.

Assuming a Noise figure of 10 dB, the minimum gains for various environments are given in Table 3. The numbers in Table 3 can be reduced somewhat by adding an external RF preamp in front of the receiver (with the receiver preamp on) to reduce the overall noise figure. For example, a 15 dB gain preamp with a 3 dB noise figure will reduce the overall Noise Figure by 6.6 dB. Adding an additional second RF preamp produces less than a ½ dB of additional improvement in the overall Noise Figure.

Robye Lahlum, W1MK, earned a BSEE degree from North Dakota State University in 1963 and an MSEE degree from Northeastern University in 1965. Robye worked for Bell Labs in the Boston, MA area for 37 years. He has been an Amateur Radio operator since 1955, and has held the call signs WØGBQ, K4JEP, W1EEF and W1MK. At Bell Labs, Robye worked with Frank Witt, AI1H, and Jerry Sevick, W2FMI. Frank is a well known antenna author and Jerry is best known for his work with transmission line transformers and his book by that name. Robye collaborated with John Devoldere, ON4UN, on the 4th and 5th editions of ON4UN's Low-Band DXing. Robye is an ARRL member.

Notes

[1] John Devoldere, ON4UN, *ON4UN's Low-Band DXing*, Fifth Edition, ARRL, 2010, p 7-115. ARRL publications are available from your local ARRL dealer, or from the ARRL Bookstore. Telephone toll free in the US 888-277-5289 or call 860-594-0355, fax 860-594-0303; **www.arrl.org/shop**; **pubsales@arrl.org.**

[2] Doug Waller, NX4D, and Jose Carlos, N4IS, 2011 Dayton Antenna Forum, "Waller Flag: Low Band Low Noise Rotatable Receiving Antenna," available at **www.kkn.net/dayton2011/dayton-2011-antenna-forum.html**.

[3] ITU Recommendation ITU-R P.372-7 is available for download in English at **www.itu.int/rec/R-REC-P.372-7-200102-S/en.**

[4] HF Interference, Procedures and Tools; North Atlantic Treaty Organization, June 2007.

[5] A. C. Fraser-Smith, Radio Science, Vol 42, 2007, RS4026, p 2.

[6] John Devoldere, ON4UN, *ON4UN's Low Band DXing*, Fifth Edition, ARRL, 2010, p 7-20.

[7] Roy Lewallen, W7EL, *EZNEC* Software, **www.EZNEC.com**.

QEX—

Dr. Sam Green, WØPCE

10951 Pem Rd, Saint Louis, MO 63146; w0pce@arrl.net

A Fully Automated DDS Sweep Generator Measurement System — Take 2

*This update describes improvements to the NJQRP DDS
and to the sweep generator measurement circuitry and construction.
Detailed instruction shows how to perform precision
measurements and present results.*

The New Jersey QRP Club's Direct Digital Synthesis (DDS) Daughtercard Kit provided the basis for the sweep generator measurement system in my Nov/Dec 2008 *QEX* article.[1, 2, 3]

This update corrects an error in Figure 5 of that article and summarizes circuit revisions. Then I will describe performance improvements to the measurement system brought about by circuit and component changes and improvements in layout and construction.

I will also summarize modifications to both the NJQRP DDS-30 and DDS-60 daughtercards, including a new change that extends the response of the output amplifiers down through the audio range to enable measurements with higher signal level at low frequencies.

The article will also show you precisely how to perform precision measurements with this versatile hardware platform and generate professional looking results with a spreadsheet program. As a bonus, I adapted this measurement system to measure reflections to check out my antennas.

Figure 1 — The complete pictorial schematic of the author's automated DDS sweep generator measurement system.

Circuit Revisions and Corrections

Figure 1 shows the complete DDS Sweep Generator Measurement System. A significant addition from the earlier article

is the hex inverter that buffers the three control lines from the parallel port to the DDS daughtercard. Another addition is a resistor to calibrate the logarithmic detector. The rest of the circuit is the same. The IC sock-

ets still appear as bottom views suitable for hand wiring.

The new control line buffers receive power from the NJQRP DDS board. In the previous configuration, power from the

[1]Notes appear on page 24.

computer parallel port reached the AD9851 DDS chip on the DDS-60 board when power was otherwise off. This prevented proper startup of the AD9851 unless one also powered down the computer or disconnected the port. This only proved to be a problem for the AD9851 on the DDS-60 and not for the AD9850 on the DDS-30, but best practice is to accommodate both.

Figure 2 is a correction of Figure 5 in the original article, which mistakenly interchanged the labels for pins 2 and 4 of the parallel port. Figure 2 in part 1 from the NJQRP website did show the correct connections to the parallel port.

In one variant, I added a second MAX110 ADC with Busy, D_{out}, and CS respectively, using previously unused parallel port pins 11 and 10 on the Status Port and pin 9 on the Data Port. All other pins are common.

In another variant, I use pin 15 as a discrete input to control measurement timing.

Logarithmic Detector Improvements

Construction enhancements significantly improve the performance of the logarithmic detector by eliminating interference. The construction technique used in part 1 was ugly construction because it was a prototype that underwent many modifications. Much of the circuitry was on perforated board with the voltage regulators mounted to the case, but the logarithmic detector suffered severely from susceptibility to radiated fields (pickup) when the DDS daughtercard was nearby, so I moved the logarithmic detector to the front panel to be close to the input connector and away from radiation from the DDS daughtercard. I added extensive power filtering to the logarithmic detector circuitry to minimize coupling through the power lead. In part 1 I suggested placing the logarithmic detector in its own metal box within the main case for extra shielding, if pickup proved to be a problem.

The impetus for this new version was to build a copy for my friend Matt Kastigar, WØXEU. This time I laid out a printed circuit board, since I had a working model and was beyond the prototype stage. Everything fit nicely onto a standard 2.5 × 3.8 inch board from ExpressPCB, including the three voltage regulators.[4] Figure 3 shows the bare board. The back side is mostly ground plane with a mirror finish, and is virtually impossible to photograph with or without a flash. I finally managed to take a photo that dis-
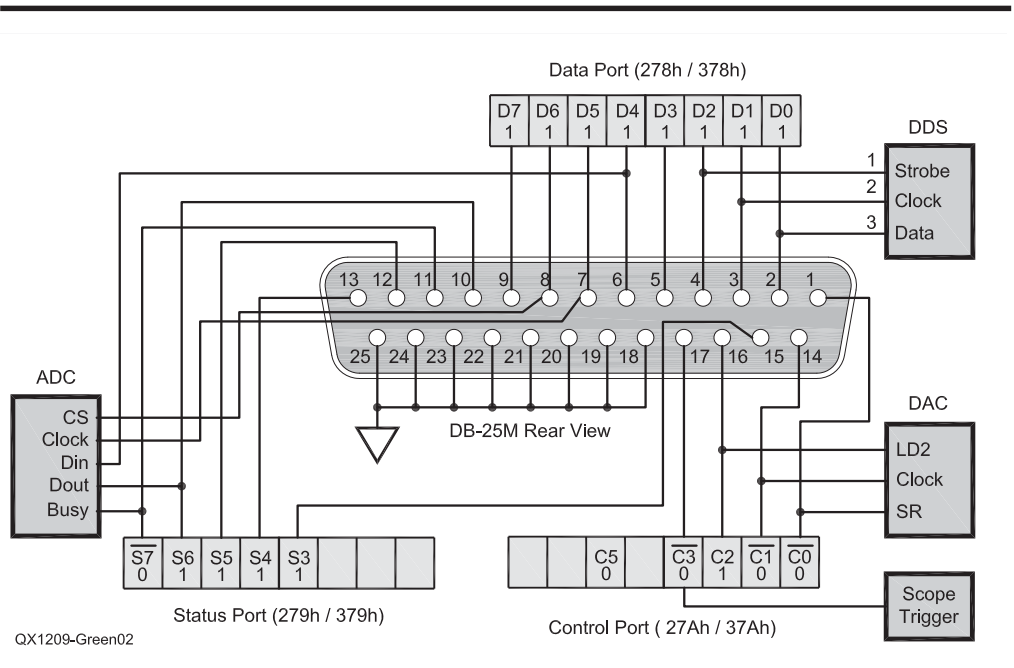
Figure 2 — This diagram shows which parallel port pins drive which functions.
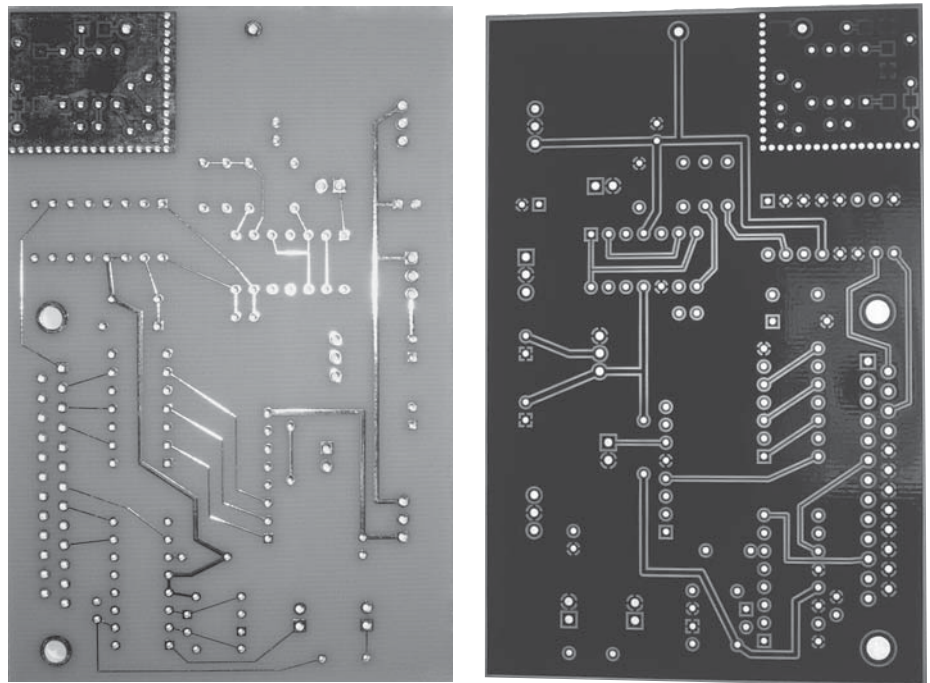
Figure 3 — Here is the bare circuit board.

plays the important features by illuminating from the opposite side.

I kept everything in the area of the logarithmic detector short and low profile to minimize pickup and used lots of ground plane. I positioned plated through holes with minimal spacing around the logarithmic detector circuitry to aid in cutting it out from the main board because I expected to have to place it within its own shielded case.

It turns out that this new configuration, with extensive grounding and low profile construction, eliminates pickup to the logarithmic detector sufficiently well that there is no measurable interference. This was a welcome improvement and a very pleasant surprise.

Rather than have the input capacitors extend much above or below the printed circuit board and provide an antenna for electro-
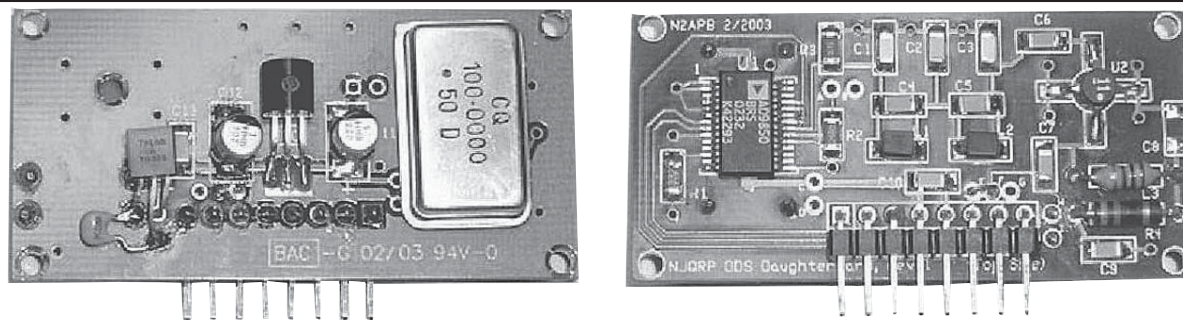
Figure 4 — These photos show the New Jersey QRP Club DDS-30 daughterboard.
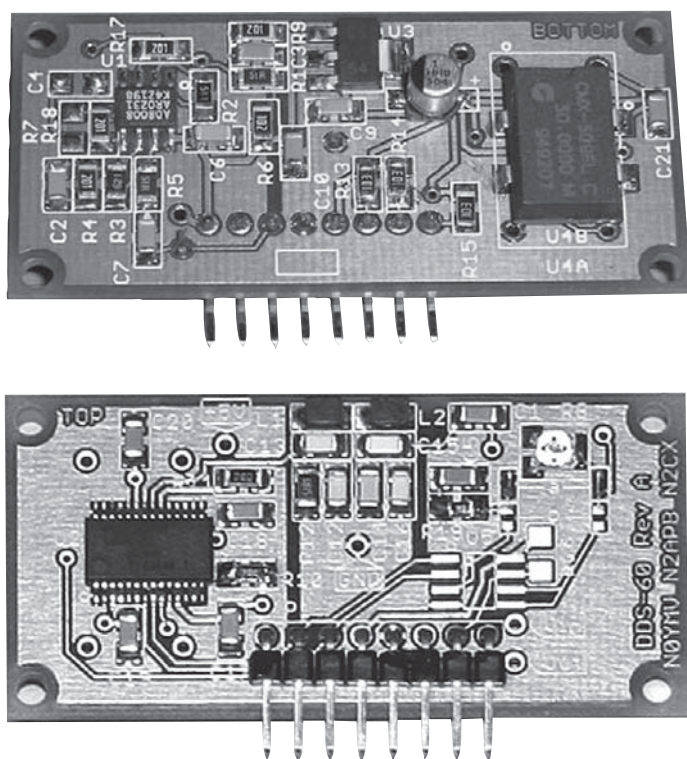
Figure 5 — These photos show the New Jersey QRP Club DDS-60 daughterboard.

magnetic interference (EMI) pickup, I used chip capacitors to keep the profile low. To achieve the most capacitance and thus keep the response down to the lowest possible frequencies, I used the highest value chip capacitors I could find and placed them both above and below the board in parallel. These chip capacitors are 47 μF each, so two in parallel yield about the same value as I used previously. I have since purchased 100 μF, 6 V chip capacitors in 1206 size from DigiKey, so the next version will extend to still lower frequency.[5] Yes, these are 100 μF nonpolarized ceramic capacitors!

NJQRP DDS Daughtercard Modifications

The NJQRP DDS daughtercards are marvelous performers. The original DDS-30 shown in Figure 4 uses the Analog Devices AD9850 DDS chip and the newer DDS-60 shown in Figure 5 uses the AD9851. There is a small programming difference between the two, so each requires a slightly different version of control software.

In part 1, I modified each of the DDS cards in order to extract a dc coupled output for performing measurements at lower frequencies than the amplified outputs can achieve. On the DDS-30, there is provision for an alternate dc coupled output that avoids the low-pass filter. There is no such provision on the DDS-60, so I connected a couple of Molex pins to the output of the low-pass filter where it is still dc coupled, just before the coupling capacitor into the amplifier.

The coupling capacitors into and out of the amplifiers in both daughtercards are all 0.1 μF, so each introduces a corner frequency just above 30 kHz that rolls off and severely attenuates the amplifier output through the audio frequency region.

Now I add high value chip capacitors to extend the low frequency response, just as with the logarithmic detector. The 1206 size chip capacitors are the same size as the 0.1 μF chip capacitors already on the daughtercards, so it is a relatively simple matter to place the new high value chip capacitors on top of the existing capacitors and solder them in place. This obviates the need to remove the original chip capacitors and avoids the risk of damaging the board to perform this modification.

For the DDS-30, attach 47 μF or 100 μF chip capacitors above C6 and C7.

For the DDS-60, attach 47 μF or 100 μF chip capacitors above C1, C2, C5, and C7. C1 and C5 are on the side with the DDS chip, and C2 and C7 are on the side with the amplifier and oscillator.

Figure 6 shows two of the high value chip capacitors added above existing capacitors on a DDS-60.

This simple modification extends the frequency response of the amplified outputs from just below 100 kHz down to below 100 Hz. Figure 7 shows the change in low-end frequency response for the DDS-30 when I added 100 μF chip capacitors onto the original 0.1 μF capacitors. The higher signal level of the amplified outputs now extends the dynamic range of measurements down through most of the audio range. This frequency range now extends sufficiently low that you may omit the direct outputs that bypass the amplifiers unless you really need

Figure 6 — This photo shows how I piggy-backed 47 µF or 100 µF chip capacitors on existing 0.1 µF chip capacitors.

Figure 7 — Here is the DDS-30 response with 100 µF chip capacitors added to two existing 0.1 µF chip capacitors.

Figure 8 — This is the DDS-60 response with 100 µF chip capacitors added to four existing 0.1 µF chip capacitors.

to generate higher level sub-audio signals.

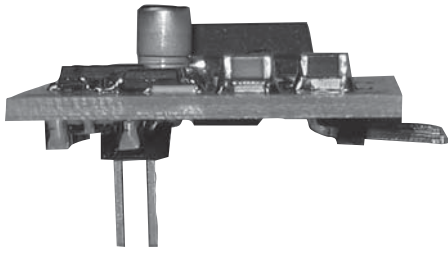Figure 8 shows the change in low-end frequency response for the DDS-60 with the corresponding modification.

Calibration and Accuracy

In preparing this article, I learned that the AD8307 logarithmic detector was measuring +8.9 dBm output power from the DDS-60 amplifier as +14.2 dBm, a sufficiently high level so that the measurement voltage from the logarithmic detector exceeded the input range of the MAX110 A/D converter. While the AD8307 handles inputs up to +16 dBm, the MAX110 A/D converter cannot handle the output voltage from the AD8307 above 2.5 V. The MAX110 accurately calibrates itself under software control, with the specified voltage reference diodes, but lower reference voltages provide a proportionally smaller input voltage range. The 1.25 V reference diodes give an input range of ±2.5 V, which the AD8307 should reach at +10 dBm. It would be better to use 1.5 V reference diodes to increase the range, but I find no suitable device between 1.255 and 1.80 V.

Since there are no suitable reference diodes, you can simply decrease the DDS-60 power level to avoid the problem. A better solution is to calibrate the AD8307, as I discuss in the next section, to reduce the erroneous output voltage to less than 2.5 V below an input power level of +10 dBm.

The frequency accuracy of the AD9850/AD9851 DDS is as good as the precision of the crystal oscillator provided with the DDS kit. If you can make a better frequency measurement of the oscillator, a calibration factor in the software readily corrects that as well.

The accuracy of the MX7543 D/A converter depends on the reference diode and offsets in the quad op amp. Results with no corrections are fairly good, though I tailor the calibration factors in software for each unit. If you use one of the recommended reference diodes and a more precise quad op amp than the LM324, you will probably need no such adjustment. Eventually I plan to program
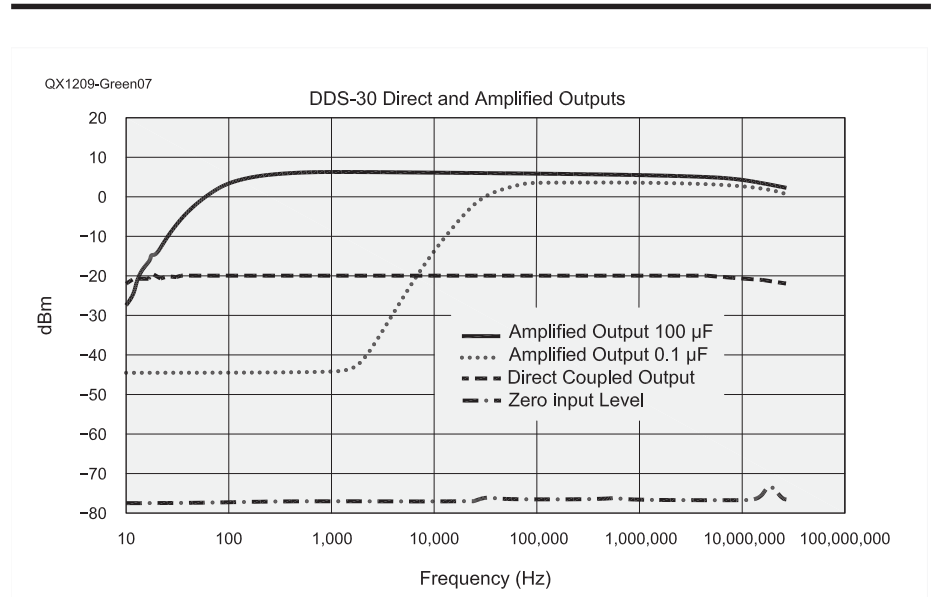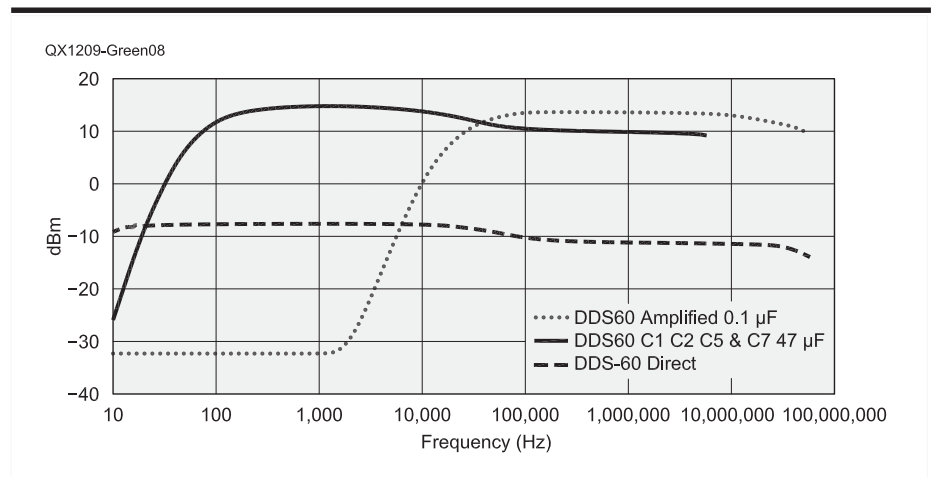
a self-calibration process that uses the A/D converter to measure and set the calibration factors for the D/A converter.

The AD8307 logarithmic detector proves to be the least accurate element. Usage here is primarily for relative measurements, so this isn't usually a problem. I use the manufacturer's nominal specification in the software to measure and display power in dBm. Nominal AD8307 specifications are 25 mV/dB and a zero volt intercept of –84 dBm. These resulted in a +14.2 dBm indication from the +8.9 dBm power measurement noted in the preceding section. That is quite a large error. The allowed range of logarithmic slope from the AD8307 data sheet is from 23 to 27 mV/dB, however, and the error I observed

corresponds to a value of 26.3 mV/dB, which is well within that range. Analog Devices discusses calibration of the AD8307 on page 19 of the data sheet, in a section titled "Slope and Intercept Adjustments." A somewhat simpler process that worked for me on two different sample devices is to tie pin 5 to pins 6 and 7 and to add a large resistor, R_{CAL}, from pin 4 to ground that I selected to decrease the measured value to the correct value. Figure 1 shows these modifications in dashed lines. For one sample of AD8307, the resistor R_{CAL} is 750 kΩ, and for another, R_{CAL} is 470 kΩ. If the sample device starts with a slope less than 25 mV/dB, this requires a calibration factor change in the software.

Figure 9 — This is the assembled sweeper board with the DDS-60 daughterboard.
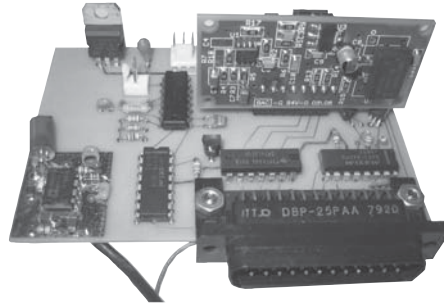
Figure 10 — This is the cable with the Molex connector.

Figure 11 — The sweeper assembly wiring details are shown in this photo.

Construction

Figure 9 shows the new printed circuit board with components. The board has a DB-25M connector to mate to the computer parallel port with a full 25 wire parallel cable. This connector also provides a means to mount the board within the case.

Connections for the DDS output(s) to the bulkhead coaxial connectors are via 2 pin Molex connectors with suitable lengths of RG-174/U coaxial cable. The coaxial cable from the bulkhead coaxial connector attaches directly to the logarithmic detector input at the printed circuit board, because I was being especially careful to keep it shielded. Other input and output signals need no additional shielding. Figure 10 shows the detail of a cable terminating in the Molex connector. Figure 11 shows the coaxial cables and other signal wires to the bulkhead coaxial connecters. Since the connectors are on the box rather than the board, you have your choice of connector type. I prefer BNC connectors, but you could use RCA phono connectors to reduce cost. SMA connectors would be overkill and are certainly less convenient.

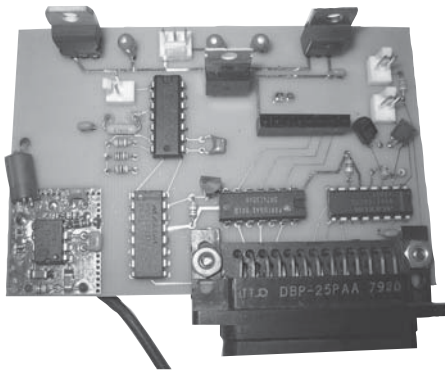The new printed circuit boards with the DDS daughtercards fit nicely into a 4.7 × 3.7 × 2.2 inch cast aluminum case. Use a larger case if you want to put the power supply inside. I use an external power supply module. All necessary regulation and power supply filtering is already on the board, but filter whatever power leads you bring into the box with feedthrough bypass capacitors to minimize emissions from the DDS source and external electromagnetic interference (EMI) to the very susceptible logarithmic detector.

Figure 12 shows the DB25M connector to the computer parallel port and the feedthrough bypass capacitors that filter the ±12 V lines from an external power supply. The most difficult part of fabrication is drilling and filing a suitably shaped hole into the case to accommodate the D connector.

Do this more carefully than I did.

Improved Dynamic Range

Figure 13 shows a significant improvement in dynamic range for the swept frequency response of the same crystal filter evaluated in part 1, with the prototype unit.

Availability of Kits

No kits are available at the time of this writing. George Heron of Midnight Design
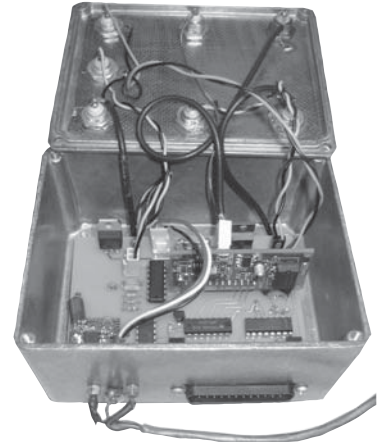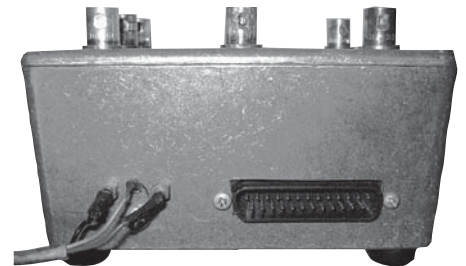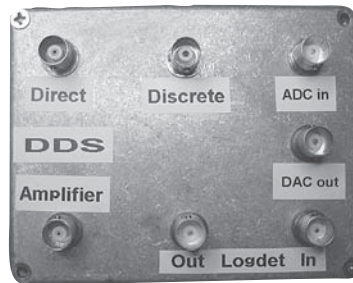
Figure 12 — Photo A shows the input and output BNC connectors. Photo B shows the case detail, with feed-through bypass capacitors and the DB25M connector.
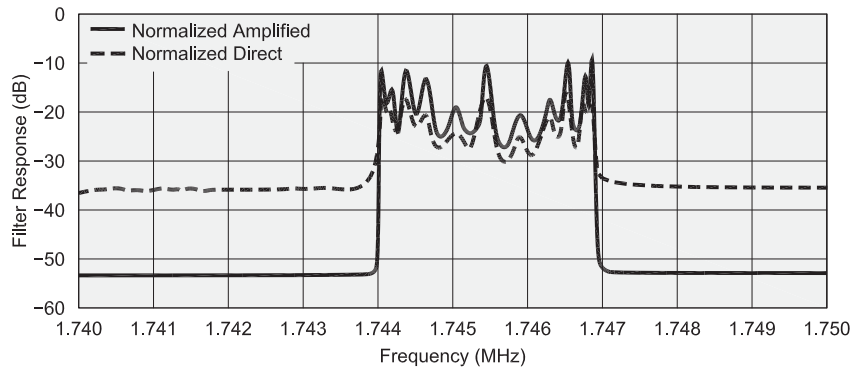
Figure 13 — Here are the frequency response sweeps of a crystal filter, showing the direct and amplified signals. Compared with the plot shown in the original article, this represents a significant improvment in dynamic range.

Figure 14 — Here is the data format of the text files that show the results from linear and logarithmic sweeps and the transfer function voltage sweep.

Solutions previously expressed some interest in providing a circuit board and kit, but nothing came of it.[6] I expect George might develop a kit if there were enough interest. The ExpressPCB board I made here is good for a trial run but is not a low cost solution. Three boards from ExpressPCB with shipping cost about $61 for nice boards without solder mask or silk screen and about $85 with these very desirable features.

Better Software from Better Programmers

In part one, I said I expected a real programmer to step up and generate a fully general version of my prototype software with a *Windows* Graphical User Interface to replace my simple console applications. That never happened. This article is an attempt to make such a project more appealing. Failing that, I need to become a better programmer and learn to write proper *Windows* drivers. UserPort software allows me to write software that controls serial and parallel ports without writing a proper driver.[7] I would rather try my hand at a higher frequency DDS.

Computer Interface Issues

One drawback is that parallel ports are disappearing from newer Personal Computers. Folks tell me I should incorporate USB. Experts tell me I can't toggle bits as fast with USB. The solution would be to incorporate a CPU into this project to control the various elements and communicate to the CPU via USB. I like that approach, but it's further out than the *Windows* driver.

Figure 15 — Here is the setup for the example measurement.
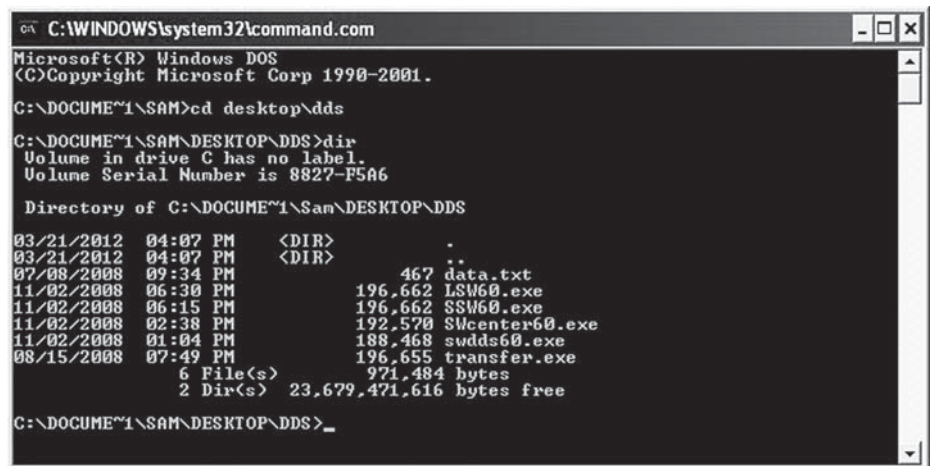
Figure 16 — Launch a DOS window and navigate to the directory with the executable programs.

Software Programs that Record Data

For the DDS-30 and DDS-60 respectively, SSW30 and SSW60 perform single linear sweeps over a specified frequency range between start and stop frequencies with a specified fixed frequency step. The swept measurement test signal from the DDS passes through a device or circuit under test and then passes to the logarithmic detector. The analog to digital converter (ADC) reads the voltage from the logarithmic detector. The application computes the resulting signal power level in dBm from the ADC voltage measurement, and writes the frequency of each measurement and the measured power to a text file whose title you must specify in the command line. The resulting text file has two columns of tab-delimited data with the titles "Frequency" and "dBm" at the heads of the two columns.

For the DDS-30 and DDS-60 respectively, LSW30 and LSW60 perform single logarithmic sweeps over a specified frequency range between start and stop frequencies with a frequency step that varies with a specified ratio of increase between steps, so that the increments increase proportionally with the frequency. This is very useful for sweeping frequency over a very wide range. Otherwise, the logarithmic sweep applications perform as do the linear sweep applications.

The Transfer application provides a swept voltage from the digital to analog converter (DAC) to drive a device or circuit under test and reads the resulting response with the ADC. The resulting text file again has two columns of tab-delimited data, but the titles at the heads of the two columns are now "Vsource" and "Voutput." Figure 14 shows example text files for these three cases.

The data that these programs record is in a format ready to use in a spreadsheet to prepare professional looking plots.

Measurement Sequence Instruction

This is the sequence of steps to acquire data and produce professional looking plots using applications that perform a single sweep and write the resulting data to a text file. These include SSW30, SSW60, LSW30, LSW60, and Transfer.

Let's perform a sample sweep and list all the steps from setup through acquisition to plot generation. The following example uses SSW60 to plot the response of the same crystal filter in Figure 13.

Setup

Connect the device under test between an output of the DDS and the input to the Logarithmic Detector as in Figure 15.

In *Windows*, click Start and then click Run to pop open the Run Dialog. Then type "cmd" or "command" into the Run Dialog and hit the Enter key to open a console win-

dow. You are now effectively in DOS and DOS rules apply.

Navigate to the folder that contains SSW60.exe. The easiest thing to do is place the executable in a folder on the desktop. Suppose you name that folder DDS. Then in the DOS console window, type "cd desktop\ dds" to enter that folder. Figure 16 shows the executables you placed within the DDS directory.

Then run SSW60.exe by typing "ssw60" and hit the ENTER key. Since you entered no name for the output text file, a necessary parameter, the program responds with directions for correct syntax, as shown in Figure 17.

When you enter at least one parameter, the program will run with default settings. If you omit the filename, the program assumes that the first entry is in fact the file name and will generate a text file with that name, even
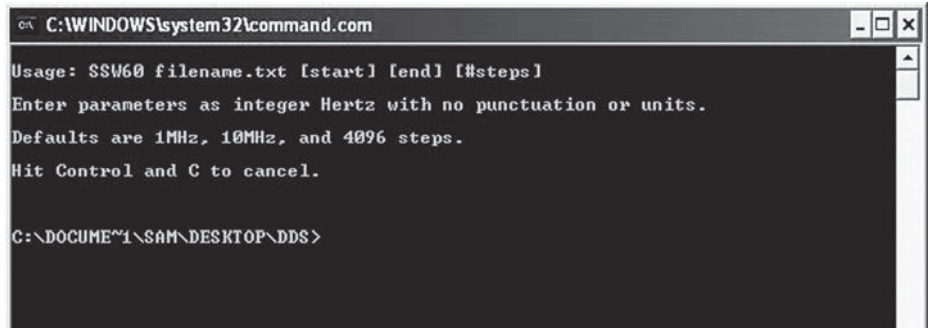

Figure 17 — Execution of a program with no parameters provides instruction in the syntax for that program.
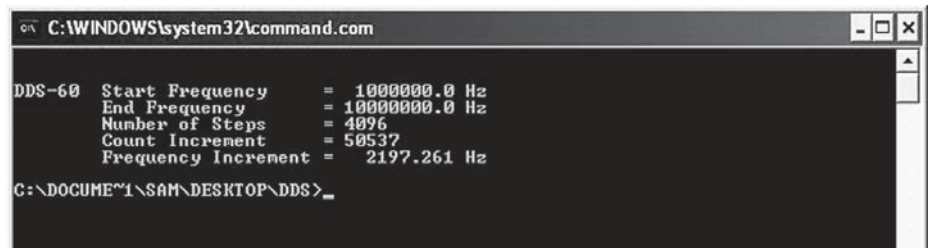

Figure 18 — Here is the SSW60.EXE display after execution.
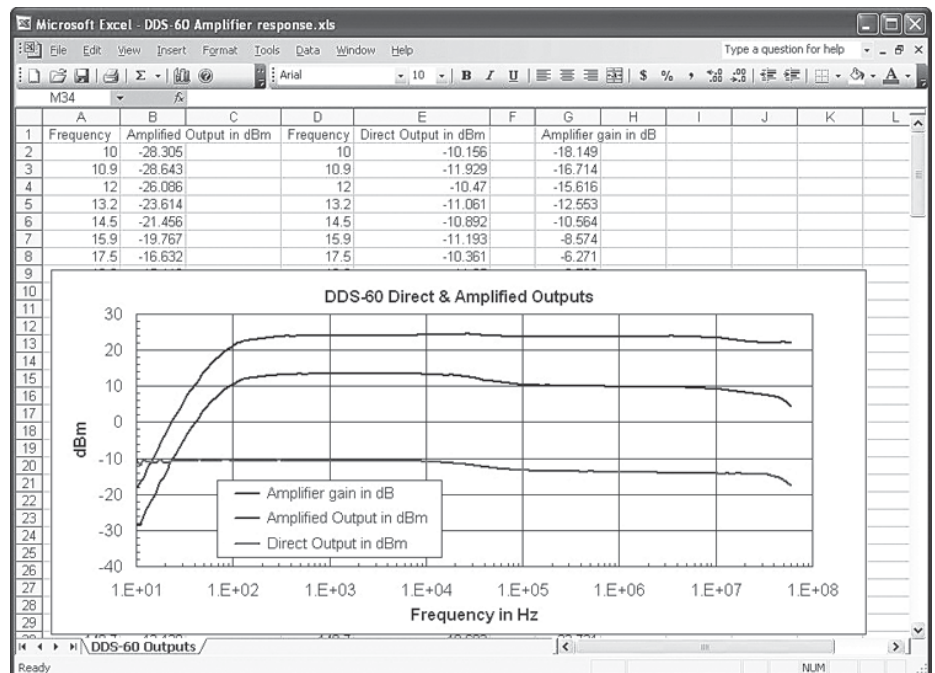

Figure 19 — Normalization of the amplified output to the direct output gives amplifier gain, as shown in the top trace.
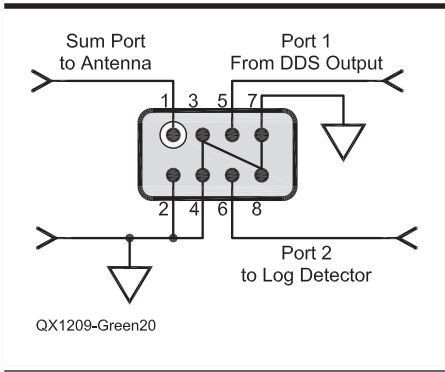
Figure 20 — A Mini-Circuits PSC-2-2 Power Splitter/Combiner enables measurement of the reflected power back into the sum port.

Reflectometer for Antenna Measurement

Just when I thought this paper was complete, I had occasion to question the condition of my antennas. I connected a Mini-Circuits Labs PSC-2-2 Power Splitter/Combiner as shown in Figure 20.[8] I put the PSC-2-2 in a small aluminum Pomona Box with BNC connectors. Port 1 connects to the DDS output. Port 2 connects to the logarithmic detector input. The sum port connects to the antenna. The signal from the DDS into port 1 only appears at the sum port, because of the inherent high isolation between ports 1 and 2.

The antenna reflects the unradiated portion of the signal back to the sum port. Equal portions of the reflected signal then appear at port 1 and port 2. The portion into port 1 goes back to the DDS but does not bother the amplifier. The portion into port 2 goes to the logarithmic detector and is proportional to the reflection coefficient of the antenna.

if you meant it to be the start frequency!

Select appropriate start and stop frequencies. Multiple tries will give you these frequencies. Figure 18 shows the screen after the program runs.

Process and Normalize Data

After execution, a text file with the data resides in the same folder. Navigate to that folder in Explorer and double click on the data file to open it. Then copy the data in the text file to the clipboard and close the text file. The sequence "Ctrl-A Ctrl-C Alt-F4" does this easily. Then paste the data into a spreadsheet program. This is a much simpler alternative than using the import external data feature of the spreadsheet program. Once the data is in the spreadsheet, it is available to plot or process.

Notice that the DDS outputs are not perfectly flat over frequency. Normalize the data for optimal presentation by performing each measurement with and without the device under test. Without the device under test, the system measures its own output power as a function of frequency as a reference. Normalize the data from the same measurement with the device under test by dividing it by the reference data. Since the data from the logarithmic detector (in dBm) is the logarithm of the data, this normalization process is simply a subtraction. The normalized result is then the loss or gain of the device under test in dB rather than dBm.

Figure 19 gives an example of the normalization procedure to display the gain of the amplifier in the DDS-60. Column B is the amplified DDS output versus frequency and column E is direct unamplified output of the DDS versus frequency. Column G is simply column B minus column E, or the quotient of amplified power divided by direct power expressed in dBm.
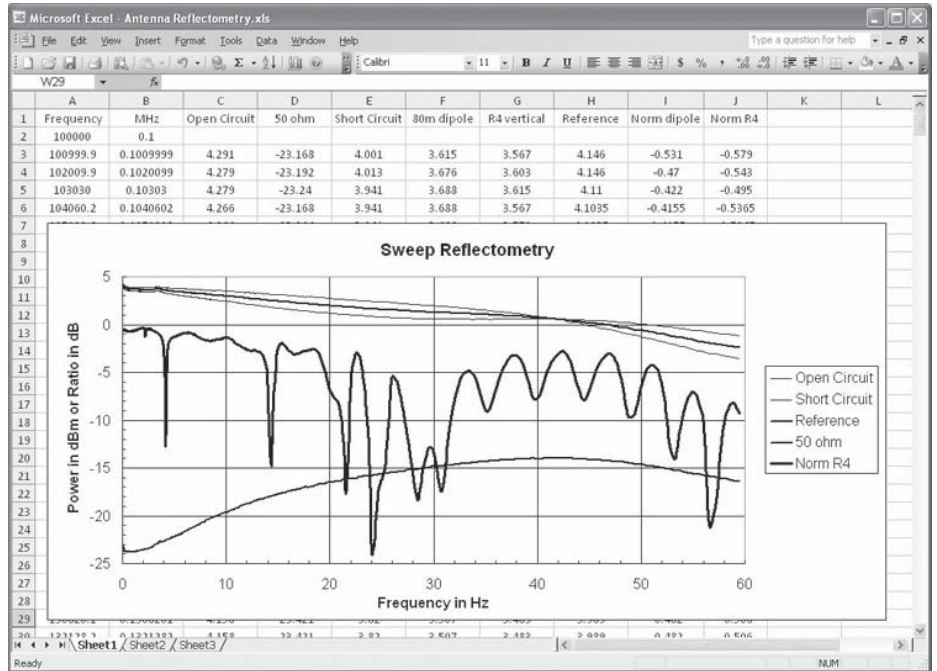

Figure 21 — Measurement of the reflected power yields the reflection coefficient.
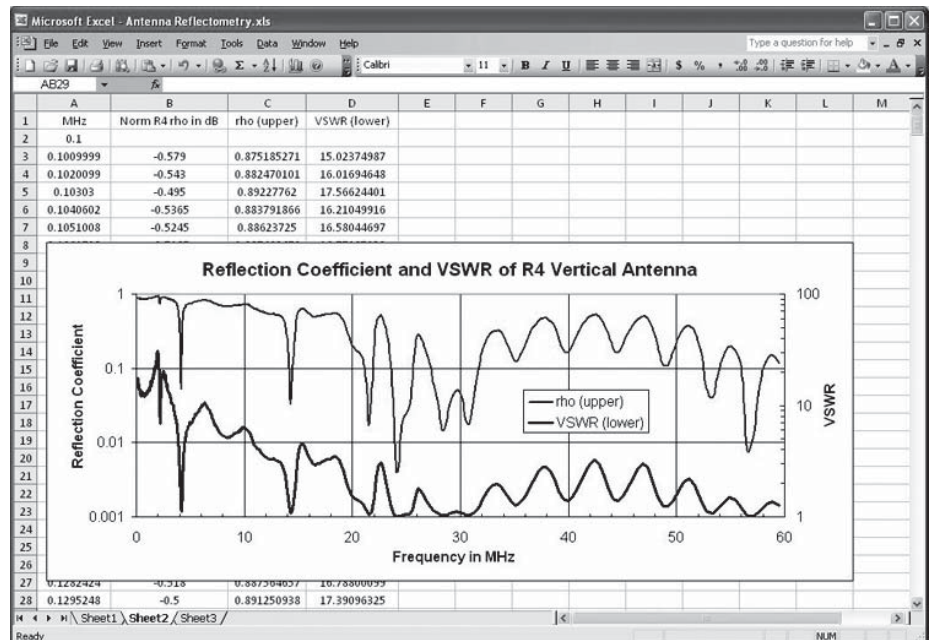

Figure 22 — This plot shows the reflection coefficient and SWR of my Cushcraft R4 vertical antenna across the HF region.

Most antenna analyzers use a bridge arrangement to compare the unknown antenna impedance to a 50 Ω reference. I spent most of a year working on an optical time domain reflectometer using a hybrid fiber-optic splitter/combiner that behaves similarly, and this approach felt familiar.

Figure 21 shows reflectometer sweep results using LSW60.exe with an open circuit, a 50 Ω termination, a short circuit and the coaxial cable to a Cushcraft R4 vertical antenna attached to the sum port of the PSC-2-2. The 50 Ω response shows the degree of isolation between ports 1 and 2. The short circuit and the open circuit cause similar total reflections over frequency with some minor variation. I use the average of these two curves as the reference value to normalize the antenna sweep. When normalized, the reflection sweep is the reflection coefficient of the antenna and feed line.

Converting the normalized reflection data from logarithmic to linear directly yields reflection coefficient. From reflection coefficient, we can calculate SWR and could similarly calculate impedance. Figure 22 shows the reflection coefficient and SWR of my R4 vertical antenna over the HF region. The reflection coefficient is the upper trace and uses the left side vertical scale that goes up to a maximum of unity for a perfect reflection. The SWR is the lower trace and uses the right side vertical scale that goes down to a minimum of unity for a perfect SWR.

I am very pleased with this initial testing as an antenna analyzer. I've always wanted one. I haven't yet evaluated any other splitter/combiners.

Software Download

All programs are available for download from the ARRL QEX files website.[9] The GNU Public License Statement is included. That statement says:

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.[10]

Dr. Sam Green, WØPCE, is a retired aerospace engineer who lives in Saint Louis, Missouri, He holds degrees in Electronic Engineering from Northwestern University and the University of Illinois at Urbana. Sam specialized in optical data communications and photonics. Sam became KN9KEQ and K9KEQ

in 1957, while a high school freshman in Skokie, Illinois, where he was a Skokie Six Meter Indian, and held a Technician class license for 36 years before finally upgrading to Amateur Extra Class in 1993. He is a member of ARRL, a member of the Boeing Employees Amateur Radio Society (BEARS), and a member of the Saint Louis QRP Society. Sam is a Registered Professional Engineer in Missouri and a life senior member of IEEE. Sam holds sixteen patents, with two more patent applications pending.

Appendix

Software details

The original NJQRP DDS-30 that operates to 30 MHz and the new NJQRP DDS-60 that operates to 60 MHz require separate versions of the DDS programs because the least significant bit in the phase word must be different. In the download package, you will find SWDDS30, SWcenter30, SSW30, and LSW30 for the DDS-30 daughtercard. You will also find SWDDS60, SWcenter60, SSW60, and LSW60 for the DDS-60. Select the appropriate software for the DDS-30 or DDS-60 by the numeric suffix.

All programs operate in a Console Window, otherwise known as a DOS Window. All require installation and configuration of UserPort to bypass *Windows* restrictions on accessing ports from simple executable programs. All programs terminate with "Control-C." "Pause" and "Enter" freeze and restart the display.

Some of the programs run once and log the data to a text file. Others run continuously and offer keystroke inputs to change the operating parameters. The details follow.

ADC.EXE

ADC will start from a command line in a DOS window or by double clicking. A usage statement appears that tells you the command line parameters to enter to decrease the resolution to 12 bits for faster measurements or increase it to 14 bits for more precise but slower measurements.

Usage:

ADC 12   12 bit resolution (plus sign)

ADC 13   13 bit resolution default

ADC1 14   14 bit resolution but slow

When running, the program displays:
• The elapsed time for each measurement, and
• The voltage measured at Input 1 of the analog to digital converter.

DAC.EXE

DAC will start from a command line only. It requires at least one parameter to function. With no parameters following the command, the program displays a usage statement that tells you proper syntax.

Usage: ADC Vstart [Vend] [#steps] [dwell]

Constant output for single argument.

Sweeps for two arguments with Vstart less than Vend,

Both between –2470 to +2470 mV in integer mV with no decimal, and

Dwell per step in ms.

With a single parameter, the output from the digital to analog converter is a constant voltage and the program ends. With two parameters, the output voltage sweeps from the first entry, which must be the most negative value to the second entry that represents the most positive value, and repeats. The possible range is about –2.5 V to +2.5 V. Enter start and end values in integer millivolts as parameters following the command to select values within this range. You may also control the number of steps and the dwell time at each.

When running, the program displays either the constant value or
• The start voltage,
• The end voltage,
• The number of steps,
• The count increment of the number the program writes to the D/A Converter, and
• The instantaneous value of the output voltage

SWDDS.EXE

SWDDS will start from a command line or by double clicking. The program displays a usage statement that tells you the proper syntax.

Usage: SWDDS [start] [end] [#steps]

With no parameters, SWDDS runs default settings to sweep the DDS from 1 MHz to 10 MHz in 4096 steps. The program sweeps from the lower start frequency to the higher end frequency recurrently until you end the program manually.

The numeric keypad allows coarse increases and decreases of start and end frequencies and number of steps. These choices make sense if you consider the positions of the keys.

"7"   Doubles the start frequency to values less than 90% of the end frequency, or raises it by 10% if the result would be higher than 90% of the end frequency.

"1"   Halves the start frequency.

"9"   Doubles the end frequency with an upper limit of 30 MHz.

"3"   Halves the end frequency to values greater than 110% of the start frequency, or Decreases it by 11% if the result would be lower than 110% of the start frequency.

"8"   Halves the number of steps.

"2"   Doubles the number of steps with an upper limit of 4096.

These are the only keys that will change operating parameters.

The D/A Converter resolution limits the number of steps to 4096. The DDS is able to provide resolution to a small fraction of a hertz if you eliminate the step count limit and the minimum step size, which in turn defeats the D/A Converter analog sweep below a step count of one.

When running, the program displays:
• The start frequency, the end frequency,
• The number of steps,
• The incremental DDS step count,
• The corresponding incremental frequency change,
• The D/A Converter sweep limits in hexadecimal numbers,
• The instantaneous output frequency,
• The incremental D/A Converter step count, and
• The instantaneous word the program writes to the D/A Converter.

I use integer math because the DDS, ADC, and DAC must receive binary numbers. At low frequencies and small frequency steps, the integer math breaks down due to rounding error. Do not expect to hit the number keys endlessly without some problems.

SWCENTER.EXE
SWCENTER will start from a command line or by double clicking. The program displays a usage statement that tells you the proper syntax.

Usage: SWCENTER [CenterFreq] [SweepWidth] [#steps]

Integer frequencies between 0 and 30000000 with no units or punctuation, and#steps = 4096 maximum.

With no parameters, SWCENTER runs default settings of a center frequency of 1 MHz and a 100 kHz sweeps width, so the default sweep range is between 950 kHz and 1150 kHz. The program sweep symmetrically from below the center frequency to above the center frequency recurrently until you end the program manually.

The numeric keypad allows fine and coarse changes of center frequency and number of steps.

"7"   Raises the center frequency by 10%.
"1"   Decreases the center frequency by 11%.
"9"   Doubles the sweep frequency range.
"3"   Halves the sweep frequency range.
"8"   Halves the number of steps.
"2"   Doubles the number of steps with an upper limit of 4096.
"4"   Decreases the center frequency by the frequency step count for adjustable fine resolution.
"6"   Increases the center frequency by the frequency step count for adjustable fine resolution.

"5"   Increases the number of steps by ¹⁄₁₆th for adjustable fine resolution.

These are the only keys that will change operating parameters. The reasoning behind "5" was to provide fine control of the total sweep time in order to adjust it to an oscilloscope time base for a stable display.

When running, the program displays:
• The center frequency,
• The total sweep width,
• The start frequency,
• The end frequency,
• The number of steps,
• The incremental DDS step count,
• The corresponding incremental frequency change,
• The instantaneous frequency,
• The incremental D/A Converter step count, and
• The instantaneous word the program writes to the D/A Converter.

SSW.EXE
SSW starts from a command line only. With no parameters, SSW prints a usage statement to explain the parameters it requires.

Usage: SSW filename.txt [start] [end] [#steps]

Enter parameters as integer hertz with no punctuation or units.

Defaults are 1 MHz, 10 MHz, and 4096 steps.

You must enter an output filename as the first parameter. If you enter anything else, a text file appears on your drive with that as the filename. The file appears in the same folder as the executable unless you redirect it using the old DOS filename rules.

With no other parameters, SSW runs default settings that sweep from 1 MHz to 10 MHz in 4096 equal steps. If you follow the filename in the command line with only a single parameter less than 10 MHz, the DDS will sweep from that start frequency up to 10 MHz. With additional parameters, you control the end frequency and number of steps. The upper limit to the number of steps corresponds to an increment of one count in the 32 bit number written to the DDS, a very tiny fraction of a hertz.

All numeric entries are decimal integer variables, so you must enter 5 MHz as 5000000 with no units, commas, spaces or any other punctuation.

The numeric keypad serves no function in this program. There is only a single sweep with a measurement and record of data at each frequency step. The useful output appears in the text file.

When running, the program displays:
• The start frequency,
• The end frequency,
• The number of steps,
• The incremental DDS step count,

• The frequency increment,
• The instantaneous frequency, and
• The measured power from the log detector at Input 2.

The program writes the instantaneous frequency and measured power to the text file.

LSW.EXE
LSW starts from a command line only. With no parameters, SSW prints a usage statement to explain the parameters it requires.

Usage: LSW filename.txt [start] [end] step-fraction-denominator]

Enter parameters as integer hertz with no punctuation or units.

Defaults are 1 MHz, 10 MHz, and 20 MHz, so each step is 5% higher in frequency.

You must enter an output filename as the first parameter. If you enter anything else, a text file appears on your drive with that as the filename. The file appears in the same folder as the executable unless you redirect it using the old DOS filename rules.

With no other parameters, LSW runs default settings that sweep from 1 MHz to 10 MHz in increasingly larger steps, proportional to the frequency. If you follow the filename in the command line with only a single parameter less than 10 MHz, the DDS will sweep from that start frequency up to 10 MHz. With additional parameters, you control the end frequency and the step ratio. The lower limit to the step ratio corresponds to an increment of one count in the 32 bit number written to the DDS, a very tiny fraction of a hertz.

All numeric entries are decimal integer variables, so you must enter 5 MHz as 5000000 with no units or commas or spaces or any other punctuation. The numeric keypad serves no function in this program. There is only a single sweep with a measurement and record of data at each frequency step. The useful output appears in the text file.

When running, the program displays:
• The start frequency,
• The end frequency,
• The percent frequency change,
• The instantaneous frequency, and
• The measured power from the log detector at Input 2

The program writes the instantaneous frequency and measured power to the output text file.

TRANSFER.EXE
TRANSFER starts from a command line only. With no parameters, SSW prints a usage statement to explain the parameters it requires.

Usage: TRANSFER filename.txt [Vstart] [Vend] [#steps]

Vstart less than Vend and both between

–2470 to +2470 mV in integer mV with no decimal or units

You must enter an output filename as the first parameter. If you enter anything else, a text file appears on your drive with that as the filename. The file appears in the same folder as the executable unless you redirect it using DOS filename rules.

With no other parameters, TRANSFER runs default settings that sweep from the full range of the D/A Converter in 4096 equal steps from about –2.5 V to +2.5V. If you follow the filename in the command line with only a single parameter less than 2500, the DDS will sweep from that start voltage up to +2.5 V. With additional parameters, you control the end voltage and number of steps. The upper limit to the number of steps corresponds to an increment of one count to the 10 bit D/A Converter.

All numeric entries are decimal integer variables, so you must enter –0.42 V as –420 with no units or commas or spaces or any other punctuation. There are additional error messages if the specified parameters lie outside of the allowed range or would cause a negative going sweep.

The purpose of this program is to apply a sweep or ramp of output voltage to a circuit under test and measure and record the response of the circuit. The output voltage behavior of a circuit in response to its input voltage is called the transfer function.

The numeric keypad serves no function in this program. There is only a single sweep with a measurement and record of data at each voltage step. The useful output appears in the text file.

When running, the program displays:
• The decimal start and end numbers written to the D/A Converter,
• The start voltage,
• The end voltage,
• The number of steps,
• The incremental D/A Converter step count,
• The instantaneous source voltage, and
• The measured voltage at Input 1.

The program writes the instantaneous source voltage and measured voltage to the output text file.

Notes

[1]There is more information about the DDS 30 Daughtercard on the New Jersey QRP Club website at: **www.njqrp.org/dds/index.html.**
[2]For information about the DDS 60 Daughercard, see George (N2APB) Heron's website at: **http://midnightdesignsolu-tions.com/dds60/index.html**
[3]Dr. Sam Green, WØPCE, "Fully Automated DDS Sweep Generator Measurement System," Nov/Dec 2008 *QEX*, pp 13-22.
[4]The Express PCB software and full informa-tion about creating and ordering circuit boards is available on their website at: **www.expresspcb.com/.**
[5]Digikey is one source of the 1206 size surface mount 100 µF capacitors I used. **http://parts.digikey.com/1/parts/1687294-cap-cer-100uf-6-3v-x5r-1206-c1206c-107m9pactu.html.**
[6]George Heron, N2APB, of Midnight Design Solutions, had expressed interest in produc-ing a kit for my original article. See **http://midnightdesignsolutions.com/sweep gen/index.html.**
[7]UserPort is a kernel mode driver for Windows NT/2000/XP that gives user mode programs access to I/O Ports. This makes it possible to access hardware directly from a normal executable in the same way as under *Windows* 95/98/ME. You can learn more about UserPort and download the file for free at: **http://hem.passagen.se/tomasf/UserPort/.**
[8]The data sheet for the Mini-Circuits power splitter/combiner that I used is available on the Mini-Circuits website: **www.minicir-cuits.com/pdfs/PSC-2-2+.pdf.**
[9]The software files for the updated sweep generator measurement system are avail-able for download from the ARRL *QEX* files website. Go to **www.arrl.org/qexfiles** and look for the file **9x12_Green.zip**.
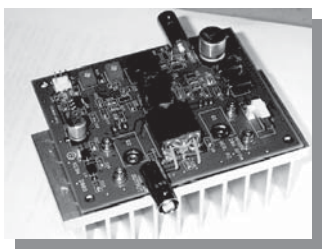[10]You can read the full text of the GNU Public License at: **www.gnu.org/licenses/**.

QEX-

Theodor A. Prosch, DL8PT

Karlstrasse 14/2, 70839 Gerlingen, Germany; **th.prosch@ieee.org**

A Minimalist Approximation of the Hilbert Transform

Digital biphase networks appear to be efficient alternatives to DFFT Hilbert transform algorithms, when sampled real signals have to be converted into quadrature signals or vice versa..

Software defined radios enjoy increasing interest, and many amateurs have started to experiment with zero-IF transceivers. You connect the transceiver I/Q input and output with the corresponding stereo ports of a computer sound card, install a software package, make some adjustments with respect to sideband suppression and carrier reduction, and you are on the air.

Normally, the digital processing of the received signals is based on algorithms that make use of the discrete fast Fourier transform (DFFT). This implies that the signal samples are processed block by block. The associated computational burden is considerable. Windowing, DFFT, filtering, and inverse DFFT demand powerful CPUs. Since the DFFT length predominantly determines the achievable suppression of the unwanted sideband, short block lengths are prohibitive. The resulting latency time and the additive noise due to numerical rounding errors are side-effects that cannot be avoided.

In this article, I will discuss a different approach. From the analog world it is well known that RC polyphase networks are adequate in order to generate quadrature signals respectively to convert I/Q signals into real signals. In practice, analog polyphase RC circuits show the draw-back that the component values must meet narrow tolerances (±0.1% or better). Furthermore, aging and temperature effects deteriorate the performance of RC networks. This is why analog polyphase networks play no significant role in the world of Amateur Radio. The same concept can be transferred into the digital

world, however, avoiding these disadvantages.[1] The resulting algorithms appear to be highly efficient and easy to implement.

A Basic Biphase Network as a Hilbert Transformer

Let us assume that an A/D converter continuously generates samples of a real signal, for example, digitizing the audio signal from a microphone. We split the path coming

[1]Notes appear on page 31.

from the digital source into two branches, where we insert all-pass filters, as shown in Figure 1.

A delay element (a tapped single element finite impulse response (FIR) filter with weight 1) is inserted in the upper branch, and a 2nd order infinite impulse response (IIR) all-pass (BiQuad, with coefficient a) in the other one. When we analyze the phase shift as a function of the normalized frequency $f = f_{Signal} / f_{Sample}$ within the interval dc to Nyquist ($f = f_{Sample} / 2$), we find that the FIR filter produces a linear phase shift in the range from 0
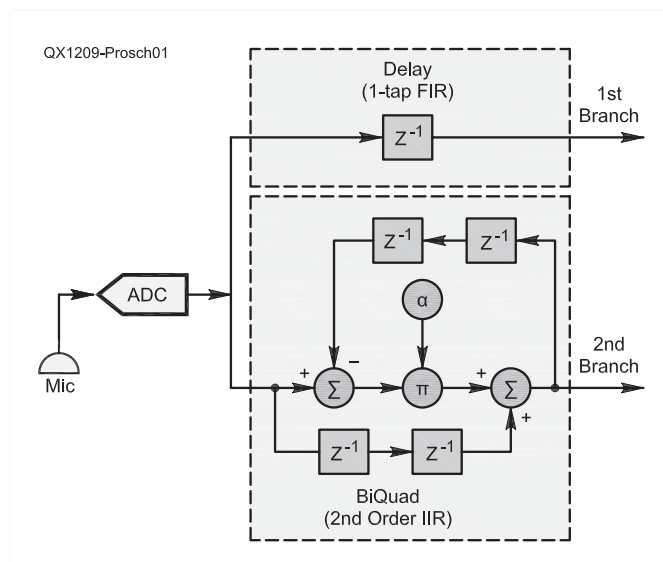

Figure 1 — This is the block diagram of a biphase network, consisting of digital all-pass filters, splitting the signal path into two branches.

to $-\pi$, whereas the BiQuad changes the phase in a nonlinear way, depending on the value of the coefficient a.[2]

$$\varphi_{BiQuad}(f) = -2 \cdot \arctan\left(\frac{(1-a)}{(1+a)} \cdot \tan(2\pi f)\right)$$

Since we are interested in generating two signals differing in phase by $\pi/2$ (regardless what frequency the input signal has), we now will have a closer look at the relative phase. By plotting the phase difference of the output signals — one delayed by the FIR, the other one shifted by the BiQuad — versus frequency, we can visualize the performance of our basic biphase network. Figure 2 shows the result, with the BiQuad coefficient a as parameter.

As you can see, the phase difference approximates $\pi/2$ over a rather wide frequency range when the BiQuad coefficient equals -0.4; the phase differences equal exactly $\pi/2$ at three distinct frequencies (the arrows indicate those occurrences). Obviously, the given combination of the FIR and the IIR all-pass filter approximates the Hilbert transform; more precisely, it approximately generates a Hilbert pair of sampled signals. Notice that by changing the BiQuad coefficient a, either the bandwidth can be increased, which causes a greater deviation from the wanted phase difference $\pi/2$, or the approximation of $\pi/2$ is improved by narrowing the usable bandwidth — the usable bandwidth and the overall phase error are not independent. In order to proceed, we have to extend our basic biphase network. (See Figures 1 and 2).

Cascading BiQuads

The refinement method appears to be straightforward. When we insert further BiQuads into the signal branches, as depicted in Figure 3, more BiQuad coefficients become available, and by properly adjusting them, the overall performance can be improved.

In order to get an indication of the extent that the cascading of BiQuads might be successful, Figure 4 shows the phase and the phase differences of two parallel all-pass filter chains, each containing 2 BiQuads plus one additional delay. What we observe is that the phase difference curve now closely oscillates around $\pi/2$, and the deviation from the ideal straight line becomes quite small. When we analyze this biphase network in the frequency domain, we find a result according to Figure 5. The outcome is quite disappointing, however, in terms of absolute values: the sideband suppression approaches 32.6 dB when the frequency of the first zero equals $(5/1000) \times f_{Sample}$, which corresponds to 240 Hz at a 48 kHz sample rate. The *ansatz* seems to be right, but more BiQuads are
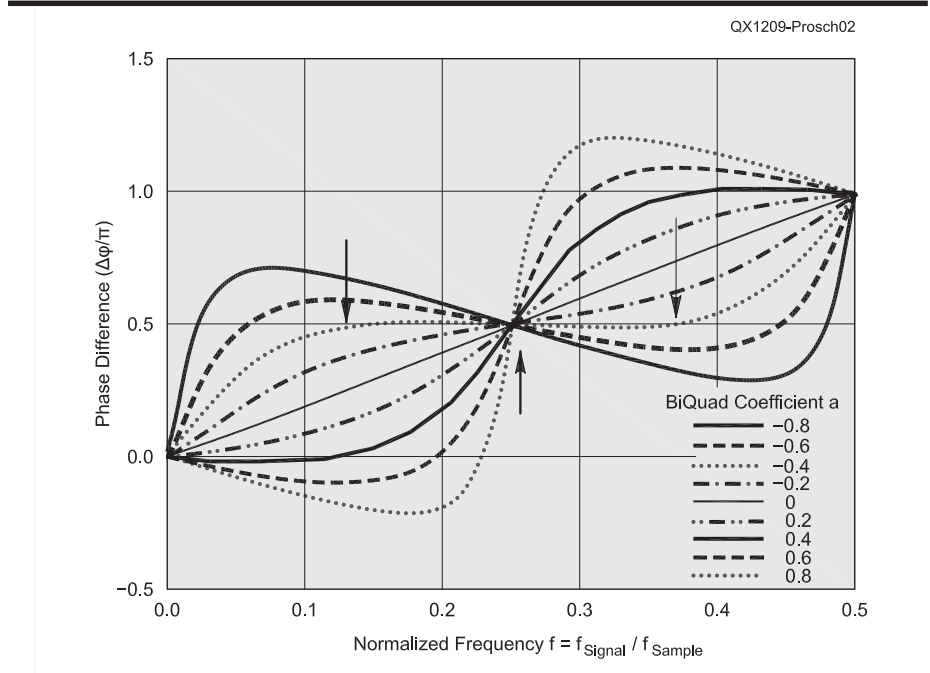

Figure 2 — Differential phase, observed across the outputs of a one-tap-FIR filter and an all-pass BiQuad-IIR filter. (Figure 1 illustrates a biphase network.) The arrows indicate the locations where the phase difference equals exactly 90° when the BiQuad coefficient a is set to −0.4.
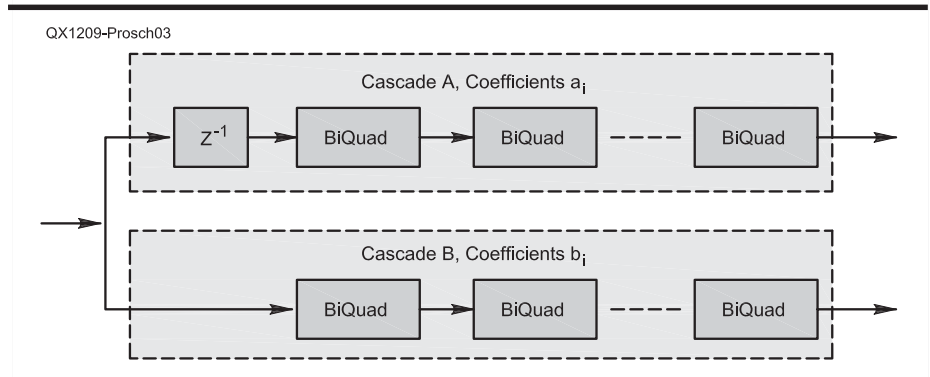

Figure 3 — We can improve the biphase network by cascading multiple BiQuad elements.
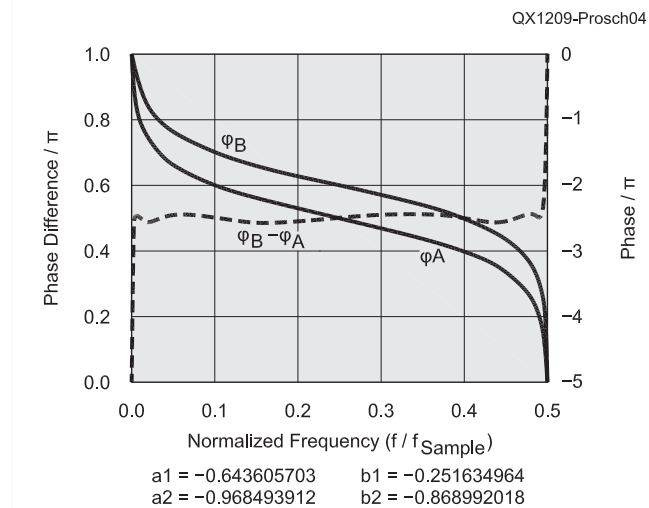

a1 = −0.643605703   b1 = −0.251634964
a2 = −0.968493912   b2 = −0.868992018

Figure 4 — This graph shows the phase and phase difference for 2 × 2 BiQuads. The inset gives the coefficients.

needed! [*Ansatz* represents the mathematical or physical assumptions that form the first order guess to establish the equations. — *Ed.*] In order to find an appropriate length of the BiQuad cascades, including the associated coefficients, some number crunching might be helpful.

In Matthew Heaton's *Parallel Allpass Filters*, we find a design algorithm for frequency selective equi-ripple filters, when digital all-pass filters are used in parallel.[3] The given MATLAB® code can easily be modified for our purpose here. We need to optimize the BiQuad coefficient space by setting the desired phase differences at the filter zeros to $\pi/2$ (instead of to π) within the objective function. This is because we take care of phase differences of 90°, and do not look for amplitude extinction. Then, using *QtOctave*, it takes a couple of minutes and the results are written to hard disc. In Figure 6, the calculated values of the sideband suppression are plotted versus those frequencies, where the first zero of the power transfer occurs. The length of the BiQuad cascades is given as a parameter. As we can see, the choice of 2×5 BiQuads is sufficient in order to achieve more than 60 dB of sideband suppression within the frequency range from $0.001 \times f_{Sample}$ through $0.499 \times f_{Sample}$. Setting the sample rate of the ADC to 48 kHz, this results in a usable frequency range from 48 Hz through 23,952 Hz, and appears to be adequate for Amateur Radio purposes. Table 1 gives three sets of BiQuad filter coefficients whose locations are indicated by circles in the SB suppression versus zero power transfer map of Figure 6.

Simulating a Biphase Network by Means of GNU Radio Companion

GNU Radio and its "*Companion*" are easy-to-use software packages for DSP experiments.[4] In order to study the performance of a digital biphase Hilbert transformer, just a couple of mouse clicks and parameter entries are required. As a basic example, Figure 7 shows the flow graph of a biphase all-pass filter network, consisting of 2×5 BiQuads plus one delay, parameterized according to Table 1, coefficient set III. Two sine wave tone generators are combined, and white noise is added in order to generate an input signal. The two branches of the all-pass filter cascades are connected with the input port of the sink via a float-to-complex number converter. For interactive experiments, the tone frequency and the amplitudes can be varied by sliders. The *GNU Radio Companion* flow graph is available for download from the ARRL *QEX* files website.[5] Maybe it is helpful as a starter for more experimental work.

A simulation result is given in Figure 8. As you can see, the sideband suppression, and the location of the 1st zero frequency agree with the specified values. The equi-ripple performance of the transformer becomes clearly visible. The broadband properties of this simple biphase all-pass filter network are remarkable and compare with a DFFT Hilbert transformer of size 1024 or greater.

Implementation

The simplicity of cascaded BiQuads becomes visible when writing program code. Let us begin with a look at Table 2, where two cascaded BiQuads per branch are taken into consideration. Six registers are allocated on behalf of the first BiQuad, and three more in support of the following one. For the moment, we assume that all registers contain reasonable information. Since the biphase network consists of two cascades, we imagine that we have a second separate register chain in parallel.

Now, for each chain, we modify the register content successively, according to the BiQuad scheme given in Figure 1. First, the sum of registers 1 and 6 is multiplied with the associated BiQuad coefficient, then the content of register 3 is subtracted, and eventually the result is stored in register 4. Then, register 7 is updated and gets the weighted sum of

Table 1
2×5 BiQuad Coefficient Sets (See Figures 3 and 6)

Coefficient Set	I	II	III
1st Zero of Power Transfer,	*0.001*	*0.002*	*0.003*
f / f_{Sample}			
Minimum Sideband-Suppression	*63.2*	*71.6*	*77.5*
(Worst Case) [dB]			
Cascade A			
a1	−0.300619362	−0.250383570	−0.222853404
a2	−0.711123048	−0.640879872	−0.595614470
a3	−0.906224525	−0.863959342	−0.832300087
a4	−0.973150645	−0.955490879	−0.940655103
a5	−0.996210654	−0.993211654	−0.990533091
Cascade B			
b1	−0.089142090	−0.071914730	−0.063038084
b2	−0.529696837	−0.459269214	−0.417480056
b3	−0.832412352	−0.774608418	−0.734309652
b4	−0.948935043	−0.920486871	−0.897795483
b5	−0.987168501	−0.977660080	−0.969358595

QX1209-Prosch05

a1 = −0.643605703   b1 = −0.251634964
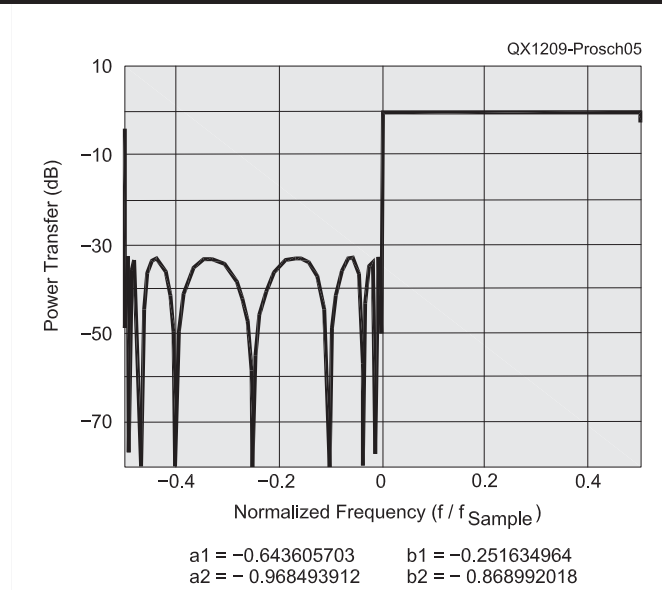a2 = −0.968493912   b2 = −0.868992018

Figure 5 — This graph shows the power transfer for 2×2 BiQuads. The inset shows the coefficients.
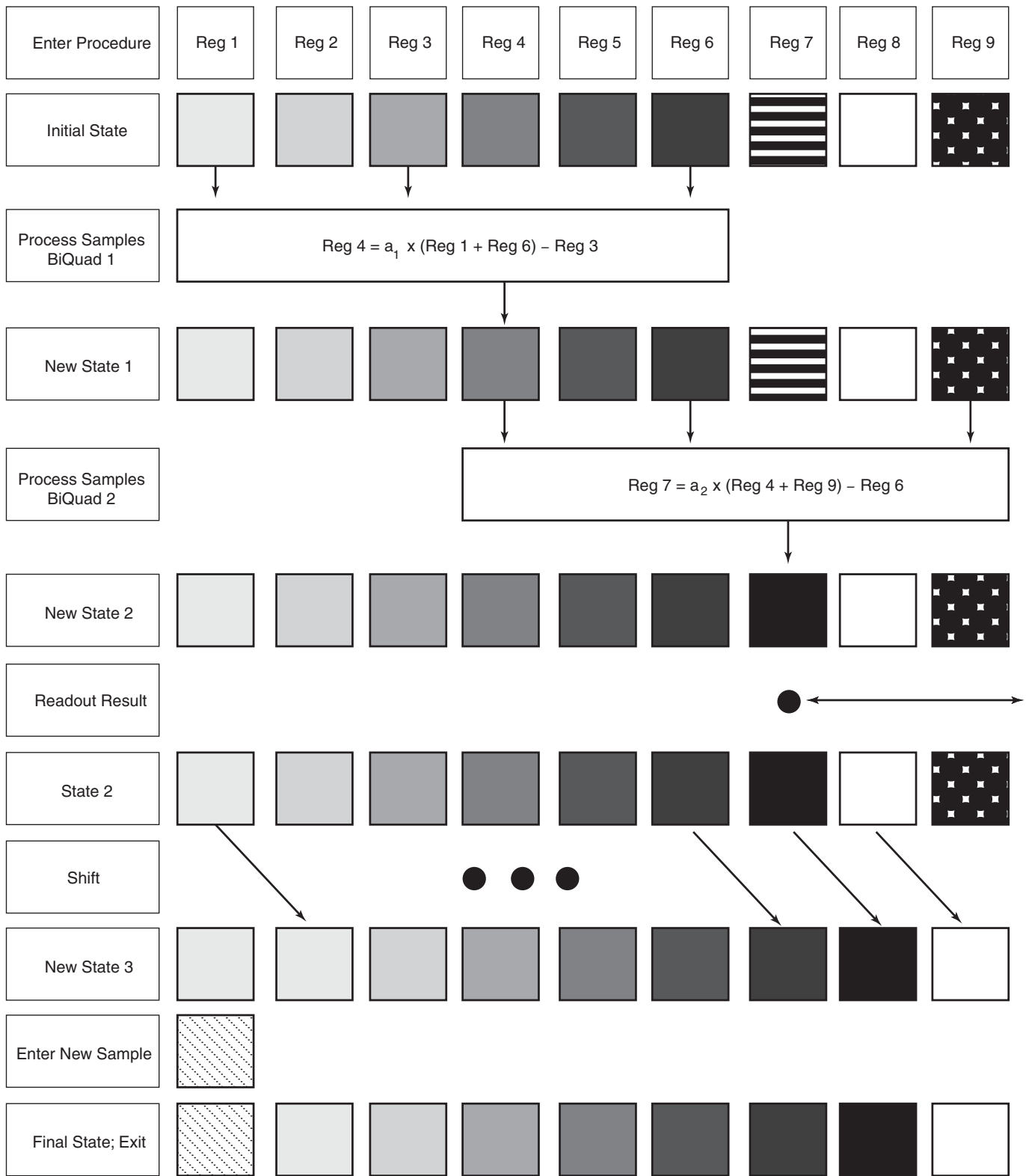
Table 2
BiQuad Algorithm; Cascade of 2 BiQuads

	Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg 6	Reg 7	Reg 8	Reg 9
Enter Procedure									
Initial State									
Process Samples BiQuad 1	$Reg\ 4 = a_1 \times (Reg\ 1 + Reg\ 6) - Reg\ 3$								
New State 1									
Process Samples BiQuad 2				$Reg\ 7 = a_2 \times (Reg\ 4 + Reg\ 9) - Reg\ 6$					
New State 2									
Readout Result							●		
State 2									
Shift	● ● ●								
New State 3									
Enter New Sample									
Final State; Exit									

Table 3
Non-optimized Piece of Pseudocode to Illustrate the Algorithm

```
...
/* for each sample value "in_ad" from the ADC
 * move "in_ad" and its delayed copy to registers         */
reg_a[0] = in_ad_delayed;
reg_b[0] = in_ad;
/* keep the sample value for one time tic
 * since we need it as delayed version                     */
in_ad_delayed = in_ad;
/* successively update the registers of two cascades,
 * each cascade consisting of N BiQuads                    */
for ( i = 0; i < N; i++ )
        {
                ii = 3 * i ;
                reg_a[ii+3] =
                koeff_a[i] *(reg_a[ii] + reg_a[ii+5]) - reg_a[ii+2];
                reg_b[ii+3] =
                koeff_b[i] *(reg_b[ii] + reg_b[ii+5]) - reg_b[ii+2];
        }
/* get the Hilbert pair, i.e. the I- and Q-samples         */
out_I = reg_a[3*N];
out_Q = reg_b[3*N];
/* shift the register content */
for ( i = 3 * N + 2; i > 0; i--)
        {
                reg_a[i] = reg_a[i-1];
                reg_b[i] = reg_b[i-1];
        }
...
```
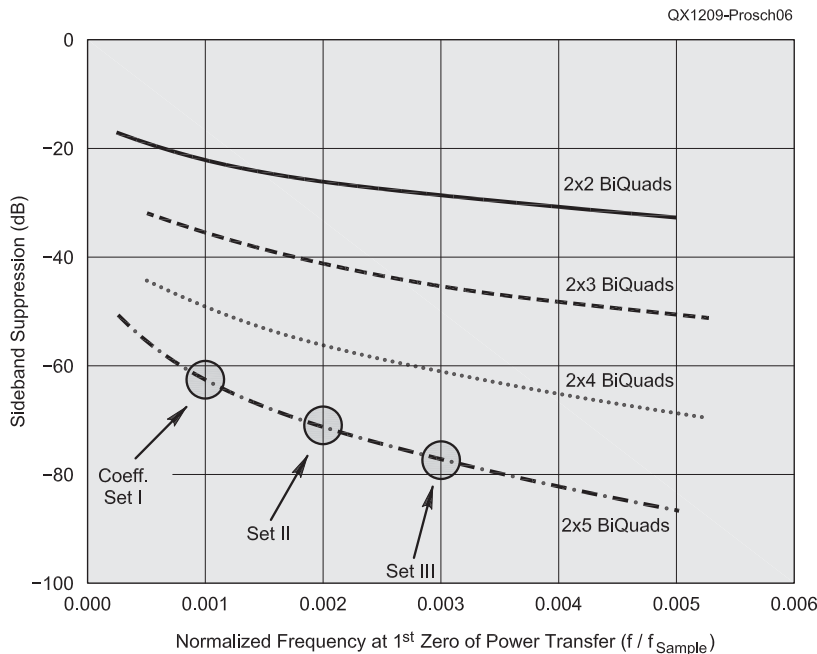

Figure 6 — Here is a graph of the sideband suppression, minimum usable frequency, and BiQuad cascade length. The BiQuad coefficients sets I through III are given in Table 1.
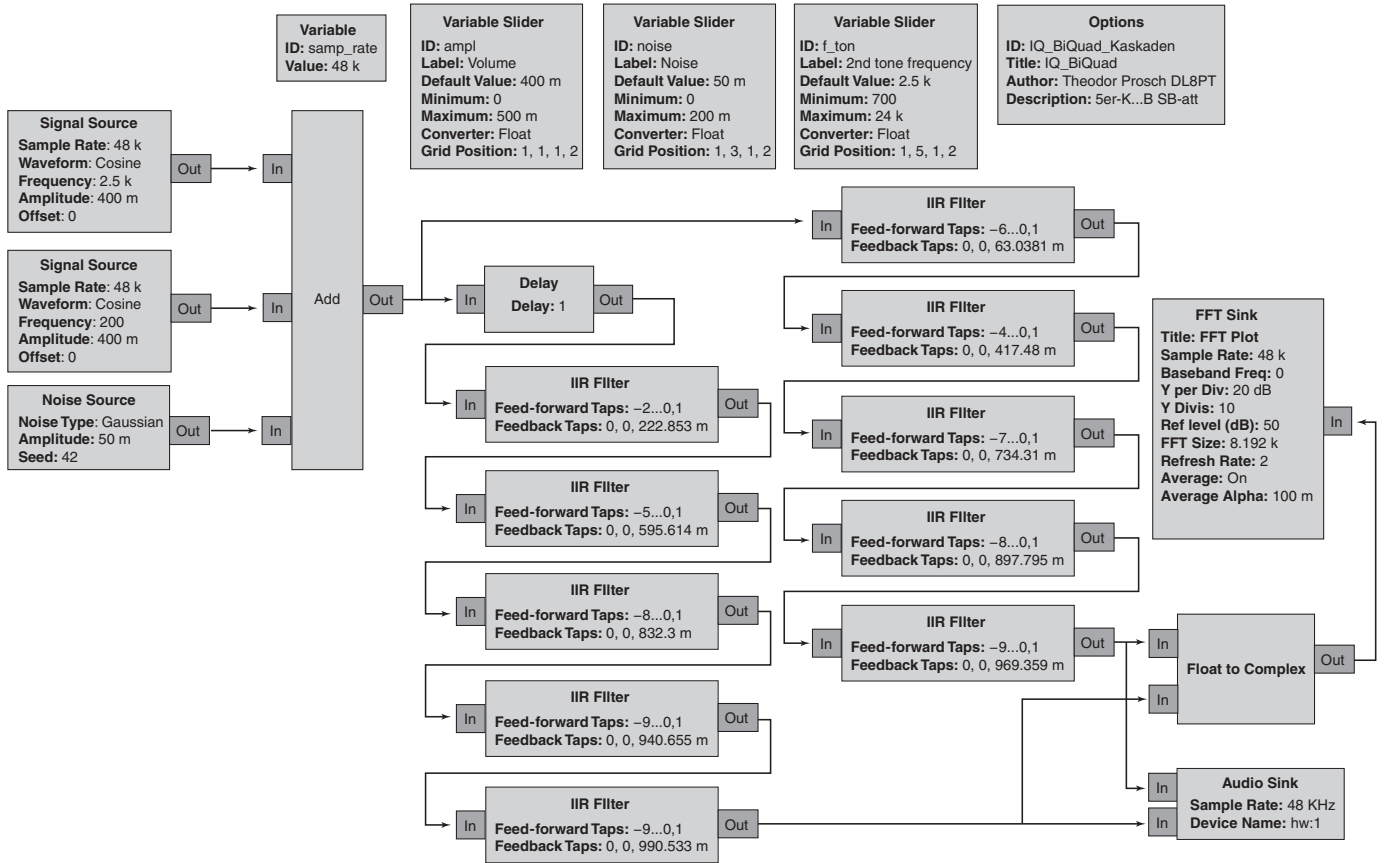
QX1209-Prosch07

Figure 7 — This chart is an example from the *GNU Radio Companion*. It is a Flow Graph, simulating a 2 × 5 BiQuad network.
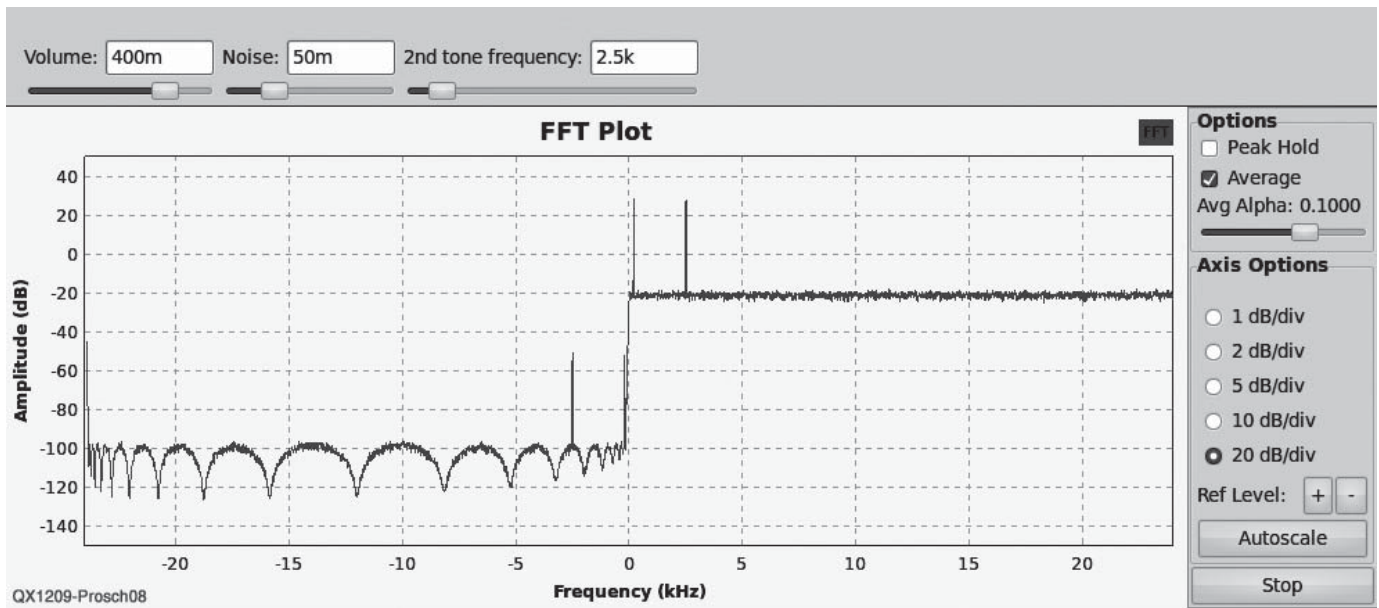

Figure 8 — Here is the simulation result from *GNU Radio Companion*, based on the flow graph of Figure 7. The input signal consists of two tones (200 Hz and 2.5 kHz), and white noise. The sideband suppression is better than 77 dB.

registers 4 and 9, minus the content of register 6. If the chains had N BiQuads, we would continue and update registers 10, 13, … 3 × N+1 accordingly.

In our example, we pick up the quadrature components, I and Q from registers 7. After having finished the add-multiply-subtract operations, we update the registers, beginning at the upper index. The register contents are shifted from left to right, chain by chain. Eventually, we transfer the latest sample value into register 1 of the first chain, and the delayed sample into register 1 of the second chain. Then the procedure repeats.

Table 3 is a non-optimized piece of pseudo code in order to illustrate the algorithm. The algorithm is initiated by setting the content of all registers to zero.

When we count the floating point operations per input sample needed to generate a Hilbert sample pair, the result is surprisingly low. In the case of 2 × 5 BiQuads, we need 10 multiplications and 20 additions. This is a very moderate computational burden, compared to a 1024 DFFT Hilbert transform.

Conclusion

Digital biphase all-pass filter cascades are capable of replacing traditional DFFT Hilbert transform algorithms. Their computational overhead is low in terms of required storage capacity and floating point operations per sample. The latency time does not depend on DFFT block lengths.

For design purposes, the sideband suppression and the usable frequency range were analyzed, whereby the required number of cascaded BiQuads is given as a parameter. Equi-ripple BiQuad cascades of length 5 are adequate for Amateur Radio purposes, when 60 dB of sideband suppression (or better) is required. Computer simulations indicate that the discussed biphase networks are numerically stable and accomplish the design specifications. For implementation purposes, I have provided a *C*-code example for BiQuad cascades of length N. Three coefficient sets are tabulated in order to allow for practical experiments.

Be careful about processing non-audio signals by means of biphase all-pass filters.

Due to the non-linear phase shift of IIR filters, the group velocity of biphase networks is not constant over the specified frequency range. There might be critical cases (such as high-order QAM or spread spectrum signals). Then, an equalizing all-pass filter should be taken into consideration, also consisting of cascaded BiQuads.

Theodor A. Prosch received a masters degree in physics from the University of Freiburg in 1972, and a doctorate from the University of Cologne in 1982. He has held research positions as a scientist at the Ionospheric Institute in Breisach, and at the Institute of Geophysics in Cologne, where he worked on remote sensing of the atmosphere, imaging polarimetry and solar radiation transfer.

In 1983 he joined Sueddeutscher Rundfunk (SDR), where he was involved in the development of wave propagation models and computer aided methods for spectrum engineering and frequency management. In 1986 he became head of the BC radio network planning section of SDR, in 1988 of Suedwestrundfunk (SWR) and in 1999 CEO of Digital Radio Suedwest GmbH (DRS), a commercial BC network operator. Since he retired from business in 2010, he enjoys his private electronics lab, plays with software defined transceivers, and operates his home made QRP station. Theodor has been a licensed Amateur Radio operator since 2006, and is a member of IEEE and the Deutsche Physikalische Gesellschaft.

Notes

[1] Katja's website has a useful discussion about the Hilbert transform and all-pass filters: **www.katjaas.nl/hilbert/hilbert.html**.
[2] Artur Krukowski, Izet Kalet, *DSP System Design: Complexity Reduced IIR Filter Implementation for Practical Applications*, Boston, Kluwer, 2003, ISBN: 1-4020-7558-8.
[3] Matthew G. Heaton, *Parallel All-Pass Filters*, Department of Computer Science and Electrical Engineering, University of Queensland, 1999.
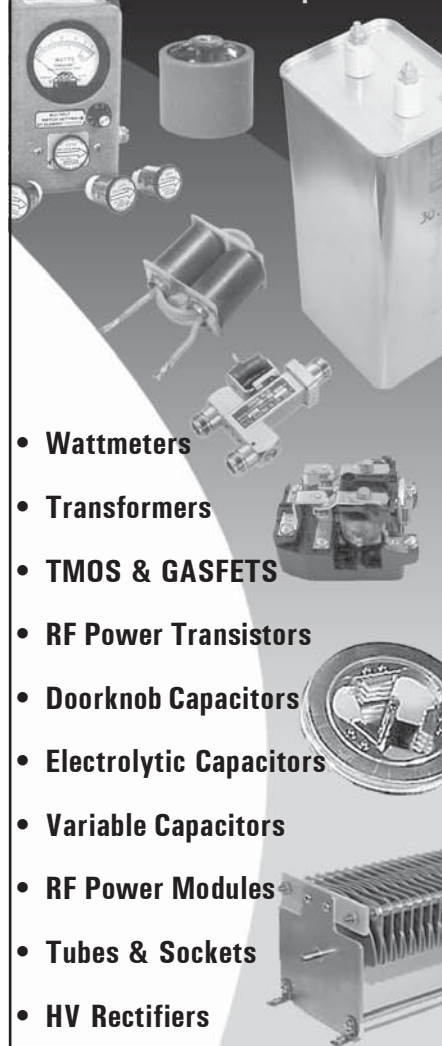[4] For more information about *GNU Radio* and the associated *GNU Radio Companion*, see: **http://gnuradio.org/redmine/projects/gnuradio/wiki**.
[5] The example *GNU Radio Companion* flow graph is available for download from the ARRL *QEX* files website. Go to **www.arrl.org/qexfiles** and look for the file 9×12_Prosch.zip.

QEX

Ray Mack, W5IFS

17060 Conway Springs Ct, Austin, TX 78717; **w5ifs@arrl.net**

SDR: Simplified

More Filter Activites

FIR Filter Recap

Figure 1 shows a graphical representation of a 5 tap FIR filter. The coefficients are those calculated for a low pass filter with a cut-off frequency of 1100 Hz with 8000 samples/s and 40 dB stop band attenuation. I used the FIR_filter_generator.exe program from the last installment.[1] Table 1 shows the coefficients in 16 bit integer and floating point representation. The formula for the filter is expressed as:

$H(z) = 0.02316\ z^0 + 0.16647\ z^{-1} + 0.27499\ z^{-2} + 0.16647\ z^{-3} + 0.02316\ z^{-4}$

I have always found "z" notation to be very confusing. It is a lot easier to visualize with a picture such as Figure 1, where each "z" value is just the sample contained in one of the shift register positions. H(z) is just what comes out of the adder at the bottom of the filter in Figure 1. H(z) is the sequence of numbers that are calculated by our DSP. We put H(z) into a DAC and low pass filter, and get H(t) which is now a continuous time function that we can hear through a speaker or watch on an oscilloscope. The notation "H(z)" is the form you will commonly see in an engineering text for the response of a filter.

Filter Response Calculation

That is all very interesting, but it is not terribly useful for figuring out if our filter is really going to do what we want. The Fourier series that corresponds to our filter is what describes the actual frequency response for all possible input frequencies. This is similar to what happens when we use the Fourier series to create a square wave:

$G(t) = \sin(2\pi ft) + 1/3 \sin(3 \times 2\pi ft) + 1/5 \sin(5 \times 2\pi ft) + 1/7 \sin(7 \times 2\pi ft) + \ldots$

If you add up all of the harmonics, you get an exact square wave. If you stop after harmonic 19, for instance, you get a waveform that is close to a square wave but shows the Gibbs phenomenon. (You might want to use *Octave* or *Gnuplot* to see what happens.) This process takes information in the frequency domain (1, 1/3, 1/5, …) and converts it to a time domain representation. We have done an inverse Fourier transform to transform frequency domain information to time domain information.

In order to get a true representation of the filter frequency response, we just need to replace each "z" with a representation of the Fourier series for the filter. The task we are doing is a Fourier
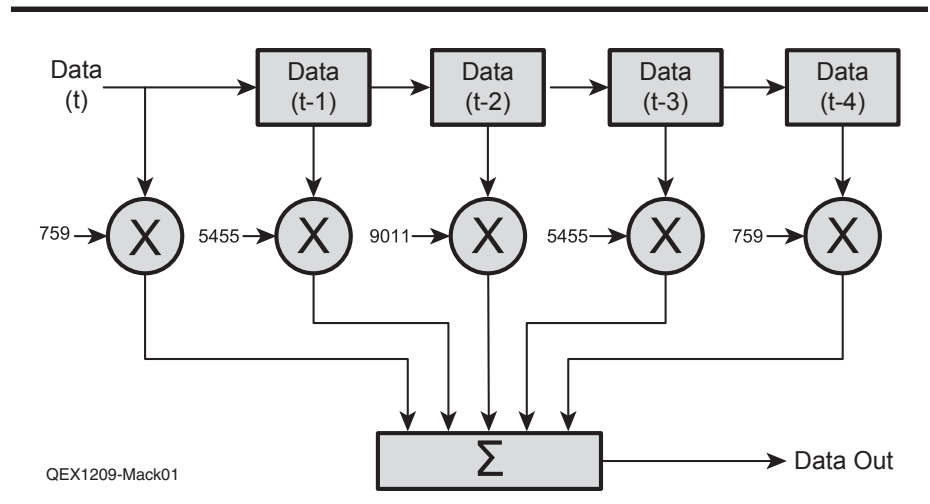
transform. Remember that the filter coefficients describe the operation of the filter in the time domain and a Fourier transform changes time domain information into frequency domain information. The transform of our filter looks like this:

$H(f) = 0.02316\ (\cos(0 \times 2\pi f) -j \sin(0 \times 2\pi f)) + 0.16647\ (\cos(1 \times 2\pi f) -j \sin(1 \times 2\pi f)) + 0.27499\ (\cos(2 \times 2\pi f) -j \sin(2 \times 2\pi f)) + 0.16647\ (\cos(3 \times 2\pi f) -j \sin(3 \times 2\pi f)) + 0.02316\ (\cos(4 \times 2\pi f) -j \sin(4 \times 2\pi f))$

Since cos (0) is one and sin (0) is zero, this simplifies to:

$H(f) = 0.02316 + 0.16647\ (\cos(1 \times 2\pi f) -j \sin(1 \times 2\pi f)) + 0.27499\ (\cos(2 \times 2\pi f) -j \sin(2 \times 2\pi f)) + 0.16647\ (\cos(3 \times 2\pi f) -j \sin(3 \times 2\pi f)) + 0.02316\ (\cos(4 \times 2\pi f) -j \sin(4 \times 2\pi f))$

Remember that each $\cos(x \times 2\pi f) -j \sin(x \times 2\pi f)$ is just a single sine wave represented in rectangular form (x and y) rather than polar form (amplitude and phase angle). In signal processing, we actually refer to rectangular form as I (the cos term) and Q (the sin term) rather than what we did in algebra class with x and y.

Putting the Software in SDR

I did an on-line search and found no program that automates the process of calculating a set of filter coefficients and displaying the resulting filter response. That doesn't mean one doesn't exist, but it does not show up in a search. *MATLAB* and *Octave* each have a function that will com-

pute the frequency response if you give it an array holding the filter values. Both require a fair amount of programming to do the computations and display. I find both as incomprehensible as most DSP math!

Since no program exists, I have created a program that incorporates the filter calculations with a Kaiser window and then displays the response. Figure 2 shows the main window for the program with representative values filled in. It allows you to enter the same information as the *C* program from the May/June column. Figure 3 shows a representative output window for the program. The program is available on the ARRL *QEX* files website, and the source code is included.[2]

I wrote the program in *Visual Basic 2010 Express* because it is the easiest environment I know to write a *Windows* program. This is not your father's *BASIC*. It isn't even very much like *Visual Basic 4* (the last one I used regularly) or *Visual Basic 6*. In 2008, Microsoft did a major re-
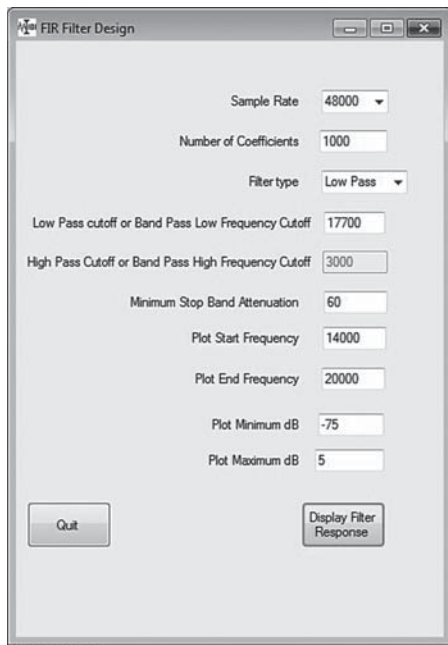

Figure 1 — This diagram is a z-space representation of a simple 5 tap FIR filter.

QEX1209-Mack01

Table 1
Coefficients for a 5 TAP FIR Filter

Coefficient	16 bit Integer	Floating Point
0	759	0.02316
1	5455	0.16647
2	9011	0.27499
3	5455	0.16647
4	759	0.02316

[1]Notes appear on page 37.

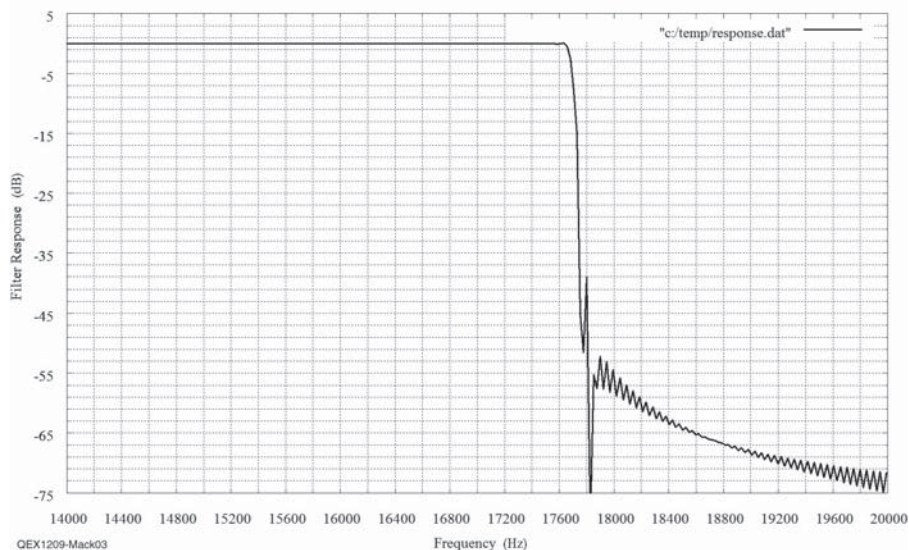Figure 2 — This screen shot is the main window of the filter response program.

Figure 3 — Here is a representative output window of the filter response program.

Figure 4 — This block diagram shows the software SSB transmit generator using the filter method.

write of *Visual Basic* to make it significantly more object oriented and to incorporate the .NET framework as an integral part of the language. Perhaps the biggest change from all previous *BASIC* systems is that all arrays start at index zero instead of index one. If you are a *C* programmer, you will find the new *Visual Basic* to be a minor change in syntax. It took me about 10 minutes to modify the syntax of the *C* console program from the last column to do the Kaiser window and coefficient calculations in *Visual Basic*.

The .NET features for doing *Windows* programs are a real boon for writing programs with one exception. I find the Chart tool to be totally incomprehensible. I was able to get a barely useful X-Y plot of the filter response after 3 days of fighting the many layers of parameters. I figured out how to do the same tasks in *Gnuplot* in about 3 hours when I first started learning that tool. Fortunately, *Gnuplot* comes with an executable image that can be run from another program to simply pop up a new window. I have incorporated that mechanism into our program. The program can still use some improvement to make it easier to have *Gnuplot* do the display. It is left as an "exercise for the reader" until I get a chance to get back to improve it. For now, we need to get back to making a radio!

An SSB Transmit Generator

Now that we have the ability to create a sine wave using the DDS program and to create a sharp cutoff filter, we can create a filter method SSB generator. The structure of our program is the same as if it were implemented in analog hardware. Figure 4 shows the block diagram of the system. The program operates in a serial fashion: first

the baseband filter limits the audio to a band of 300 Hz to 3 kHz, second it computes the phase value for the carrier, third is the multiplication for the balanced mixer, and finally the undesired sideband is removed. The filter response program shows that the low audio cutoff is only useful with 200 taps or more. At 100 taps the rejection is only on the order of 12 dB below 100 Hz. Likewise, the opposite sideband filter requires on the order of 700 to 1000 taps to give approximately 60 dB of opposite sideband suppression. The large number of taps also makes the skirts very steep, so that we can use the filter to also further reduce any carrier feed through.

There are a number of compromises we could make if we were going to make a real transmitter. The first is setting the lower frequency limit for audio. Simply using a dc block in the analog portion of the audio chain will set a lower boundary on the frequency. The response will be zero at 0 Hz and rise very rapidly to the frequency we set. This reduces the need for a sharp cutoff in DSP. The close in rejection of audio above 3 kHz is 45 dB or more. Additionally, there

is almost no energy above 3 kHz in the human voice, so energy in that region will likely be at least 60 dB below the lower frequencies after filtering. Limiting the higher frequencies allows us to use a 6 kHz wide sideband selection filter instead of the normal 3 kHz filter to get better skirt response. A low pass or high pass filter would also work and give approximately the same skirt response, but we want to be sure to eliminate any residual energy at baseband in the case of a lower sideband transmission. The wider bandwidth limits the lower frequency for our carrier. We want the carrier frequency to be as high as possible in order to limit image response when we up convert to our final RF signal. This experimental transmitter is not really suitable as a real transmitter because the CODEC limits the highest frequency to 20 kHz. It is truly just an audio CODEC. When I have more time, I would like to go back to the Blackfin Stamp so that I can use the DAC08 at 1 MHz sample frequency for transmit and build an ADC board that can also sample near 1 MHz. My goal is to make a 6 m sideband rig to fill in
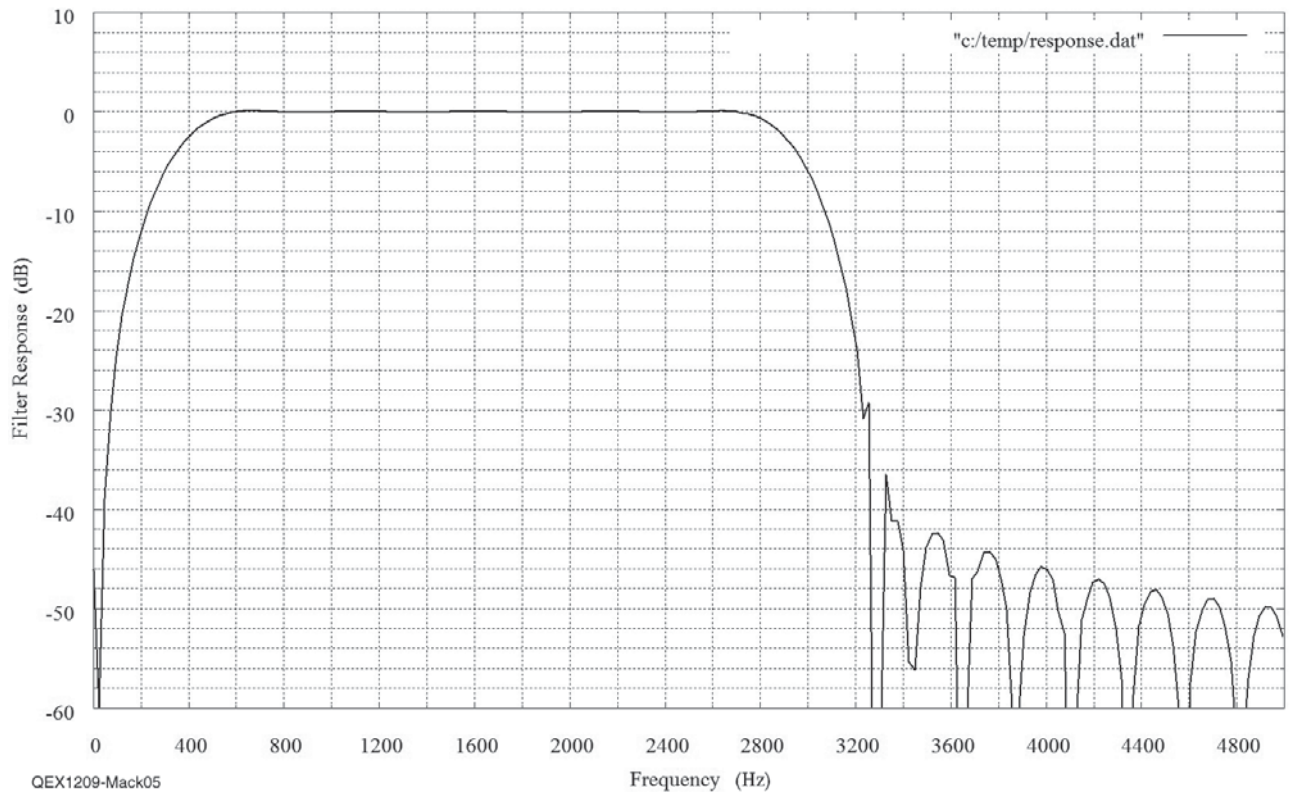
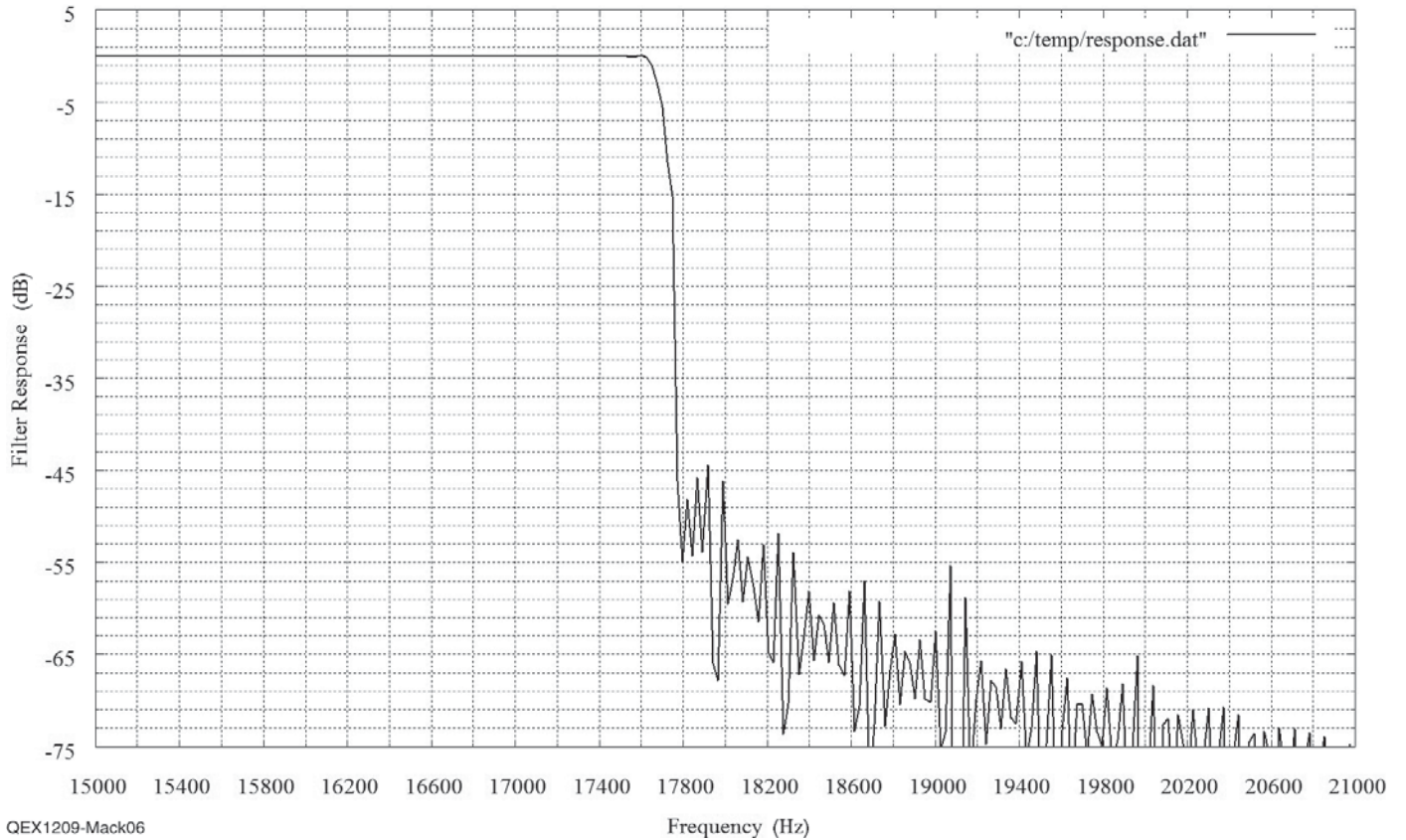**Figure 5 — This graph is the response of the 200 tap baseband filter.**

Figure 6 — This graph is the response of the 700 tap sideband selection filter for a carrier at 18 kHz. It shows the response of the opposite sideband.
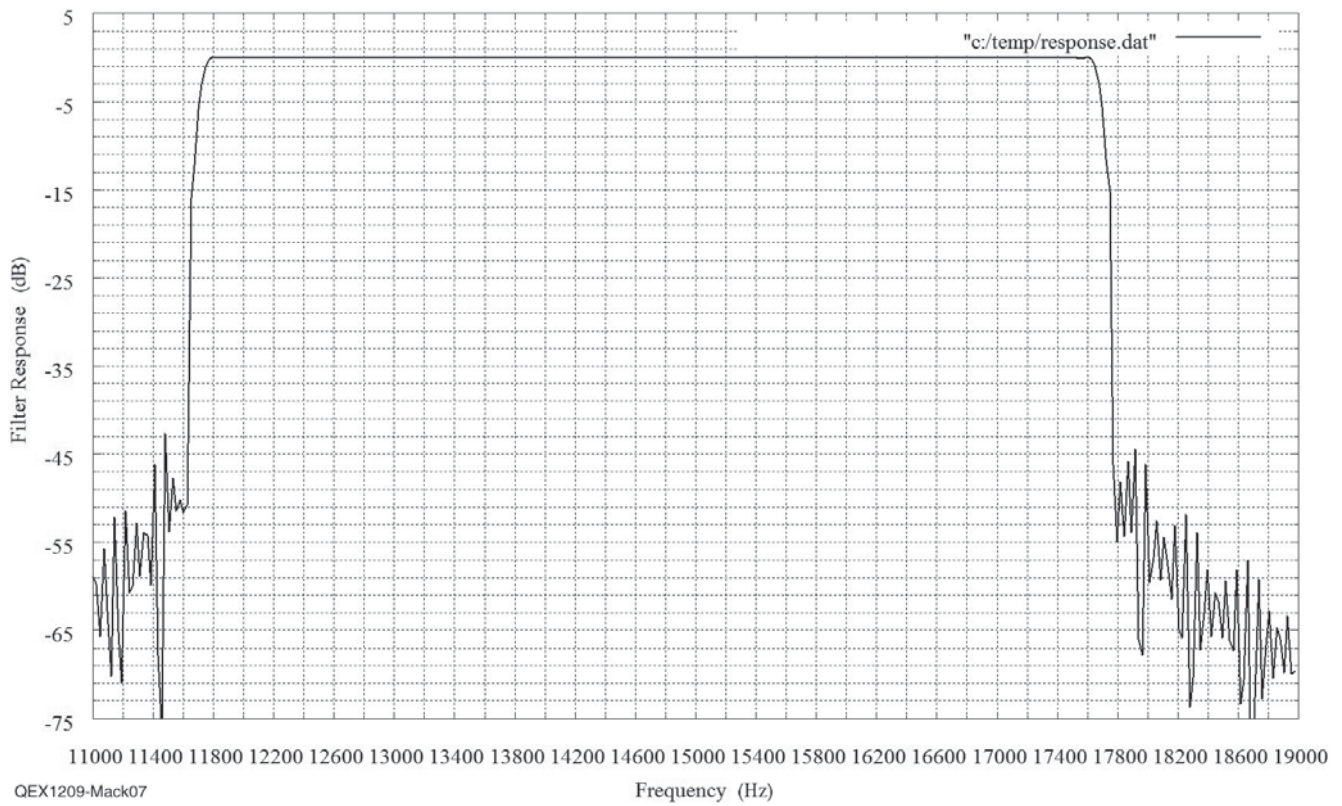
Figure 7 — Here is the response of the 700 tap filter, showing a wider frequency view. The filter is 6 kHz wide to allow for a steep skirt on the carrier side. The 6 dB cutoff point is set to 300 Hz away from the carrier.
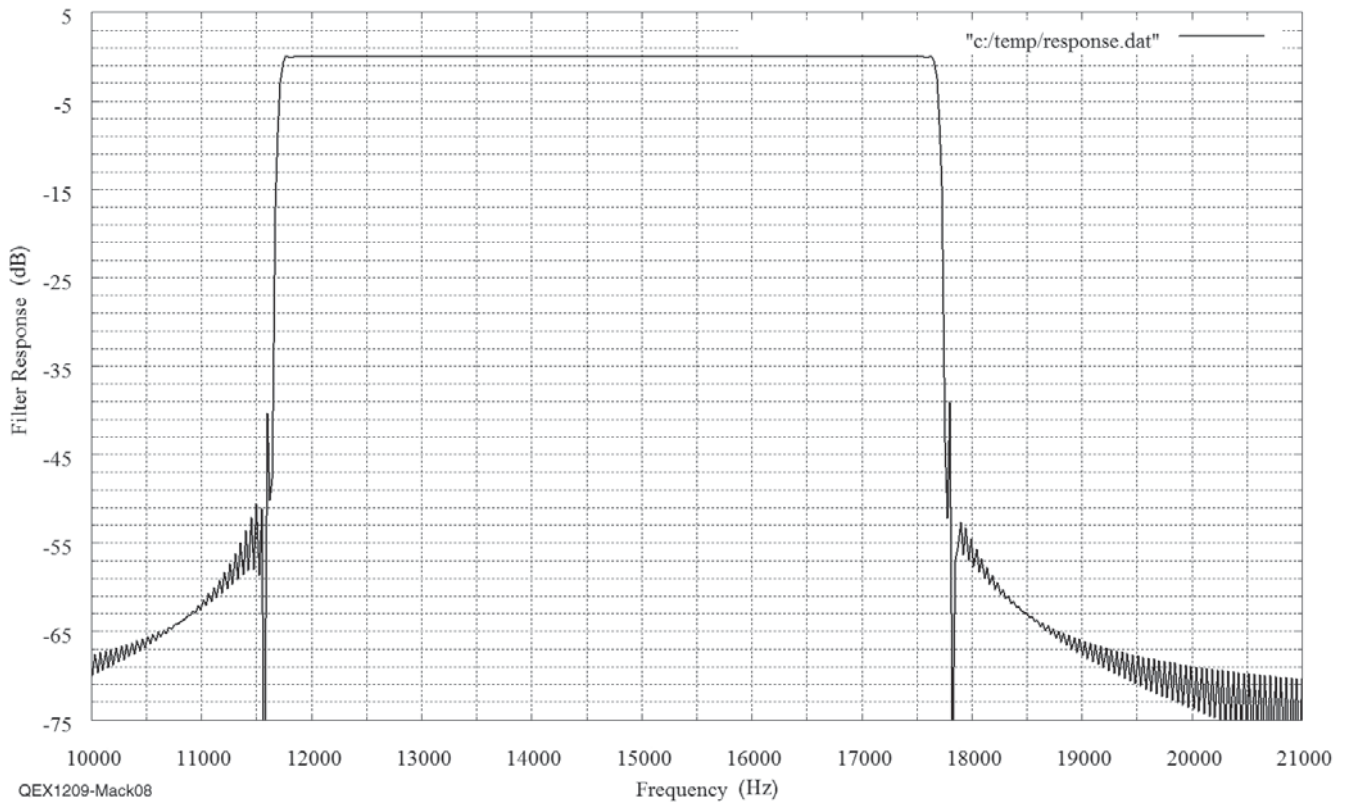

Figure 8 — The response of a 1000 tap filter is represented in this graph. It shows the trade off of more calculations versus out of band rejection.
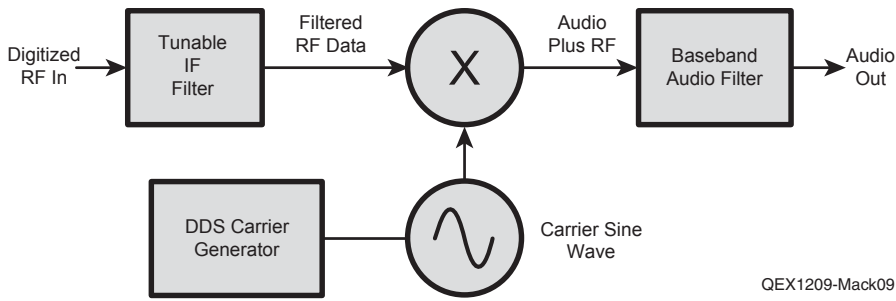
Figure 9 — Here is a block diagram of the SSB receiver using DSP.

Figure 10 — Odd and even square waves in the time domain.

one of the holes in my station.

We have a practical limit with respect to the number of taps in the filters. Each tap requires one multiply-accumulate operation, which is an MAC in the DSP world. (Unfortunately, MAC also means Media Access Control if you are a networking hardware person and an object lesson in why three letter initials can be a bad idea!) The DSP is capable of one MAC for each MHz of clock frequency for each portion of the hardware chain. The C5535 has two multipliers in the chain, so it is capable 200 million MACs (TI uses the initials MMAC) if the processor is running at 100 MHz and the library software uses both multipliers. Our experiments are running the CODEC at a 48 kHz sample rate, so our transmitter with 200 taps for audio and 1000 taps for sideband selection will need 57.6 MMACs to do its job. Regular software also requires one or two instructions per clock cycle. If we take the worst case of one multiplier used and one instruction per cycle, we have approximately 40 per cent of the DSP left over for regular computing with a 100 MHz clock. The best case is 70% of capacity left over.

An SSB Receiver

Figure 9 shows how we can reverse the steps above to create an SSB receiver. We simply run the "RF" from a down converter into one channel of the CODEC. We set the "RF" filter frequency to select the desired signal, multiply it with the output of our DDS generator, and then filter the resulting audio. The audio filter produces better audio if it is a band pass filter than just a low pass filter because the RF filter will allow some opposite sideband energy to pass at very low frequencies.

Odd and Even Functions

We need to get a little closer to the math in preparation for dealing with the 90° phase shift that is important to a lot of DSP operations. At the beginning of this column, we looked at the Fourier series for our filters in both the time domain and the frequency domain. The series is a general case where the phase and amplitude are arbitrary. The case of a square wave is more than just a curiosity. Figure 10 shows two different square waves with appropriate time scale.

They are identical in frequency and amplitude and both extend from negative infinity to positive infinity. They differ in phase by 90°, though. The top waveform is called an even function because the value of the waveform at 1 second is the same as the value at −1 second. The bottom waveform is called an odd function because the value of the waveform at −1 second is equal to the value at 1 second but multiplied by −1. This has implications for the Fourier series for the two waveforms. The even function has the Fourier series:

G(t) = cos (2πt) − ⅓ cos (3 × 2πt) + ⅕ cos (5 × 2πt) − ⅐ cos (7 × 2πt) + … The odd function has the Fourier series:

G(t) = sin (2πt) + ⅓ cos (3 × 2πt) + ⅕ cos (5 × 2πt) + ⅐ cos (7 × 2πt) − …

Our implementations of FIR filters have always been even functions in the frequency domain. For that reason, the filter coefficients have always been mirror images around the center. In the 5 tap example, tap one was the same as tap three and tap zero was the same as tap four. Since it is an odd size, you can think of tap 2 being the same for zero and "minus zero" since it is exactly in the center. The coefficients have exact mirror image pairs for an even number of coefficients. Figure 11 shows the ideal filter response of the 5 tap filter for positive and negative frequencies.

Mathematicians call certain phenomena "degenerate cases." A point is a degenerate case of a circle: it has a radius of 0. In DSP we have a degenerate case called
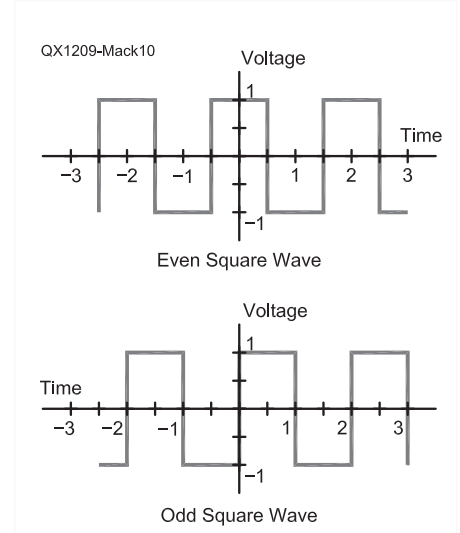
an all pass filter. It passes all frequencies with unchanged amplitude. There are two cases of all pass filters, however: one is odd and one is even! Figure 12 shows the two responses.

We generate the filter constants for an FIR by taking the Fourier transform of the frequency response. The transform of our even all pass filter is another degenerate case. To get out exactly what you put in, you just multiply each sample by one (cos 0). This is shown in the top of Figure 13, where we end up with a single coefficient. When we put in a cosine wave we get the cosine wave back out. Our odd all pass filter has a Fourier series that contains only sine terms rather than cosine terms. Those coefficients correspond exactly to our Fourier series for the odd square wave. The positive values are 0, ⅓, 0, ⅕, 0, ⅐ … Since it is an odd function, the values for our DSP implementation will be −⅐, −0, −⅕, −0, −1/3, −0, −1, 0, 1, 0, ⅓, 0, ⅕, 0, ⅐.

The important implication for the odd function all pass filter is that putting in a cosine wave at any frequency will produce a sine wave with the exact frequency of the
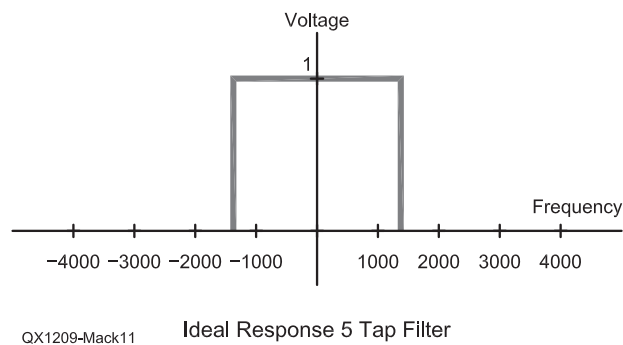

Figure 11 — This is the Ideal filter response for the 5 tap filter, showing its even order characteristic.
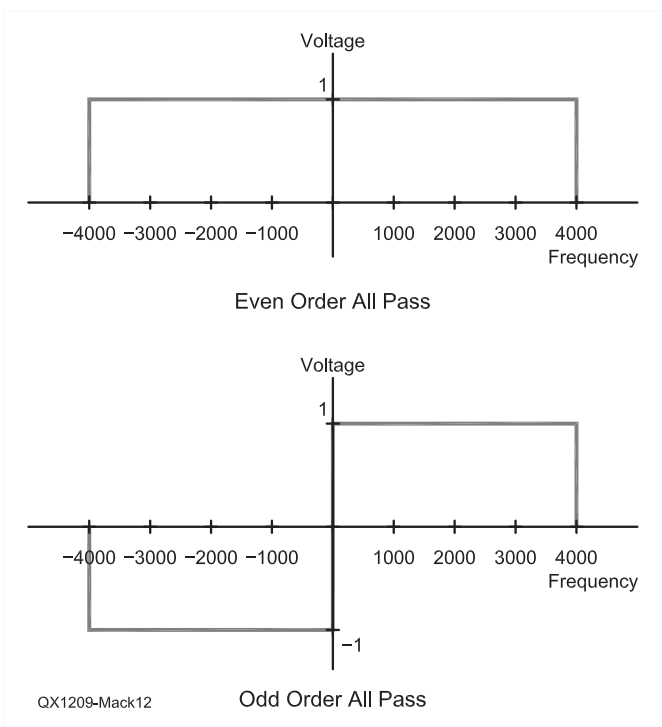
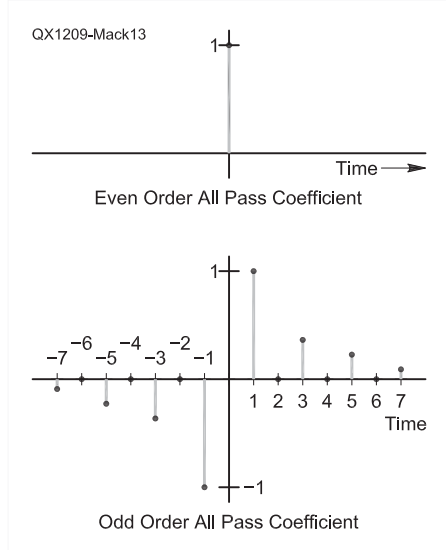Figure 12 — Here are even order and odd order all pass filters.

Figure 13 — The filter coefficients for even order and odd order all pass filters are shown on this graph.

Figure 14 — This is the amplitude response of a 247 tap Hilbert Transform with a 48 kHz sample rate.

input. This is an exact 90° phase shift! The odd order all pass filter is called a Hilbert Transform when implemented in DSP.

The exact 90° phase shift is exactly what we need for many RF signal generation tasks. There is no free lunch, however! Remember that there is that nasty discontinuity in the frequency response at dc. We saw before that any sharp change in the frequency domain causes the Gibbs phenomenon to appear. It is the same here. We get a constant phase shift, but the amplitude is not constant and rings at the frequency of the discontinuity (0 Hz and ½ fs in this case). We saw before that increasing the number of filter coefficients to a very large number will compress the ringing in the frequency response to a small portion of the total, but will not eliminate the 8% overshoot. A useful Hilbert transform will require a large number of coefficients to move the bulk of the amplitude error below our lowest frequency of interest. Another limitation of the Hilbert transform is that it requires an odd number of coefficients.

The Phasing Method

Chapter 11 of *Experimental Methods in RF Design* presents a good description of the use of the phasing method in DSP for an 18 MHz transceiver.[3] Chapter 9 describes the theory of the phasing method with equations to show how amplitude error and phase error affect opposite sideband suppression.

The heart of the phasing method is the same amplitude but 90° phase difference between the two channels. Any deviation from exactly 90° and any amplitude imbalance cause less than perfect opposite sideband suppression. The rule of thumb is that 0.1 dB of amplitude imbalance plus one degree of phase error will limit sideband suppression to –40 dB. Those limits are representative of what is possible with analog components with temperature fluctuations and normal component variations. DSP eliminates the issues with component changes from ideal. We get response that is only limited by the precision of the ADC and DAC and the number of taps we choose to implement. Since a Hilbert Transform does not have phase error, the opposite sideband suppression is determined solely by amplitude imbalance. Figure 14 recreates the example from Chapter 11 and shows that a 247 tap filter at 48 kHz sample rate will have 0.02 dB amplitude imbalance near 300 Hz. That will yield opposite sideband suppression of 52 dB.

Every other coefficient of the Hilbert Transform is zero, as is the center, so a 247 tap transform will only need 123 multiply-accumulate operations if implemented efficiently. This is a significant savings over the filter method example given earlier, which required 1000 multiply-accumulate operations for equivalent performance. The TI library contains a Hilbert transform function, but it is not clear if it implements a cycle saving algorithm different from an FIR filter. The software for this issue does not contain any phasing examples. I hope to include that next time.
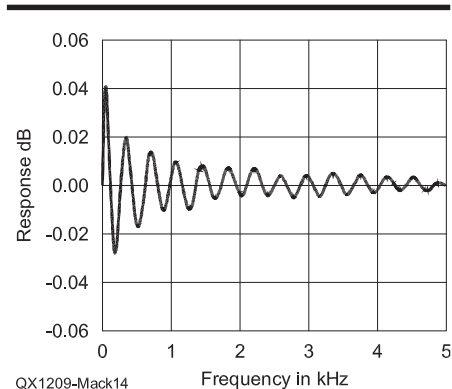
Notes

[1]Ray Mack, W5IFS, "SDR:Simplified, Filter Design Program," May/Jun 2012 *QEX*, pp 40-44. The software files described in that column are available for download from the ARRL *QEX* files website. Go to **www.arrl.org/qexfiles** and look for the file **5x12_Mack_SDR.zip**.
[2]The software for this column is available for download from the ARRL *QEX* files website. Go to **www.arrl.org/qexfiles** and look for the file **9x12_Mack_SDR.zip**.
[3]Wes Hayward, W7ZOI, Rick Campbell, KK7B, and Bob Larkin, W7PUA, *Experimental Methods in RF Design*, The American Radio Relay League, 2009, ISBN: 978-087259-923-9. ARRL Publication Order No. 9239, $49.95. ARRL publications are available from your local ARRL dealer or from the ARRL Bookstore. Telephone toll free in the US: 888-277-5289, or call 860-594-0355, fax 860-594-0303; **www.arrl.org/shop**; **pubsales@arrl.org**.

Letters to the Editor

A New Antenna Model (Jul/Aug 2012)

Dear Glenn,

Your "New Antenna Model" was an interesting article. Thanks for sharing your ideas, but I do have a question.

Is Figure 4D (all the way down) on page 10 (SWR and Return Loss), is the vertical axis correct? I think RL is defined as: RL = –20 log SWR + 1 / SWR – 1

Therefore,

SWR 2 = 9.5 dB RL;

SWR 1.2 = 20.8 dB RL,

SWR 3 = 6 dB RL,

SWR 6 = 2.9 dB RL and so on.

So it appears to me to be somehow shifted on the right vertical axis.

On the left axis, SWR should not start at SWR = 0 but SWR = 1.

Or am I missing something?

— 73, Andy Hansen, HB9CVQ (DK2VQ, AK4IG), Switzerland; **hb9cvq@hispeed.ch**

Dear Andy,

No, you're not missing something, it's wrong. Thanks for the catch!

It's also correct to state RL as $(1 + \rho)$ / $(1 - \rho)$ where ρ is the reflection coefficient. This gives the same numbers as your equation, so $\rho = 0.5$ means a voltage reflection coefficient of 0.5, which is an SWR of $(1 + 0.5) / (1 - 0.5) = 3$ and a return loss = $-20 \log(0.5) = 6$ dB.

During graphics layout for publication in *QEX* the left axis got re-labeled. As published, Figure 4D has an SWR less than unity shown at the bottom of the left axis and some of the data seems to go down there! If we can perfect this, it might be useful for negative noise figure receivers! In fact, the axis should have been labeled 1 to 10 rather than 0 to 10 as published.

Some labeling corrections on Figure 8 also didn't make it into the published version. Part A should indicate that image is for a ½ λ dipole, B is a 1 λ dipole, C is a 3/2 λ dipole and D is a 2 λ dipole. I've put a couple of these corrections up on my website at **www.sonic.net/~n6gn.** If anyone else finds other problems, please let me know and I'll try to correct them there.

— 73, Glenn Elmore, N6GN, 446 Halter Ct, Santa Rosa, CA 95401; **n6gn@sonic.net**

Hi Larry,

The article "A New Antenna Model" by Glenn Elmore, N6GN, in the Jul/ Aug 2012 issue of *QEX* was very interesting and thought provoking, however, there is one aspect with which I would like to take issue. It is his statement that "power is coupled into the radiating tip where the radiation resistance is located." I am not sure whether he really believes this or whether it is a merely a statement of the way his model portrays the operation. Any way, it is totally untrue for a real antenna that radiation takes place at the tip, but unfortunately his statement has been taken at face value in at least one antenna forum on the Internet. I would therefore like to set the record straight.

There is one basic fact we know about radiation and it was summed up by the famous theoretical physicist, Richard Feynman, when he said "If you wiggle an electron, other electrons around it feel a force." So the basis of radiation is that we need to accelerate electrons, or more generally charged bodies. Notice that simply moving the electron at a constant velocity is not sufficient for radiation, as we know since direct current does not radiate.

We normally produce radiation by accelerating charges in a *conductor*, but in principle we do not *need* a conductor. For instance if we have a charged body and swing it around on the end of an insulator (a piece of string) then this will radiate. In practice such an experiment would be difficult to do because we cannot rotate the charge very fast and so an extremely large charge would be necessary to produce any measureable radiation. But the principle stands, that radiation merely needs the acceleration of charges or electrons.

Turning now to antennas, *these are merely devices for accelerating charge.* Conductors are very handy here because they have a lot of free electrons, which move through the conductor with relative ease and so are easily accelerated. Glenn Elmore's article correctly points out that the ends of wire antennas have a very high voltage. Of course that is *exactly* what is needed to accelerate the electrons, and why a normal antenna configuration is so good at doing this. The high voltage does not radiate, however, it is the accelerated electrons that radiate, and these are in the high current part of the antenna (since a current is the movement of electrons).

It is worth a comment at this point on fields. No one knows *how* an accelerating electron produces a force on another, and so we have invented fields to help us explain what is happening. We cannot prove that these fields exist, however. We might try to measure them, and for this we take a small conducting dipole or loop and move it around the radiating antenna and measure a voltage. But the voltage is a result of the force on the electrons in our measurement loop, and proves only what we knew already that "if you wiggle an electron, other electrons around it feel a force." It does not prove that there are fields. Unfortunately, some workers have taken the field concept to outlandish lengths, as evidenced by the "cross field" antenna, and if we analyze this to find where in the antenna it is accelerating electrons, it is evident that it will not work, at least not in the way the authors claim.

So remember, when you put up a small wire antenna in the garden and accelerate its free electrons with a few watts of power, the electrons in another antenna 1000 miles away will feel a force, which the receiver at that end can detect. Who would have believed that this would be possible? I have been involved in electromagnetics all my adult life and I still think it is magic!

— 73, Alan Payne G3RBJ; **paynealpayne@ aol.com.**

Hi Alan,

Thanks so much for writing. I'm pleased that this model is prompting questions. You ask some good ones.

Richard Feynman does describe acceleration of an electron as producing a force on other charges as a fundamental principle and that this occurs whether the charge is in a conductor or not. He also posits as similarly fundamental, however, the theory of superposition. (Richard Feynman, *Lectures on Physics*, Vol. I, 12-9). It is truly an amazing thing that we can accelerate charges with our amateur radio transmitters and see the effects at great distance. It is also amazing that the effects from multiple electric charges, whether static, moving steadily or accelerating, can be added together without any consideration of interaction among them to obtain the result predicted by field theory.

As a result of superposition, mere acceleration of a charge does not mandate radiation *in the far field.* Far field effects are described by the sum of the effects from all charges and these may add to zero. It is for this reason that ideal transmission lines can be understood not to radiate. In operation, coaxial cable or balanced line, for example, each have accelerating charges which can produce force on other charges. Superposition allows that the effects of all charges must be considered, however, in determining the effect on a distant (test) charge as modelled by field theory. For ideal transmission lines, the effects of the moving charges in the two different conductors cancel, such that the total effect can and does sum to zero. Thus, they do not radiate even though individual charges are being accelerated within them.

The same is true for the model I have presented. It describes an antenna element as a surface wave transmission line having axial and longitudinal symmetry in regions

far from the center and ends. The net far-field force produced by all of the accelerating charges is zero. Only from regions where there is asymmetry is there any net far-field effect (radiation) produced. As intense as they may be in the high current regions along the SWTL, the axial and longitudinal fields are symmetric — each field "line" has an "opposite twin" that cancels its effect in the far field.

With regard to a model and our beliefs, I'd like to point back to the article, to the statement in the article, "All models are wrong, but some are useful." If we ever believe that we comprehend the final, precise and complete answer — if we even believe that such might be attainable — I think we limit ourselves. Models are useful and their utility can be seen in the ways they allow us to expand our understanding, extend our applications and enjoy the world we see around us. The model I have presented portrays antennas as wave devices in a manner that fits what we measure and currently understand. Whether we use field theory, theory of potentials, QED or something else, I hope that our exploration of antennas and our enjoyment of the magic of Amateur Radio will never be limited by "is" but will have the freedom of "acts like." It's truly wonderful that there is always more for us to discover!

— 73, Glenn Elmore, N6GN; **n6gn@sonic.net**

New Results on Shortening Beverage Antennas (Jul/Aug 2012

Hi Larry,

In the Jul/Aug issue, in the article by Christoph Kunze, DK6ED, on page 31, the two 100 μF electrolytic capacitors in the K9AY control box are shown with reversed polarity. The 100 μF capacitor in the middle of the page, for the relay coil, is also shown incorrectly.

— 73, Kenneth Hansen, KB2SSE, 10 Maple Rd, Ringwood, NJ, 0745; **kb2sse@arrl.net**

Hi Kenneth,

Thank you for pointing out that error on the schematic diagram of Figure 15. That error was ours and not the author's, and we apologize for reversing the polarity of those electrolytic capacitors. Too often we just assume that the negative side of the capacitor would go to the ground connection, but that isn't the case here, given the connections to the diodes.

— 73, Larry Wolfgang, WR1B, QEX Editor; **lwolfgang@arrl.org**
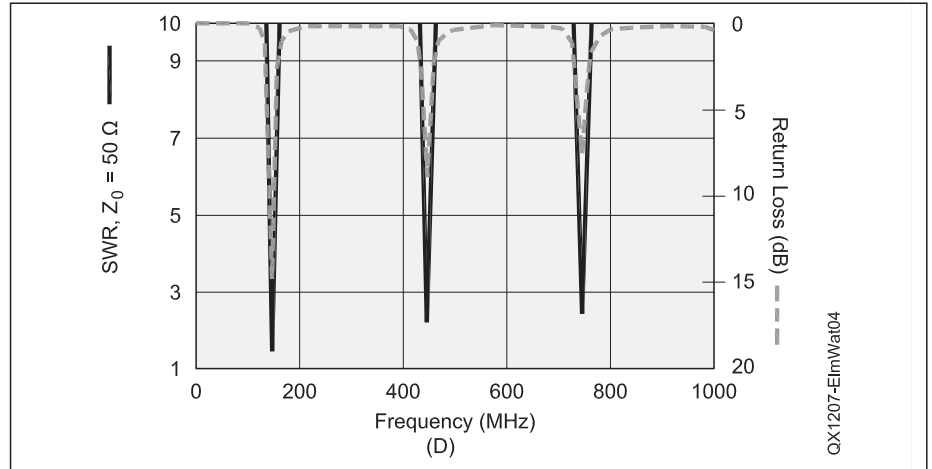
Figure 4D — This is the corrected version of Figure 4D from "A New Antenna Model," Jul/Aug 2012 QEX, p 10. Note that the SWR scale along the left edge of the graph has been corrected to show an SWR of 1 at the bottom of the graph.

QEX

Upcoming Conferences

Microwave Update 2012

October 18 – 21, 2012
Biltmore Hotel Santa Clara
2151 Laurelwood Road
Santa Clara, CA 95054

Microwave Update (MUD) is an annual event held since 1985. The 50 MHz and Up Group of Northern California is pleased to host the 2012 event. MUD is a conference dedicated to microwave equipment design, construction, and operation. It is focused on, but not limited to, Amateur Radio on the microwave bands. There are technical presentations all day Friday and Saturday, with an antenna measurement range and outdoor flea market Sunday morning.

Thursday Tours

The hosted tour at this year's Microwave Update is the Stanford Dish. There will be two sessions (morning and afternoon) for the tour, and there is limited space for each one. Only 25 people per session can be accommodated.

If you would like to attend the Dish tour, when you register for MUD please select check boxes for Dish Morning or Afternoon. If you could attend either, please check both Morning and Afternoon boxes.

Banquet Speaker

Dr Thomas Lee will be the speaker at the banquet Saturday evening. He will talk about the history of radio and the influences of Amateur Radio on its development, plus the need for our future contributions. Dr Lee is currently the director of the Microsystems Technology Office at DARPA.

There is more information about the conference on the MUD website at **http://microwaveupdate.org**.

2012 AMSAT Space Symposium and Annual Meeting

October 26 – 28, 2012
Holiday Inn Orlando-International Airport
5750 T. G. Lee Blvd,
Orlando, FL 32822
1-407-851-6400

AMSAT announces that the 2012 AMSAT Space Symposium will be held on Friday, October 26th through Sunday, October 28th, 2012.

Features include:

Space Symposium with Amateur Satellite Presentations

Operating Techniques, News, & Plans from the Amateur Satellite World

Board of Directors Meeting open to AMSAT members

Meet Board Members and Officers

Annual General Membership Meeting

Annual Banquet—Keynote Speaker and Door Prizes

Call for Papers

Dan Schultz, N8FGV, has issued a second call for papers for the 2012 AMSAT Annual Meeting and Space Symposium. Proposals for papers for publication, symposium presentations, poster presentations and equipment and operating demonstrations are invited on any topic of interest to the amateur satellite community. We request a tentative title of your presentation as soon as possible, with final copy to be submitted by October 1 for inclusion in the printed proceedings.

Abstracts and papers should be sent to Dan Schultz, N8FGV, at **n8fgv@amsat.org**.

There are still prime slots available on the speaker's schedule. The AMSAT symposium depends on *you* to make it happen; without speakers there is no symposium. Please consider doing a presentation, even if you have never done so in the past. We are especially interested in papers and presentations involving our educational partnerships with K-12 and university level classrooms. Education is quickly becoming critical for our future rides to space.

A Monday trip to Kennedy Space Center is planned. Please call or e-mail Office Manager, Martha Saragovitz (**martha@amsat.org**) and let her know if you are interested.

QEX-