**Victor A. Kean, Jr., K1LT**

6205 Plummer Rd, Carroll, OH 43112; vkean@k1lt.com

# Beam Steering on 160 Meters

*Change the direction of your antenna array electronically with a computer and software defined radios.*

My lot is a tall, narrow rectangle, 500 feet east to west by 1300 feet north to south. Long Beverage antennas to the northeast and southeast fit nicely, but Beverages oriented due east and west are just too short. A broadside array of phased verticals complements my geography, since the broadside array receives at right angles to the axis along the verticals. See Figure 1. While I could have settled for just two directions from my array, for somewhat more effort I can electronically rotate the main beam of the array. Furthermore, developing a steerable array allows me to combine two of my passions: ham radio and computing.

## Array Design

The simplest phased array is a row of identical antennas with uniform spacing. If the spacing is about ½ λ, the array is called a "broadside array" because the array has maximum directivity in a direction perpendicular to the line of antennas. If the spacing is about ¼ λ or less, the array is called an "end-fire array" because the array has maximum directivity along the line of antennas. Note that a broadside array is bidirectional, while the end-fire array is usually unidirectional. The spacing between antennas of a broadside array can be greater than ½ λ, but there is no advantage to exceeding about 80% of a wavelength. If the spacing between the antennas is reduced much below ½ λ, the directivity declines considerably.

A broadside array of short verticals is bidirectional, as shown in Figure 2. The array elements can be any antenna, however. If each element of the array is unidirectional, then the overall array will be unidirectional, a principal called *pattern multiplication*.[1] One of the simplest unidirectional small antennas is a pair of verticals in an end-fire configuration. A K9AY loop would be another simple unidirectional antenna, and a Beverage antenna would be a good candidate as well.

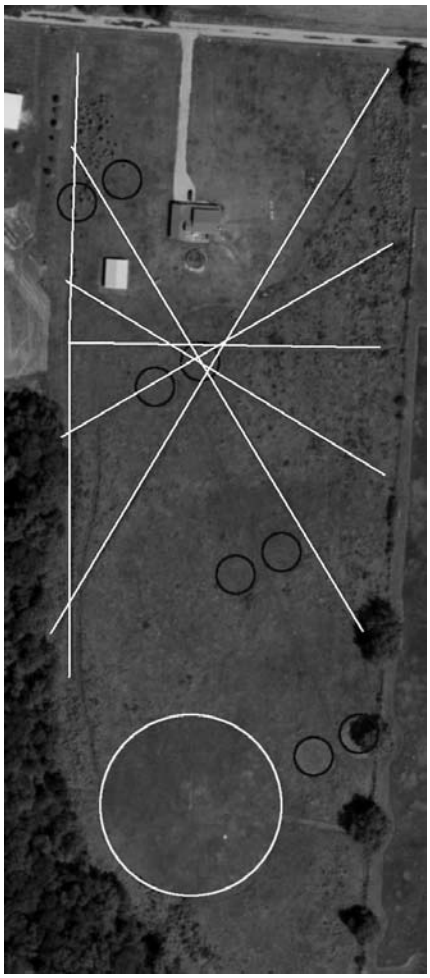


**Figure 1 — K1LT Antenna Layout — The lines along the West (left) edge of the field and the lines crossing in the middle are two-wire Beverages. The large circle at the bottom of the photo marks the location of the transmitting antenna and radial field, and the smaller circles mark the locations of phased array antenna elements and radials. North is straight up. The broadside array could not be aligned broadside to Europe without shrinking the array, which would seriously degrade performance. Aerial photograph used with permission of Digital Globe.**

A 2 element end-fire array has a cardioid pattern as shown in Figure 3. To obtain a cardioid pattern, one must arrange for the two elements to be out-of-phase when a signal arrives from the direction of the null. If both antenna elements have resistive feed impedances and are properly matched to their feed lines and the feed lines are equal length, then one may insert a phasing line with a time delay equivalent to the physical separation and subtract one signal from the other. Also, a simple relay can move the phasing from one feed line to the other, allowing the null direction to be rotated 180°. See *ON4UN's Low Band DXing* by John Devoldere for more information about 2 element end-fire arrays.[2]

Each end-fire array comprises a single element of the overall phased array. In other words, the antenna I built is a 4 element broadside array of 2 element end-fire arrays. I chose to implement a 4 element array since my processing electronics supports 4 inputs (see below). Also, 4 elements fits reasonably well on my land. Finally, Tom Rauch, W8JI, suggests that ionospherically propagated signals are often not very coherent over more than about 1½ λ, which is the approximate length of a 4 element broadside array.[3]

## Phasing

Making a broadside array from a set of identical antennas in a line is quite simple. One must feed each element with an equal amplitude signal, in phase with one another. Since we are receiving and not transmitting, we take power from each antenna and sum the signals in phase.

The following discussion about phasing and steering assumes array elements with little or no mutual coupling. I present the explanation that validates this assumption in the section about antenna element design.

The traditional phasing method requires

[1]Notes appear on page 39.

proper feed-point impedances matched to equal length feed lines that all connect at a common point. For example, if each antenna exhibits a feed-point impedance of 50 Ω purely resistive, then we can use any length of feed line to a common point and connect the feed lines in parallel. The paralleling of N antennas reduces the combined impedance by a factor of N. Therefore, additional impedance matching would be required. Although it's not required, an N-way combiner would guarantee graceful performance degradation in the event of failure of one or more elements of the array.

Figure 4 shows that the equal element amplitude scheme produces an array with maximum directivity, although the pattern has many side lobes. If the element amplitude scheme follows a binomial pattern, such as 1:2:1 for 3 elements or 1:3:3:1 for 4 elements, then the array pattern has no side lobes but somewhat less directivity, as in Figure 5. *Low Band DXing* shows a 4 element broadside array using a 1:2:2:1 element amplitude scheme, which produces a pattern with directivity and side lobes that are a compromise of the 1:1:1:1 and 1:3:3:1 schemes, shown in Figure 6.

## Steering

A broadside array can be steered by applying a progressively increasing phase shift to each element. For example, by applying phase shifts of approximately 0, 10, 20, and 30° to the elements of a 4 element array, the direction of maximum directivity is shifted about 3° towards the element with the largest phase shift. Figure 7 shows the pattern from Figure 4 steered by about 3°.

Figure 8 shows the relationship between the steering angle and the phase offset. The formula for that relationship is:

$$\varphi_n = \frac{(n-1)\bullet 2\pi \bullet \text{spacing}}{\text{wavelength}} \bullet \sin(\text{steering angle}) \quad \text{[Eq 1]}$$

where $\varphi_n$ is the phase delay required for the n-th element, "n" is the element number, and "spacing" is the distance between elements in the same units as the wavelength. The steering angle is the difference between the array major axis (long dimension) and the desired signal bearing. In other words, broadside is 90°. Note that when n is 1, the first element, the phase delay is zero. The phase delay applied to each element merely accounts for the extra distance with respect to the first element that the incoming wave front must traverse in order to keep the signal induced in each of the elements in phase. Thus the first element's phase shift is zero because we use it as the reference. The simplest way to apply a phase shift is to use a delay line, which is easily made from coaxial cable. We can steer in multiple directions by using switches or relays to select from a set of delay lines. The amount of switching required for more than two or three directions is prohibitive, however.

The phase shift does not have to be applied to the feed line. The phase shift need only be applied before the signals from each element are summed. Therefore, the phase

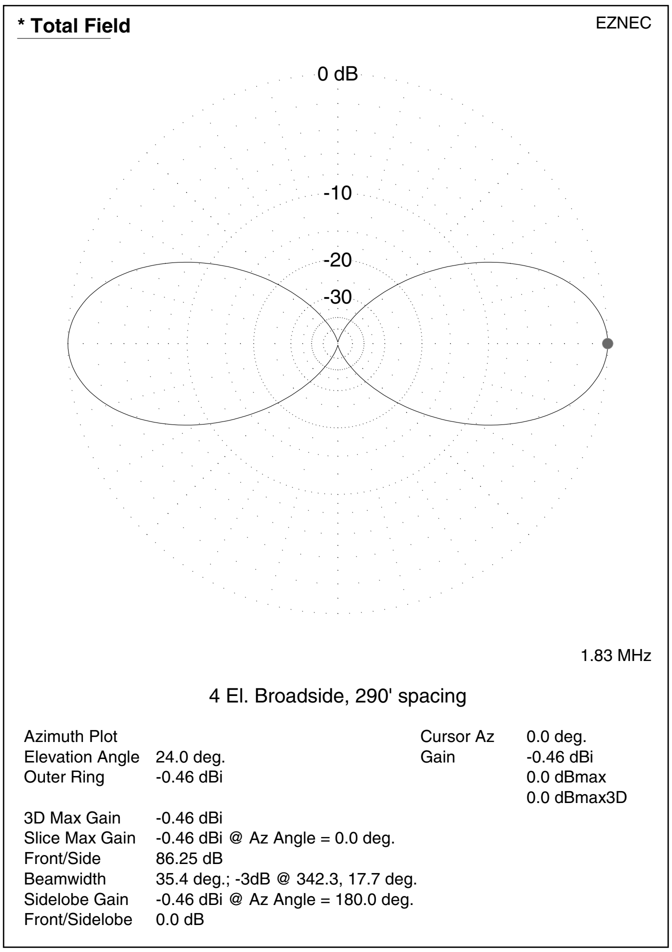



**Figure 2 — Azimuthal pattern of a 4 element broadside array of short verticals with binomial feed point amplitudes.**

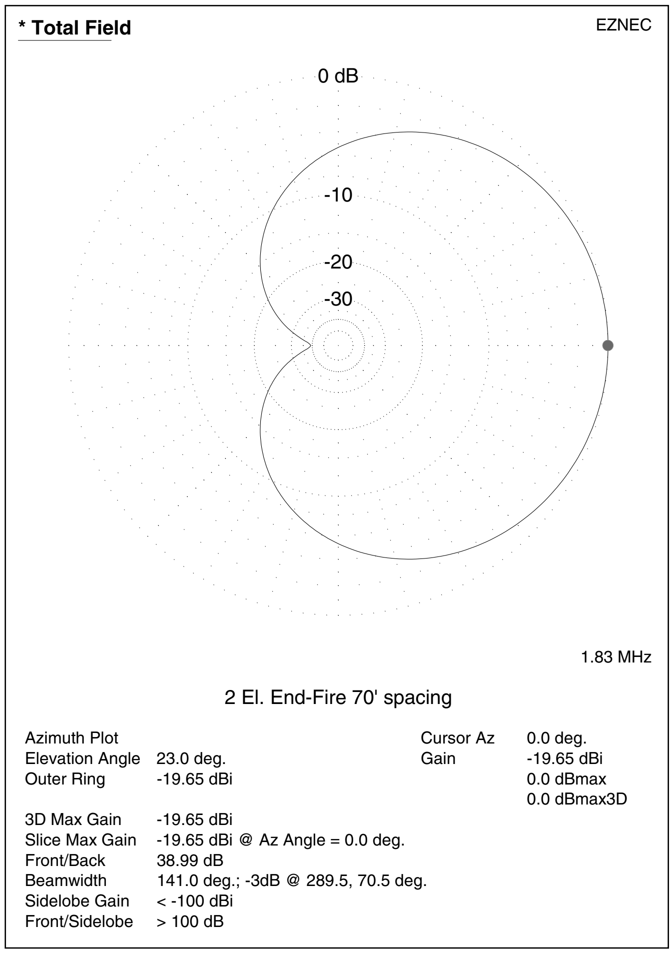



**Figure 3 — Azimuthal pattern of a 2 element end fire array of short verticals with equal feed point amplitudes and 136° phase difference demonstrating the cardioid pattern.**

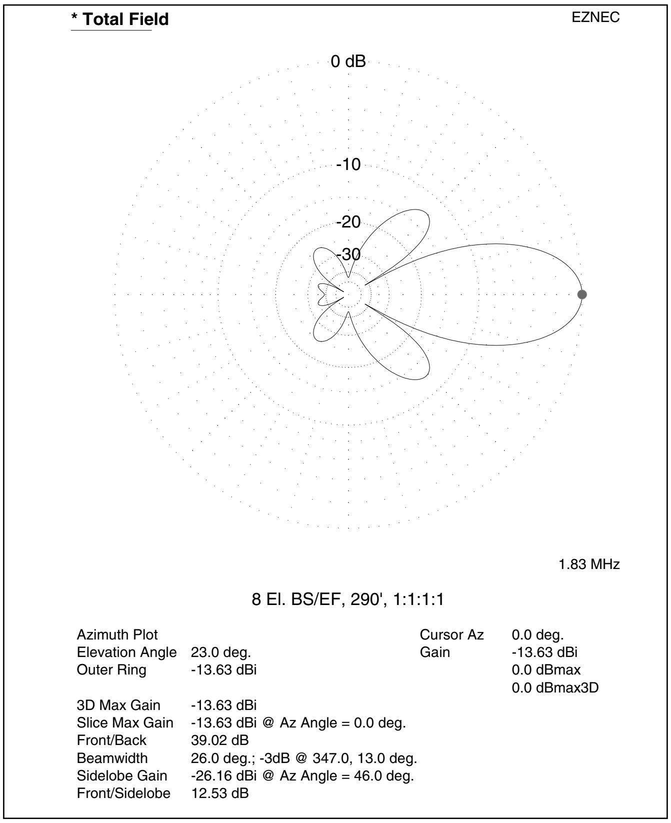



**Figure 4 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire (BS/EF) arrays with equal feed point amplitudes.**

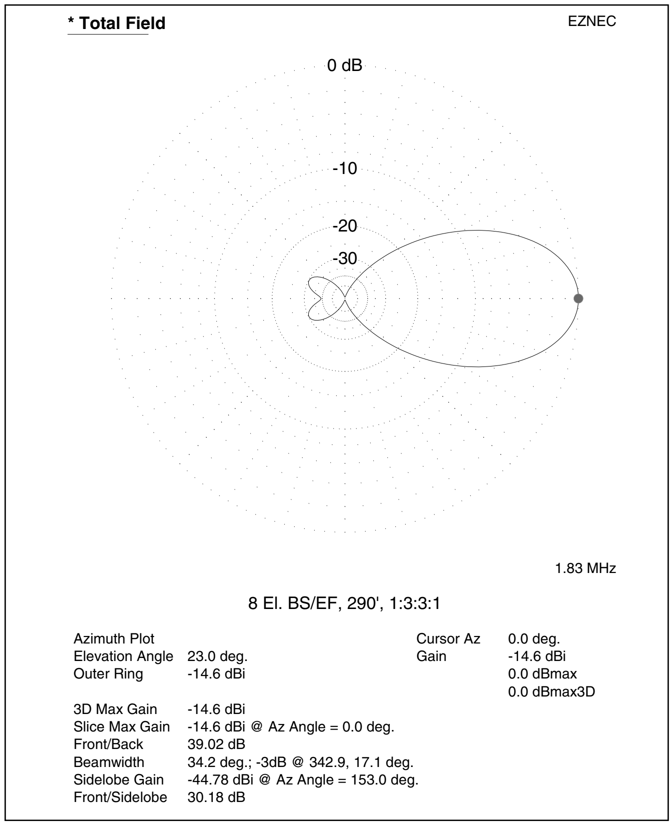



**Figure 5 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with the binomial (1-3-3-1) feed point amplitude arrangement.**

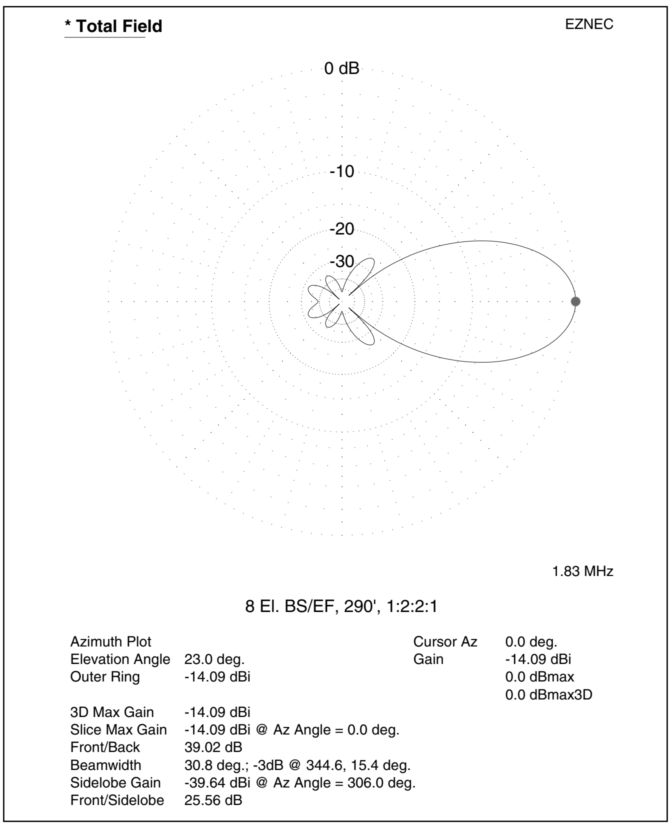



**Figure 6 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with the 1-2-2-1 feed point amplitude arrangement from *ON4UN's Low Band DXing*.**

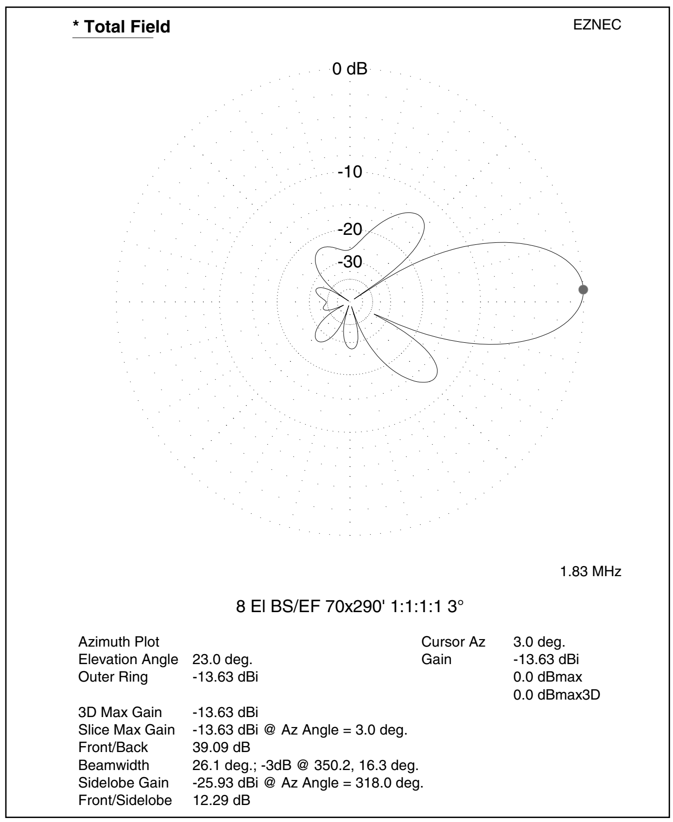



**Figure 7 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with equal feed point amplitudes and progressively increasing phase offsets of about 10°.**

shift can be applied with software after the signal is converted from the analog to the digital domain.

## Steered Array Performance

A broadside array has a certain amount of directivity that depends upon the number of elements and the element spacing. Steering the array tends to decrease directivity. The more you steer the beam away from broadside, the less directivity you get. See Figure 9 and Table 1 for the uniform feed array and see Table 2 for the binomial feed array. Also, the more you steer the array, the more bidirectional the array tends to become.

At about 60° of steering, the largest secondary lobe actually exceeds the size of the main lobe. Finally, the direction of maximum gain fails to keep up with the steering angle. This phenomenon occurs because the end-fire array has considerably less gain at right angles to its main lobe. Note that the worst directivity occurs near 70° of steering.

A better element design might switch



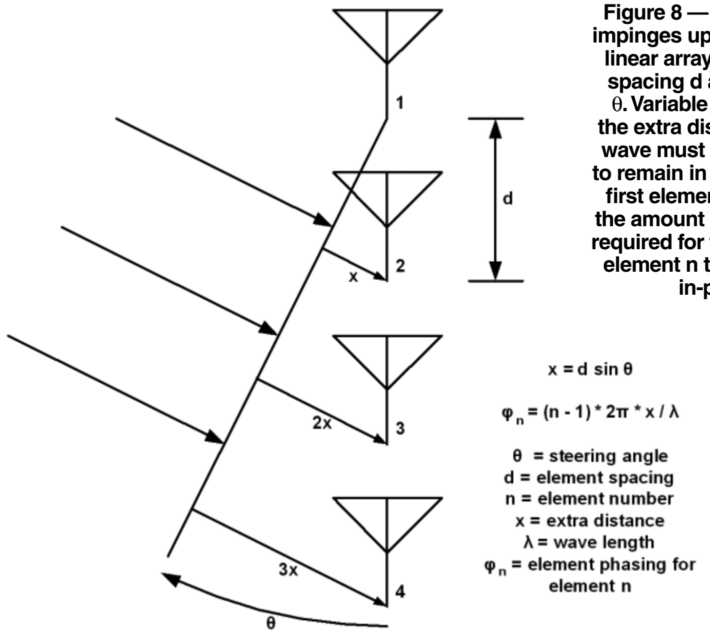



**Figure 8 — A plane wave impinges upon a 4 element linear array with element spacing d at an angle of θ. Variable x represents the extra distance that the wave must travel in order to remain in phase with the first element. Angle $\varphi_n$ is the amount of phase delay required for the signal from element n to be summed in-phase.**

**Table 1**

**Performance data for a 4 element broadside array of 2 element end-fire arrays with equal feed point amplitudes steered from 0° to 90° in 10° increments.**

**Maximum Directivity**

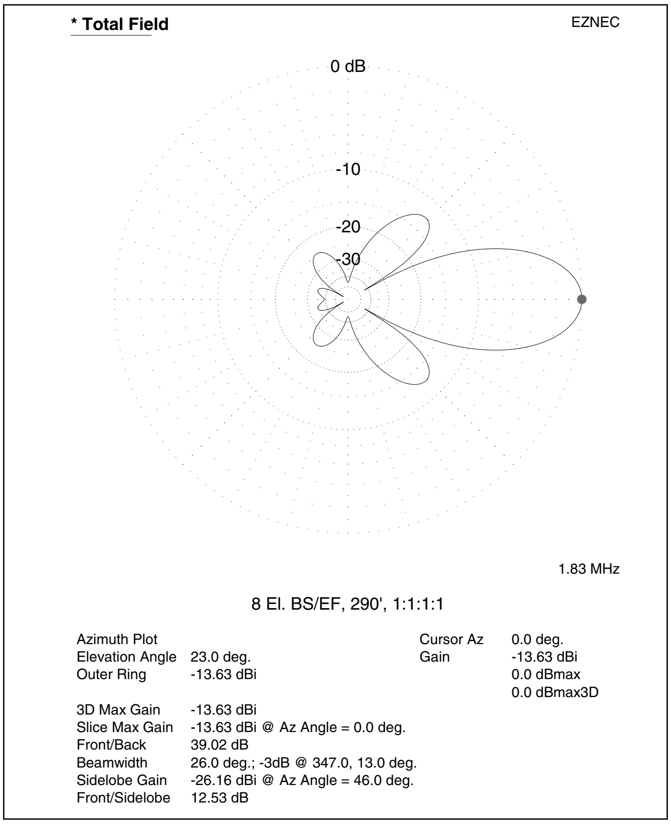
| *Steered Angle* | *Average Gain* | *Peak Gain* | *RDF* | *RDF change from best* | *Direction of Maximum RDF* | *Beam Width* |
|---|---|---|---|---|---|---|
| *Degrees* | *dB* | *dB* | *dB* | *dB* | *Degrees* | *Degrees* |
| 0 | –29.08 | –13.63 | 15.45 | 0.00 | 0 | 26.0 |
| 10 | –28.99 | –13.69 | 15.30 | –0.15 | 11 | 26.5 |
| 20 | –28.86 | –13.87 | 14.99 | –0.46 | 21 | 27.9 |
| 30 | –28.74 | –14.18 | 14.56 | –0.89 | 31 | 30.2 |
| 40 | –28.49 | –14.64 | 13.85 | –1.60 | 41 | 34.3 |
| 50 | –28.20 | –15.23 | 12.97 | –2.48 | 50 | 40.0 |
| 60 | –28.03 | –15.94 | 12.09 | –3.36 | 58 | 46.1 |
| 70 | –28.01 | –16.39 | 11.62 | –3.83 | 64 | Wider than secondary lobe |
| 80 | –28.03 | –15.96 | 12.07 | –3.38 | 68 | |
| 90 | –28.05 | –15.83 | 12.22 | –3.23 | 69 | |

**Table 2**

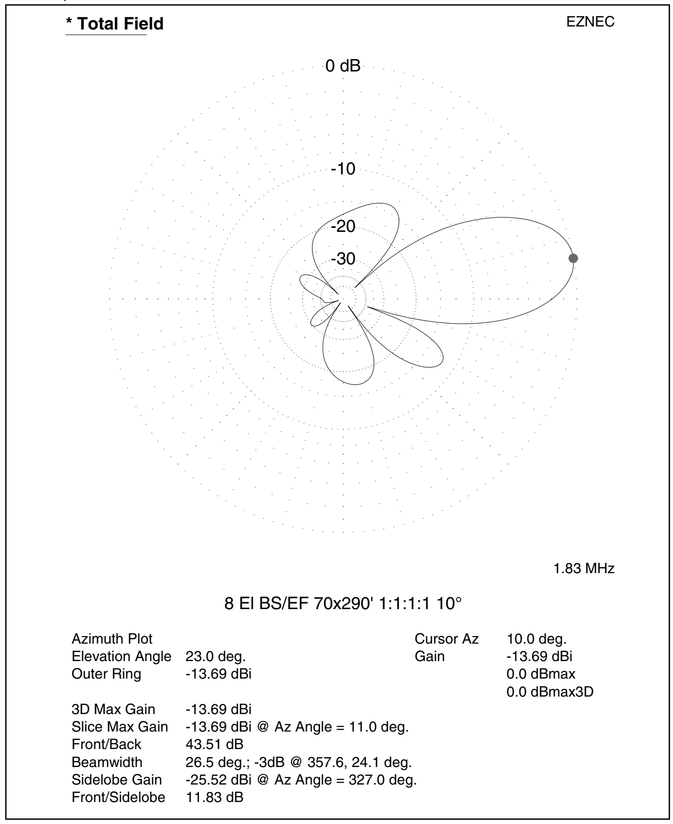
**Performance data for a 4-element broadside array of 2-element end-fire arrays with binomial (1-3-3-1) feed point amplitudes steered from 0° to 90° in 10° increments.**

**Maximum Directivity**

| *Steered Angle* | *Average Gain* | *Peak Gain* | *RDF* | *RDF change from best* | *Direction of Maximum RDF* | *Beam Width* |
|---|---|---|---|---|---|---|
| *Degrees* | *dB* | *dB* | *dB* | *dB* | *Degrees* | *Degrees* |
| 0 | –28.98 | –14.52 | 14.46 | 0.00 | 0 | 34.2 |
| 10 | –28.92 | –14.58 | 14.34 | –0.12 | 10 | 34.8 |
| 20 | –28.78 | –14.76 | 14.02 | –0.44 | 20 | 36.2 |
| 30 | –28.55 | –15.05 | 13.50 | –0.96 | 30 | 39.2 |
| 40 | –28.29 | –15.46 | 12.83 | –1.63 | 40 | 43.6 |
| 50 | –28.08 | –16.00 | 12.08 | –2.38 | 48 | 48.7 |
| 60 | –27.98 | –16.63 | 11.35 | –3.11 | 55 | 52.6 |
| 70 | –27.96 | –17.04 | 10.92 | –3.54 | 60 | Wider than secondary lobe |
| 80 | –27.97 | –17.70 | 10.27 | –4.19 | 63 | |
| 90 | –27.98 | –17.87 | 10.11 | –4.35 | 64 | |

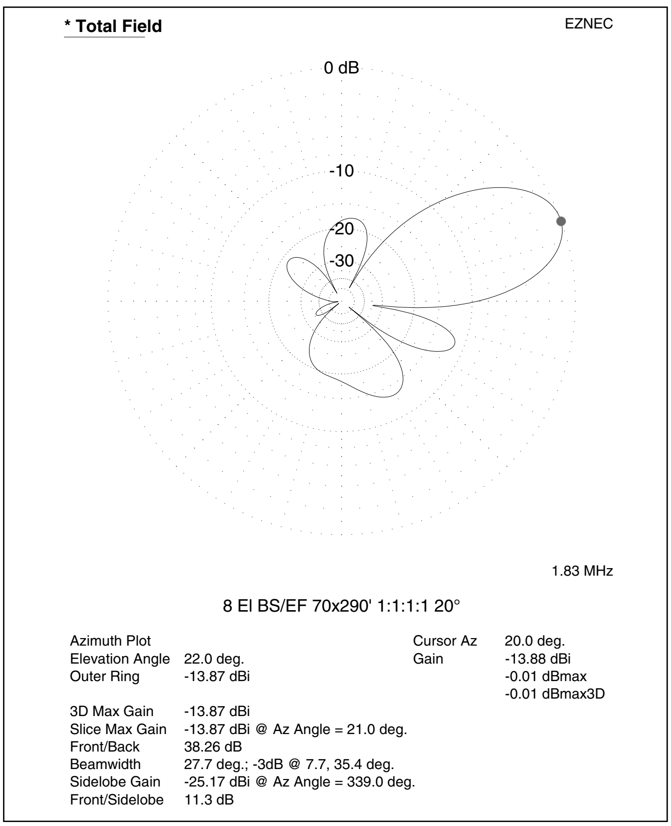



(A)

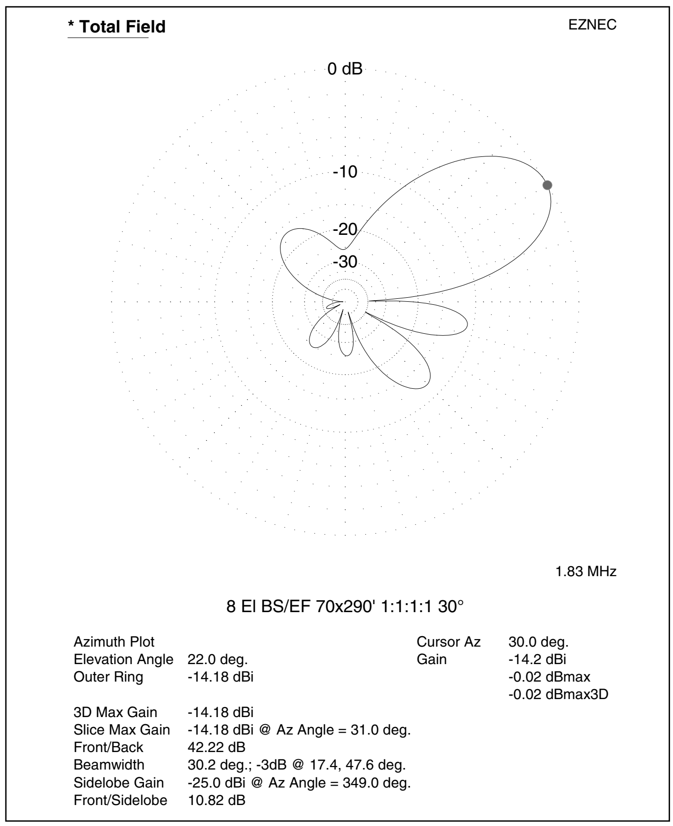



(B)

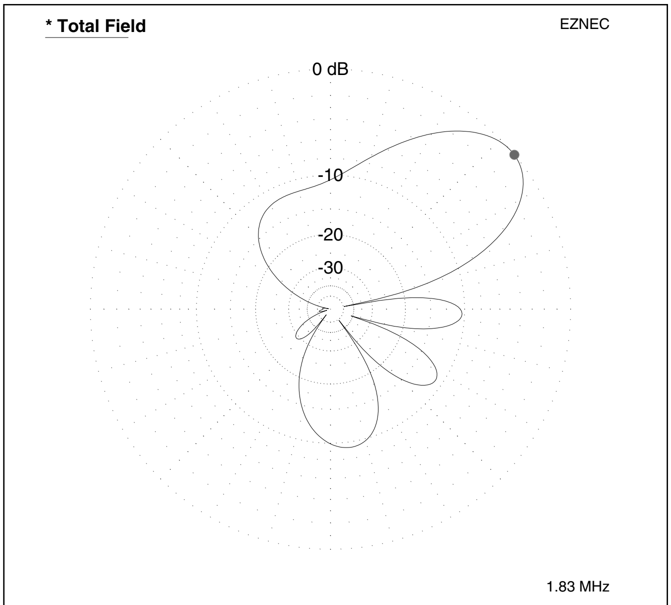
**Figure 9 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with equal feed point amplitudes steered from 0° to 90° in 10° increments. Note how the peak of the main lobe follows the outline of the cardioid pattern in Figure 3. Part A is at 0°, Part B is at 10°, Part C is at 20° and so on. Part J is at 90°.**




(C)



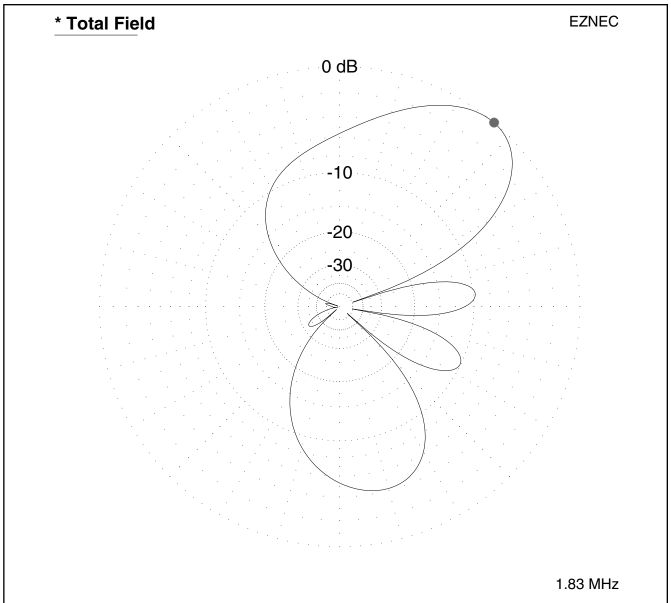

(D)




8 El BS/EF 70x290' 1:1:1:1 40°

| | | | |
|---|---|---|---|
| Azimuth Plot | | Cursor Az | 40.0 deg. |
| Elevation Angle | 21.0 deg. | Gain | -14.65 dBi |
| Outer Ring | -14.64 dBi | | -0.01 dBmax |
| | | | -0.01 dBmax3D |
| 3D Max Gain | -14.64 dBi | | |
| Slice Max Gain | -14.64 dBi @ Az Angle = 41.0 deg. | | |
| Front/Back | 28.84 dB | | |
| Beamwidth | 34.0 deg.; -3dB @ 26.3, 60.3 deg. | | |
| Sidelobe Gain | -23.94 dBi @ Az Angle = 278.0 deg. | | |
| Front/Sidelobe | 9.31 dB | | |

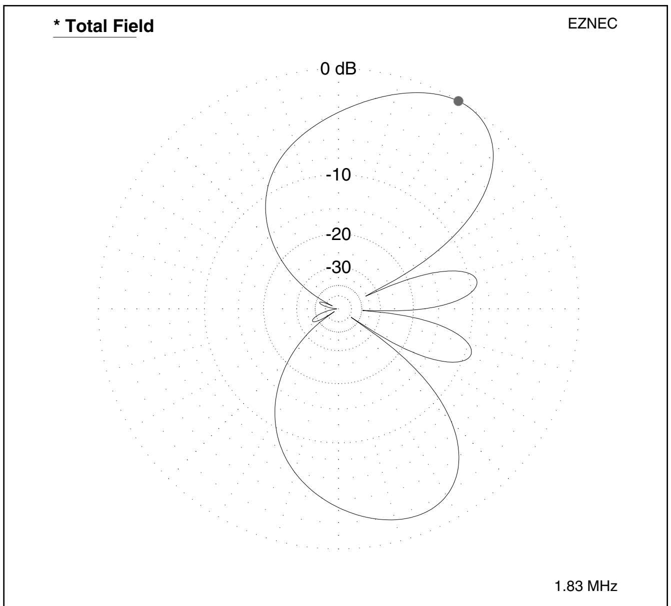
(E)




8 El BS/EF 70x290' 1:1:1:1 50°

| | | | |
|---|---|---|---|
| Azimuth Plot | | Cursor Az | 50.0 deg. |
| Elevation Angle | 21.0 deg. | Gain | -15.23 dBi |
| Outer Ring | -15.23 dBi | | 0.0 dBmax |
| | | | 0.0 dBmax3D |
| 3D Max Gain | -15.23 dBi | | |
| Slice Max Gain | -15.23 dBi @ Az Angle = 50.0 deg. | | |
| Front/Back | 24.6 dB | | |
| Beamwidth | 40.1 deg.; -3dB @ 34.4, 74.5 deg. | | |
| Sidelobe Gain | -19.41 dBi @ Az Angle = 285.0 deg. | | |
| Front/Sidelobe | 4.18 dB | | |

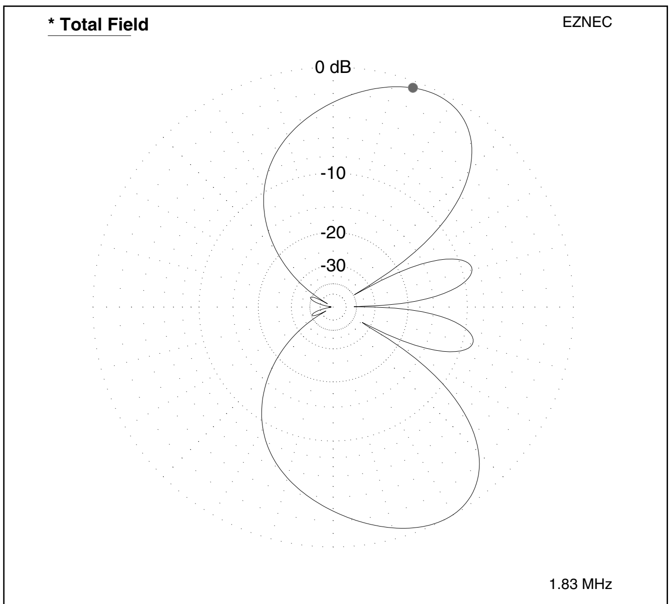
(F)




8 El BS/EF 70x290' 1:1:1:1 60°

| | | | |
|---|---|---|---|
| Azimuth Plot | | Cursor Az | 60.0 deg. |
| Elevation Angle | 20.0 deg. | Gain | -15.96 dBi |
| Outer Ring | -15.94 dBi | | -0.02 dBmax |
| | | | -0.02 dBmax3D |
| 3D Max Gain | -15.94 dBi | | |
| Slice Max Gain | -15.94 dBi @ Az Angle = 58.0 deg. | | |
| Front/Back | 11.9 dB | | |
| Beamwidth | 45.8 deg.; -3dB @ 41.1, 86.9 deg. | | |
| Sidelobe Gain | -17.33 dBi @ Az Angle = 292.0 deg. | | |
| Front/Sidelobe | 1.39 dB | | |

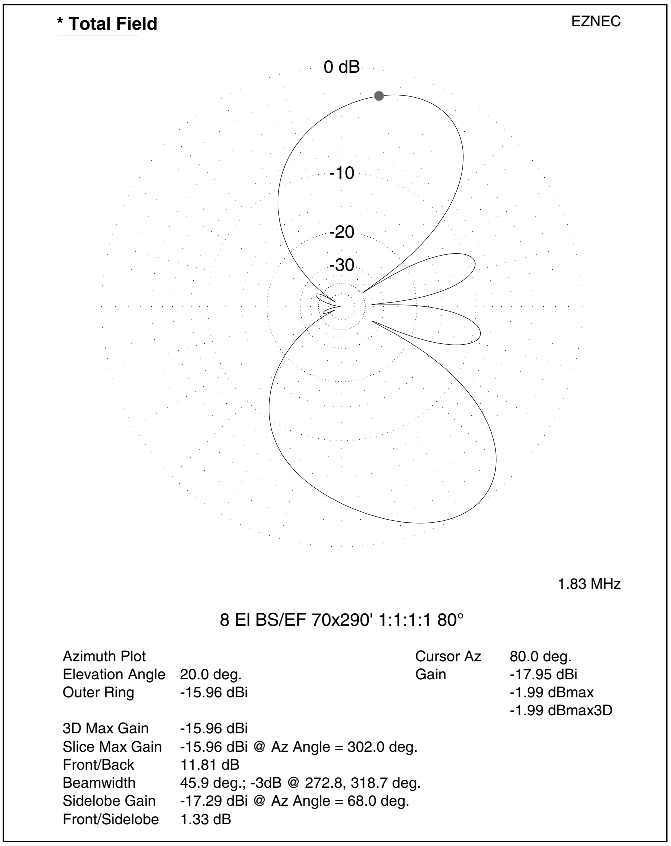
(G)



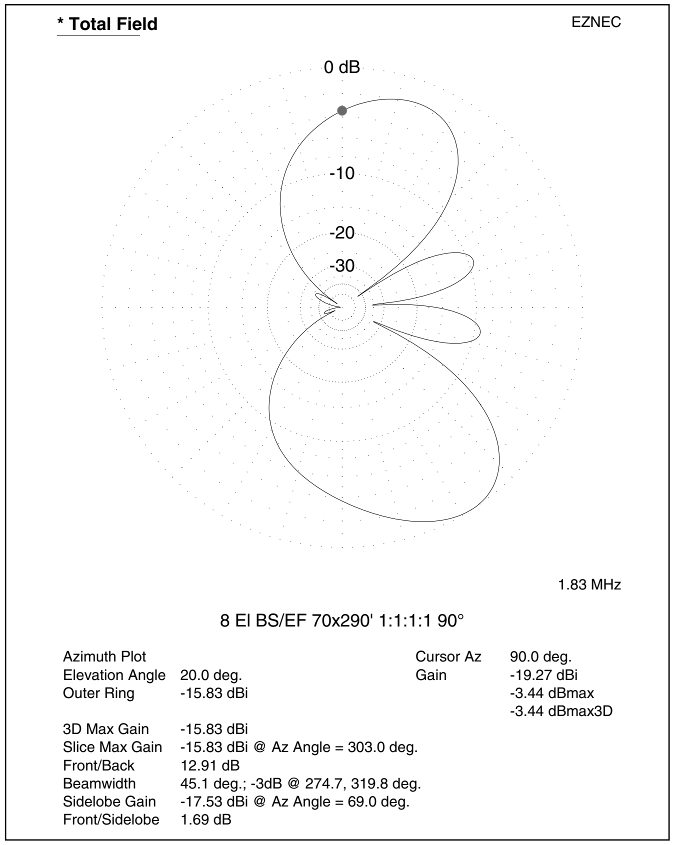

8 El BS/EF 70x290' 1:1:1:1 70°

| | | | |
|---|---|---|---|
| Azimuth Plot | | Cursor Az | 70.0 deg. |
| Elevation Angle | 19.0 deg. | Gain | -16.88 dBi |
| Outer Ring | -16.39 dBi | | -0.49 dBmax |
| | | | -0.49 dBmax3D |
| 3D Max Gain | -16.39 dBi | | |
| Slice Max Gain | -16.39 dBi @ Az Angle = 298.0 deg. | | |
| Front/Back | 8.81 dB | | |
| Beamwidth | 47.3 deg.; -3dB @ 268.5, 315.8 deg. | | |
| Sidelobe Gain | -16.68 dBi @ Az Angle = 64.0 deg. | | |
| Front/Sidelobe | 0.29 dB | | |

(H)




(I)




(J)

the 2 element end-fire array into a broadside array. This would allow the overall array to retain some directivity at high steering angles, which it would otherwise lose. The additional directivity would come at the expense of front-to-back ratio, although the front-to-back ratio is pretty poor in either case. An even better element design might be a pair of K9AY loops at right angles, which can be switched to any of four directions.

## Element Design

Each element of the phased array is itself a smaller phased array — in this case an end-fire array. Some of the end-fire array design considerations, such as mutual coupling, carry over to the larger array.

Mutual coupling complicates the establishment of a desired current in each element of a phased array. Mutual coupling is greatly minimized if the array elements are small relative to wavelength and have large losses. Also, large losses coincide with low *Q*, which in turn coincides with broad bandwidth, which finally results in stable phase relative to the other elements. We want as little reactance as possible at the feed point of each array element, and we want that reactance to be stable with respect to the weather and other disturbances.

One of the simplest practical antennas for 160 meters is a short vertical. Tom Rauch, W8JI, suggests a desirable short vertical design that has the following attributes: short for minimal mutual coupling, large top hat to maximize radiation resistance, and a length chosen to simplify impedance matching with common components.[4] The short vertical minimizes mutual coupling because the radiation resistance is very low. The top hat consists of four wires the same length as the vertical segment, at an angle of 45° from vertical. The top hat wires can also function as guy lines if necessary. Finally, the length is chosen so that a common inductor can be used to resonate the antenna. When all of the wires are 23 feet long, then a 33 μH inductor in series with about 2000 pF of capacitance brings about resonance at 1.86 MHz. This antenna has a radiation resistance of about 1 Ω, so that a series 68 Ω resistor in addition to the resistance of the inductor provides a good match to inexpensive 75 Ω feed line. Figure 10 shows one of many ways to build a short vertical.

As stated previously, a 2 element end-fire array achieves a deep null by combining the signals from the 2 elements out of phase. The easiest way to do that is to delay the signal from one of the elements by a length of coax equivalent to the distance between the elements.

My end-fire-array-element verticals are an arbitrary 70 feet apart. I want the null to be elevated slightly for sky wave signals, and I chose 20° of elevation. Elevating the null direction reduces the apparent distance between the elements by the cosine of the elevation angle, which gives 65.78 feet at 20°. I used Belden 1829A, which is an RG-6 equivalent cable, for my phasing delay line. 1829A has a velocity factor of 83%. Therefore, the length of the delay line is only 83% of the working distance. Thus, I need 53.7 feet for my phasing line. My actual phasing lines are 54 feet, based on the table from ON4UN's book and the fact that Belden paints distance markers on the coax every 2 feet.[5]

Since the velocity factor of coax can vary, even when cut from the same spool, I checked the phase delay of each phasing line with an oscilloscope that has a time delay measurement feature. Each of the phasing lines measured within 5% of the desired delay.

By using short verticals with a minimum of mutual coupling, and swamping the feed point impedance with a resistor, the verticals

**Figure 10 — One way to build a short vertical. This antenna consists of three 12 foot 2×4s screwed together with deck screws so that the bottom two 2×4s overlap the top 2×4 by 2 feet. The bottom 2×4s are mounted 1 foot high on a bolt through an 8 foot 2×4 buried in the ground so that the overall height is 23 feet. An AWG no. 10 copper wire runs the length of the 2×4s and joins four 23 foot top hat wires that also guy the structure. The 2×4 buried in the ground provides extra rigidity so that the vertical is less likely to break in the middle, as the first several did. Ice on the top-hat wires enhances visibility.**

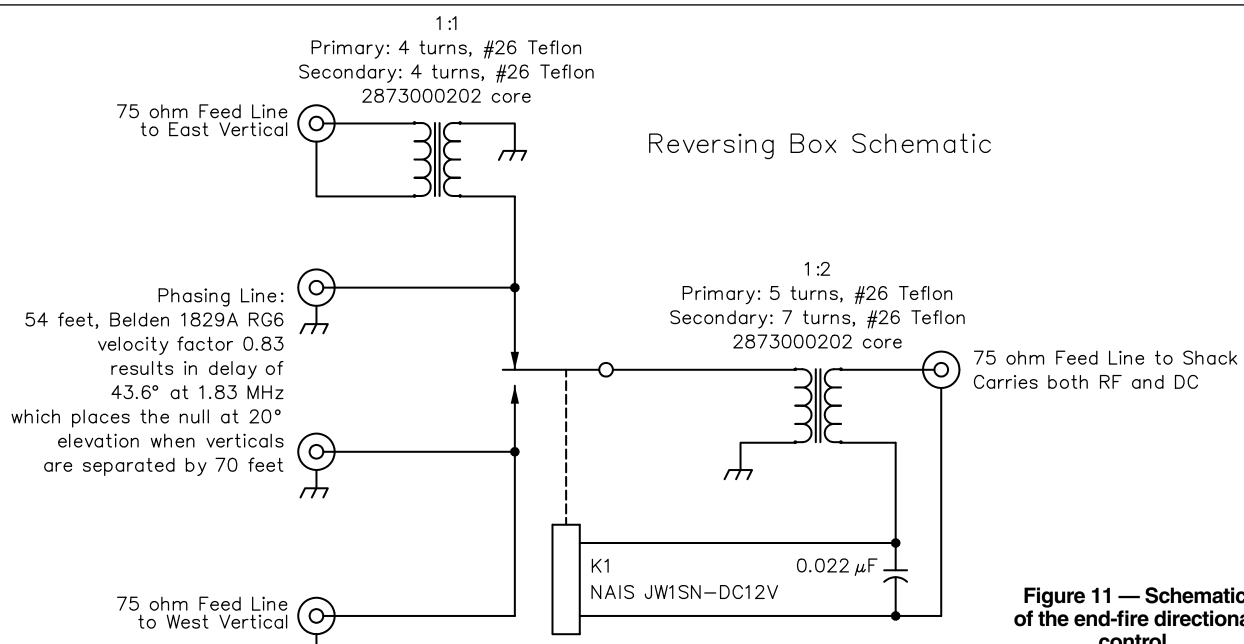



**Figure 11 — Schematic of the end-fire directional control.**

can simply be fed in parallel with confidence that the power from each vertical will add in phase. A 1:1 transformer inverts the phase of one element feed line, so that subtraction occurs rather than addition. In order to reverse the direction of the cardioid pattern, a relay switches the phasing cable into either the east feed line or the west feed line. Finally, a 2:1 matching transformer steps up the impedance to match the feed line back to the shack. The box that contains the direction relay and matching and inversion transformers also houses the terminals for the phasing line. See the schematic in Figure 11 and a photograph in Figure 12.

## Feed Lines

Although some elements are closer to the shack than others, each element is fed with 1000 feet of hamfest-bought RG6 equivalent cable. The attenuation and phase shift of each feed line should be as nearly the same as possible. Although I used surplus coax, I recommend that you obtain high quality coax from a reputable manufacturer. I ultimately added additional chunks of coax to match the time delays (phase shifts) of each feed line so that proper performance of the antenna array can be observed without electronic compensation. A linear phased array is more sensitive to feed line phase variations than to feed line amplitude variations. With high quality coax, compensation should not be necessary.

## SDR Components

I used Software Defined Radio (SDR) technology to implement beam steering. The SDR approach converts the analog signal to a digital signal as soon as possible and uses software to obtain radio receiver functionality. The beauty of this approach is that very simple hardware permits an otherwise complex operation to be performed with very simple software. Richard G. Lyons does an excellent job describing how digital sampling supports software signal processing.[6, 7] Refer to Figure 13 for the system block diagram.

My system contains 4 Softrock v6.1 receivers, designed by Tony Parks, KB9YIG.[8] Each receiver consists of a local oscillator, a divider chain, and a quadrature-sampling detector. Without going into great detail about how a quadrature-sampling detector works (see the description in the *QEX* article "A Software Defined Radio for the Masses, Part 1," by Gerald Youngblood, K5SDR),[9] the Softrock receiver converts an RF signal into 2 baseband (audio) signals, by convention called "I" for "in-phase" and "Q" for "quadrature", which are fed into a computer for further processing via software.

There is one receiver for each antenna element. One receiver is completely assembled,

**Figure 12 — End-fire directional control built into the cover of a "marine grade" plastic electrical box. Be sure to drill holes in the box to allow moisture out.**

while the other 3 receivers omit the local oscillator and divider chain. Instead, the first receiver supplies the in-phase and quadrature sampling clocks so that all of the receivers are synchronous or "phase locked." Observe Figure 14 for a schematic of the modified 160 meter Softrock v6.1 receiver. The schematic also shows slightly modified band pass filter component values. These component values provide a slightly better match to 75 Ω feed lines. The schematic also shows the clock signals that are daisy-chained from the first receiver to the other receivers, and also shows which components need not be populated on the slave receivers.

The four receivers are mounted in a metal box for shielding and handling convenience. I also used a direct digital synthesis VFO in place of the local oscillator on the first receiver.[10] Replacing the crystal oscillator with a VFO makes tuning the AM broadcast band possible. The AM broadcast band is a rich source of test signals. See Figure 15 for a photograph.

The I and Q outputs from each receiver are connected to an M-Audio Delta 1010LT pro-audio sound card. The Delta 1010 is the big brother of the popular Delta 44 and has 8 analog inputs and 8 analog outputs, as well as a bunch of digital audio inputs and outputs, which are of no use here. The Delta 1010 typically costs only about 35% more than the Delta 44.

A generic *Linux* computer hosts the Delta 1010 and the SDR software. My hardware consists of an Athlon XP 2000+ on a generic 2002 vintage motherboard. The software is *Ubuntu* 8.04 with the "real-time" package, and some additional packages described below. The "real-time" package allows *Linux* to provide better real time performance.

The Jack Audio Connection Kit (*JACK*) is an audio server. *JACK* collects audio from sources such as device drivers and directs the audio to applications according to the user's direction. *JACK* allows audio applications to be chained together so that the output of one application becomes the input of another, without introducing latency.

The *sdr-core* program by Frank Brickle, AB2KT, and Bob McGwier, N4HY, provides the actual receiver functionality.[11] *SDR-core* is modular, in that this program has no user interface and depends upon *JACK* for the audio interface.

The *sdr-shell* program by Edson Pereira, PU1JTE, provides the user interface for *sdr-core*.[12] Through this interface the operator sets the receive frequency, the mode, the bandwidth, the AGC settings, and all of the other receiver attributes.

All of these software components of a typical *Linux*-based SDR receiver are undergoing active development by their authors, including the documentation. I strongly recommend checking the DttSP-Linux group on Yahoo for the latest information.[13]

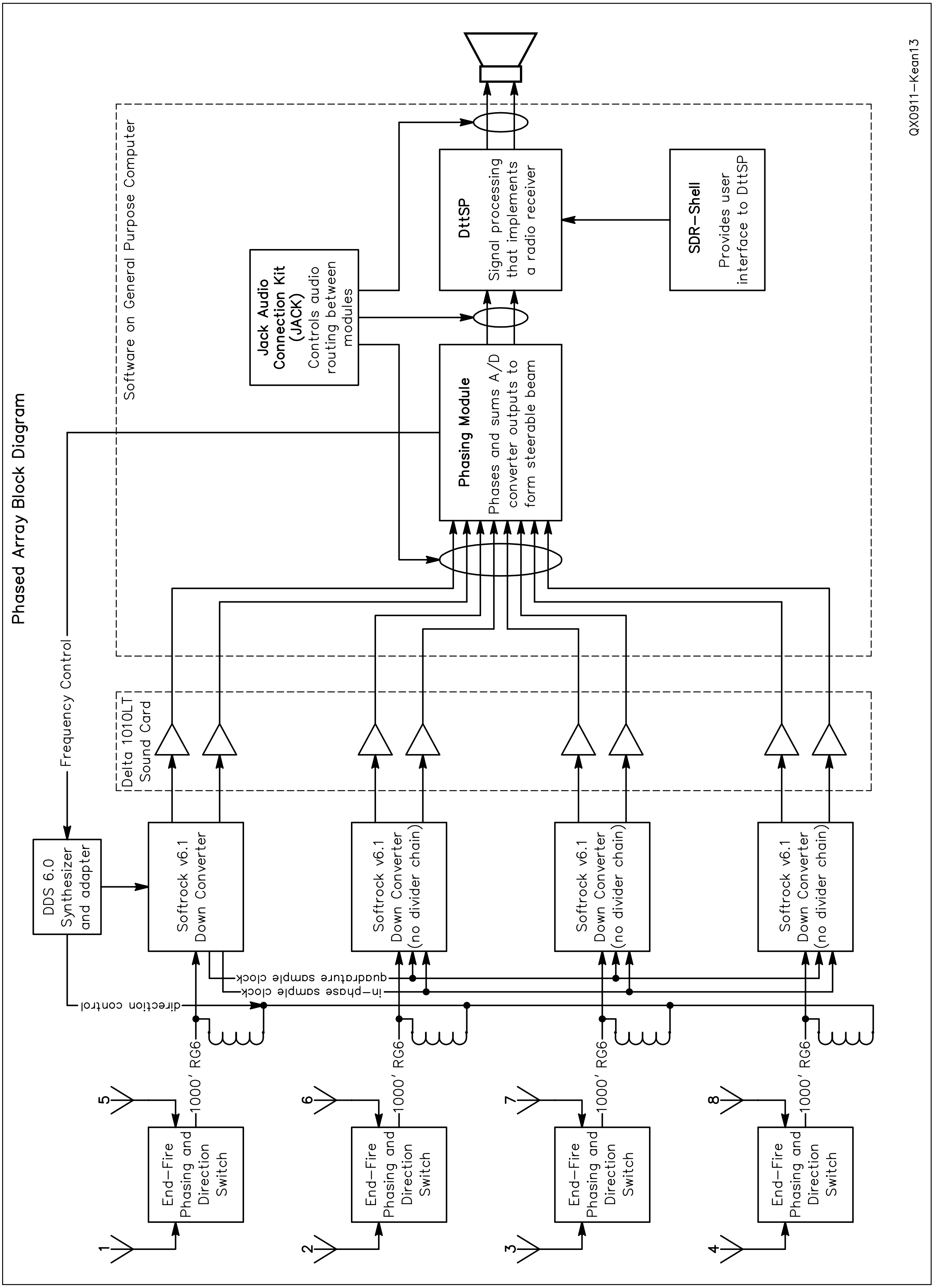



Figure 13 — The system block diagram.

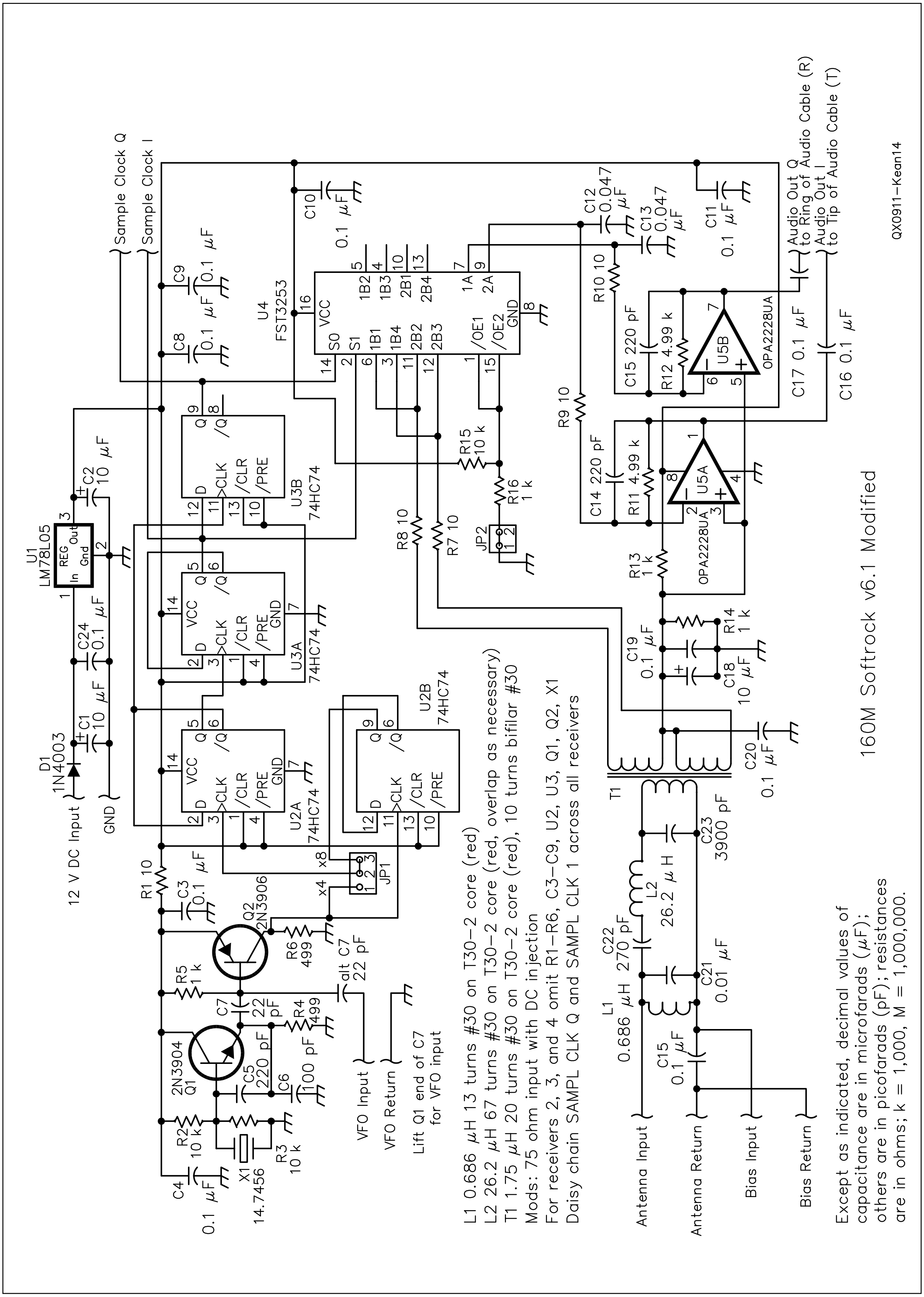



**Figure 14 — This schematic shows the modified 160 meter Softrock v6.1 receiver.**

## Phasing Software Calculations

*SDR-core* requires only a single I and Q input, whereas the four antenna elements via the Delta 1010 supply four sets of I and Q. What's missing? The module that I wrote, which I call "*Phasor*," applies the phase shift, sums the four sets of I and Q signals, and produces a single I and Q set as output, which becomes the input for *sdr-core*.

The mathematics behind beam steering for a linear array such as the broadside array is actually fairly simple. (See Note 1.) First, number the elements from north to south starting at zero. Next, apply a phase shift to each element according to Equation 1:

$$\varphi_n = \frac{(n-1)\bullet 2\pi \bullet \text{spacing}}{\text{wavelength}} \bullet \sin(\text{steering angle}) \quad \text{[Eq 1]}$$

where "n" is the element number, $\varphi_n$ is the phase shift for element "n," and "spacing" is the distance between elements in the same units as the wavelength. The steering angle is the difference between the array minor axis (short dimension) and the desired signal bearing. Note that the steering angle is a mathematical angle, which counts counterclockwise starting from due east, whereas hams use a navigational bearing, which counts clockwise starting from due north. So, to convert a bearing to a steering angle, subtract the bearing from 90°.

The software assigns the amplitude value for each element according to the type of pattern desired by the operator. For a broadside array with maximum directivity, $A_n$ is 1. For a broadside array with no sidelobes but about 1 dB less directivity, $A_n$ follows a binomial pattern. For three elements, the pattern is 1:2:1 and for four elements, the pattern is 1:3:3:1. That is, $A_1 = 1$, $A_2 = 3$, $A_3 = 3$, and $A_4 = 1$. Note that the popular 1:2:2:1 feed pattern for four elements produces a compromise between maximum directivity and minimum side lobes.

Then, apply the phase shift to the signal from each receiver. Recall that a direct conversion software radio requires two analog inputs for the I and Q signals. (See Note 9.) The I sample is the magnitude of the real part of a complex number and the Q sample is the magnitude of the imaginary part of that complex number. As a mathematical expression, one complex sample of data from the received signal is:

$$X_n = I_n + j\,Q_n \quad \text{[Eq 2]}$$

In other words, when quadrature sampling, $I_n$ is sufficient to represent the amplitude of the signal from receiver n at some point in time, but $X_n$ includes both the amplitude and phase of that signal. (See Note 7.)

Applying the element phasing is merely multiplying the complex sample value from each receiver with the complex phasing value for that element and adding them up. Convert the separate amplitude and phase numbers into a complex number:

$$C_n = A_n \bullet \cos\varphi_n + A_n \bullet j\sin\varphi_n \quad \text{[Eq 3]}$$

where $A_n$ and $\varphi_n$ are the individual element amplitudes and phases as determined above.

Multiply each receiver's signal $X_n$ by the corresponding element factor $C_n$ and sum them all:

$$R = X_1 \bullet C_1 + X_2 \bullet C_2 + \ldots + X_n \bullet C_n \quad \text{[Eq 4]}$$

The result R is a complex number that is converted back to real numbers by recalling that:

$$R = I_r + j\,Q_r \quad \text{[Eq 5]}$$

and extracting the $I_r$ and $Q_r$ values. If complex numbers are too complicated, you can consider the process in terms of real numbers and a little additional trigonometry:

$$I'_n = A_n \bullet \sqrt{(I_n^2 + Q_n^2)} \bullet \cos(\varphi_n + \tan^{-1}(Q_n / I_n)) \quad \text{[Eq 6]}$$

$$Q'_n = A_n \bullet \sqrt{(I_n^2 + Q_n^2)} \bullet \sin(\varphi_n + \tan^{-1}(Q_n / I_n)) \quad \text{[Eq 7]}$$

$$I_r = I'_1 + I'_2 + \ldots + I'_n \quad \text{[Eq 8]}$$

$$Q_r = Q'_1 + Q'_2 + \ldots + Q'_n \quad \text{[Eq 9]}$$

where $I'_n$ and $Q'_n$ are intermediate values and $\tan^{-1}$ is the inverse tangent (arc-tangent) function.

The resultant $I_r$ and $Q_r$ values are output to the software defined radio software. Note that all of the calculations above have to occur for each and every sample as the sound card delivers them, 96,000 times per second, per receiver. That's a lot of computation, which is why you'll need a fairly modern computer!

Because of the modularity provided by the *JACK* architecture, the *sdr-core* program is completely unaware that there is any additional upstream processing. Thus *sdr-core* performs its processing and delivers the received signal to your speakers.

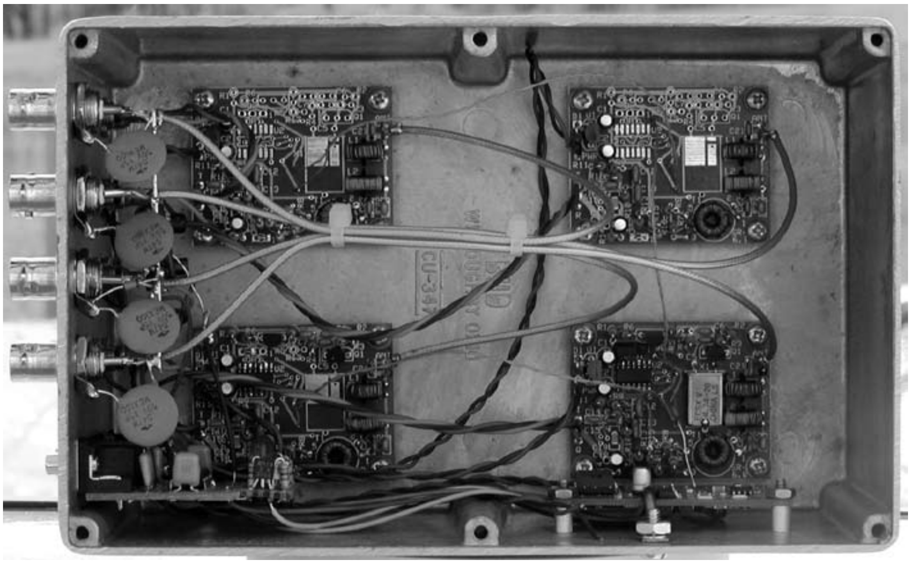


**Figure 15 — Four Softrock v6.1 Receivers and a DDS-60 VFO in a cast aluminum box. The box also contains a serial interface circuit to drive the VFO controls and the direction control relays via an RS-232 port.**

## Phasing Software

Although the calculations are fairly straightforward, a usable *phasor* module requires operator input, mainly to obtain the desired steering angle. The operator may also choose among several array patterns, ranging from the uniform feed, which produces maximum directivity, to the binomial feed, which eliminates side lobes. Also, each element can be individually routed to the receiver so that element performance can be compared and localized noise sources identified. Finally, the operator needs to execute the calibration procedure, and the program displays the calibration factors, which may suggest whether or not the hardware is functioning optimally.

Figure 16 is a screen capture of *sdr-shell* and *phasor* running together. *JACK* and *sdr-core* are also running, but their only visible manifestations are a few messages in the terminal session window. The *phasor* window shows the current bearing, the approximate antenna pattern (which shows you where the nulls and side lobes point), and calibration data. The window also shows the individual element amplitude and phase settings.

The *phasor* source code is available via Google Code.[14] *phasor* is written in the *C99* dialect of the *C* language and requires a number of libraries for support, such as the GNU Scientific Library for matrix algebra, the *JACK* library for audio plumbing, the FFTW3 library for Fast Fourier transforms, and others. The software repository includes build documentation and a brief user manual.

## Calibration

Despite considerable effort to make all of the antennas as identical as possible, and to make each of the Softrock receivers as identical as possible, I still found considerable variation from one antenna and receiver to another. In order to retain as much performance as possible, I had to compensate for the receiver and antenna variability with a calibration procedure.

I divided the calibration procedure into two phases: one step to calibrate the receivers so that identical inputs produce identical outputs, and a second step to calibrate the antennas so that each antenna and receiver presents the same output for a perfectly broadside signal.

I calibrate the receivers with a signal generator and a Mini-Circuits 4 way combiner/splitter module. *Phasor* has a "receiver calibration" mode where the program measures the amplitude and phase of the strongest signal and calculates and stores the correction factors. Currently, I calibrate at 1830 kHz and assume a constant correction for the entire band. However, most of the phase and amplitude error comes from the pass band filter at the input to each receiver. This error is very likely to be frequency dependent, so a better scheme would be to measure the shape of the amplitude and phase error curves, and then store the coefficients of a fitted polynomial. Some future version of *phasor* may support frequency dependent calibration.

Once the receivers are calibrated, the antennas can be calibrated. I have tried several techniques to calibrate the antennas, including transmitting from a special calibration antenna at the middle of the array, receiving a well-known signal such as W1AW or an AM broadcast station, and transmitting in-band from a chosen off-site location. So far, the last method works best.

I set up a simple (not particularly efficient) mobile station in the car. I put the phased array receiving system in recording mode, and drive to a convenient location about a mile away that is directly in front of the array. I transmit for a minute or so, and return home.

I then play back the recording with *phasor* in "antenna calibration" mode. Again the program calculates and stores the amplitude and phase correction factors.

Note that the program cannot distinguish between errors due to receiver differences or antenna differences. Thus, receiver calibration occurs first, so that during antenna calibration all of the remaining error can be attributed to the antennas.

## Benefits

The obvious benefit is a fairly directive, steerable receiving system for 160 meters. The main lobe is sharp enough that I can see a difference between two headings with only a 15° difference.

While the main lobe is very broad, the pattern nulls are fairly sharp. Often, one can steer the array to minimize QRM without attenuating the desired signal. For example, from Ohio, QRN from thunderstorms in the Southeastern US is often quite strong. When listening to Europe, the first null in the pattern points to the southeast, attenuating QRN. The phased array hears Europe in the presence of this QRN *much better* than my Beverage antennas oriented for Europe.

Although eight short verticals require considerably more effort to erect than a 900 foot wire, they take only slightly more room. The entire array fits into an area 120 feet by 920 feet, including radials. The equivalent directivity from an array of Beverages would require 340 feet by 850 feet, and that gets you only two directions.

Spatial information can be recorded and replayed later. During the 2007 CQ 160 Meter CW contest, I recorded several minutes of raw data from the output of the Delta 1010. When played back, I can steer the antenna "after the fact" (except for throwing the relays, of course). Thus, when I listen to a QSO embedded in QRM, I can adjust the receiving direction to maximize the desired signal or minimize the QRM. Recording requires eight channels at 96,000 samples of 4 bytes each per second, however, which when multiplied out comes to 3 megabytes per second. It's a good thing that hard drives are relatively cheap.

The linear array is not necessarily the best use of real estate. A circular array seems quite attractive, since one could steer all the way around the compass without any significant loss of directivity. Also, a circular array would avoid relays, which would permit the real-time recording of all available signals. Circular arrays have a severe drawback, however: for the same number of elements as a linear array, the side lobe suppression is very poor in comparison. For example, eight elements in a circle 200 feet in diameter with an additional element in the center fed for maximum directivity has side lobes that are only 8.4 dB down from the main lobe. Also, the main lobe beam width is 33°. Finally, of the total of 7 side

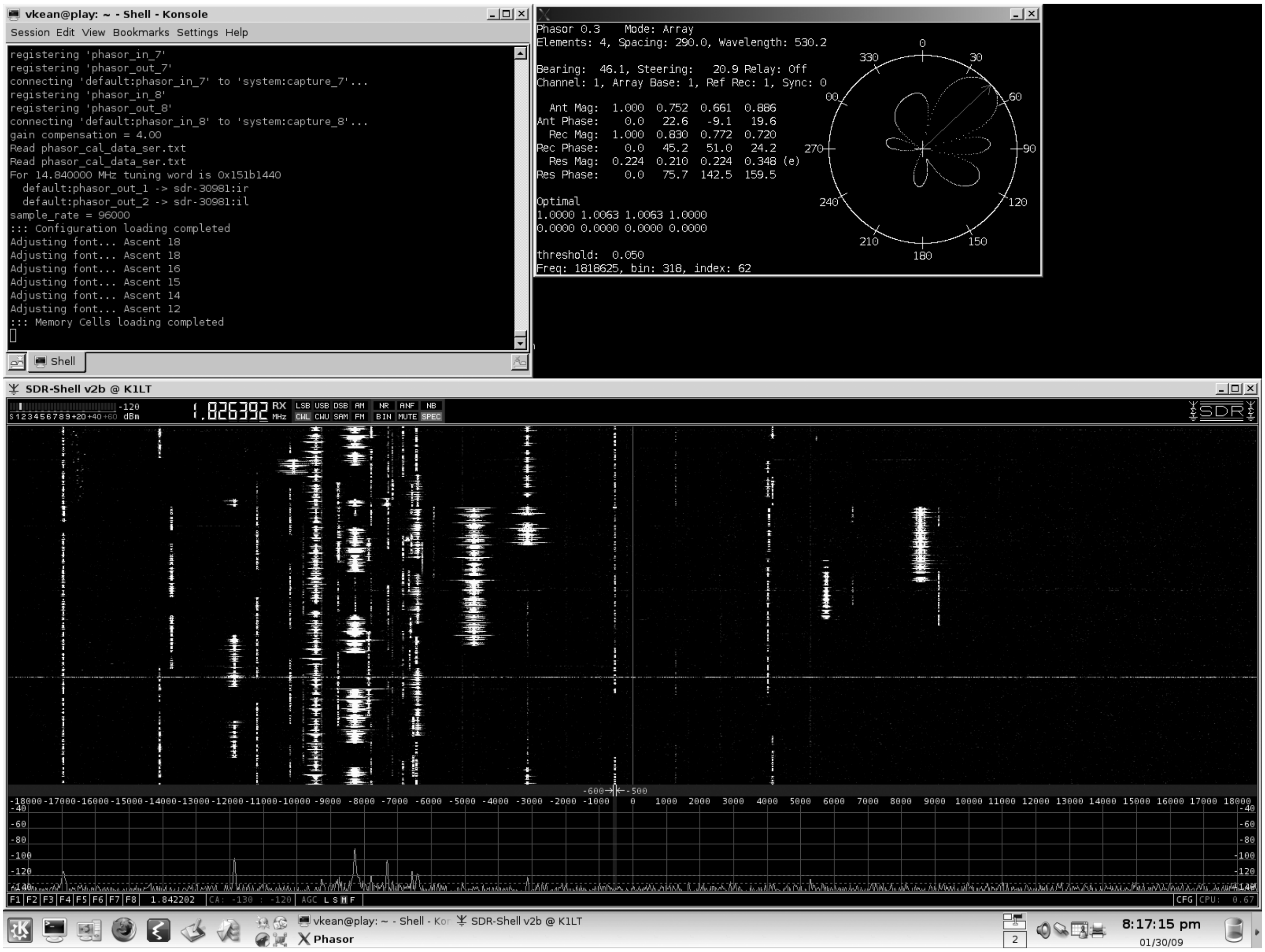



**Figure 16 — Here is a screen capture of *phasor*, *sdr-core* and *sdr-shell* in operation. The window in the top center is *Phasor*, the window in the top left is a terminal session that initiated all of the programs, and the window that occupies the bottom two thirds of the screen is *sdr-shell*. The signal just to the left of the vertical line marking the center of the waterfall display is SM5EDX working K4WM.**

lobes, the smallest (off the back) is down only 17 dB. By comparison, a 4 element broadside array of 2 element end-fire arrays fed for maximum directivity has maximum side lobes 13 dB down and a 28° beamwidth.

Fortunately, there are other circular patterns with better performance, at the expense of a more complex layout. For example, 2 concentric circles, 5 elements on the outer circle, 3 elements on the inner circle, and one in the center has maximum side lobes 13 dB down and a 45° beamwidth.

Software defined radio brings several advantages to traditional radios. First, a spectrum display is available. During a contest, the spectrum display is invaluable for finding an empty frequency on which to call CQ. When chasing DX, one can easily find the last caller to work a station in a monstrous pileup. Second, any arbitrary pass-band filter becomes available with a couple of mouse clicks (or push buttons, when I get around to changing some software), with a bandwidth as narrow as 20 Hz with no ringing. Finally, one can tune to the next signal by merely clicking on the spectrum display. When searching and pouncing, I often skip the bright traces that indicate strong (likely) local signals, and look for the dim traces that usually correspond to desirable DX signals.

## Disadvantages

The principal disadvantage of beam steering via SDR is the fact that the end result is a complete receiving system, rather than just an antenna. Thus, I had some difficulty integrating this receiving system into my traditional contest station. The initial solution required that the operator adapt to knob-less tuning and the uncoordinated separate receiver and transmitter. After a few contests of practice, I stopped using my Beverage antennas (except for my JA Beverage) and virtually ignored the receiver in the ICOM IC-765. I did "mis-calibrate" the frequency display on the SDR receiver to indicate my typical receive frequency rather than "zero beat" frequency, so that I could put the transmitter on the receiver frequency by matching the digital readouts. Fortunately, *sdr-shell* includes "mouse gestures" as a tuning aid, which mostly overcomes the lack of knobs.

After a few contests manually tuning the transmitter, I modified *sdr-shell* to send the transmit frequency via a TCP connection to some free virtual serial-port software on my logging computer. The virtual serial-port software combines the data stream from the TCP connection with the data stream from the logging program and sends the data to the transceiver via a physical serial port. Thus, clicking on a signal on the spectrum display puts both receiver and transmitter on frequency instantly.

The second main disadvantage was that in order to gain adequate selectivity, I had to accept significant receiving latency. In other words, every received signal was about 20 milliseconds late in comparison to the analog radio. A 20 millisecond delay doesn't sound like much, but I can't listen to the transmitter side-tone and my own received signal at the same time without going insane. To reduce the latency, one would either have to sacrifice selectivity, which is unacceptable in the contest environment, or dramatically increase the sampling rate, which requires more computational power (not so bad) and a drastically more powerful sound card. Actually, sound cards with a sufficient number of channels and a high enough sampling rate are not readily available without resorting to laboratory grade equipment. Hopefully, the march of technology will resolve this issue soon. For now, I turn off the transmitter side-tone.

The third disadvantage is the lack of dynamic range, which is exacerbated by the direct conversion architecture. Because the entire 1800-1896 kHz band must pass through the nominally 20-bit analog to digital converters, the available dynamic range is reduced in comparison to a narrow-band architecture. Fortunately, at my location, there are rarely any super strong signals, and most of the time I am able to receive, even in a busy contest, without any overload. The exception to my good fortune are those times when strong "line noise" occurs, which consumes a considerable amount of dynamic range, because of the broadband nature of noise. When the noise appears, then strong signals appear to splatter. The solution to this problem requires a more sophisticated software radio architecture. The digital down-conversion architecture uses over-sampling to increase the effective number of bits, which in turn improves dynamic range. This architecture gets very expensive, however, when replicated four times.

## Future Opportunities

There are several improvements to the smart antenna that I have not had time yet to explore. For example, one can generate more than one beam simultaneously. Besides the immediate benefit of listening in more than one direction at a time, the additional beams can be used to subtract an offending signal from the overall pattern. For example, by placing a beam over thunderstorms to the southeast, I might be able to remove the QRN that leaks into the main beam from a side-lobe.

Another possibility is automatically finding signals after CQing. Rather than manually pressing keys or clicking compass points to rotate the array to find a caller, the software could, in principal, listen in all directions simultaneously, and present the operator with a signal with the antenna already properly positioned. Likewise, the receiver could automatically tune and filter that signal.

These opportunities appear to require only an investment in software development. Once someone demonstrates the appropriate algorithm, then everyone can benefit from the effort. Hopefully my work will inspire others to take the next step.

## Notes


[1]M. T. Ma, *Theory and Application of Antenna Arrays.* New York: John Wiley & Sons, 1974.

[2]John Devloldere, ON4UN. *ON4UN's Low-Band DXing*, 4th edition, ARRL, 2008. pp 7-5 to 7-8.

[3]Tom Rauch, W8JI. "Beverage Length," Online Posting. 19 Oct 2000, Topband, 30 Jan 2009 **lists.contesting.com/_topband/2000-10/msg00123.html**.

[4]Tom Rauch, W8JI. "Small Vertical Arrays," Online Posting, 27 Feb 2008, **www.w8ji.com/small_vertical_arrays.htm**.

[5]John Devloldere, ON4UN, *ON4UN's Low-Band DXing*, 4th edition, ARRL, 2008. pp 7-5 to 7-8.

[6]Richard G Lyons, *Understanding Digital Signal Processing*, New York: Addison-Wesley Longman, Limited, 1997.

[7]Richard G Lyons, "Quadrature Signals: Complex, But Not Complicated," *DspGuru: DSP Central*, Nov 2008, 02 Feb 2009, **www.dspguru.com/info/tutor/quadsig.htm**.

[8]Tony Parks, KB9YIG, "Softrock v6.1 Schematic," Yahoo groups file, 25 Aug 2006, Yahoo Softrock40, 7 Jan 2009, **http://f1.grp.yahoofs.com/v1/kLyzStSna0dvh5bRqhv4gMPWrWWJhEz41VC0e8kQTo8I45IRVoIvUsrY_M2bgsQspQSS30lf_gfta7Ydg_cHMU16C3jMBA/%27A2%20SoftRock%20v6.0%20docs/SR_v6.1_schematic_8_25.pdf**.

[9]Gerald Youngblood, AC5OG (now K5SDR), "A Software-Defined Radio for the Masses, Part 1," *QEX*, Jul/Aug 2002, pp 13-21.

[10]The American QRP Club, "DDS-60 Kit from the AmQRP Club," 30 Jan 2009, **www.amqrp.org/kits/dds60/**.

[11]The Comprehensive GNU Radio Archive Network (CGRAN), "DttSP". 30 Jan 2009, **https://www.cgran.org/wiki/DttSP**.

[12]Edson Pereira, PU1JTE, "SDR-Shell," 14 Jan 2008, 02 Feb 2009, **http://ewpereira.info/sdr-shell/**.

[13]Dttsp-linux, 2 Feb 2009 **http://groups.yahoo.com/group/dttsp-linux/?yguid=2942992**.

[14]Victor A. Kean, Jr., K1LT, "*phasor*," *Google Code*, 03 Feb 2009, **http://code.google.com/p/phasor/**.


*Victor Kean, K1LT, is an ARRL Life Member, and a Software Engineer from the Central Ohio area. He was first licensed in 1969 as WN1LKU and has been working with computers ever since his high school days. He obtained his BS in Electrical Engineering from The Ohio State University in 1976 and was very active with the Ohio State Radio Club, W8LT. He has worked with microprocessor controlled repeater networks, packet radio networks, and most recently, digital radio. He has always been interested in the 160 meter band, and has operated nearly every ARRL 160 meter contest since 1973. When he is not contesting on 160 or reading Slashdot, Victor is likely to be preoccupied with his 5 year old daughter.*