$5

QEX

November/December 2009

www.arrl.org

A Forum for Communications Experimenters

Issue No. 257

VE5FP built this 0.1 MHz to 200 MHz signal generator based on the NimbleSig III DDS circuit designed by VA7TA.

ARRL The national association for AMATEUR RADIO
225 Main Street
Newington, CT USA 06111-1494

KENWOOD

Listen to the Future

TS-480

The Perfect Remote Base Transceiver

Straight Out of the Box!

- **The perfect internet base transceiver - straight out of the box!**
- **Easy to operate.**
- **The size makes it great for base, mobile or portable operation.**
- **Free VoIP/Control software downloads at Kenwoodusa.com.**
- **Incredible RX specifications.**
- **No expensive sound card interface needed.**

KENWOOD U.S.A. CORPORATION
Communications Sector Headquarters
3970 Johns Creek Court, Suite 100, Suwanee, GA 30024
Customer Support/Distribution
P.O. Box 22745, 2201 East Dominguez St., Long Beach, CA 90801-5745
Customer Support: (310) 639-4200 Fax: (310) 537-8235

www.kenwoodusa.com
ADS#16509

JQA-1205 091-A
ISO9001 Registered
Communications Equipment Division
Kenwood Corporation
ISO9001 certification

QEX (ISSN: 0886-8093) is published bimonthly in January, March, May, July, September, and November by the American Radio Relay League, 225 Main Street, Newington, CT 06111-1494. Periodicals postage paid at Hartford, CT and at additional mailing offices.

POSTMASTER: Send address changes to: QEX, 225 Main St, Newington, CT 06111-1494
Issue No 257

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 860-594-0200
Fax: 860-594-0259 (24 hour direct line)
e-mail: **qex@arrl.org**

Subscription rate for 6 issues:

In the US: ARRL Member $24, nonmember $36;

US by First Class Mail:
ARRL member $37, nonmember $49;

International and Canada by Airmail: ARRL member $31, nonmember $43;

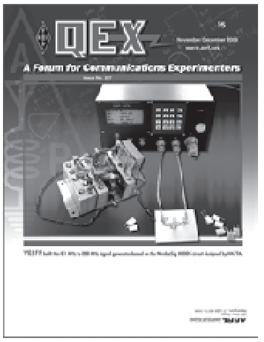
Members are asked to include their membership control number or a label from their QST when applying.

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.

November/December 2009

About the Cover

Jim Koehler, VE5FP, built a 0.1 MHz to 200 MHz signal generator based on the NimbleSig III DDS circuit designed by Tom Allldread, VA7TA. He includes a design and construction information, as well as some test application suggestions.



In This Issue

Features

Columns

Index of Advertisers

The purpose of *QEX* is to:

1) provide a medium for the exchange of ideas and information among Amateur Radio experimenters,

2) document advanced technical work in the Amateur Radio field, and

3) support efforts to advance the state of the Amateur Radio art.

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and letters for publication in *QEX* should be marked Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be submitted in word-processor format, if possible. We can redraw any figures as long as their content is clear. Photos should be glossy, color or black-and-white prints of at least the size they are to appear in *QEX* or high-resolution digital images (300 dots per inch or higher at the printed size). Further information for authors can be found on the Web at **www.arrl.org/qex/** or by e-mail to **qex@arrl.org**.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the Editor or the League. While we strive to ensure all material is technically correct, authors are expected to defend their own assertions. Products mentioned are included for your information only; no endorsement is implied. Readers are cautioned to verify the availability of products before sending money to vendors.

Larry Wolfgang, WR1B

lwolfgang@arrl.org

Empirical Outlook

I am writing this editorial shortly after returning from the 28th ARRL/TAPR Digital Communications Conference, held in Chicago again this year on September 25-27. The excitement and enthusiasm of the attendees at a DCC is hard to imagine until you have experienced it. Friday and Saturday are chock full of technical presentations, with some time squeezed in to meet and talk with other attendees as well as checking out the nifty displays in the Demonstration Room — better known as the "Play Room!" If you haven't attended one of these conferences, you should add it to your future plans.

Somehow, newly elected TAPR President Steve Bible, N7HPR, had twisted my offer to lead one of the Saturday introductory sessions about surface mount construction into the first full presentation of the DCC on Friday morning. I will readily admit that I was quite nervous about giving that talk because most of the attendees are quite experienced at SMC, while I am a raw beginner. Everyone was kind, and they went easy on me, however. I picked up a few tips and ideas to try as a result of later discussions, too.

Scotty Cowling, WA2DFI, (newly elected TAPR Vice President) had the idea to spin off my talk by setting up a soldering station in the Demo Room, so anyone who had not tried soldering surface mount components could give it a try. He brought some surplus circuit boards as well as a roll of resistors and a few 208 pin FPGA ICs! (And I thought the ½ inch square, 64 pin IC I soldered onto the board for my NUE-PSK modem was a challenge!) I was surprised by the number of attendees who wanted to practice soldering a few surface mount resistors onto a board. There were even a few brave souls who tackled the FPGA ICs. Patience is definitely required!

There were many other interesting projects and displays in the Demo Room. Most of them were highlighted by technical presentations during the DCC. Bill Brown, WB8ELK, displayed several of the transmitters he flies on his high altitude balloon launches. Bill's presentation at the Saturday evening banquet was most interesting and inspiring. Bob Bruninga, WB4APR, had an APRS and text messaging system display. Bob also gave a presentation about the universal ham radio text messaging initiative. David Bern, W2LNX, had his PS/2 Keyer on display. David uses a PIC microcontroller to take the Morse code input from a keyer paddle and emulate a PS/2 keyboard and mouse for computer text entry and cursor control. Expect to read more about that project in a future issue of *QEX*!

John Hansen, W2FS, gave a presentation about using the ZigBee protocol and XBee Pro modules to create a virtual serial cable. This is a very convenient way to remotely program your mobile transceiver from a computer inside your house. Of course there are many other possible applications for this system! He had his system on display in the Demo Room, too. ICOM and Kenwood had small displays set up, as did Don Arnold, W6GPS, with the AvMap GPS.

There were a number of other displays throughout the weekend, and of course many more very interesting technical presentations, along with several introductory sessions on Saturday. You could learn about digital Amateur TV, several experiments with packet radio networks and an update on the high performance software defined radio (HPSDR) project. I can't recount all of the presentations here. You'll have to attend a DCC yourself to get the full flavor of the conference.

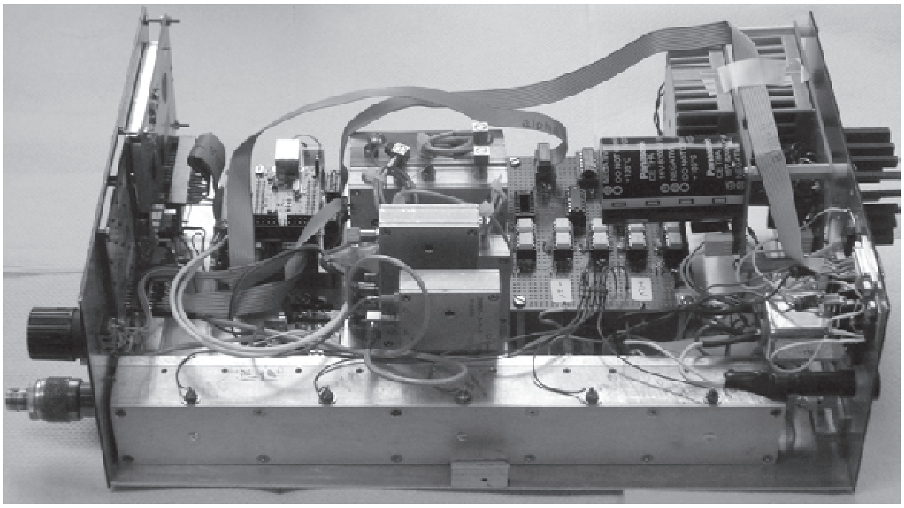
Watch the Upcoming Conferences listings in *QEX* for the dates and locations of many other interesting technical conferences throughout the year. There are regional VHF/UHF/Microwave conferences, Microwave Update and the AMSAT Space Symposium each year. Pick one or several that sound interesting to you, and plan to attend. Better yet, plan to give a presentation to share your ideas with others at the conference. Most of all, have fun with ham radio!

Jim Koehler, VE5FP

2258 June Road, Courtenay, BC V9J-1X9 Canada; jim.koehler@usask.ca

A DDS-Based Signal Generator

The author built a versatile piece of test equipment using the NimbleSig III DDS by Tom Alldread, VA7TA.



A peek inside the signal generator.

If you were to canvas several Amateurs who like to build equipment at home, and ask them what test equipment is most necessary, you would get many answers, but most would agree that a signal generator is somewhere near the top of the list. For myself, I would put it third after a VOM and an oscilloscope.

Nowadays, most hams of an experimental inclination make do with older, surplus commercial equipment. These instruments are very good and very inexpensive relative to their original cost. To get the features I want in a signal generator, however, it would still be more expense than I want to pay, and even then it might be difficult to find an instrument that meets all my requirements.

The features I want in a signal generator are:

1) Good frequency stability;
2) Low phase noise;
3) Frequency accuracy of 1 Hz or better;
4) Accurate output levels;
5) Built-in RF detector with good accuracy;
6) Computer interface for control and data recovery;
7) The ability to scan in frequency and to display the response of filters.

The advent of direct digital synthesis (DDS) has greatly expanded the possibilities of what a home experimenter can do to make stable, accurate signal sources of laboratory quality. Recently Tom Alldread, VA7TA, has developed NimbleSig III (NS3), a DDS-based module.[1] It provides the basis for a signal generator that fulfills all my requirements. In addition, it provides two independent RF outputs, which greatly increases its versatility.

Before I describe my design in more detail, let me first list some of the attributes of the DDS that make it so suitable for signal generators:

1) Accuracy and precision of the frequency;
2) Accuracy of the output level;
3) The ability to amplitude and frequency modulate the signal with great precision.

The precision of the frequency is determined by the minimum frequency step size; this is usually much less than one hertz. The absolute accuracy of these output frequencies is determined only by the accuracy of the clock source.

The fact that the amplitude of the output waveform is determined by a digital-to-analogue (D/A) converter means that one can set the output amplitude to an accuracy that is determined by the number of bits in the D/A module inside the DDS circuit and the linearity of that module. In the NS3 *QEX* article, VA7TA discussed at length the accuracy with which both AM and FM were produced.

There is a downside to using a DDS for a signal source and that is that the upper limit to the frequency is determined by the particular DDS circuit and its maximum clock frequency. Modern commercial DDS integrated circuits have a clock limit of about 1 GHz resulting in a maximum un-aliased output frequency of about 400 MHz. NS3 uses a 500 MHz clock so its upper limit is about 200 MHz. For much of the sort of experimentation that I do, that is more than good enough.

A minor additional downside is the presence, in the output spectrum, of spurious output signals. Again, VA7TA has discussed this in his articles. For NS3, these spurious signals are down at least 60 dB relative to the main output. While the presence of such spurs may be a problem if you are using a DDS as a local oscillator in a receiver, they

[1]Notes appear on page 13.

are not usually a significant problem in a signal generator.

This article will not be a detailed description with complete circuit diagrams but rather a discussion of the necessary elements in the design.

The Basic Signal Generator

A block diagram of a generic DDS-based signal generator is shown in Figure 1. The clock determines both the stability and the absolute accuracy of the signal generator so its design is fundamentally important.

The controller provides the instrument to human interface and is, in some ways, the most difficult part of the project to design. For example, I have limited bench space so I did not want a really large instrument that could have many controls. On the other hand, limiting the number of knobs and switches can make the instrument awkward to use. If you have to push too many buttons to change a few settings, it becomes tiresome very quickly.

Since my desired signal generator has a built-in power detector, there must be some display of recorded power; I opted to use a 64 × 128 pixel graphics display, which allowed me to simulate a running graph showing signal levels and other graphical displays.

Updating a graphical display many times per second, even one this small, takes some processing power, so the controller must be fairly fast. A major consideration in the choice of microprocessor is the software that is available for programming it. Many years ago I started doing hand assembly for some of the first 8 bit microprocessors, but nowadays I want a good compiler and all the bells and whistles of a good development platform. I chose to use an NXP LPC2138 or LPC2148 because there is a good, professional, inexpensive (free!!) *C* compiler for them (the GNU compiler) and there is a very low-cost development platform for that compiler; the Rowley Associates *Crossworks* package.[2] Rowley has a full version of this package available for non-commercial use at a very reasonable price. I've never begrudged buying tools for my workshop and I view buying tools like this development system in the same light. *Crossworks* allows you to debug using a JTAG interface, so you can examine internal registers and observe data variables in the program while it is running in the actual microprocessor. JTAG also is used to load the program into microprocessor flash memory. It makes for a very convenient and easy way to design and debug microprocessor firmware.

The NXP LPC2138/48 is an ARM7 processor, runs internally at 60 MHz, executes most instructions in a single clock cycle and has 32 K bytes of RAM and 512 K bytes of flash program memory. With this large program memory, one can write a program using many of the better programming techniques because one is not constantly constrained

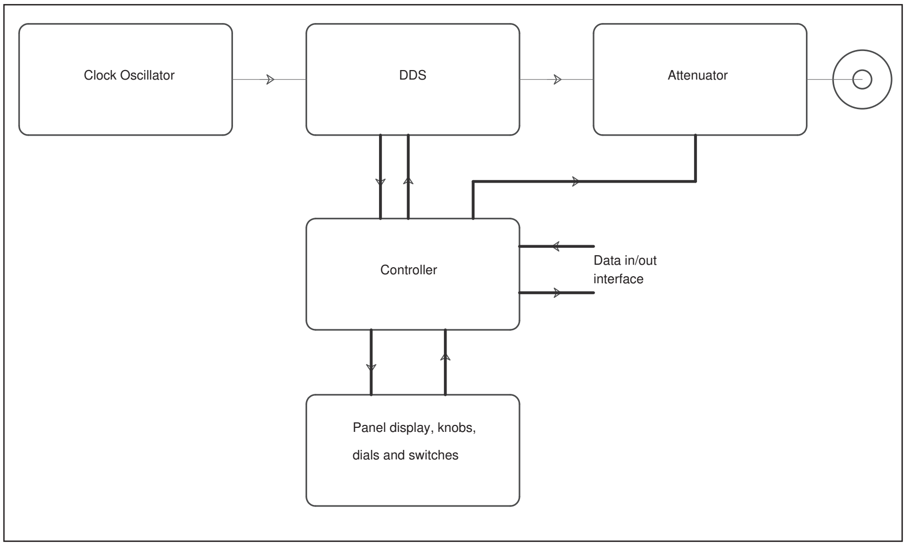


Figure 1 — A block diagram of the DDS based signal generator.

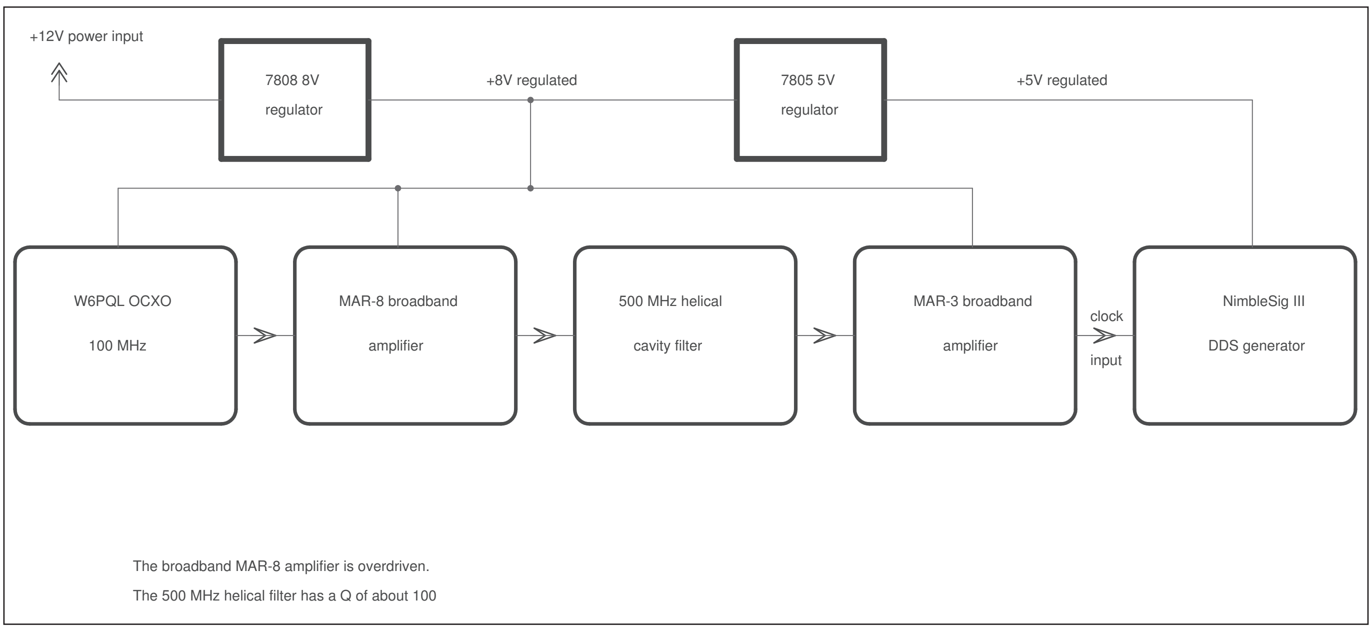


Figure 2 — A block diagram of the clock module used in the signal generator. The MAR8 amplifier is overdriven to produce harmonics of the 100 MHz crystal frequency, resulting in a 500 MHz reference signal.

by having to consider how much memory you're using. It is a far cry from when I was writing assembler programs for an Intel 8080 using 2708 EPROMs, which each had 1 K bytes of memory.

There are two RF outputs of NS3 and I opted to have just one of them go through an electronically actuated attenuator. I originally considered using a manual attenuator salvaged from a good grade commercial signal generator. For example, one often sees offers for non-working models of good quality signal generators on eBay. Many of these have perfectly good output attenuators and the prices are very low. I wanted to have an electronically controlled attenuator, however, because I could envision several operating modes in which I wanted the controller to be able to set output level. Also, with an electronically controlled attenuator, you do not require the additional front panel space for a knob and dial.

Let me now discuss the block elements from Figure 1 in some detail.

The 500 MHz Clock

The clock in the DDS used in NS3 is an internal 500 MHz oscillator, which is phase-locked to a 25 MHz TCXO. VA7TA has found that, at room temperature, this results in a very accurate, stable frequency. Because the clock is phase-locked, however, one would expect it to generate more phase noise than a good external clock. I decided to use the external clock option for NS3. I built a 100 MHz Butler oscillator from a kit by W6PQL.[3] This oscillator, using a good quality 100 MHz overtone crystal, provides a low phase noise source for the clock. A block diagram of the whole clock circuit, including the NS3 module, is shown in Figure 2.

The output of the oscillator is connected to the input of a broadband amplifier. The amplifier is overdriven and so generates harmonics of the 100 MHz input signal. The output of this amplifier is then passed through a helical cavity filter tuned to 500 MHz. The filtered output is then amplified to get the 500 MHz signal up to the +7 dBm level needed by the DDS chip inside NS3. Figure 3 shows the circuit used in both the first and second amplifier; they differ only in the actual MiniCircuits MAR series of amplifier used and the resistor that sets the bias for the stage.

The 500 MHz helical cavity filter was designed using the nomograph in recent issues of *The ARRL Handbook*.[4] To keep the loaded Q high, the input and output are very loosely coupled capacitively. The loss due to this loose coupling is compensated for by the second amplifier.

Figure 4 is a photo showing the entire assembly mounted on an aluminum plate. I have a large number of milled aluminum cases that I found at a scrap yard in Phoenix; they originally housed a MiniCircuits RF power divider/joiner. I used these little enclosures to separately make the 100 MHz oscillator (two cases bolted together) and one enclosure for each amplifier. The RF connections between the cases are made using RG-188A/U cable and SMB connectors. The large brass screw heads seen on each box were just to fill unused 10-32 tapped holes in the enclosures. The NS3 circuit board was

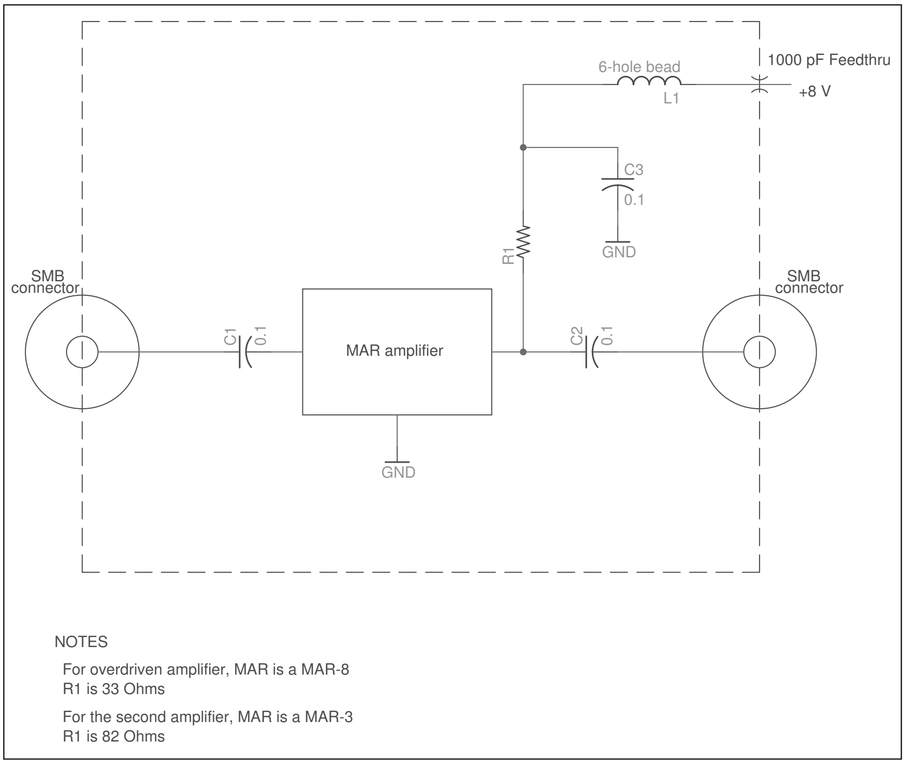


Figure 3 — This schematic diagram shows the amplifier circuit used in both the MAR3 and the MAR8 amplifiers. The overdriven amplifier, which generates harmonics of the crystal oscillator, uses a MAR8. The second amplifier uses a MAR3 device.

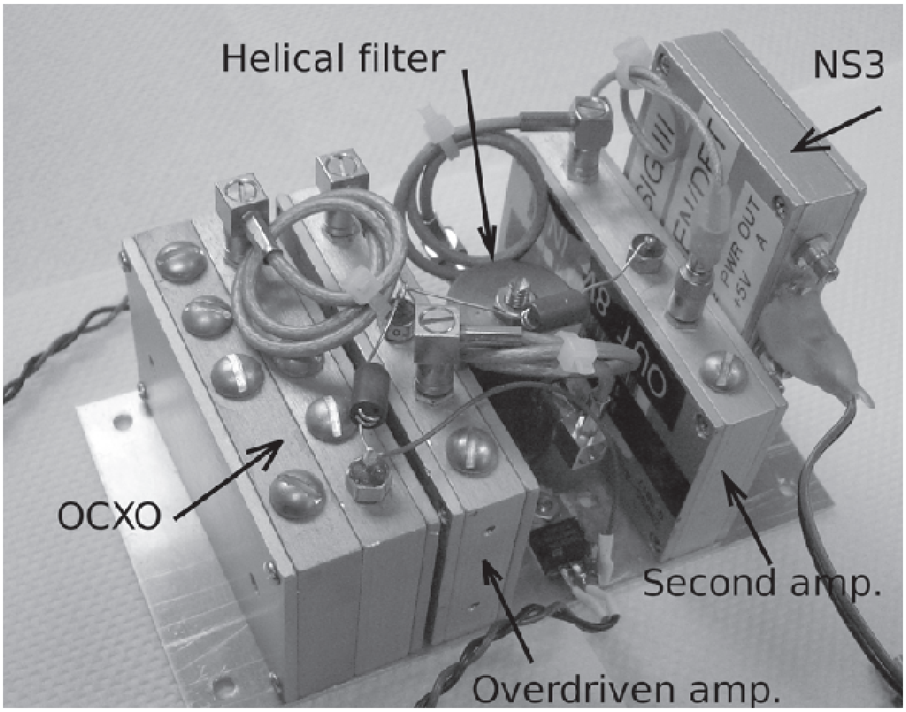


Figure 4 — This photo shows the modules used to make the 500 MHz clock for the DDS.

designed to fit in one of these cases and it is mounted adjacent to the second amplifier.

DDS Module

This module is just the NS3. The word *just* is an understatement since VA7TA has done a tremendous amount of work in making an Analog Devices AD9958 into a very simple to use and very flexible RF signal source. He has implemented amplitude and frequency modulation and a host of other features into a module that is controlled through a serial logic level interface. I won't describe NS3 in detail; instead, I just refer you to his original *QEX* articles.

The Attenuator

I haunt hamfest flea markets and, whenever I see an electronically controlled attenuator at a reasonable price, I buy it. They are not common but neither are they uncommon. Sometime in the past, I had acquired a nice looking attenuator that looked to be of good quality. The fellow I bought it from thought that it might have been from an HP instrument of some sort. It had an SMA connector at the input end and a Type-N connector at the output end; the output end obviously meant to protrude through the panel of some instrument. It had five sections; 10 dB, 20 dB and three of 40 dB. Each step was controlled by a latching relay, which was connected to a feed-through capacitor on the attenuator

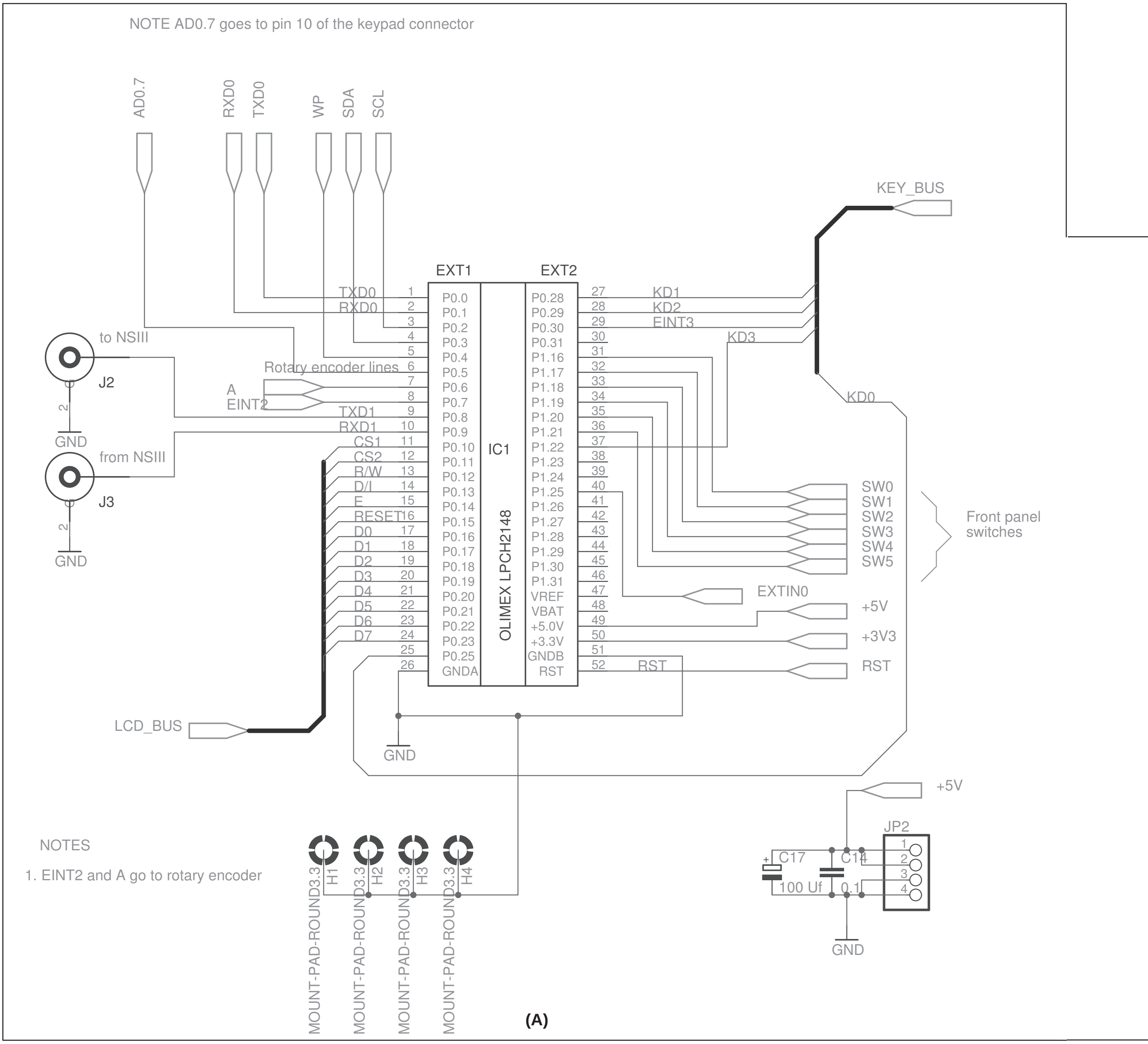


Figure 5 — The schematics of the various sections of the signal generator are shown here. Part A is the main CPU. Part B is the UART interface circuit. Part C shows the connections to the front panel. Part D shows the graphics LCD module and Part E is the external EEPROM memory. Part F is the keypad and decoder schematic. The port expander section is given in Part G.

body. I found that +12 V to the relay inserted the attenuation step and –12 V removed it. The fact that the input connector was an SMA type made me think that the attenuator was designed for at least the VHF range and, sweeping it on a wave analyzer, I found that it was quite flat from 4 MHz through 200 MHz. There's no reason to believe that it isn't flat down to close to dc.

I won't discuss the design of the interface between the attenuator and the ARM7 controller because no-one else is likely to want to duplicate it. I implemented it, wire-wrapped on "perf board," using 74LS family logic since this 5 V logic can be driven by the 3.3 V logic level controller output lines. The logic is designed so that the controller can control the attenuator with just five output signals; three lines for an address to select any one of the five sections, one line for direction (that is giving an output of either +12 or –12 V) and one to select or "activate" this decoder logic. I found that the latching relays needed to have the voltage applied for more than 50 ms to reliably switch an attenuator section in or out so, after the address and direction lines are set, the select line is held high for 50 ms to actuate the latching relays.

The Controller

Olimex manufactures a very small controller board using the NXP LPC2148 microprocessor; their part number is LPC-H2148.[5] It is designed to be a daughter board on a larger board and I decided that I could not make one like it for the purchase price. The LPC2148 is essentially identical to the LPC2138 except that it has a built-in USB interface. I did not use this USB port.

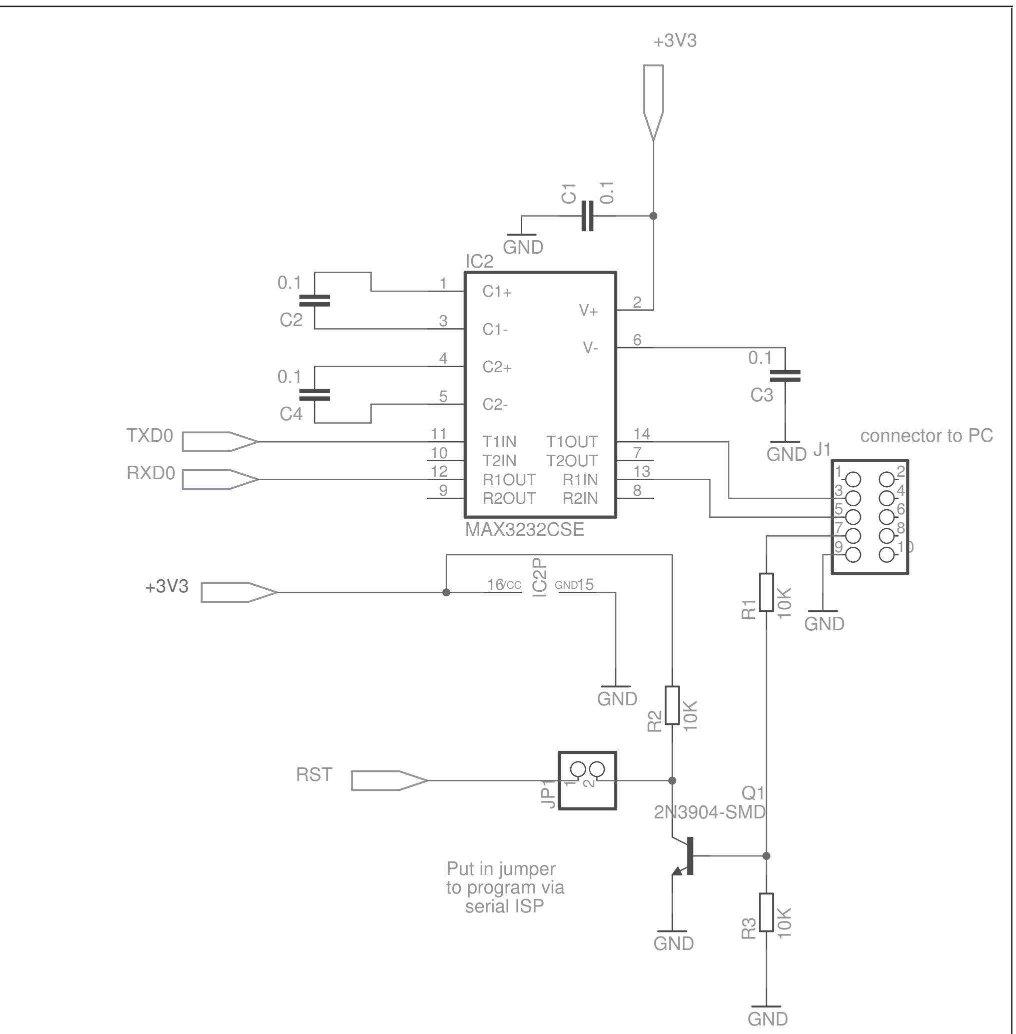
I designed a mother board to take this daughter board. The mother board circuit, shown in Figure 5, provides a keypad encoder and a number of 10-pin headers for connection to the rest of the signal generator and a 20-pin connector to the graphical LCD module. NS3, in its case, has SMB connectors going to the serial logic-level I/O, so I put circuit-board-mount SMB connectors on the motherboard for the serial connections. There are also low-pass RC filters for up to eight front panel switches.

The LPC2138/48 has two UARTs; one is used to communicate with NS3 and the other, via an RS-232 level converter, is connected to a 9-pin Type-D connector on the rear panel for communication with an external computer or terminal.

Some may question the use of a separate keypad encoder since, with microprocessors as controllers, polling a keypad is usually handled in software. I decided, however, that the additional cost of this IC was not significant compared to the work I would have to do to implement software polling. Another consideration was that a keypad encoder would use up just five lines of controller (four address lines plus one DATA_AVAILABLE) whereas software polling would need to accommodate the 8 pins of the keypad.

The Front Panel Human Interface

The design of the front panel is the most important aspect of the whole design. All too often, this is considered only briefly or superficially. I recall many years ago at work, I bought a very good quality, very expensive signal generator at a time when digitally controlled instruments were just becoming available. The frequency could be set in 1 Hz steps. It had a nice digital panel display with large bright numerals. To set the frequency, however, you first used a knob (a rotary encoder) to brighten specific digits; the brightened digit moved to the left or right as you rotated the knob. This allowed you to "select" a particular digit and, holding down a push-button switch you could change that value with the same rotary knob. This control method was used to set either the frequency or the attenuation level. On the face of it, this seemed to be a very neat and tidy way of saving some panel space but, in practice, it was unbelievably awkward to use! A common task like looking at and listening to the output of a receiver as you tuned the generator through the received passband was virtually impossible to do. I found that I preferred to use an old, 1950s vintage Measurements Corporation Model 80 signal generator, which had two knobs; one for tuning and the other for the attenuator. I had the Model 80 on my workbench along with an HP frequency counter. I never turned the Model 80 off, so the output, although not completely stable in frequency, was stable enough for most of the things I needed to do. The expensive signal generator was used rarely and only when the superior frequency stability was needed; it spent most of its working life sitting unused



(B)

on a shelf in the storeroom. The moral of the story is that convenience of the user is exceedingly, indeed overwhelmingly, important in making a useful instrument.

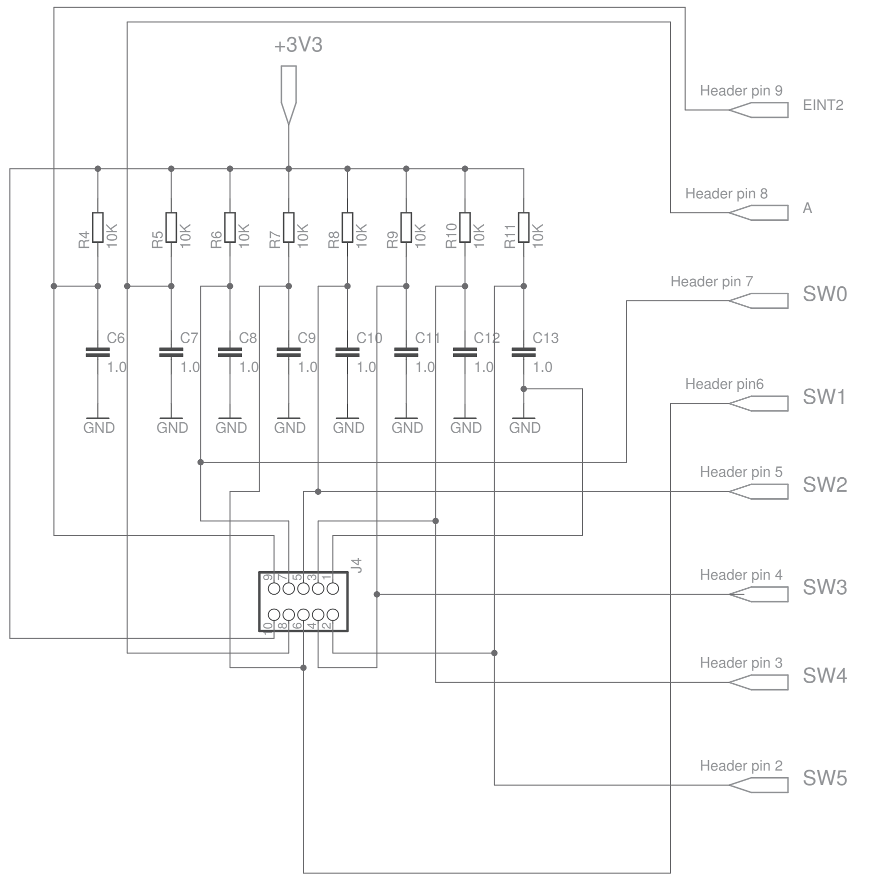
For this design, I decided to use a keypad to set the frequency. A close-up photo of the keypad is shown in Figure 6. To set a frequency such as 52050100 Hz, you press the number buttons; as each is pressed the accumulated number is shown in the center of the graphical display in large numbers. Then you press the ‘Hz’ button and then either “A” or “B” depending on which of the two NS3 outputs you want to change. To get 50 MHz from output B, you would just press “5” “0” “MH” and “B.” The remaining keypad button, marked “Sto” is used to store all the present settings of the signal generator; the frequencies, attenuator level, mode, and so on, into the internal EEPROM memory so that when you next turn on the generator, it will have all the same parameters it had when you pressed the “Sto” button.

The rotary knob goes to a simple non-optical rotary encoder, which is used to change the signal generator frequency in steps of a size that is set separately. The step size can be set to any frequency interval; the default is 100 Hz. Turning this rotary knob moves both frequencies in steps of 100 Hz. Having both frequencies step when the knob is rotated means that you could, for example, use output B for the local oscillator when testing a receiver using the attenuated A output as the signal source. The difference between these two outputs would be set to the intermediate frequency.

The rotary encoder has an additional switch, which is actuated by pushing the knob inwards towards the panel. I used this switch to set the attenuator. When the knob is pushed in, the display shows the present signal level and, by rotating the knob, you can increase or decrease this level in 1 dB steps. The displayed output level is shown in both dBm and in voltage.

There are two push-button switches marked “Menu” and “Set.” Pushing the Menu button moves you through a menu that is used to display various options. It, the “Set” switch and the keypad are used to change the mode of the signal generator; to change the modulation, scan limits, etc. I won’t go into a lot of detail about this except to say that it is important to think through how this is done before you implement it in software.

Most things one wants to consist of a number and an action. My own feeling is that the most intuitive way to do this is to enter the number first and then do the action. The example cited earlier about how to set the frequency illustrates this; first the frequency is entered via the keypad and the action consists of pressing either the “A” or “B” button.



(C)

Note: Header goes to front panel switches and rotary encoder

Pin 10 of header is +5V, pin 1 is GND

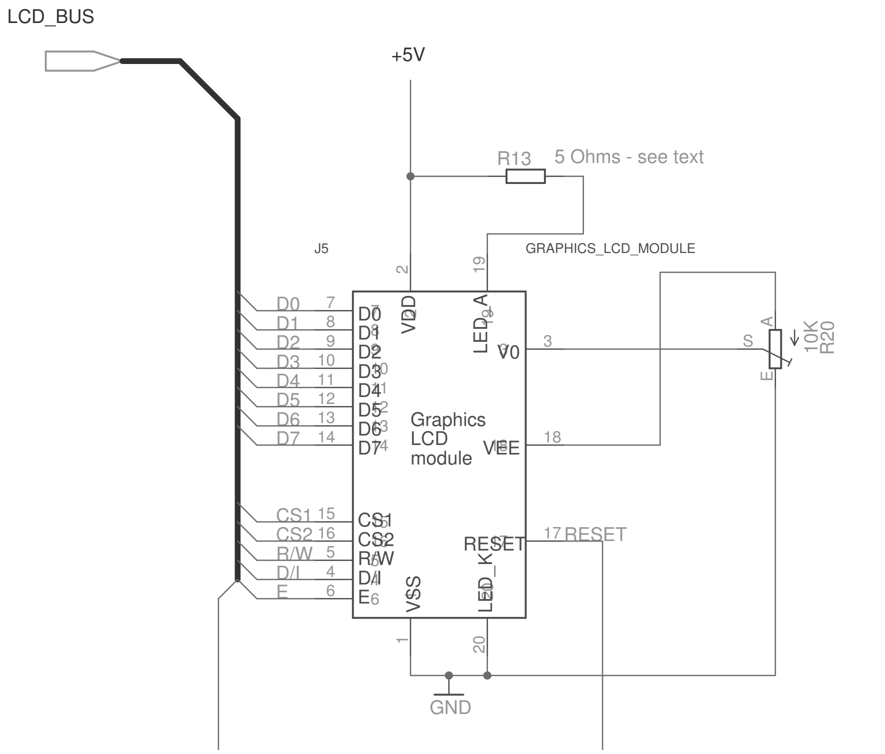
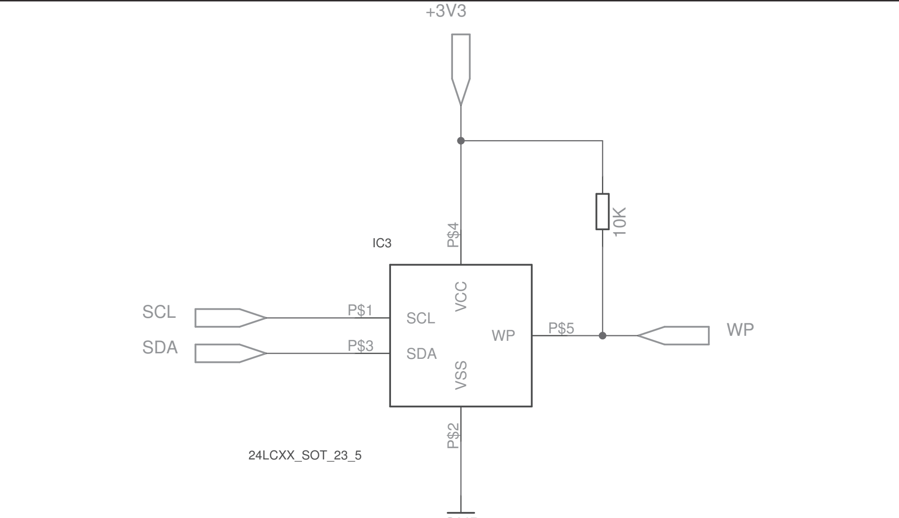
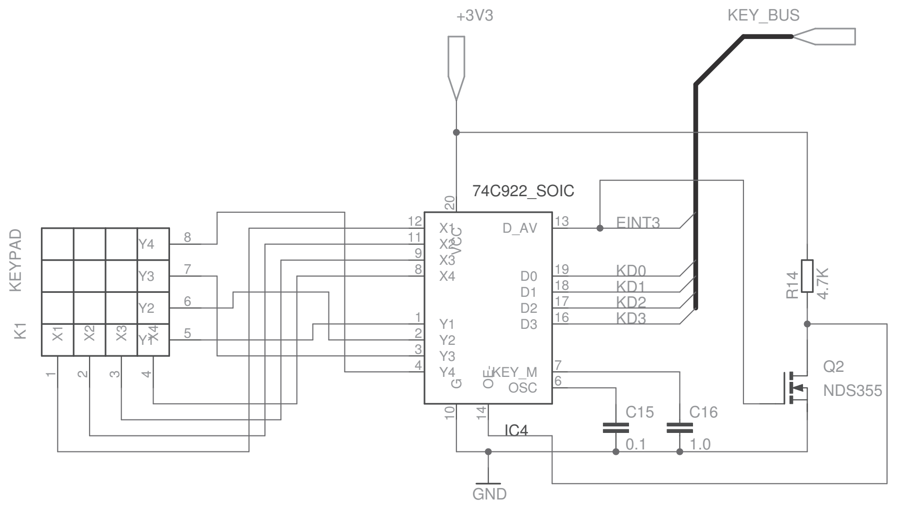


Figure 5 (continued)

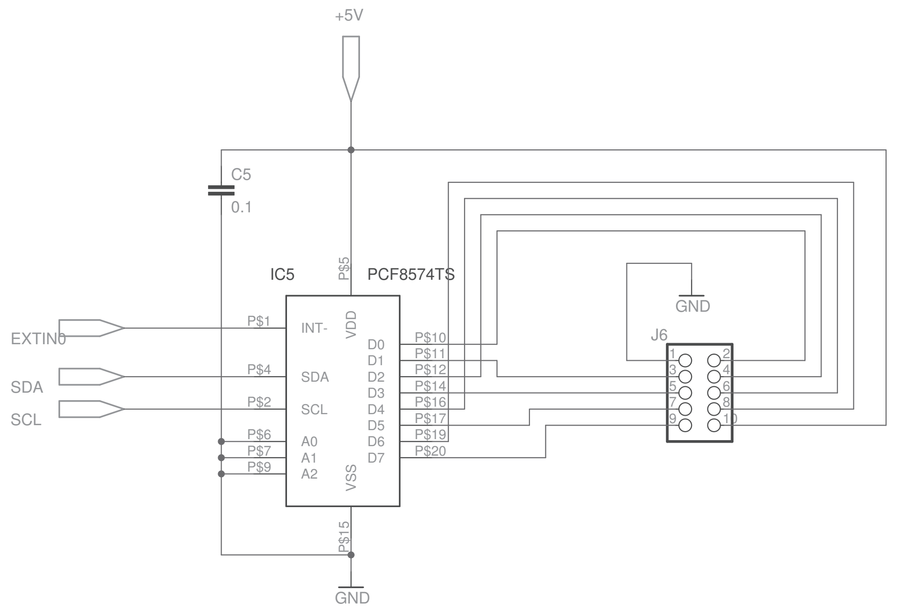
(D)



(E)



(F)



(G)

This may seem obvious but there is an alternative philosophy in which you first select the action you intend to do, then enter the number. For example, I could have used the menu button to select (and display) a number of intended actions such as "Set frequency A," then, when I got to the desired action, could have entered the number on the keypad. It doesn't matter which philosophy you choose, but it is important to be consistent in that philosophy for all the various instrument settings.

Operating Modes

There are five general operating modes; CW, AM, FM, Scan and Xtal. The first three are obvious. When turned on, the instrument always comes on in the CW mode, with the graphical display showing a running graph of the detector level. If you wish to change to any of the other modes, you do it using the Menu and Set buttons; if you choose AM, for example, you will be prompted to set the amplitude modulation percentage and frequency; the default values will be whatever was last stored (via the "Sto" button) in the EEPROM memory. FM is done similarly.

The "Scan" mode option prompts you to set a lower and upper frequency limit as well as a scan time. The "Xtal" mode is one I use to measure the parameters of crystals; I will discuss this briefly later in this article.

The Display

The graphical display uses a widely available 64 × 128 pixel module; a number of manufacturers make essentially identical modules. It has built-in LED back-lighting and a KS0108 controller. The KS0108 controller basically lets you turn individual pixels, specified by an address, on and off. The very hard work of writing code to display basic text and lines has been done by Martin Thomas.[6] I wrote a "higher level" set of procedures for displaying line graphs, text boxes, and so on, which use Martin's procedures. In my implementation, the data for the graphics is double buffered. There is a 1024 byte buffer (8 bytes × 128 columns) that contains an image of the actual display. There is also an identical input buffer that is written to by the graphical routines. A background interrupt routine, triggered by a timer every 40 ms compares the image buffer to the input buffer. If there is a difference in any byte, the image buffer is changed to make it identical to the input buffer, and at the same time this new change is written to the display itself. So, the graphical display is refreshed 25 times per second. In each refresh cycle, only those pixels in the display that need to be altered are done. This is the fastest way to update the graphical display; completely rewriting it every time actually takes longer due to the response of the KS0108 controller. Having the display double buffered means that in the program itself, you can just write to the input buffer without having to consider the timing because that is all done in the background in interrupt routines.

The Scan Procedure

The display has 128 pixels horizontally so every scan will have 128 frequency steps. The vertical display is in dB, which is a logarithmic scale so it is reasonable to have the horizontal frequency display logarithmic as well. This is easily done and the only parameters needed to describe a scan are the lower and upper frequencies and the time desired for the scan. The serial interface to NS3 is at 115200 baud so it takes roughly 87 µs to send a single 8 bit character with a one-bit start and stop. Since the "frequency" command to NS3 can require as many as 13 characters (including the final carriage return), one must budget 1.13 ms per command. This determines the theoretical lower limit to the sweep period for any scan; 144 ms.

The serial output from the controller is buffered and driven by interrupts. What this means is that, after the program seemingly sets the frequency, the serial output just starts and it will take another 1.13 ms to finish. Then, and only then, NS3 will set the frequency. So, to make sure NS3 has completed the frequency change and that the AD8307 output has settled, I set a fixed minimum delay of 2 ms after commanding the frequency change but before sampling the detector output level. Therefore, the minimum time taken to make a scan is 256 ms; about four scans per second. I find that this is tolerable. I have not included, in this timing, the additional time need to update the display, convert the A/D output into dB and then change the appropriate pixels. Because these computations just take a few microseconds, and the display update is done in the background while the serial output is being buffered out and while the timer is running to provide the delay, these operations do not have a significant effect on the scan rate.

NS3 has a built in detector with several options for how many individual samples to take. I wanted to have more control over the detection process and also to have a lower detection floor than that of NS3, so I opted to build a separate detector in a separate enclosure and to use the controller (one of its A/D inputs) to make a level measurement. The detector uses the ubiquitous AD8307 and I selected the output low-pass filter capacitor so that the time constant was about 100 µs.

One of the parameters that is set for the scan mode is a scan time. Obviously, this cannot be less than 256 ms. If it is larger than that, however, the time is divided by 128 and this time is added to the minimum 2 ms allocated between samples. My timer routine works in steps of 1 ms. (I could have used finer steps but didn't.) This results in a coarseness of the scan time setting of 128 ms. Thus, if you select a scan time of 500 ms, the closest available times will be 384 or 512 ms. In this case the program would select 512 as being the closest.

While the instrument is scanning, pressing the SET button at any time causes it to finish the scan it is doing, then send out, via the serial RS-232 port, a table of values of the frequencies and detected signal levels of that scan. After that is done, it then recommences scanning as before.

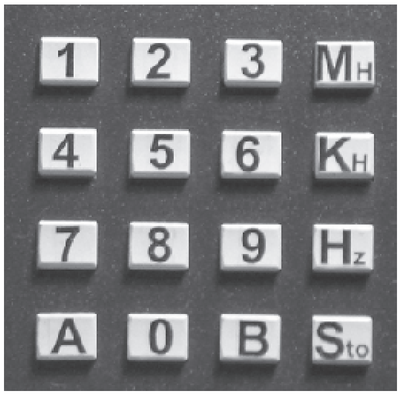


Figure 6 — Here is a close-up photo of the keypad used to control the signal generator.

The Frequency Calibration

Tom Alldread has made frequency calibration for NS3 very easy. He has built a routine into the firmware that allows you to accurately set the output frequency by comparison with some standard. The easiest standard to use is WWV, and the procedure is fully described in Part 3 of his article. (See Note 1.) In my case, having an oven-controlled oscillator as the clock source, I left the instrument running for an hour or so before doing this calibration.

The Attenuation and Detector Calibration

NS3 has commands that will set the output levels in steps of just 0.1 dB to as low as –20 dBm; that is 10 dB below the nominal –10 dBm output of the unit. These levels are very precise because it is all done digitally inside the AD9958 DDS IC. The attenuator I had, of unknown provenance, had steps of 10 dB. The problem was to calibrate the detector and to determine exactly what size the attenuator section attenuations were.

This can be done, conceptually, as follows (I will describe how I actually did essentially this but in a more efficient way later). With all attenuator sections out (no attenuation), the output levels of the detector are measured at –10 dBm and –20 dBm; those levels being set by NS3. Then with NS3 set to give an output of –10 dBm, the 10 dB attenuator section is switched in and the resulting detected

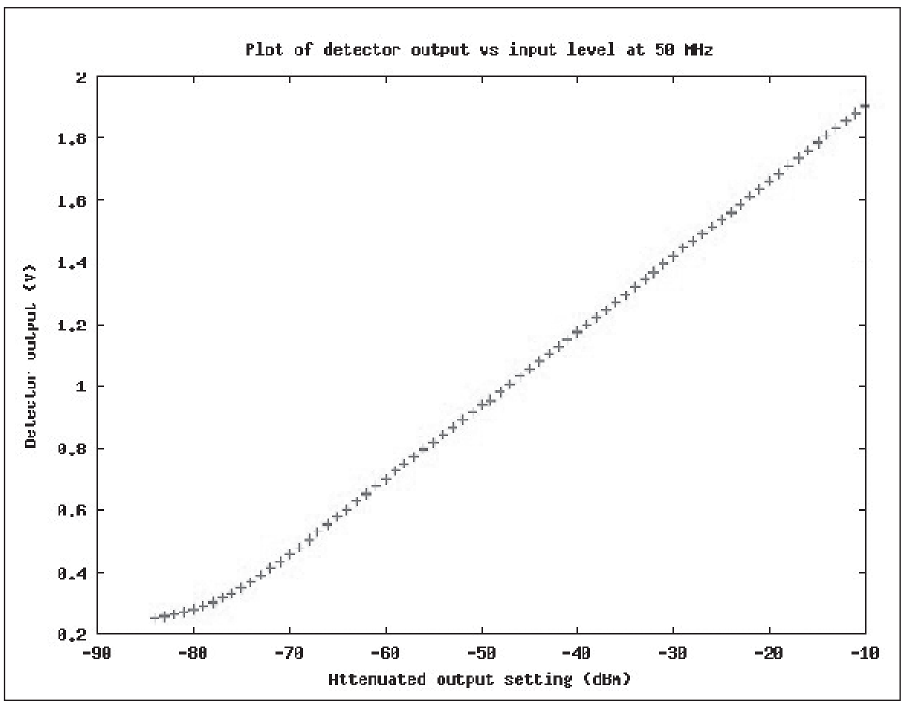


Figure 7 — This graph shows the plot of the detector output level (in volts) versus the output level set by the attenuator routine.

level compared to the level when NS3 was set to –20 dBm with this attenuator section switched out. Any difference can be attributed to the error in the attenuator section. Once you know that, you can repeat the process now with the 20 dB step of the attenuator, and so on and so on until all the sections have been calibrated. Once you know the true attenuations of all the attenuator section, you are then in a position to calibrate the detector. According to the AD8307 data sheets, the absolute calibration of the output is slightly frequency sensitive. The slope of the response is the same for all frequencies, however. So, measuring the response carefully at any one frequency (by using the now calibrated attenuator) in order to determine the slope, it is possible to make accurate relative level measurements by assuming the same slope applies to all frequencies.

To do the overall detector/attenuator calibration, I added a mode to the MENU specifically for this. The attenuator output is connected to the detector input and, when this mode is activated, the output level is changed, in 1 dB steps, from –10 dBm to -85 dBm and the mean A/D measured levels, averaged over 100 samples at each level, are sent out the serial port. I repeated this "scan" three times because there is some noise in the A/D determination. I captured this output in my main desktop computer, wrote it as a file and then plotted this response with a program called "gnuplot." Gnuplot is available for both *Windows* and *Linux* machines, and it is free![7] Figure 7 shows the plot of detected level (in volts) versus the output level set by my attenuator routine. This figure looks very similar to the plots shown in the AD8307 data sheet — as it ought to!

You can see that the straight line (in the logarithmic plot) response is pretty good down to about –72 dBm. The scatter, too small to see in this plot, is due partially to the granularity of the 10 bit A/D converter in the controller and partially to the ripple in the AD8307 response.

Figure 8 shows a truncated version (input levels greater than –72 dBm) of the same plot but with a least-square best-fit to a straight line superimposed on the plot. Gnuplot has the capability of doing this fit and produces the coefficients of the fit. That is, if we assume the detected power level, P, is related to the measured A/D level, v, then the two are related by an equation:

$$P = A + Bv \quad \text{[Eq 1]}$$

Gnuplot gives you the values for A and B, which give the best fit to this equation.

Looking at this plot, recall that the 1 dB steps from –10 dBm to –19 dBm are set by precise internal levels within NS3. The step from –19 dBm to –20 dBm is one in which the 10 dB attenuator section is switched in and the NS3 output is set back to –10 dBm. If there was an error in this 10 dB attenuator step, we would expect to see a discontinuity in the plot of points going from –19 to –20 dBm. Similarly, the step from –29 to –30 dBm is where the 10 dB attenuator section is switched out and the 20 dB attenuator section is switched in. Any discontinuity in this step would be attributed to the difference in errors of the 10 dB section and the 20 dB section. My calibration routine sets output levels from –10 dBm to –85 dBm. This means that the 10, 20 and the first of the 40 dB sections are being switched in and out over this range.

Figure 8 is too coarse to show any observable discontinuities but, having fitted the line,

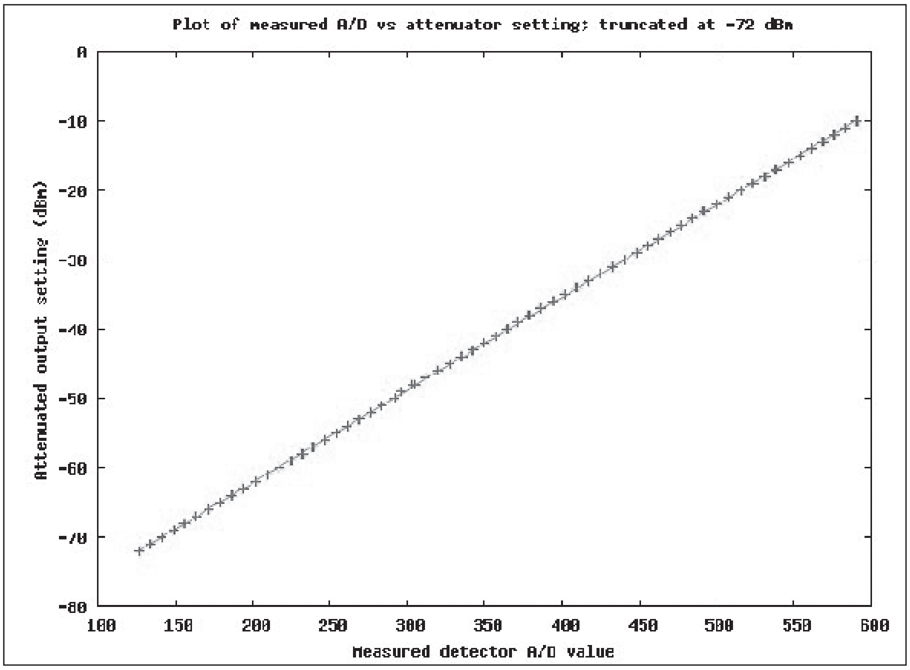


Figure 8 — This graph shows a truncated version (input levels greater than –72 dBm) of the Figure 7 plot, but with a least-square best-fit to a straight line superimposed on the plot.

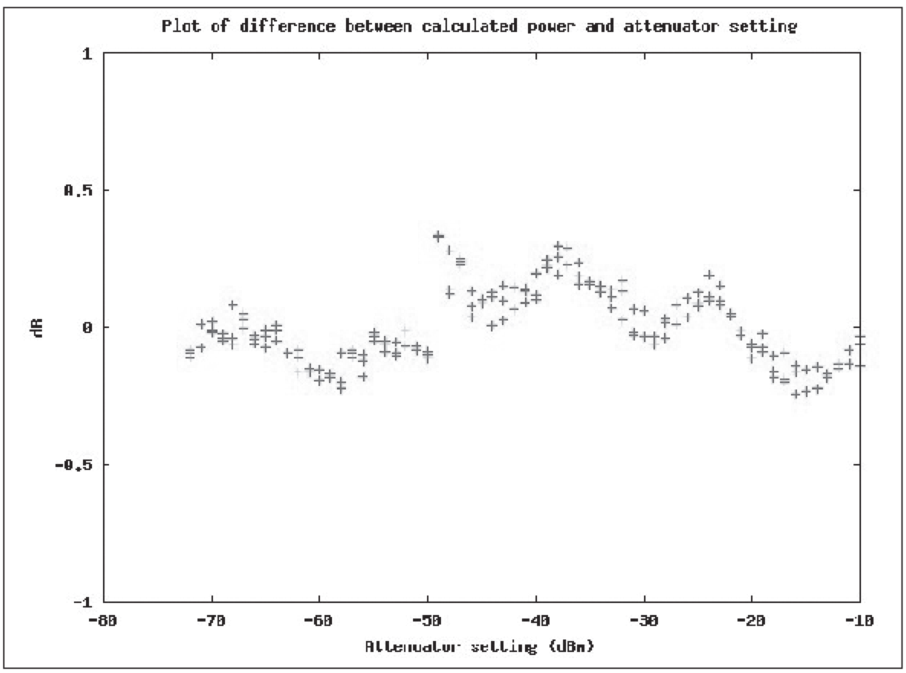


Figure 9 — This graph is a plot of the difference between the calculated output power and the attenuator setting. Several discontinuities stand out on this plot. The most obvious is at the –49/–50 dBm attenuator setting.

one can then plot the differences between the best fit line and the actual data. This is shown in Figure 9. As you can see, there are some discontinuities at the –19/–20, –29/–30, –39/–40, –49/–50 regions of the graph; the largest of these is at the –49/–50 step. The region from –20 to –30 also looks a bit high; that is when the 10 dB section is being switched in. If I assume the 10 dB section is 10.0 dB, the 20 dB section is 20.2 dB and the 40 dB section is 40.0 dB and go through the same procedure again (finding the best fit and then plotting the differences), I get Figure 10. Here, the most obvious cause of the differences is just the unavoidable ripple in the AD8307 output. I could probably make some more minor corrections to the attenuator levels but it likely isn't worth it. The observed scatter of ±0.2 dB is a measure of the combined accuracy of the output attenuator setting and the accuracy of the detector.

I was lucky to have an attenuator as accurate as this one. With a more inaccurate attenuator, the correction procedure would be longer and more tedious but, nevertheless, still possible. The overall accuracy of the attenuator calibration is based upon the accuracy of the 0 to 9 dB steps set within NS3.

In my program, to measure a power level, I make a number of A/D conversions of the detector output and average them to reduce the effect of noise. Let this number be v. Then, the detected power level, P, in dBm is given by Equation 1.

Before leaving the topic of the attenuator and detector, I should mention that I did not measure the attenuation of the other two 40 dB sections. At –60 dBm, for example, the output level is only about 224 µV. The next 40 dB attenuator section would only be used to get to –80 dBm, which is about 22 µV. Once you get down to microvolt levels or sub-microvolt levels, the leakage around the attenuator can make determining the exact output voltage, to an accuracy of just a few percent, very difficult and so the exact calibration of these other two 40 dB sections really isn't that important. Also, the observed accuracy of the 10, 20 and first 40 dB sections made me think that the other two 40 dB sections would likely be accurate enough, Even an error as large as 1 dB at the –100 dBm level only corresponds to a change in output of about 0.2 µV.

The Crystal Parameter Measurement Mode

I have previously written a *QEX* article discussing the methods used to measure crystal parameters.[8] In that article, I described, in some detail, an algorithm that can be used to measure crystal parameters in a fixture in which they are connected in shunt with a resistance.

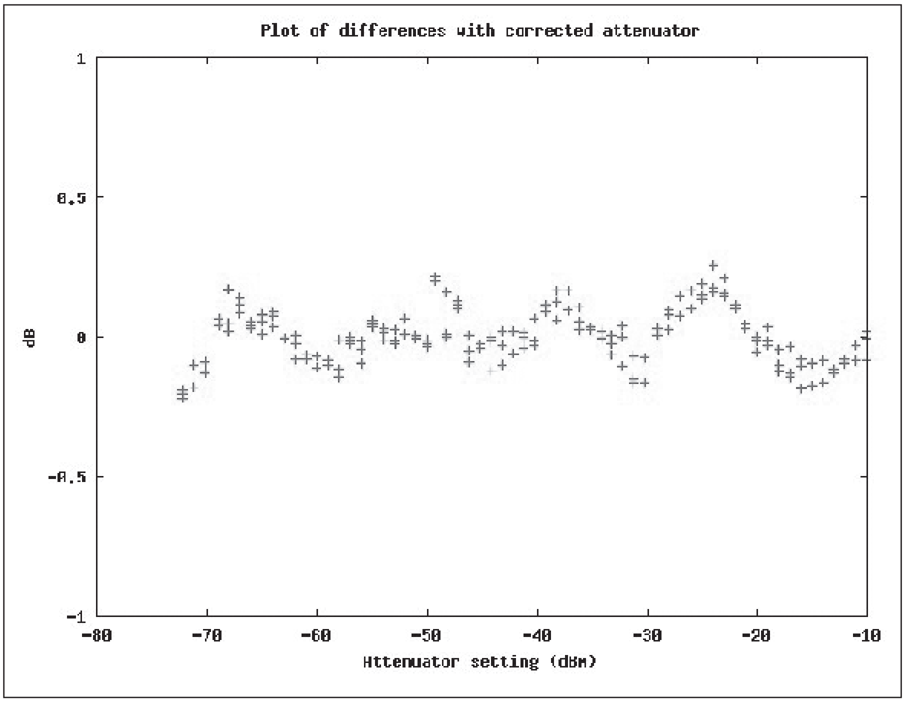


Figure 10 — After applying the best-fit algorithm to the detector output versus attenuator set levels, correcting for the true attenuation of different steps of the attenuator and plotting the difference, this graph results.

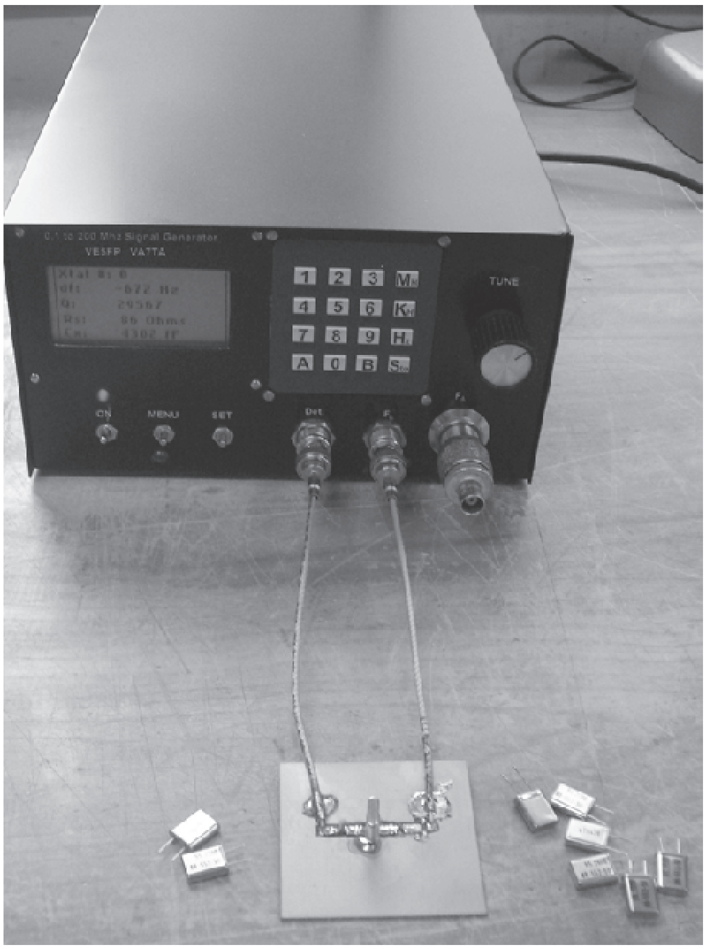


Figure 11 — Here, the signal generator is being used to measure crystal parameters.

Alan Bloom, N1AL, subsequently wrote a Letter to the Editor of *QEX*, pointing out that the frequency of minimum impedance is not the same as the self-resonant frequency of the crystal but is changed by the parallel capacitance of the crystal and its stray capacitance as it is mounted in a circuit. This change is not trivial, as it might be several hundred hertz for an overtone crystal in the VHF region, and so needs to be taken into account.

The difficulty with determining the amount of this correction is that it requires a fairly precise measurement of the total capacitance in parallel with the series-resonant crystal circuit. Typically the crystal parallel self-capacitance is something like 5 pF and the measurement fixture may add a similar amount for a total in the neighborhood of 10 pF. Figure 11 shows my test setup. Ideally, we would like to know this to an accuracy of 0.1 pF in order to determine the frequency correction with an accuracy of about 1%.

This subject is too complex to go into here and is not really relevant to the signal generator alone but I have been doing a bit of modeling of crystal responses in the shunt mode and comparing the modeled response with the measured response of real crystals. I think I have discovered a technique for determining the correction with sufficient accuracy and I hope to write another article about it in the future.

Summary

This signal generator is a laboratory quality instrument that I can set to any frequency between 100 kHz and 200 MHz to an accuracy of 1 Hz. The frequency stability is determined by an oven-controlled crystal oscillator and is probably on the order of one part in ten million. Crystal oscillators age and so it will probably need to be recalibrated once a year or so.

The power detector is accurate to about ±0.2 dB over a range from –10 dBm to –70 dBm. It was calibrated at a single frequency but, for relative power measurements at any given frequency, it should retain the same relative accuracy.

The generator can scan in frequency over any arbitrary range (within the 0.1 to 200 MHz range of the instrument) and display the output on a log-log plot of signal level versus frequency.

It is a really useful instrument being well-suited for measuring filter responses, for tuning filters, for aligning receivers, and so on. It is also suitable for making more critical measurements such as those needed to determine crystal parameters and to accurately measure the response of crystal filters.

Jim Koehler was first licensed as VE5AL at age 15 in 1952. He attended university in Canada and Australia, and became a professor of physics and engineering physics at the University of Saskatchewan, retiring in 1996. He and his wife moved to Vancouver Island after retirement. He dabbles in photography, electronic design and generally enjoys the wonderful scenery and relaxed lifestyle.

Notes

[1]Thomas M. Alldread, VA7TA, "NimbleSig III — Part 1," *QEX*, Jan/Feb 2009, pp 3-20; Part 2 of this article appeared in the Mar/Apr 2009 issue of *QEX* (pp 16-25) and Part 3 appeared in the May/Jun 2009 issue of *QEX* (pp10-20).

[2]Download the "Crossworks for the ARM" package at **www.rowley.co.uk**.

[3]See **www.w6pql.com/vhf_ocxo.htm**.

[4]Mark Wilson, K1RO, Editor, *The ARRL Handbook*, 2010 edition, ARRL, 2009, ISBN: 0-87259-144-1; ARRL Publication Order No. 1441, $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**.

[5]See **www.olimex.com**.

[6]Martin Thomas has written a complete package for the LCD2xxx processor and a graphical display using the KS0108 controller. See **www.siwawi.arubi.uni-kl.de/avr_projects/arm_projects/#lpc_glcd_dcf**.

[7]See **www.gnuplot.info/**.

[8]Jim Koehler, VE5FP, "Some Thoughts on Crystal Parameter Measurements," *QEX*, Jul/Aug 2008, pp 36-41.

[9]Alan Bloom, N1AL, Letters to the Editor, *QEX*, Sep/Oct 2008, 41-42.

QEX

UNITED STATES POSTAL SERVICE

Statement of Ownership, Management, and Circulation (All Periodicals Publications Except Requester Publications)

1. Publication Title: QEX
2. Publication Number: 0 8 8 6 – 8 0 9 3
3. Filing Date: October 1, 2009
4. Issue Frequency: Bi-monthly: Jan/Mar/May/July/Sept/Nov
5. Number of Issues Published Annually: 6
6. Annual Subscription Price: $24.00
7. Complete Mailing Address of Known Office of Publication (Not printer) (Street, city, county, state, and ZIP+4®): 225 Main Street, Newington, Hartford County, CT 06111-1494
Contact Person: Amy Hurtado
Telephone (Include area code): 860-594-0257
8. Complete Mailing Address of Headquarters or General Business Office of Publisher (Not printer): 225 Main Street, Newington, Ct 06111-1494
9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor (Do not leave blank)
Publisher (Name and complete mailing address): Harold Kramer, 225 Main Street, Newington, CT 06111-1494
Editor (Name and complete mailing address): Larry Wolfgang, 225 Main Street, Newington, CT 06111-1494
Managing Editor (Name and complete mailing address): Larry Wolfgang, 225 Main Street, Newington, CT 06111-1494
10. Owner (Do not leave blank. If the publication is owned by a corporation, give the name and address of the corporation immediately followed by the names and addresses of all stockholders owning or holding 1 percent or more of the total amount of stock. If not owned by a corporation, give the names and addresses of the individual owners. If owned by a partnership or other unincorporated firm, give its name and address as well as those of each individual owner. If the publication is published by a nonprofit organization, give its name and address.)

Full Name	Complete Mailing Address
American Radio Relay League, Inc.	225 Main St. Newington, CT 06111-1494

11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities. If none, check box → ☒ None

Full Name	Complete Mailing Address

12. Tax Status (For completion by nonprofit organizations authorized to mail at nonprofit rates) (Check one)
The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes:
☒ Has Not Changed During Preceding 12 Months
☐ Has Changed During Preceding 12 Months (Publisher must submit explanation of change with this statement)

PS Form 3526, September 2007 (Page 1 of 3 (Instructions Page 3)) PSN 7530-01-000-9931 PRIVACY NOTICE: See our privacy policy on www.usps.com

13. Publication Title: QEX
14. Issue Date for Circulation Data Below: Sept/Oct08-July/Aug09 Sept/Oct 09

15. Extent and Nature of Circulation		Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue Published Nearest to Filing Date
a. Total Number of Copies (Net press run)		8003	7900
b. Paid Circulation (By Mail and Outside the Mail)	(1) Mailed Outside-County Paid Subscriptions Stated on PS Form 3541 (Include paid distribution above nominal rate, advertiser's proof copies, and exchange copies)	4990	5529
	(2) Mailed In-County Paid Subscriptions Stated on PS Form 3541 (Include paid distribution above nominal rate, advertiser's proof copies, and exchange copies)	0	0
	(3) Paid Distribution Outside the Mails Including Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Paid Distribution Outside USPS®	1464	1480
	(4) Paid Distribution by Other Classes of Mail Through the USPS (e.g. First-Class Mail®)	516	481
c. Total Paid Distribution (Sum of 15b (1), (2), (3), and (4))		6970	7490
d. Free or Nominal Rate Distribution (By Mail and Outside the Mail)	(1) Free or Nominal Rate Outside-County Copies included on PS Form 3541	76	70
	(2) Free or Nominal Rate In-County Copies Included on PS Form 3541	0	0
	(3) Free or Nominal Rate Copies Mailed at Other Classes Through the USPS (e.g. First-Class Mail)	31	8
	(4) Free or Nominal Rate Distribution Outside the Mail (Carriers or other means)	93	93
e. Total Free or Nominal Rate Distribution (Sum of 15d (1), (2), (3) and (4))		200	179
f. Total Distribution (Sum of 15c and 15e)		7170	7669
g. Copies not Distributed (See Instructions to Publishers #4 (page #3))		839	231
h. Total (Sum of 15f and g)		8008	7900
i. Percent Paid (15c divided by 15f times 100)		97.21%	97.67%

16. Publication of Statement of Ownership
☐ If the publication is a general publication, publication of this statement is required. Will be printed in the Nov/Dec 09 issue of this publication.
☐ Publication not required.

17. Signature and Title of Editor, Publisher, Business Manager, or Owner: Publisher/CEO
Date: October 1, 2009

I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanctions (including fines and imprisonment) and/or civil sanctions (including civil penalties).

PS Form 3526, September 2007 (Page 2 of 3)

Horst Steder, DJ6EV
Forstbachstrasse 13, 35625 Huettenberg, Germany;
h-g.steder@freenet.de

Jack A. Hardcastle, G3JIR
8 Norwood Grove, Rainford, St. Helens,
Great Britain WA11 8AT; g3jir@liv.ac.uk

Crystal Ladder Filters for All

The authors share their comprehensive crystal filter design program.

Crystal ladder filters have been used in home construction projects for a very long time and radio amateurs have developed ever more sophisticated design strategies for their production. Almost any group of crystals, of the same nominal frequency, when connected into a ladder network will produce a band pass frequency response, but designing a really useful filter is more demanding. The purpose of this article is to show you how to produce filters with a more predictable performance using a new, freely available software package. The method is based on the equations of Milton Dishal, and is fully self-contained; there are no complicated calculations, approximations, or filter tables to look up.[1] The program takes over all the many complex procedures involved, and so enables the constructor to concentrate fully on the selection of the filter components and its construction. A ZIP file with the program, help files and a brief explanation is available for download from the ARRL *QEX* Web site.[2] After downloading the ZIP file, extract the files to a separate directory on your hard drive. Do not create that directory in the "Program Files" or "Documents and Settings" directories on your hard drive.

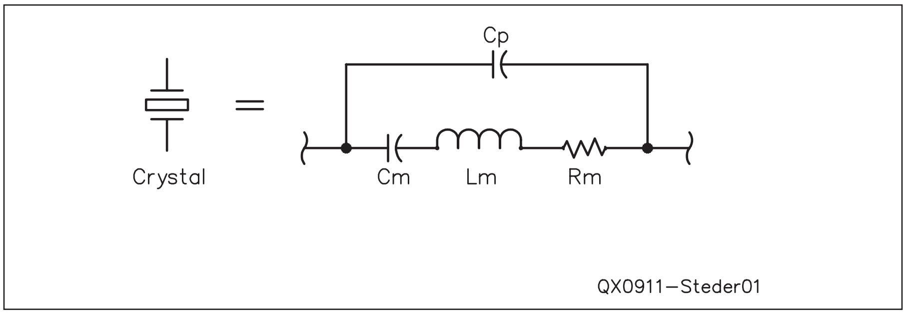


Figure 1 — The mechanical properties of a quartz crystal can be represented by the electrical equivalent components shown above.

First Steps

Before starting any design process you need to know the properties of the materials you are going to use, and crystals are no exception. In fact, they are much more complicated than more familiar components such as capacitors or inductors, or indeed than L/C tuned circuits.

Crystal Parameters

The equivalent circuit of a crystal, shown in Figure 1, consists of the motional inductance, Lm, and the motional capacitance, Cm, in series with the loss resistance, Rm; which all combine to produce a series resonance. Together with Cp they produce a parallel resonance as well. Because of this double resonance, a crystal connected in the series circuit of Figure 2 will produce a sharply peaked response that is followed at a slightly higher frequency, by a very sharp dip.

Therefore, a ladder filter with crystals as the series elements will always show a steeper slope on the high frequency side. This makes it ideal for making lower-sideband SSB filters. By assembling a filter with a sufficient number of crystals, however, the slope on the low frequency side can be made steep enough to use the filter for upper-sideband operation as well. This can be seen in Figure 3. The Dishal program is also a very convenient tool for determining the necessary number of crystals for a desired specification.

Measuring Cp

This is the capacitance of the crystal's electrodes, together with the unavoidable associated circuit strays. It can easily be measured at low frequency by using a bridge or a component test set. It has a very low value, of the order of 3 to 6 pF, so it is necessary to arrange to make this measurement when the crystal is mounted in a similar position to its final assembly in the filter, in order to produce a realistic value for use by the program.

Determining Lm and Cm

Lm and Cm cannot be measured directly, so they have to be obtained by alternative means. For people with limited test facilities the classical method of doing this, which requires precise measurements of phase shift (see Note 6), may be replaced by one of the alternatives listed in the Notes.

The first of these methods, which was suggested by G3UUR, tests the crystal in a Colpitts oscillator.[7] The result of that measurement is evaluated by the routine "Xtal/G3UUR," located in the "Xtal" drop-down menu. This then provides values for Lm and Cm.

The second, and easiest of these alternative methods, was described by Wes Hayward W7ZOI (see Note 7) and also by K8IQY.[13] Here, the crystal is treated as a very high-Q tuned circuit and by measuring the –3 dB bandwidth and the loss resistance, Rm, we can calculate the motional inductance. This method has the advantage of not needing accurately calibrated capacitors, but it does need an accurate 3 dB attenuator and a finely-tuned, stable signal generator. This

[1]Notes appear on page 18.

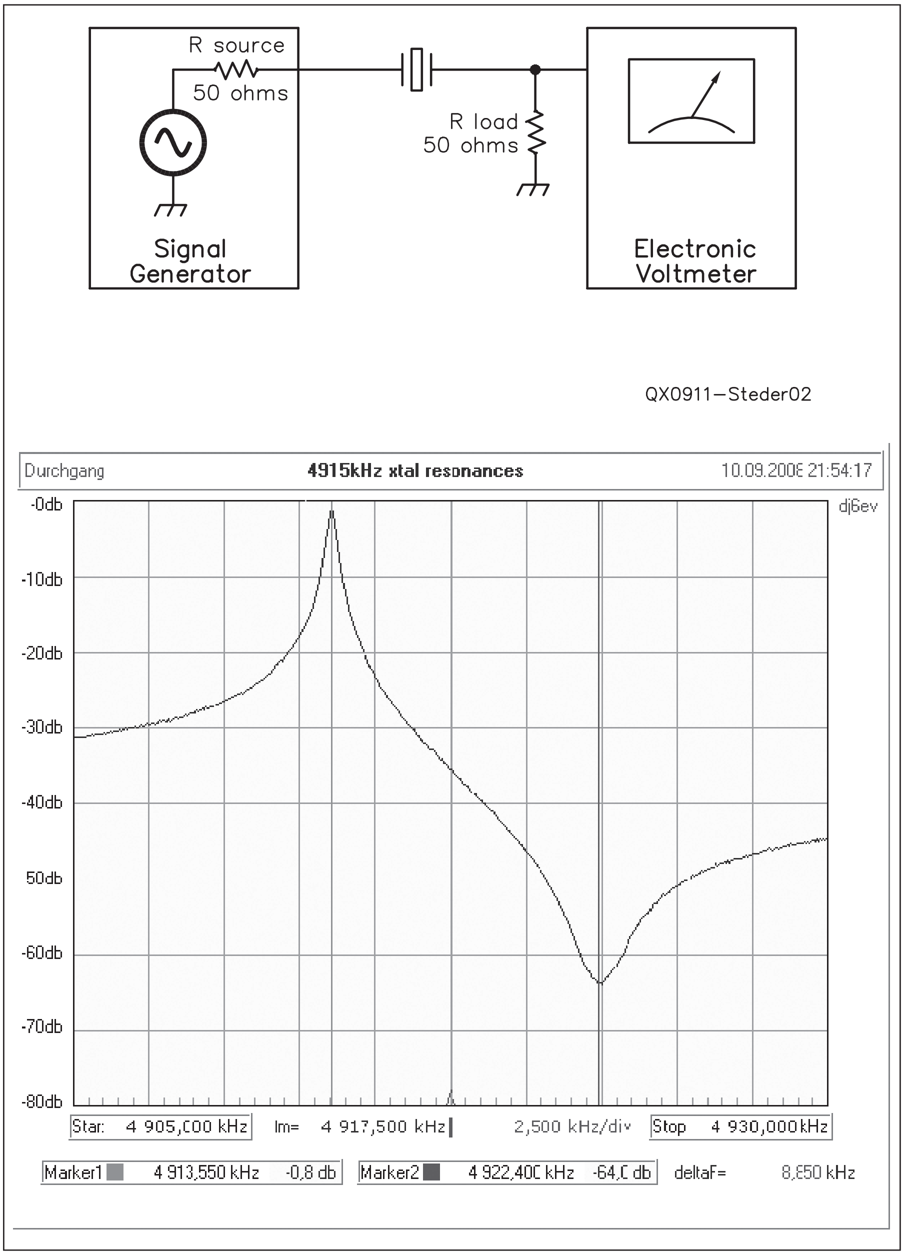


Figure 2 — Part A shows the 50 Ω test circuit to show the properties of a crystal. Part B shows the frequency response of crystals, showing series and parallel resonance.

may easily be home-brewed by using one crystal from your batch of crystals in a variable crystal oscillator circuit (see Note 13). Again, the calculation routine is located in the "Xtal" menu as "3 dB-Method."

Whichever measurement method you decide to use, you need to check the complete batch, making a note of Fs, Cm and Rm for each crystal, as well as Cp. All of these will be needed later as input to the computer program. It is particularly important to take notice of Rm, because it will allow you to weed out any low-Q (high Rm) samples. Also, at this stage it is worth selecting the highest and lowest frequency crystals for use as LSB and USB carrier crystals.

A Typical Filter

For an SSB filter, we suggest that you use crystals within the range 4 to 11 MHz for a first attempt. These are cheap and readily available because they are produced in large quantities for computer and communications applications.

It goes without saying that care needs to be taken in the selection of crystals. One cannot assume that crystals with equal frequencies and the same case style will show the same motional inductance. This can vary significantly from manufacturer to manufacturer and can even vary from batch to batch. It is therefore mandatory not only to measure the frequency but also the motional inductance or capacitance of each crystal. As an example of this variation see Tables 1 and 2. Table 1 is a group of crystals purchased in Germany and Table 2 refers to the crystals supplied with an Elecraft K2 kit.

Table 1

Measured Parameters for Crystals Purchased in Germany

Fs = 4913.57 kHz
Lm = 69.7 mH
Cm = 15.053 fF
Cp = 3.66 pF

Note**:** For the benefit of anyone unfamiliar with the unit fF is the abbreviation for femtofarads and 1 fF is equal to one thousandth of a picofarad.

Table 2

Measured Parameters for Crystals Supplied with an Elecraft K2 Kit

Fs = 4913.57 kHz
Lm = 76 mH
Cm = 13.805 fF
Cp = 3.65 pF

The Design Program

A very detailed description of the facilities provided within the program is given in the "Help" file so only the major features will be described here. This article gives sufficient information to get you going. The initial input display is illustrated in Figure 3.

The program assumes identical crystal parameters. Therefore the standard procedure is to select crystals from the list of measured items that show the best match for fs, Lm (or Cm) and Cp, calculate their average values and insert them into the program.

The first parameter to be entered is Cm, the motional capacitance, in femtofarads (or Lm, the motional inductance in millihenrys). Then enter Fs and Cp, which were measured earlier. The 3 dB bandwidth is your own choice and depends on whether you are making an SSB or a CW filter. In our case the value used was 2.4 kHz. This allows for a little bandwidth "shrinkage" due to the finite crystal Q. Too large a figure here will result in the program giving an error message informing you of the maximum achievable bandwidth. We chose to design a 0.2 dB ripple Chebyshev filter using 8 crystals. The final entry is the frequency span to be displayed. In this example it is 10 kHz, but larger and smaller bandwidths can be handled. Then all you need to do is to press the "Calculate" button and all the component values required are presented on the left-hand panel, together

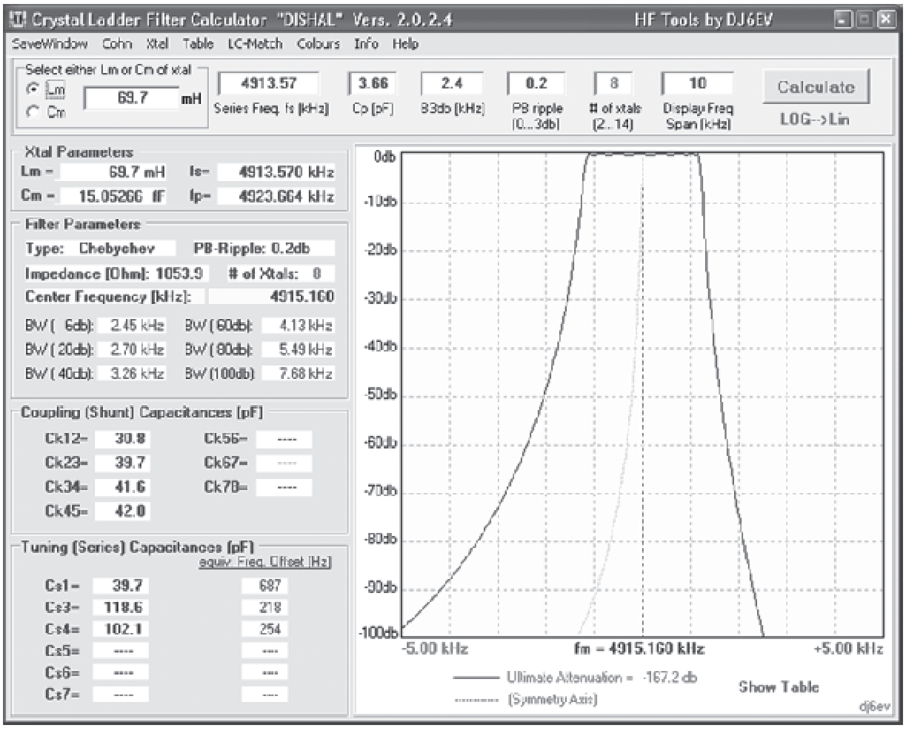


Figure 3 — Filter design program user interface.

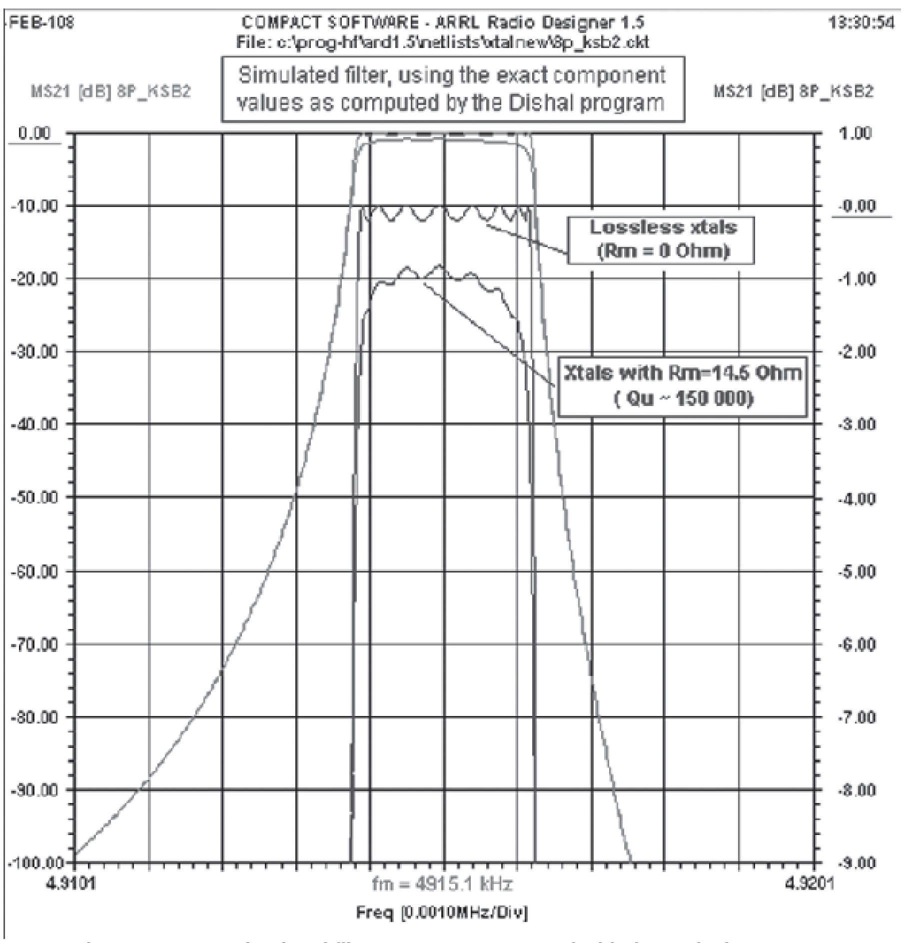


Figure 4 — ARD simulated filter response compared with theoretical response.

with a graph of the frequency response on the right.

The top tool bar has a drop-down menu "Table," which allows this graph to be replaced by a table of insertion loss. This enables you to make an easy comparison between the theoretical response and your subsequent measurement of your own filter.

As a guide to what you may expect to measure, the parameters of the crystals used in the Figure 3 filter were as listed in Table 1. We suggest you try out the program by entering these values into the program yourself and press "Calculate."

Amateur Radio Designer

In order to confirm the validity of the program and its underlying algorithms, a number of filters have been constructed. Also, they have been simulated using the *ARRL Radio Designer* software formerly marketed by ARRL. A typical simulation response is shown in Figure 4 and of particular note is the ability of *ARD* to simulate the effect of losses within the filter caused by the finite Q of the crystals. Note how the edges of the passband become more rounded and the ripple is no longer of equal amplitude across the passband.

Some Design Refinements

Ideally, the crystals used in this type of filter should all be absolutely identical but unless you have a large batch from which to select, you will have to use the closest you can to this ideal. An acceptable tolerance would be approximately ±2% of the design bandwidth. For an SSB filter, this is around ±50 Hz but only ±10 Hz for a CW filter. Don't despair, however, there is a special feature in the program to help you to compensate for these deviations from the ideal. This is in the bottom left-hand corner of the results panel, and displays the frequency shift produced by placing the calculated value of capacitance in series with each crystal. If you look at the initial measurements you made on the crystals, you may be fortunate enough to find crystals that already deviate from the ideal by a similar amount. If so, they can be used in the filter without any additional series capacitor at all. Otherwise you need to calculate the size of the capacitor required to pull it onto the target frequency. There is a facility for "Xtal tuning" included under the "Xtal" drop-down menu, which calculates the size of this capacitor for you. A very detailed description of the use of this facility is given in the "Help" file.

Because the filter components are dependent upon your choice of such details as bandwidth and passband ripple you will find that making small changes to these can result in capacitor values that are sufficiently close to standard preferred values not to need further trimming. Even if you are not that fortunate, you should usually be able to make the required capacitor using no more than two capacitors in parallel.

The filter impedance also depends upon your choice of input parameters, but this can be matched over a wide range by adding an extra inductance and capacitance at both ends of the filter. Again, this calculation is incorporated into the program, maintaining our aim of providing you with a complete design package. To this extent it is very educational to make changes to the input values, just to see what effect they have on the calculated components. The program will tell you when you exceed the limits of practical designs.

The measured insertion loss will always be higher than is shown in the design response, which assumes perfectly lossless components. This is mainly due to the Rm of the crystals, but the capacitors also add a small contribution. Similarly, additional insertion loss is incurred if impedance matching networks or transformers are introduced. Binocular transformers like the BN2402-43 or similar cores usually introduce less loss than toroidal cores because of their much better coupling coefficients.

It is very important that the inductive reactance of these transformers is at least ten times higher than the required filter impedance, otherwise the filter will show distortion of the passband and increased ripple because of its sensitivity to reactive loads. Similarly, the termination impedances should not differ from the calculated value by more than –5 to +10%. Otherwise, this will cause the passband shape and ripple to deviate significantly from the desired format. It has been noted that narrow band CW filters are more tolerant of these departures from the ideal than are wider bandwidth filters.

Further Refinements

If you insert a value of zero as the crystal's shunt capacitance in the program, it will calculate component values for the type of filter described in Note 7, where the crystals have been shunted by inductors that resonate with Cp. This type of filter has a frequency response that is inherently symmetrical in both stop bands, and for some applications such as spectrum analyzers, it may justify the additional complexity. It is educational to observe the effect that this neutralization of Cp has on the value of the coupling capacitors.

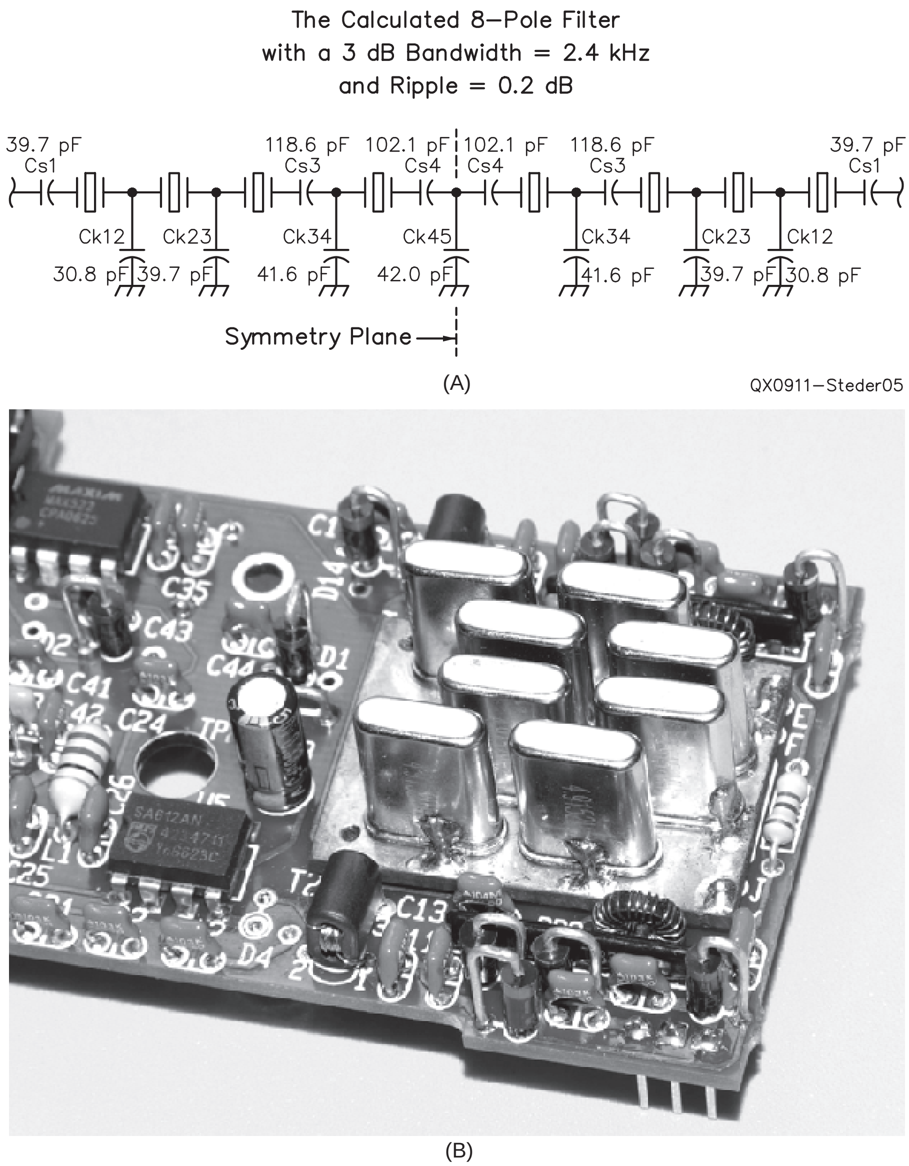


Figure 5—An 8-pole Chebyshev filter circuit is shown at A. Part B shows the filter installed in an Elecraft K2 transceiver.

Filter Design and Choice of Filter Frequencies

We recommend that the design value for the passband ripple be kept fairly low, on the order of 0.1 to 0.5 dB. This is because, as detailed earlier, the real filter will show an increased ripple due to the inevitable component tolerances and losses. Filters with higher ripple do, however, provide better shape factors, so it may be a price worth paying, particularly if you are not prepared to select more accurate components.

Generally the choice of intermediate frequency is decided by other factors in the overall system design, but the limitations of crystal filter design must also be taken into account at that stage. Some of these considerations are summarized below.

For SSB filters in particular, lower frequencies such as 4915 kHz used in the popular K2 transceiver have the advantage that the crystal Qu (unloaded Q) is normally greater than that of higher frequency crystals. Also, the filter band edges are much closer to the calculated values because the relative bandwidth b3/fm is higher for lower frequency IF values. Furthermore, higher frequency filters show more rounding of the passband edges and have a higher insertion loss, all due to the lower Q and the lower relative bandwidth, b3/fm. Unfortunately, lower frequency filters have a more asymmetrical stopband response and you need to use a higher-order filter to obtain the required selectivity. Life is never easy for the filter designer! All these conflicting factors are easily demonstrated by the program, and it is worth experimenting just for its educational value, even if you never intend to make a filter.

It is perhaps an obvious point, but it is worth stressing that the choice of an IF that is within or near to a broadcast band will make it almost impossible to avoid breakthrough of the powerful broadcast signals.

Construction Hints

One of the most important requirements is the careful and accurate measurement of *all* components, not only the crystals themselves. The capacitors should not only be measured accurately for their value but should also be tested for their HF quality, especially when using small ceramic capacitors. Painstaking care pays off by producing very good filters.

If SMD (surface mount) capacitors are used in the filter, they will exhibit an additional capacitance to ground unless the ground-plane is removed from below them. These values can only be estimated, and will reduce the effective bandwidth of the filter if they are not deducted from the calculated values. This is less critical when the coupling capacitances exceed values of 100 pF (for instance in CW filters).

Ladder filters are friendly, and if the passband curve down to around –10 dB fits the design expectations, the stopband response will also follow the computed values. The stopband rejection is heavily influenced by the degree of input-to-output isolation, and careful physical layout and screening of the filter components is mandatory if the potential full performance is to be realized.

Based on experience, the crystal cases should be grounded using the shortest possible connections to minimize stray inductance. Double-sided circuit boards are best, with as many through connections as possible for the ground planes. All leads in the layout should be as short and wide as possible to keep stray inductance and resistance very low. Similarly, all the shunt components should be grounded directly. Failure to do so results in additional coupling, which the program is not designed to take into account, and will produce unpredictable passband distortion.

It should be noted that typical crystal cases, such as HC49-U, often have small "bumps" where the leads enter the can. To ensure rigid installation, the crystals should sit comfortably on the ground plane of the circuit board, and this will be facilitated if appropriate holes (about 0.5 mm deep) are carefully drilled into the surface of the board. Otherwise vibration will eventually cause the leads to fatigue and fracture at this point.

Figure 5A shows the circuit diagram of the filter designed in Figure 3. Figure 5B shows it installed in an Elecraft K2 transceiver.

And Finally

The computer program that has been described here provides a design tool suitable for use by designers with little or no prior experience in filter design. By making this program freely available we hope more people will be tempted into the highly enjoyable home-brew facet of amateur radio. You can find the latest program version and any updates on the Warrington Amateur Radio Club (WARC) Web site.[14]

Jim Koehler presented an automated method of measuring crystal parameters in a Jul/Aug 2008 *QEX* article.[15] A much greater number of excellent articles have been published in *QST* and *QEX*, and their exclusion is not to be regarded as a value judgment. It is just that we have chosen those which complement our article by describing some features in greater detail than we could fit in here. In fact, a much extended list of crystal filter references is included on the Warrington Amateur Radio Club (WARC) Web site (see Note 14).

Notes

1 Milton Dishal, "Modern Network Theory Design of Single-Sideband Crystal Ladder Filters," *Proceedings of the IEEE*, Sep 1965.

2 The crystal ladder filter design program is available for download from the ARRL QEX Web site. Go to **www.arrl.org/qexfiles** and look for the file **11x09_Steder-Hardcastle.zip**.

3 J. A. Hardcastle, G3JIR, "Some Experiments with High-Frequency Ladder Crystal Filters," *Radio Communication*, RSGB, Dec 1976, pp 896-898, Jan 1977, pp 28-29, Feb 1977, pp 122-124, Sept 1977, 687-688. A summary of this material was published by ARRL in the Dec 1978 issue of *QST*, pp 22-24.

4 J. A. Hardcastle, G3JIR, "Ladder Crystal Filter Design," *Radio Communication*, RSGB, Feb 1979, pp 116-120. This article was reprinted by ARRL in the Nov 1980 issue of *QST*, pp 20-23.

5 J. A. Hardcastle, G3JIR, "Computer-Aided Ladder Crystal Filter Design," *Radio Communication*, RSGB, May 1983, pp 414-420.

6 J. A. Hardcastle, G3JIR, "Quartz Crystal Parameter Measurement," *QEX*, Jan/Feb 2002, pp 7-11.

7 Wes Hayward, W7ZOI," Refinements in Crystal Ladder Filter Design," *QEX*, June 1995, pp 16-21. This article was reprinted in *QRP Power*, ARRL Publication No. 210.

8 Doug DeMaw, W1FB, "A Tester for Crystal F, Q and R," *QST*, Jan 1990, pp 21-23 (with Feedback in Mar 1990 *QST*).

9 Wes Hayward, W7ZOI, "Designing and Building Simple Crystal Filters," *QST*, July 1987, pp 24-29.

10 Wes Hayward, W7ZOI, "A Unified Approach to the Design of Crystal Ladder Filters," *QST*, May 1982, 21-27 (with Feedback in July 1987 *QST*).

10A Also see "An Oscillator Scheme for Quartz Crystal Characterization," **www.w7zoa.com**.

11 Randy Evans, KJ6PO, "Crystal Parameter Measurement and Ladder Crystal-Filter Design," *QEX*, Sept/Oct 2003, pp 38-43.

12 Jack Smith, K8ZOA, "Designing the Z90's Gaussian Crystal Filter." *QEX*, May/June 2007, pp16-26.

12B Also refer to **www.cliftonlaboratories.com**.

13 Crystal filter test equipment: **www.k8iqy.com**.

14 Warrington Amateur Radio Club: **www.warc.org.uk**.

15 Jim Koehler, VE5FP, "Some Thoughts on Crystal Parameter Measurement," *QEX*, July/Aug 2008, pp 36-41.

Note: The articles from Notes 4, 5 and 6 are available on the Warrington Amateur Radio Club Web site at **www.warc.org.uk**. The articles from Notes 8, 9 and 10 were reprinted in *W1FB's Design Notebook*, ARRL publication No. 126, ISBN:0-87259-

Next Issue in *QEX*

Thomas Dean, KB1JIJ, describes "A Digital FHSS Transmitter" that he designed as a project for a digital logic class at the United States Military Academy. Using an Altera DE2 Development and Education Board with a Cyclone II field programmable gate array (FPGA), Cadet Dean created a software defined frequency hopping spread spectrum (FHSS) transmitter. The article presents a demonstration of the kind of system that can be contained on a single IC, and describes how it was created.

Rudy Severns, N6LF

PO Box 589, Cottage Grove, OR 97424; n6lf@arrl.net

Experimental Determination of Ground System Performance for HF Verticals Part 6 Ground Systems for Multiband Verticals

How much will the signal strength and feed point impedance change as radials are added?

The first five parts of this series have focused on ground systems for a single-band vertical (mostly on 40 m).[1, 2, 3, 4, 5] This part of the series will address multiband radial systems, and give us an opportunity to see if the performance equivalence shown earlier between a large number of radials lying on the ground and a few elevated radials will hold with a multiband radial system.

The experiments were performed in two phases. The first was for radials lying on the ground and the second was for elevated radials. These represent two typical scenarios for amateurs: in other words. "Do I put the antenna in the back yard or up on the roof?" These are quite different arrangements, so the discussion is divided into two parts, beginning with the radials lying on the ground surface and then moving on to elevated radials.

The Test Antenna

For this series of tests I used a SteppIR III vertical antenna. The SteppIR has the advantage that its height can be adjusted to be resonant anywhere between 40 m and 6 m. The height adjustment is motorized and controlled remotely, so it is very convenient for tests on multiple bands.

[1]Notes appear on page 24.

Test Frequencies

Most of the measurements were taken at spot frequencies of 7.2, 14.2, 21.2 or 28.5 MHz. I did make a limited number of measurements across each band, however, and some of those results will be discussed.

Radial System Configurations

For these experiments I made up four sets of thirty two ¼ λ radials, one set for each band (40, 20, 15 and 10 m). The radial lengths are given in Table 1 along with the corresponding free space ¼ λ. As is the usual practice, the radials are a few percent shorter than the free space ¼ λ. The radials were fabricated from AWG no. 18 stranded, insulated wire.

Table 1
Description of Radial Lengths

Frequency (MHz)	*Free-Space ¼ λ (Feet) / (Inches)*	*Radial Length (Feet)*
7.2	34.2 / 410	33
14.2	17.3 / 208	16.8
21.2	11.6 / 139	11.3
28.5	8.63 / 104	8.4

Table 2
Total Length of Wire in Each Configuration.

Configuration	*C1*	*C2 and C8*	*C3*	*C4*	*C5*	*C6*	*C7*
Total Wire (ft)	2240	280	560	1056	528	264	132

Table 3
Transmission Gain (S21) in dB for Each Configuration Relative to C2 (0 dB).

Frequency (MHz)	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*
7.2	+0.9	0.0	+0.2	+0.9	+0.4	+0.1	–3.2
14.2	+0.8	0.0	+0.3	+1.0	+0.5	–0.6	–1.8
21.2	+0.3	0.0	+0.3	+0.8	+0.2	–1.1	–2.6
28.5	–0.6	0.0	0.0	+0.4	–0.5	–1.3	–3.8

During the experiments I used several different configurations:

C1) Sets of 32 single-band radials, one set at a time. In this way I had an optimized ¼ λ vertical over a ground system of thirty two ¼ λ radials on each band. These antennas were then measured individually on the appropriate single band.

C2) Four ¼ λ radials on each band (16 total radials), connected *all at the same time*.

C3) A repeat of C2 except using eight radials for each band (32 total radials) .

C4) Thirty two 33 foot radials.

C5) Sixteen 33 foot radials.

C6) Eight 33 foot radials.

C7) Four 33 foot radials.

C8) For some elevated radial tests, I used four ¼ λ radials on each band, *one set of radials at a time*. The set of four was chosen for the particular band.

C1 and C8 were used for comparison purposes in that they represent a monoband antenna on each band. Obviously with a multiband antenna you would not run out to the antenna and change the radials whenever you changed bands! But this can give us feeling for any compromise in going from monoband to multiband verticals.

C2 represents the most common multiband ground system in general use both for elevated and ground surface radial systems, and so it was an obvious choice. I could have chosen many other possible combinations but those I did choose are at least reasonable. In particular I wanted to show that a few long radials (C6 and C7) don't work very well whether on the ground or elevated. Table 2 shows the total length of wire in each configuration.

Radials Lying on the Ground

The experimental results for radials lying on the ground are shown in Tables 3, 4 and 5. In Table 3 the values for S21 are in dB *relative* to the measured S21 value for C2 (0 dB). This was done to make it easier to compare each configuration to the de facto standard (C2).

The results for C7 show the same problem when used with a multiband vertical as shown earlier for a single band vertical — the ground loss is very high. Increasing the radial number from 4 to 32 (from C7 to C4) shows improvement.

C2 is our "standard" ground system (at least in practice) and we can see that its performance in comparison to the other configurations is quite good. It is true that individual sets of 32 radials on each band (C1) are somewhat better (except on 10 m, for which I have no explanation!) but the compromise is less than 1 dB. Even though C2 has only four radials cut for 40 m, the other twelve shorter radials seem to take up most of the slack, and we do not see the very poor performance that four radials by themselves displayed. By doubling the number of radials in C2 to eight for each band (C3), we see some improvement over C2, although it's only a fraction of a dB.

The best performer is C4, which is 0.4 to 1 dB better than C2, depending upon the band. C4, however, requires almost four times as much wire. If we cut the amount of wire in half (C5) we still have some improvement over C2 (with the exception of 10 m). C3 and C5, which use approximately the same amount of wire, behave very similarly.

In the final analysis it appears that the standard ground system (C2) works just

Figure 1 — Here is a view of the vertical with elevated radials.

Table 4
Physical Height of the Vertical for Each Frequency and Ground System Configuration.

Configuration Frequency (MHz) (Inches)	*Free Space ¼ λ (Inches)*	*C1 (Inches)*	*C2 (Inches)*	*C3 (Inches)*	*C4 (Inches)*	*C5 (Inches)*	*C6 (Inches)*	*C7*
7.2	410	391	406	394	391	386	371	369
14.2	208	201	202	201	198	199	200	201
21.2	139	137	137	137	137	137	137	138
28.5	104	103	102	102	102	102	103	104

Table 5
Measured Feed Point Impedances With the Vertical Height Adjusted for Resonance at the Test Frequency.

Configuration Frequency (MHz)	*C1 (Ohms)*	*C2 (Ohms)*	*C3 (Ohms)*	*C4 (Ohms)*	*C5 (Ohms)*	*C6 (Ohms)*	*C7 (Ohms)*
7.2	40.0	54.4	51.7	40.0	43.5	56.3	92.4
14.2	35.1	50.0	44.5	42.7	51.2	62.4	85.8
21.2	36.0	40.5	38.4	42.0	48.9	66.3	102.9
28.5	34.4	48.2	39.3	43.8	51.6	67.8	105.6

fine. You can add more wire and get some improvement but whether that improvement is worthwhile depends on the user.

As shown in Tables 4 and 5, there is some interaction between the tuning or resonant height of the vertical and the individual ground system configurations. We've seen this effect in earlier experiments. The heights shown are a bit of an approximation. The control unit display for the SteppIR gives the length of the tape (the vertical conductor) above a certain point but between that point and the actual ground radial plate there is approximately another 12 inches of wire. The wire is bent within the base housing so you can't assign an accurate additional length. I have used 12 inches as a reasonable approximation.

The measured feed point impedances are given in Table 5.

Elevated Radials

Having four sets of 32 radials (one set on each band) on hand from the ground surface tests I decided to use these same radials for the elevated radial tests. With the exception of C1and C3, I used the same configurations (C2, C4-C7) for the elevated tests. In the elevated radial testing, I used C8 in place of C1. Like C1, C8 is not practical, being a series of monoband verticals, but it serves as a reference against which to judge the compromise from using a multiband radial system. For comparisons between elevated and ground surface radials I have added a column (C1) to Tables 7 and 9 for the *on-the-ground* data associated with C1. We will use these when we discuss elevated versus ground radials.

A photograph of the experimental arrangement for the elevated radial tests is shown in Figure 1.

Because of the need for easy access to the radial base plate to make the many changes in radial configuration, I had to place the base of the antenna only 6 feet above ground.

Six feet high for the base is a bit low if we want to improve the feed point match by sloping the radials downward. In free space the input impedance of a 4-radial ground-plane antenna is about 22 Ω. As we bring the antenna closer to the ground, the impedance will vary around this number but in general is well below 50 Ω. Often the SWR will be high. One common means to improve the match is to slope the radials downward from the base, which raises the feed point impedance and lowers the SWR. Because of the limited height at the center, I could only lower the outer ends of the radials a small amount. Keep this in mind when we look at the measured impedances and SWR plots.

Experimental results are given in Tables 6, 7 and 8. A few of the columns have blanks. These are cases where that configuration, on that band, performed so poorly as to be unac-

Table 6
Transmission Gain (S21) in dB for Each Configuration Relative to C2 (0 dB).

Frequency (MHz)	*C2 (dB)*	*C4 (dB)*	*C5 (dB)*	*C6 (dB)*	*C7 (dB)*	*C8 (dB)*
7.2	0.0	–-0.1	–0.2	–0.2	0.0	0.0
14.2	0.0	+0.2	–0.8	–4.0	—	+0.2
21.2	0.0	+0.4	+0.2	+0.2	—	+0.4
28.5	0.0	+1.1	+1.8	+0.7	—	+0.2

Table 7
Physical Height of the Vertical for Each Frequency and Configuration.

Configuration Frequency (MHz)	*Free Space ¼ λ (Inches)*	*C1 (Inches)*	*C2 (Inches)*	*C4 (Inches)*	*C5 (Inches)*	*C6 (Inches)*	*C7 (Inches)*	*C8 (Inches)*
7.2	410	391	403	397	397	400	403	403
14.2	208	201	208	190	180	150	—	208
21.2	139	137	143	142	143	145	—	142
28.5	104	103	104	100	97	88	—	104

Table 8
Measured Feed Point Impedances with the Vertical Height Adjusted for Resonance.

Configuration Frequency (MHz)	*C2 (Ohms)*	*C4 (Ohms)*	*C5 (Ohms)*	*C6 (Ohms)*	*C7 (Ohms)*	*C8 (Ohms)*
7.2	43.4	42	41.0	42.1	43.0	43.0
14.2	34.2	38.9	41.1	83.9	—	33.9
21.2	36.8	52.3	49.5	48.4	—	31.4
28.5	23.9	34.8	38.3	73.2	—	24.5

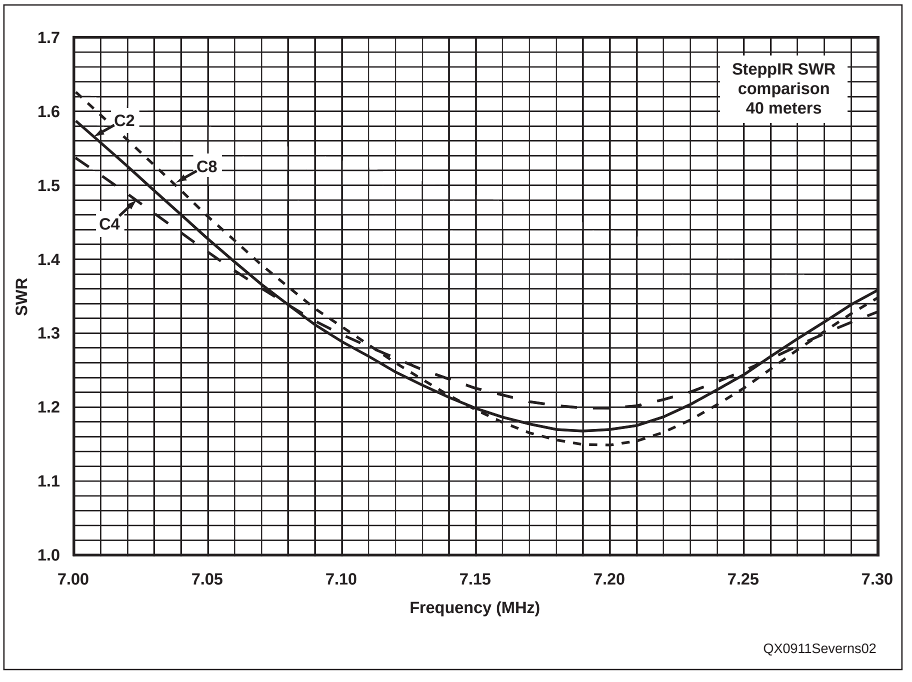


Figure 2 — Feed point SWR comparison on 40 m.

ceptable and I didn’t see any point in recording that information.

From the data in Table 6, the standard multiband radial system (C2) appears to work very nearly as well as C4 and it takes only a quarter as much wire! The only band on which C4 appears to have a significant advantage is 10 m. C2 is also very close to C8 so there is very little compromise from the monoband case. As we move to fewer long radials (C5-C7) we see there is an immediate problem on 20 m, where the gain starts to fall quickly. From Table 7 we see that on 20 m the resonant height of the vertical starts to change radically as we go to fewer long radials, so clearly there is some funny business going on. This is related to the fact that ¼ λ radials on 40 m are close to ½ λ long on 20 m. Except on 20 m, C5 and C6 seem to be okay on 15 and 10 m, but by the time we get to C7 (four 33 foot radials) the performance was so poor I haven’t even entered the data. The four long radials don’t even work well on 15 m, where they are close to ¾ λ long.

From a loss point of view there appears to be little advantage to using anything other than the standard four radials cut for each band (C2). There is, however, the question if there is any matching (SWR) improvement from using more wire — for example C4 instead of C2. Figures 2 through 5 show a comparison of the feed point SWR between C2, C4 and C8 on the four bands.

On 40 m the differences are insignificant. On the higher bands we see little difference between C2 and C8. C2 is behaving pretty much as we would expect. However, C4 does seem to offer some improvement above 40 m. It is especially noticeable on 15 m, where the 32 radials are all near ¾ λ resonance. From some of my earlier work I was not surprised that increasing the number of radials beyond four did not give much improvement in S21, but I was expecting to see much flatter SWR curves. This just doesn’t seem to happen on 40 m but does appear on 15 m with ¾ λ radials.

We should keep in mind that the feed point impedances and associated SWR will be affected by the height above ground, which in this case is very low. For well

Modifying the Ground Radial Connections on the SteppIR

Before conducting the experiments, I modified the ground radial connection on the standard SteppIR and also made up a special feed line choke that would have an impedance greater than 1000 Ω on all bands.

As the SteppIR comes from the manufacturer, it has a single no. 12 brass machine screw to which the ground radials can be attached. I felt this was not adequate and certainly not very convenient for the many radial changes necessary during the experiments. I changed the single brass no. 12 screw to a pair of ¼-20 machine screws spaced about 6 inches apart, as shown in Figure 1A.

I then fabricated an aluminum disk with fifteen ¼-20 bolts with wing-nuts around its perimeter. The disk was attached to the base of the SteppIR housing as shown in Figure 2A.

For all the measurements in the experiments, but particularly for the elevated radial measurements, I wanted to have a common mode choke (balun) in the feed line and the cabling at the base of the antenna. The choke I used is shown in Figure 3A. The choke has 6 turns of RG8X coaxial cable wound on two stacked type 43 cores (Fair-Rite #2643803802, available from Mouser Electronics). Also shown in the picture is the probe from the HP4815A vector impedance meter used for impedance measurements. The measured shunt impedance was between 2 and 3 kΩ from 7 through 30 MHz.

Figure 4A shows both the chokes installed at the base of the antenna.

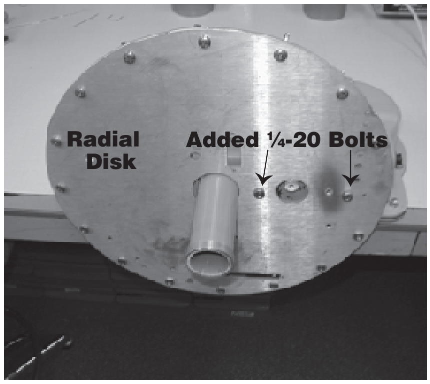


Figure 2A — SteppIR with radial disk attached.

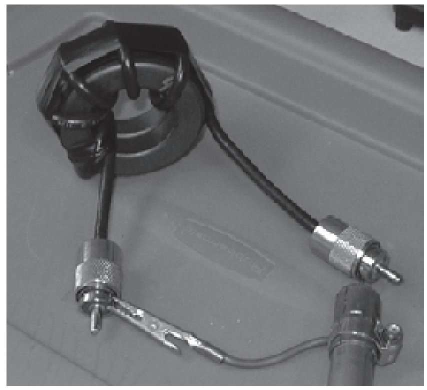


Figure 3A — Common mode choke for the feed line.

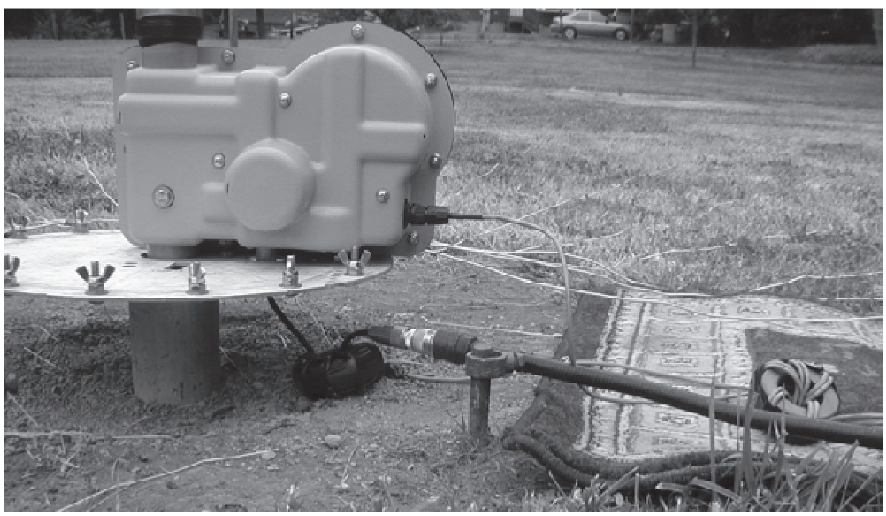


Figure 4A — Chokes installed in the feed line and control cables at the base of the antenna.

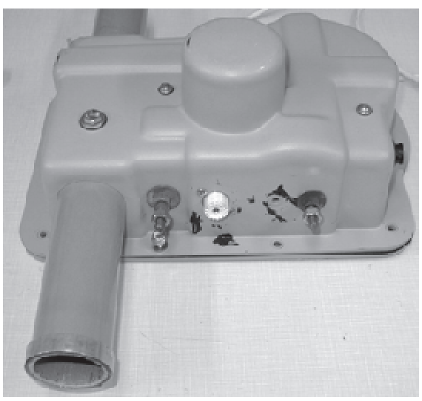


Figure 1A — Modified radial attachment scheme for the SteppIR.

elevated radials, where the slope can be adjusted to provide a better match, the results may be much better than shown here.

Elevated Versus Ground Radials

Another key question is "How do the elevated radial systems compare to a large number of radials on the ground on each band?" Table 9 makes that comparison using the results from this series of experiments. C1 is used as the reference (0.0 dB).

C1 uses radials lying on the ground surface and C2, C4 and C8 are elevated. When we compare the signals for C1 to those for C8, which is a direct comparison between four elevated radials against 32 ground surface radials, one band at a time, we see only small differences: four elevated radials seem to perform much the same as large numbers of ground surface radials. This is in keeping with what we saw in Part 3, only now extended to bands from 40 through 10 meters.

C4 (which is thirty two elevated 33 foot radials) is also only marginally different from C1 and C2 except on 10 m, where the difference is 1.4 dB. Considering it has four times the wire, I doubt it's justified.

Some Final Comments

In summary, I don't see any compelling reason to use more than four radials on each band for a multiband vertical. The "standard" system (C2) does in fact seem to work well. If you want to lay out or hang up more wire, you can get some small improvement but generally the maximum improvement seems to be on the order of 1 dB or less, although the improvement might be somewhat higher over poorer soil than mine. In a way, this was a bit of a disappointment. It would have been nice to discover some magic new ground system for multiband verticals, but that was not to be. All I've really accomplished is to show that the old standard works just fine, and it appears that a few elevated radials can work as well as a large number of on-the-ground radials! Be careful, however! As I pointed out earlier in the series, elevated monoband radial systems with only a few radials are very susceptible to local effects that can cause unequal radial currents, which can degrade performance.

Keep in mind when comparing the data in this part with some of the data reported in earlier parts of this series, that this set of measurements were made in mid-summer when the temperature had been 85° and 108° F over the preceding month. The soil will have dried out considerably compared to that for most of the earlier experiments. This can cause the impedance and S21 measurements to vary substantially between seemingly identical experiments. This is why I emphasized in Part 1 the need to do all com-

Table 9
Transmission Gain (S21) in dB for Each Configuration Relative to C1 (0 dB).

Frequency (MHz)	*C1 (dB)*	*C2 (dB)*	*C4 (dB)*	*C8 (dB)*
7.2	0.0	+0.2	+0.1	+0.1
14.2	0.0	+0.1	+0.3	+0.3
21.2	0.0	–0.5	+0.4	–0.1
28.5	0.0	–0.3	+1.1	–0.1

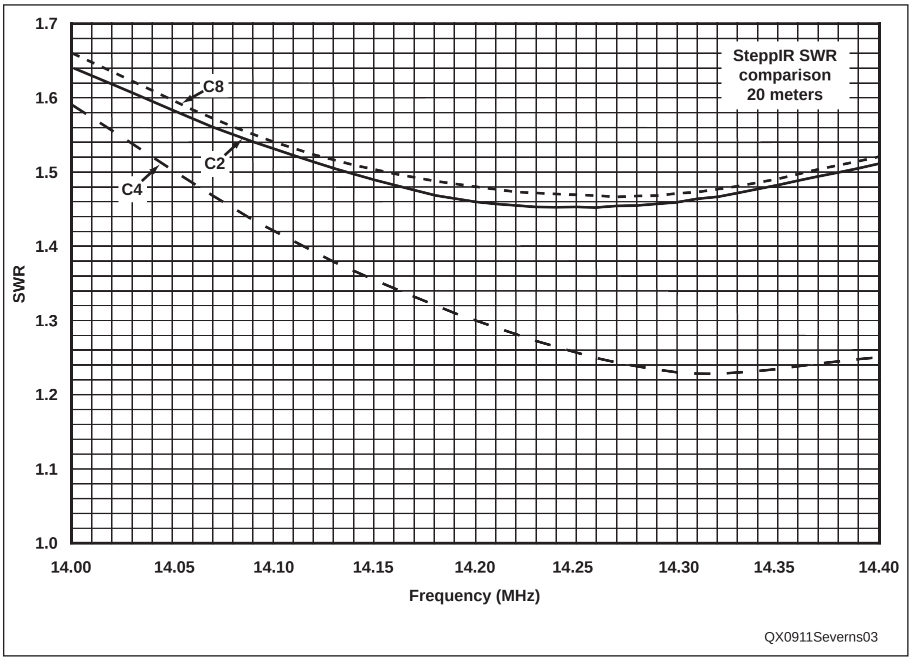


Figure 3 — Feed point SWR comparison on 20 m.

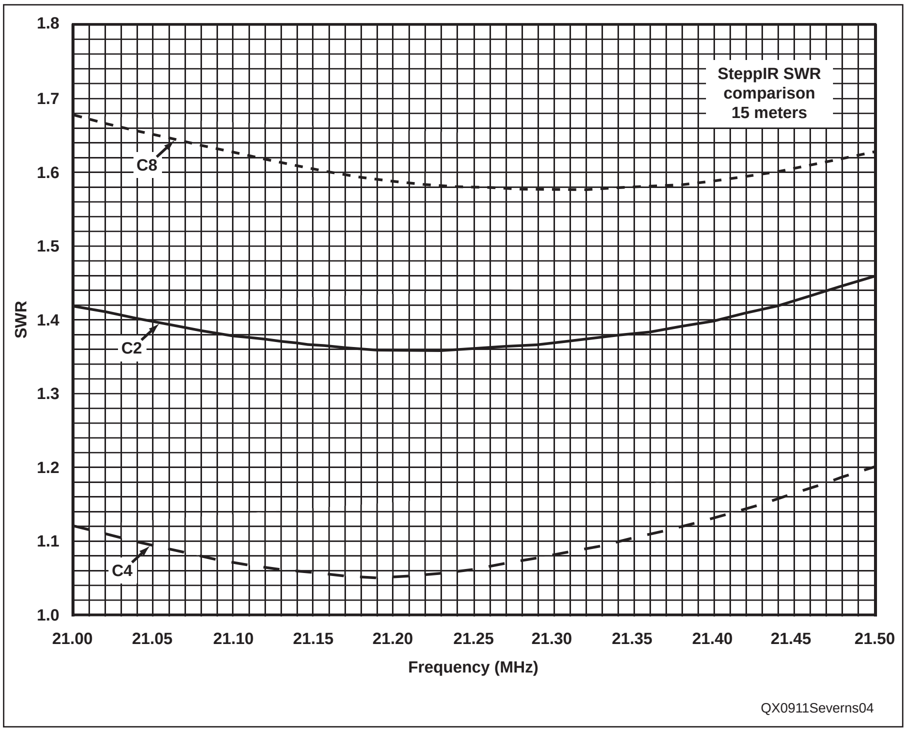


Figure 4 — Feed point SWR comparison on 15 m.

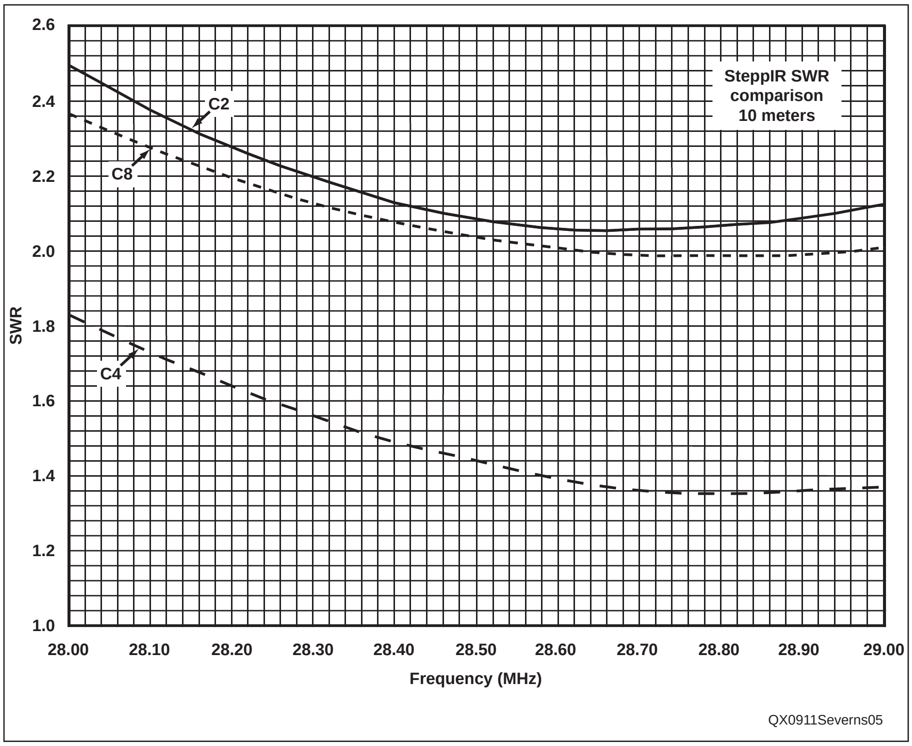


Figure 5 — Feed point SWR comparison on 10 m.

parison experiments in as short a time interval as possible. This sensitivity to changes in ground characteristics is also the reason I have emphasized that the specific numbers derived from these experiments must *not* be taken as absolutes. They are intended only to show the trends in performance between different ground systems. In addition, the frequency range in this series of tests goes much higher than those for the earlier experiments. The soil characteristics at a given location and time will vary with frequency.[6] In other words, your mileage may vary!

Despite the extensive experimental work reported in this series there will still be many unanswered questions regarding ground systems for verticals. Answers will have to be deferred to future experiments and computer modeling. Hopefully, others will be inclined to join in this effort making their own contributions. Of course not all questions have to be answered experimentally. As some of this work has indicated, *NEC* modeling can shed a lot of light on many questions, although in the end it's always more convincing if there is at least some experimental confirmation.

Acknowledgement

I would like to express my appreciation to Mike Mertel, K7IR, for the loan of the SteppIR vertical antenna used in these experiments. That antenna made the experiments much easier. I would also like to thank Mark Perrin, N7MQ, for his help at some key points in the experiments, when another hand was really helpful.

Rudy Severns, N6LF, was first licensed as WN7WAG in 1954 and has held an Extra class license since 1959. He is a consultant in the design of power electronics, magnetic components and power-conversion equipment. Rudy holds a BSE degree from the University of California at Los Angeles. He is the author of two books and over 80 technical papers. Rudy is an ARRL Member, and also an IEEE Fellow.

Notes

[1]Rudy Severns, N6LF, "Experimental Determination of Ground System Performance - Part 1," *QEX*, Jan/Feb 2009, pp 21-25.

[2]Rudy Severns, N6LF, "Experimental Determination of Ground System Performance - Part 2," *QEX*, Jan/Feb 2009, pp 48-52.

[3]Rudy Severns, N6LF, "Experimental Determination of Ground System Performance - Part 3," *QEX*, Mar/Apr 2009, pp 29-32.

[4]Rudy Severns, N6LF, "Experimental Determination of Ground System Performance - Part 4," *QEX*, May/June 2009, pp 38-42.

[5]Rudy Severns, N6LF, "Experimental Determination of Ground System Performance - Part 5," *QEX*, Jul/Aug 2009, pp 15-17.

[6]Rudy Severns, N6LF, "Measurement of Soil electrical Parameters at HF," *QEX*, Nov/Dec 2006, pp 3-9.

QEX

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.

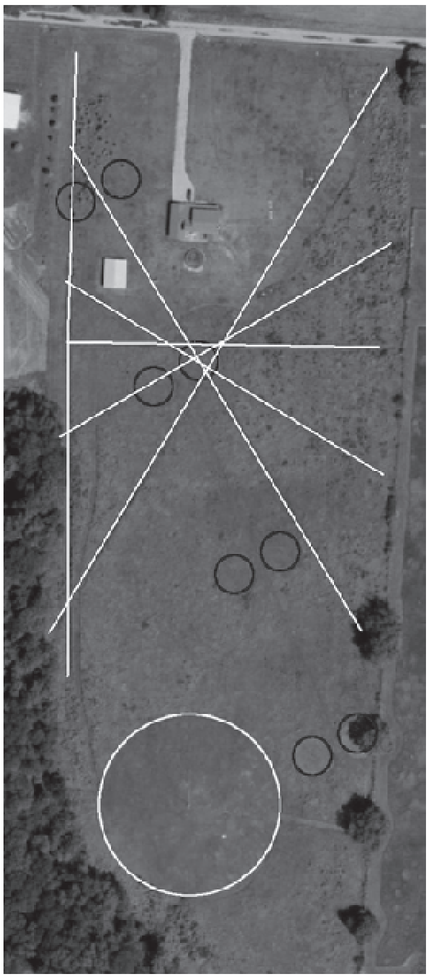


Figure 1 — K1LT Antenna Layout — The lines along the West (left) edge of the field and the lines crossing in the middle are two-wire Beverages. The large circle at the bottom of the photo marks the location of the transmitting antenna and radial field, and the smaller circles mark the locations of phased 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:

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

where φ_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

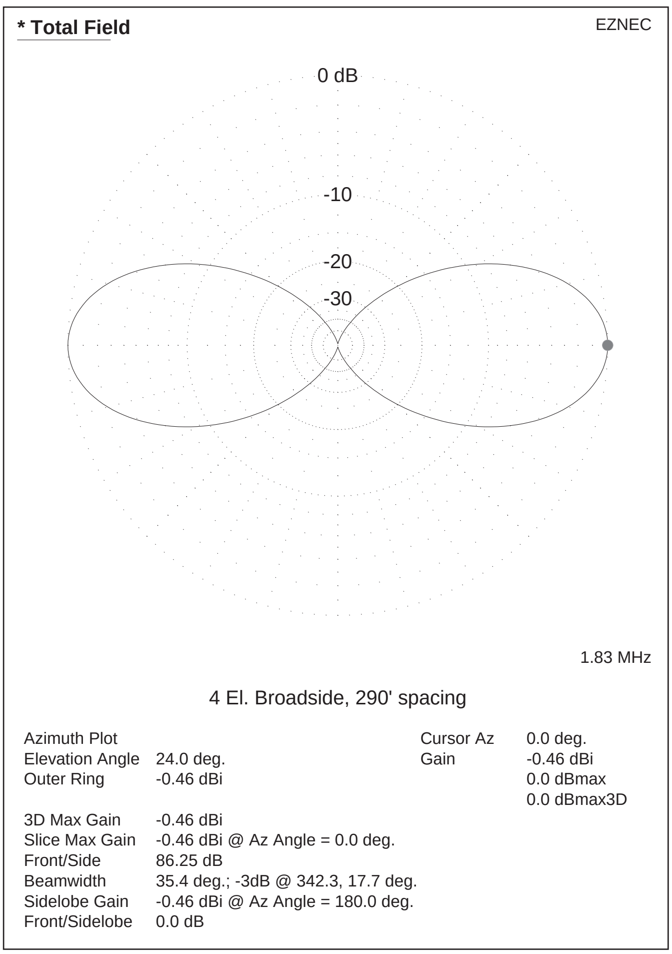


Figure 2 — Azimuthal pattern of a 4 element broadside array of short verticals with binomial feed point amplitudes.

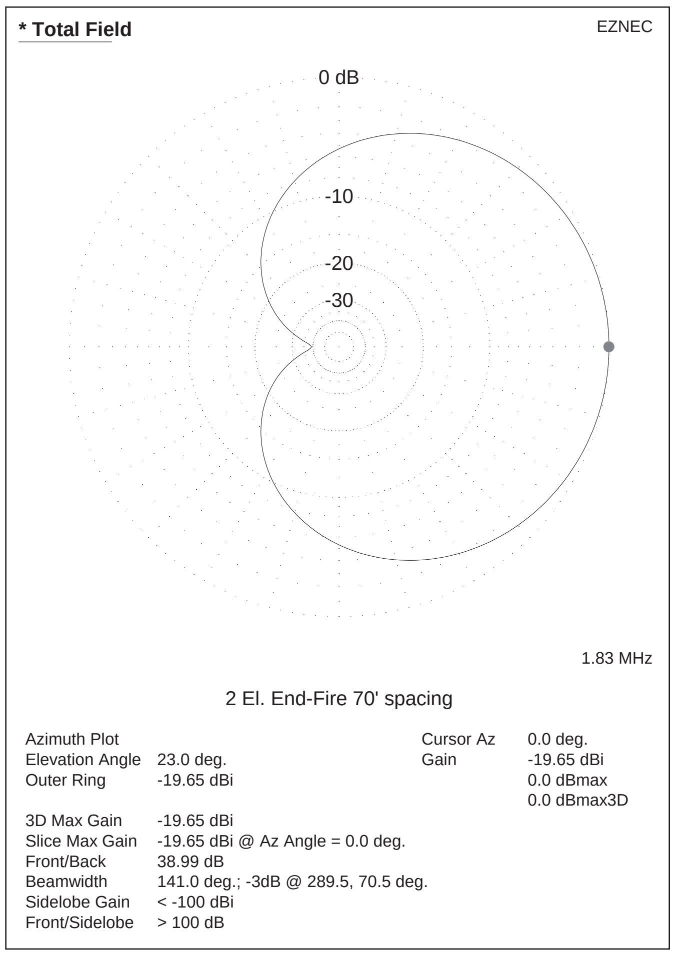


Figure 3 — Azimuthal pattern of a 2 element end fire array of short verticals with equal feed point amplitudes and 136° phase difference demonstrating the cardioid pattern.

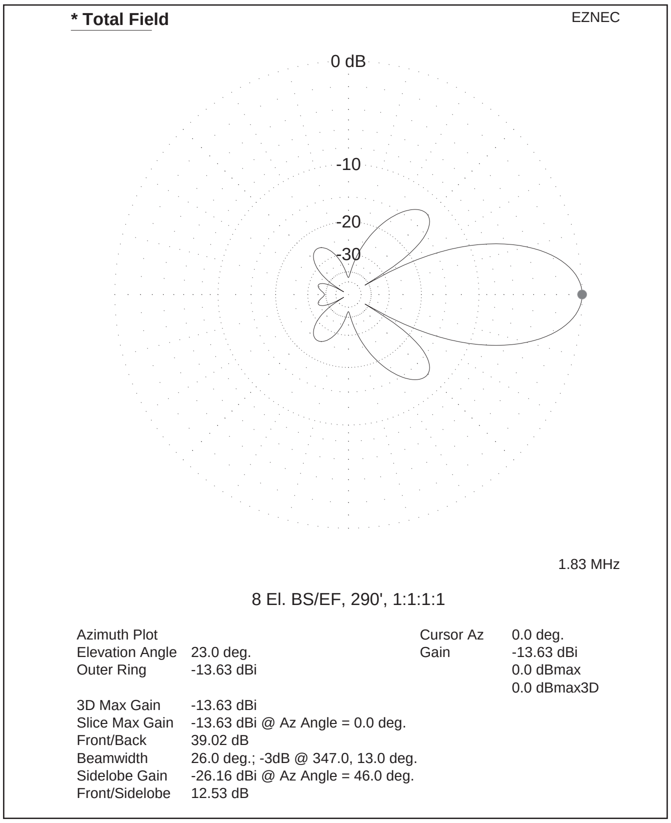


Figure 4 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire (BS/EF) arrays with equal feed point amplitudes.

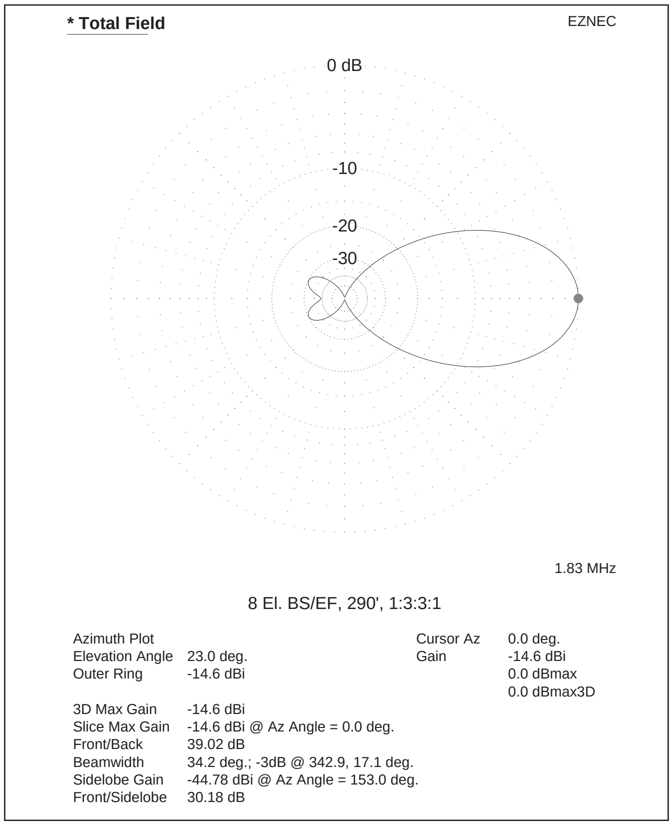


Figure 5 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with the binomial (1-3-3-1) feed point amplitude arrangement.

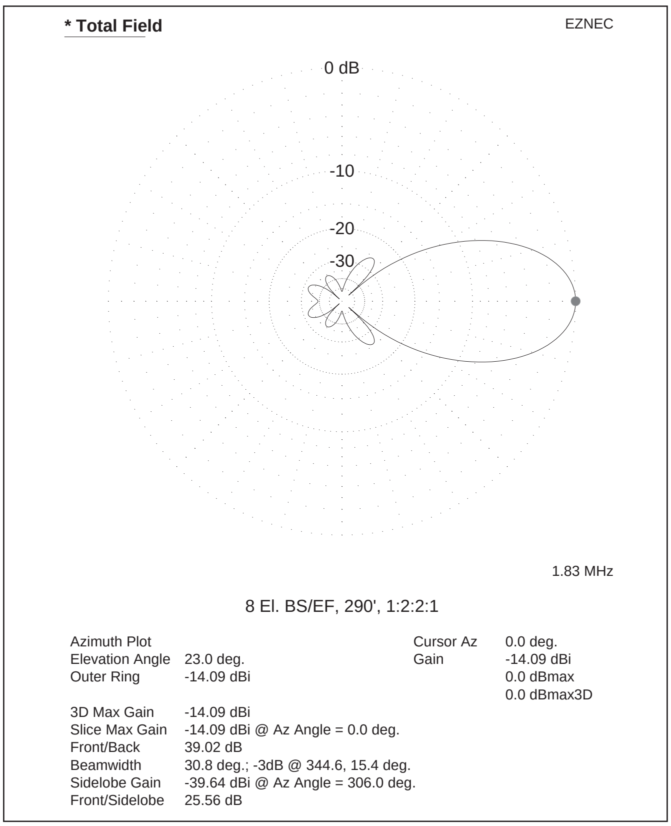


Figure 6 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with the 1-2-2-1 feed point amplitude arrangement from *ON4UN's Low Band DXing*.

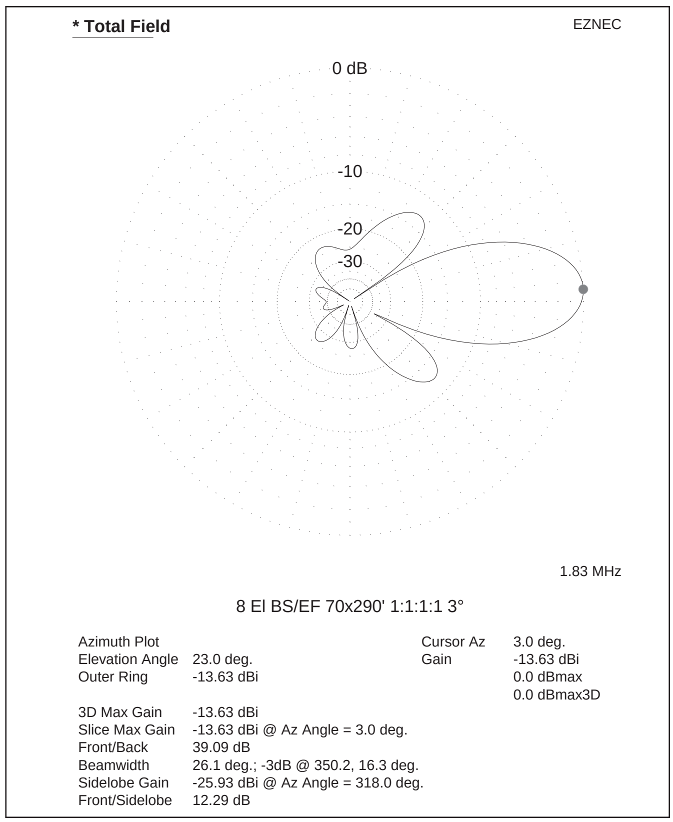


Figure 7 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with equal feed point amplitudes and progressively increasing phase offsets of about 10°.

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

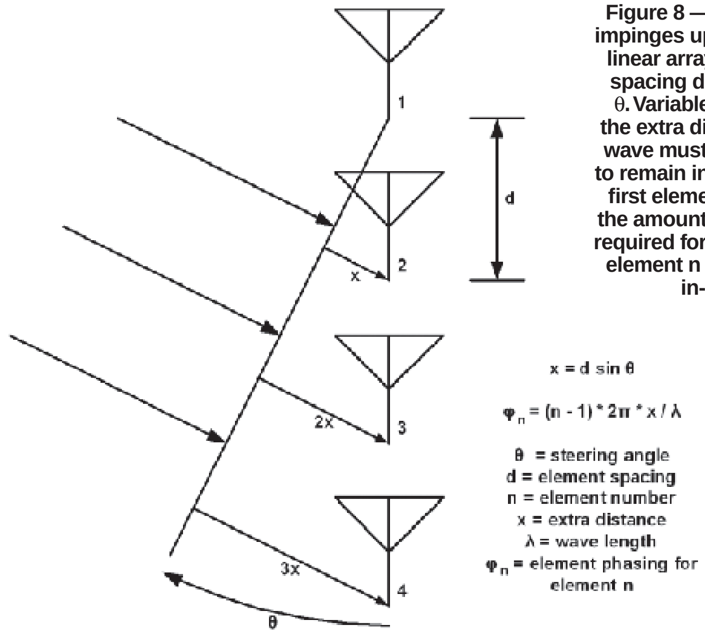


Figure 8 — A plane wave impinges upon a 4 element linear array with element spacing d at an angle of θ. Variable x represents the extra distance that the wave must travel in order to remain in phase with the first element. Angle φ_n is the amount of phase delay required for the signal from element n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

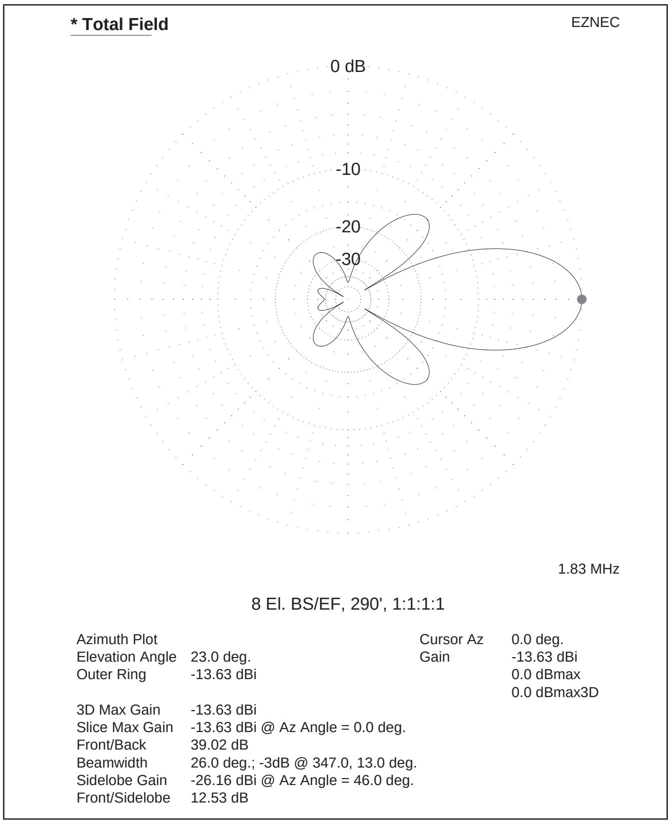
Steered Angle	*Average Gain*	*Peak Gain*	*RDF*	*RDF change from best*	*Direction of Maximum RDF*	*Beam Width*
Degrees	*dB*	*dB*	*dB*	*dB*	*Degrees*	*Degrees*
0	–29.08	–13.63	15.45	0.00	0	26.0
10	–28.99	–13.69	15.30	–0.15	11	26.5
20	–28.86	–13.87	14.99	–0.46	21	27.9
30	–28.74	–14.18	14.56	–0.89	31	30.2
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

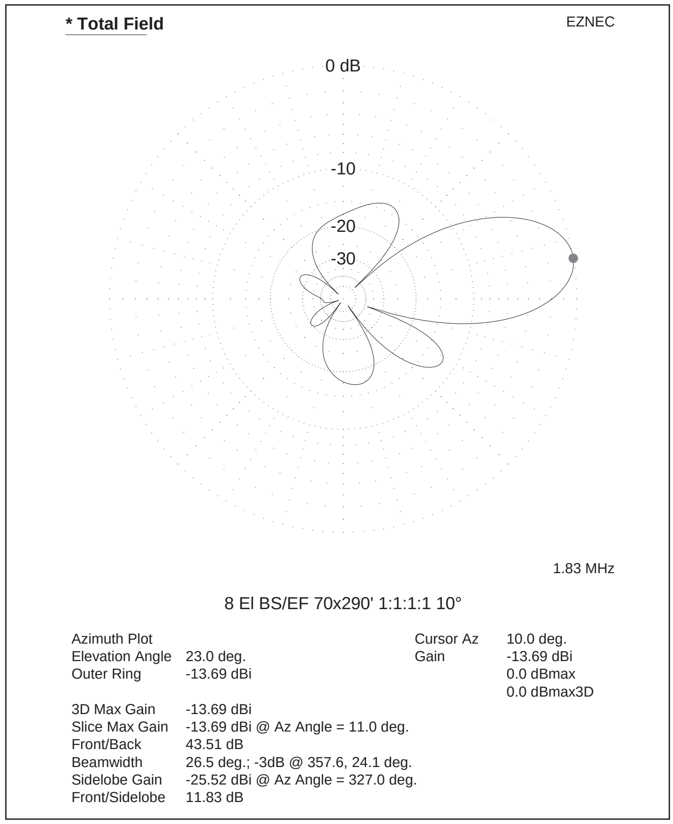
Performance data for a 4-element broadside array of 2-element end-fire arrays with binomial (1-3-3-1) feed point amplitudes steered from 0° to 90° in 10° increments.

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	

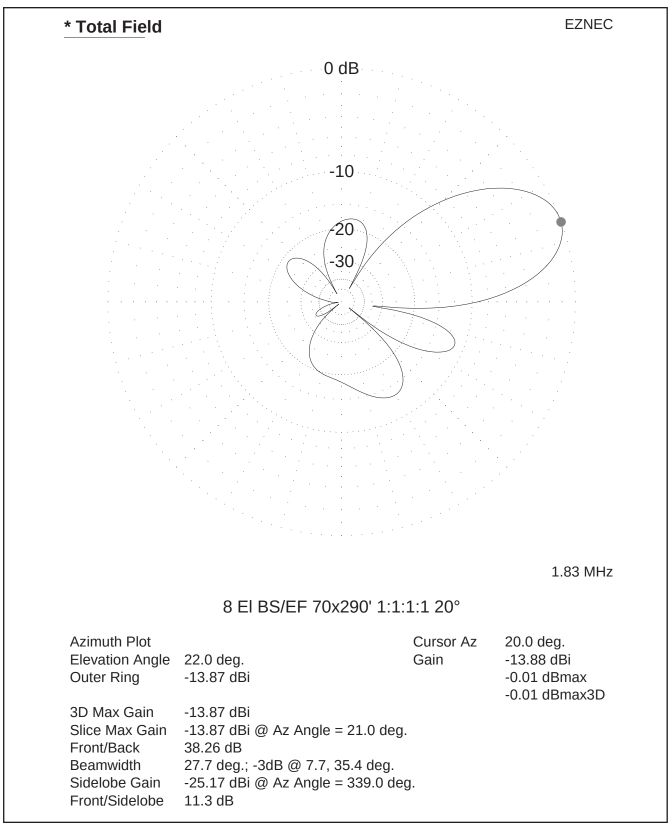


(A)

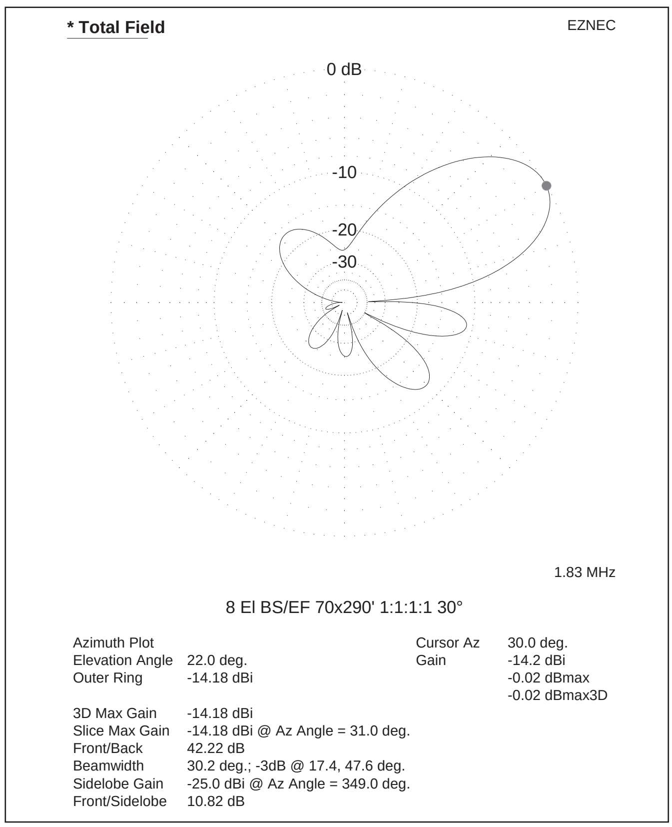


(B)

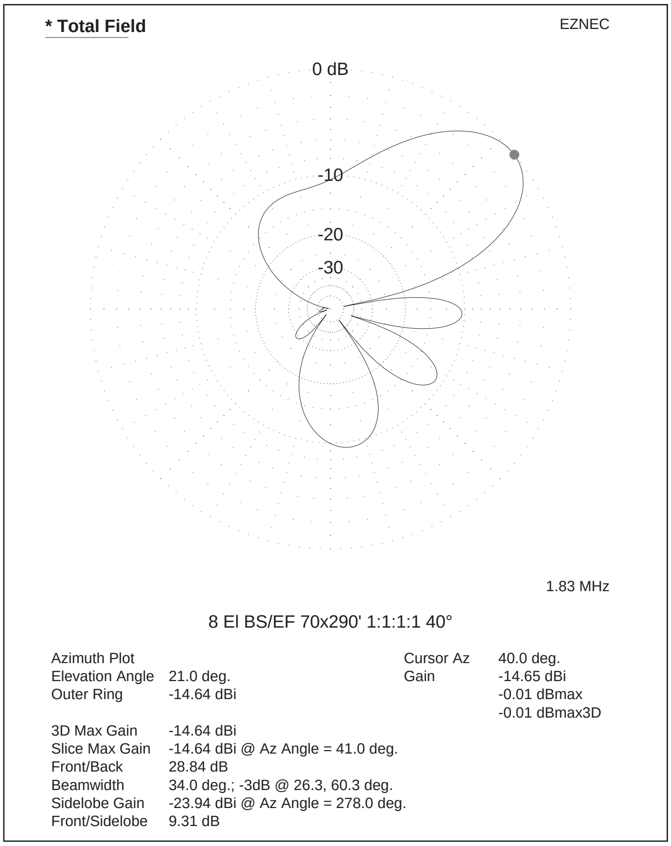
Figure 9 — Azimuthal pattern of a 4 element broadside array of 2 element end-fire arrays with equal feed point amplitudes steered from 0° to 90° in 10° increments. Note how the peak of the main lobe follows the outline of the cardioid pattern in Figure 3. Part A is at 0°, Part B is at 10°, Part C is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



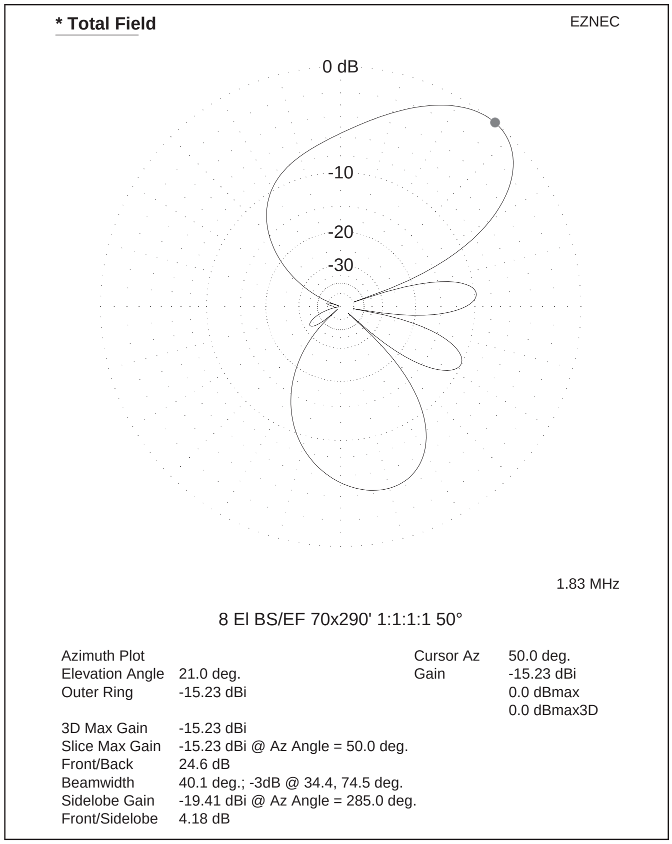
(C)



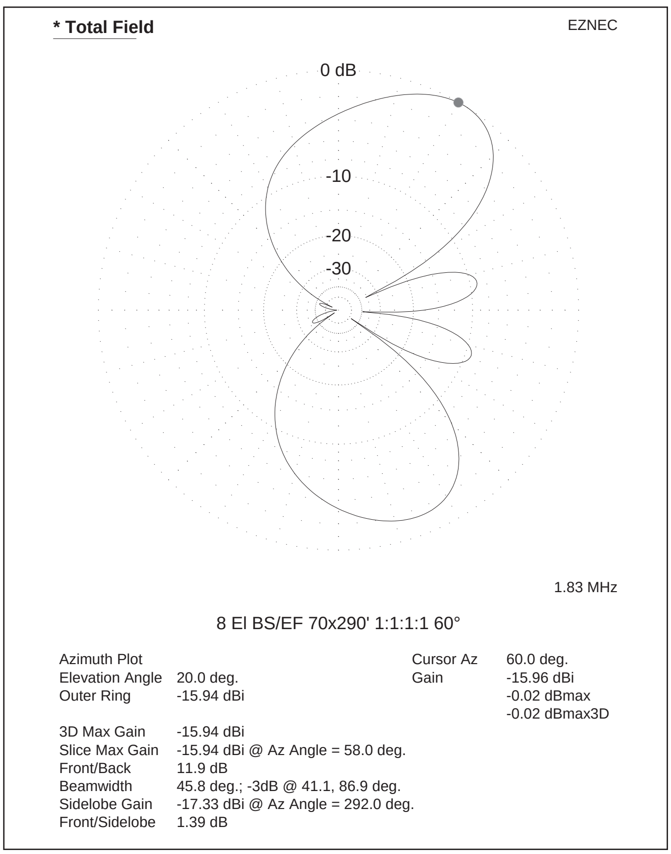
(D)


* Total Field
EZNEC
0 dB
-10
-20
-30
1.83 MHz
8 El BS/EF 70x290' 1:1:1:1 40°
Azimuth Plot
Elevation Angle 21.0 deg.
Outer Ring -14.64 dBi
Cursor Az 40.0 deg.
Gain -14.65 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

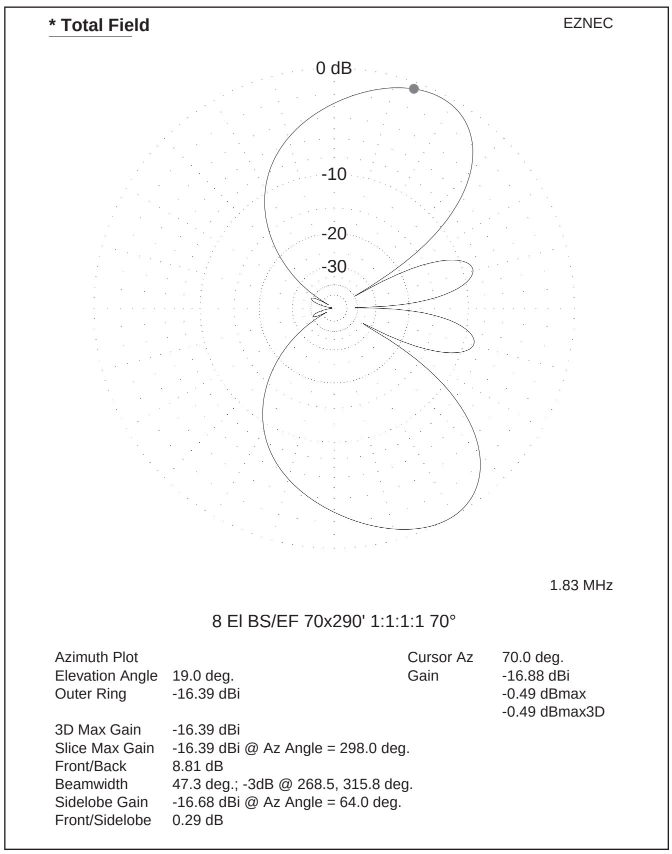
(E)


* Total Field
EZNEC
0 dB
-10
-20
-30
1.83 MHz
8 El BS/EF 70x290' 1:1:1:1 50°
Azimuth Plot
Elevation Angle 21.0 deg.
Outer Ring -15.23 dBi
Cursor Az 50.0 deg.
Gain -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

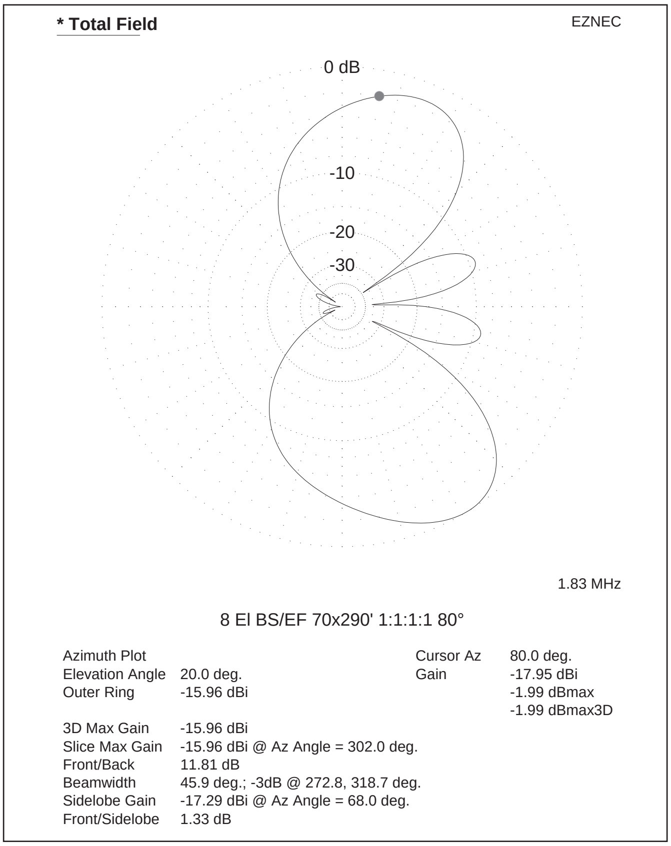
(F)


* Total Field
EZNEC
0 dB
-10
-20
-30
1.83 MHz
8 El BS/EF 70x290' 1:1:1:1 60°
Azimuth Plot
Elevation Angle 20.0 deg.
Outer Ring -15.94 dBi
Cursor Az 60.0 deg.
Gain -15.96 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

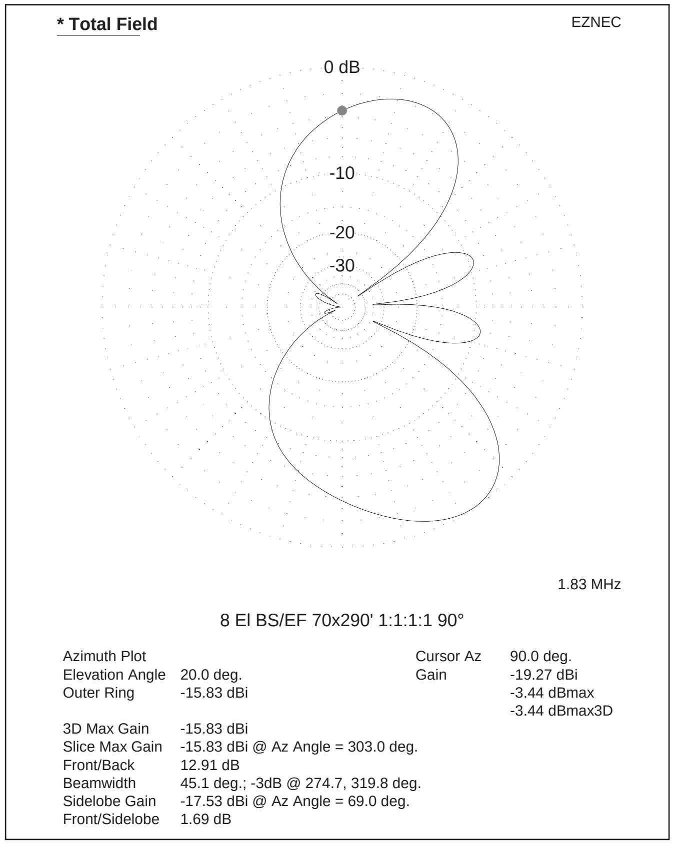
(G)


* Total Field
EZNEC
0 dB
-10
-20
-30
1.83 MHz
8 El BS/EF 70x290' 1:1:1:1 70°
Azimuth Plot
Elevation Angle 19.0 deg.
Outer Ring -16.39 dBi
Cursor Az 70.0 deg.
Gain -16.88 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.

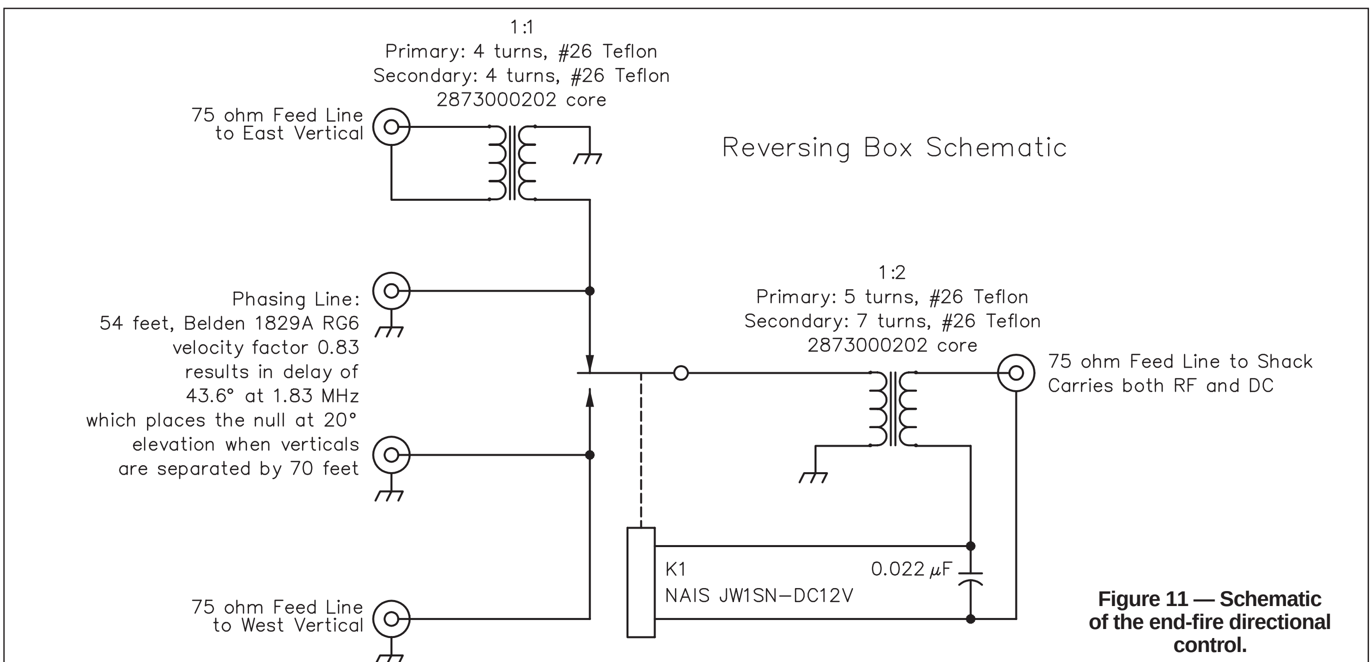


Figure 11 — Schematic of the end-fire directional control.

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.

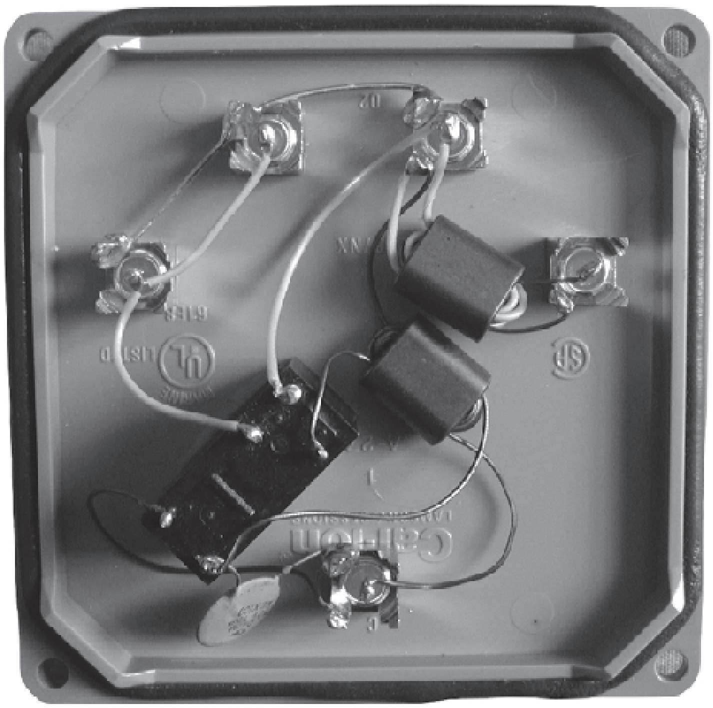


Figure 12 — End-fire directional control built into the cover of a "marine grade" plastic electrical box. Be sure to drill holes in the box to allow moisture out.

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, 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]

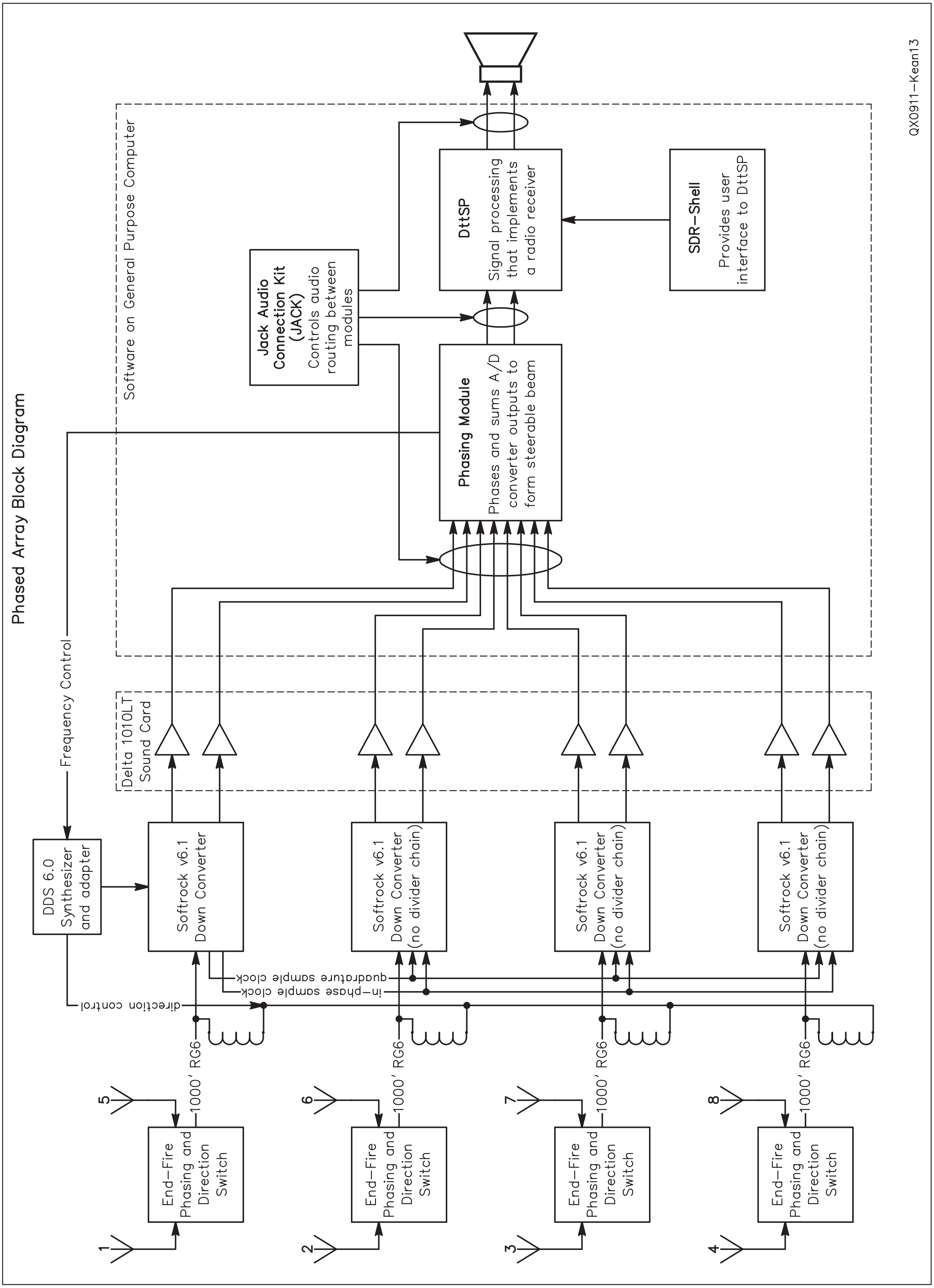


Figure 13 — The system block diagram.

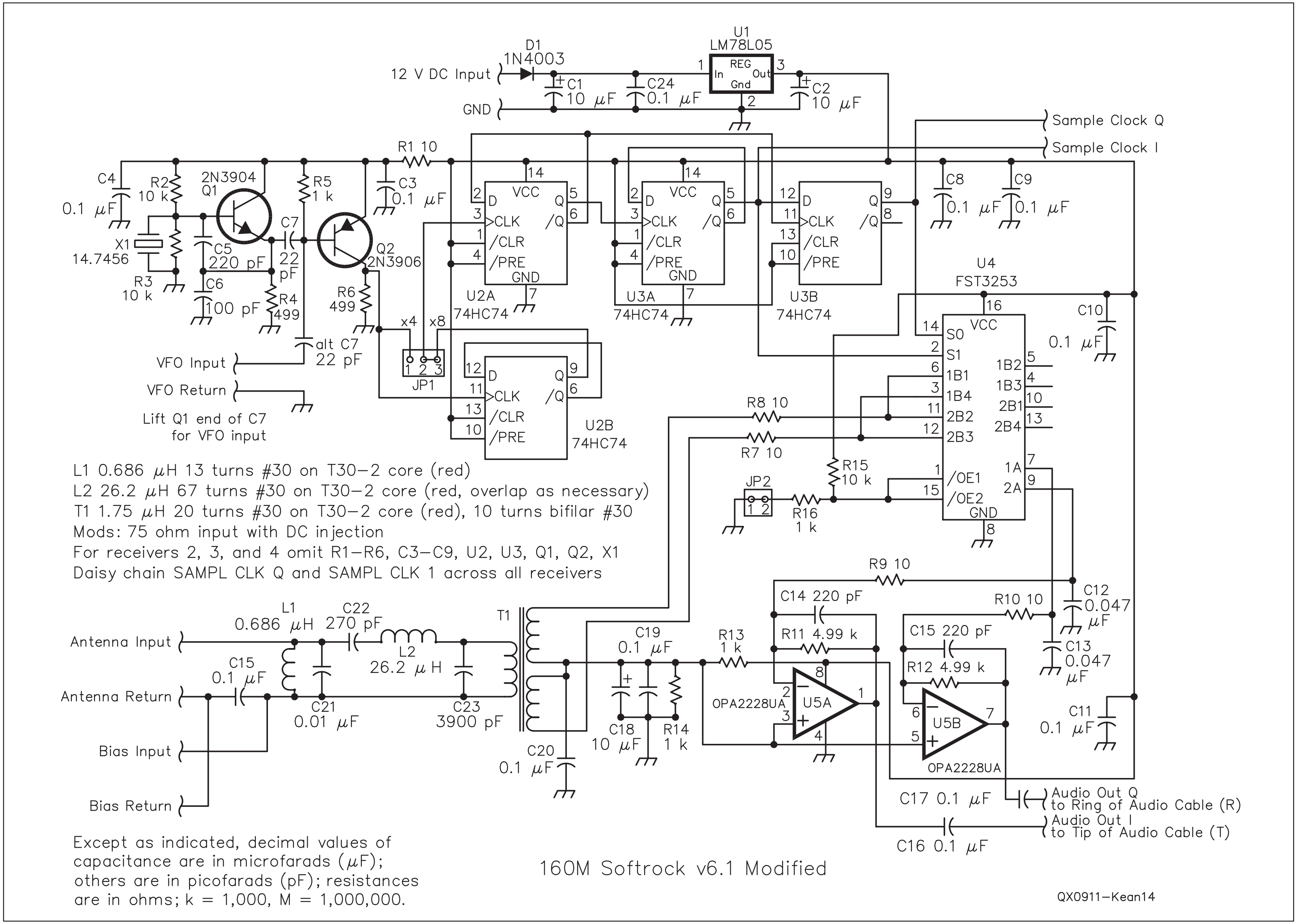


Figure 14 — This schematic shows the modified 160 meter Softrock v6.1 receiver.

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, φ_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 φ_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.

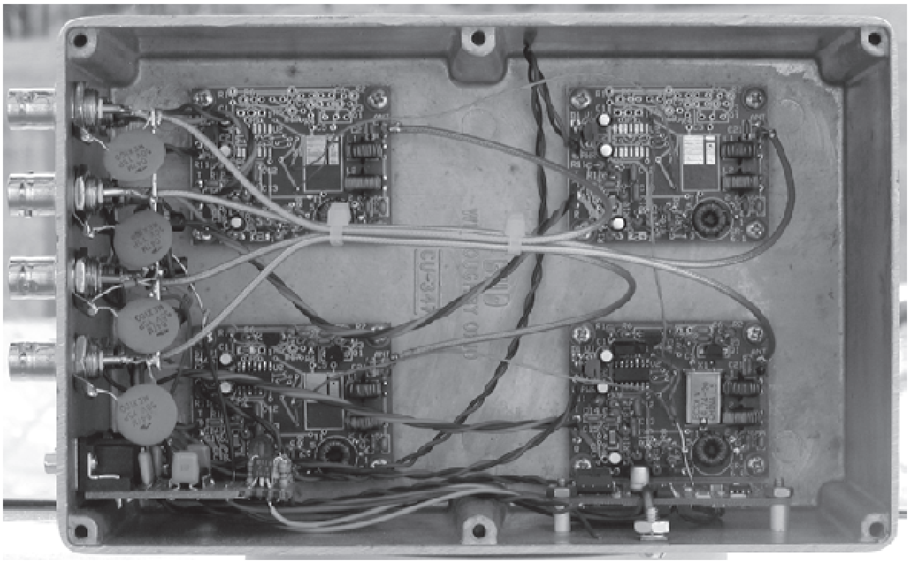


Figure 15 — Four Softrock v6.1 Receivers and a DDS-60 VFO in a cast aluminum box. The box also contains a serial interface circuit to drive the VFO controls and the direction control relays via an RS-232 port.

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

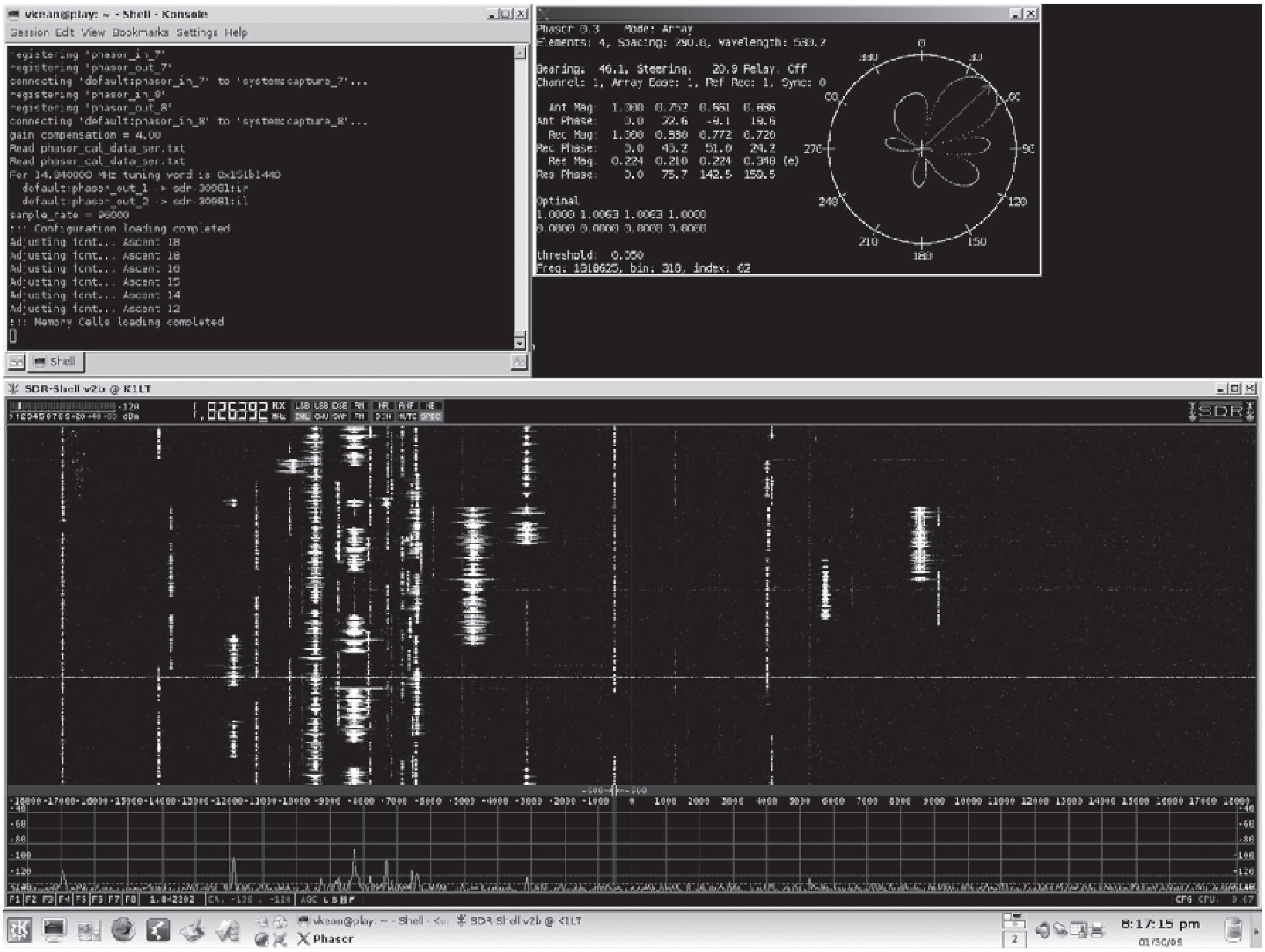


Figure 16 — Here is a screen capture of *phasor*, *sdr-core* and *sdr-shell* in operation. The window in the top center is *Phasor*, the window in the top left is a terminal session that initiated all of the programs, and the window that occupies the bottom two thirds of the screen is *sdr-shell*. The 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/kLyzStSna0dvh5bRqhv4gMPWrWWJhEz41VC0e8kQTo8l45IRVoIvUsrY_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.

QEX

Paul Wade, W1GHZ

161 Center Rd, Shirley, MA 01464; w1ghz@arrl.net

Waveguide Filters You Can Build—*and* Tune Part 1 A Tour of Filters

In this first of a 3 part series, the author reviews basic filter design parameters and terminology

Let's take a short tour of filters, skipping the deep math. Before we can talk about filters, though, we must start with resonators, the building blocks for filters. Common resonators include LC circuits, transmission line sections, waveguide cavities, and quartz crystals.

There are also mechanical resonators, which may be easier to visualize than the invisible workings of an electrical resonator. Pluck a guitar string, or tap a suspended pot lid, and an audible tone will be produced for a few seconds. The mechanical resonator has been excited with mechanical energy; the energy is stored as a resonance and slowly released as sound. Good resonators will produce a pure tone for a longer time.

Another example of a mechanical resonator is a pendulum. A good pendulum will swing for a very long time with a constant period, or time to complete one swing back and forth. The period of the pendulum swing is determined by the length or the pendulum. The amplitude of the swing will decay slowly, due to friction and air resistance, but the period does not change — the frequency is constant (frequency is the inverse of period). The stored energy is dissipated very slowly. In electrical terms, a pendulum is a high-*Q* resonator, where *Q* is defined as the ratio of stored energy to energy dissipated.

To produce something useful, some energy must be extracted from a resonator. The guitar string produces sound to make music, while a pendulum may be coupled to a clock mechanism to tell time. When energy is extracted, the resonator decays faster — the *Q* has been reduced.

If we add energy to the resonator as fast as it is being extracted and dissipated, it can continue indefinitely. We could blow gently on the pendulum each time it starts a downward swing, but we must time it correctly — energy at the wrong frequency may be counterproductive. Of course, we could use a mechanical or electrical signal to time the addition of energy — I have a clock whose pendulum will run for eight days just by winding up a spring and allowing the mechanism to extract a tiny bit of energy from the spring for each tick.

The electrical equivalent of the clock mechanism is called feedback, adding energy to a resonator to make an oscillator.

Microwave Resonators

Typical microwave resonators are sections of transmission line: odd multiples of an electrical quarter-wavelength shorted at one end and open at the other, or multiples of an electrical half-wavelength either shorted at both ends or open at both ends. The transmission line may be coaxial, with an inner and outer conductor of various shapes, or waveguide formed by a conductor. These transmission line structures are often called cavities. Planar structures on dielectrics are also used. The shapes need not be regular or symmetrical, but odd shapes will complicate calculations.

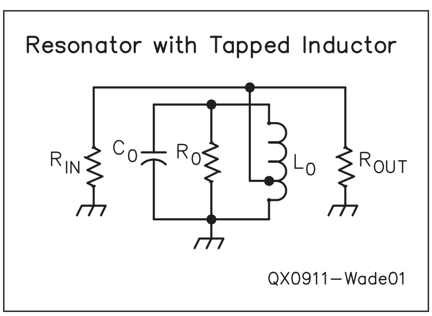


Figure 1 — This is the schematic diagram of a single L-C resonator with tapped inductor.

Whatever the configuration, a single resonator is equivalent to a parallel LC circuit (often called a tank circuit for reasons lost in antiquity) like Figure 1. At some frequency, the reactance of the capacitor, X_C, will be equal to the reactance of the inductor, X_L, and at that frequency the circuit is resonant. It will "ring" at this frequency if excited by an impulse.

Any real resonator has some intrinsic loss, shown as the R_o in the circuit. This loss determines the intrinsic *Q*, or unloaded *Q*, of the resonator: $Q_U = R_o / X$. Since the reactances are equal, either X_L or X_C may be used.

When the resonator is connected to a circuit, the resistance added by the circuit appears in parallel with the intrinsic R_o, so the total *R* must be lower than R_o, reducing the *Q* to the loaded *Q*, Q_L. For example, suppose that we make the connection by tapping down on the inductor ¼ of the turns from the bottom. You might recall that the impedance

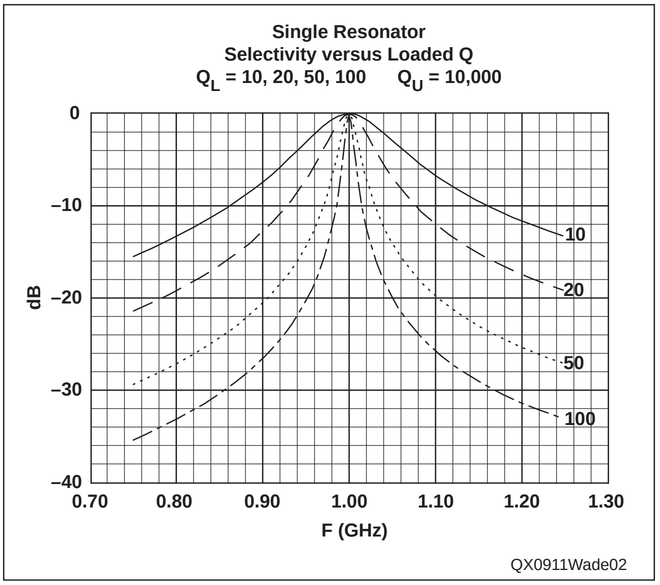


Figure 2 — This graph shows the selectivity of a single resonator versus loaded *Q*.

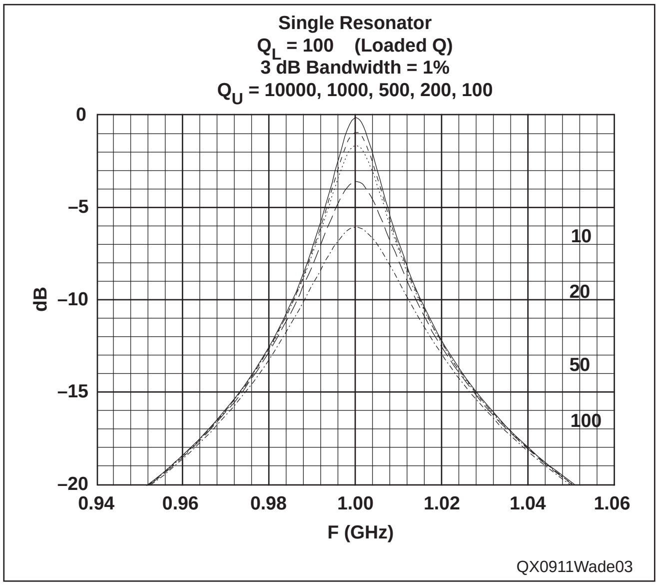


Figure 3 — A graph of the selectivity of a single resonator with loaded *Q* = 100 and varying unloaded *Q*.

ratio is the square of the inductor turns ratio; the turns ratio is 4, so the impedance ratio is 16. If we connect a 50 Ω circuit to the resonator, then the added resistance is 16 × 50 Ω = 800 Ω. If the intrinsic R_o were 10,000 Ω, then the resultant R would be 740 Ω. For an arbitrarily chosen reactance X = 200 Ω, Q_U = 10000 Ω / 200 Ω = 500, while the loaded Q_L = 740 Ω / 200 Ω = 3.7, a significant reduction.

Selectivity

The selectivity of a resonator is determined by its loaded Q_L. The 3 dB bandwidth — the difference between the frequencies at which the response is reduced by 3 dB — is simply stated by Equation 1.

$$BW_{3\,\text{dB}} = \frac{Frequency}{Q_L} \quad \text{[Eq 1]}$$

Figure 2 makes the effect clear — low Q_L resonators are not very selective, while high- Q_L resonators are quite sharp. The graph is centered at 1 GHz to make it easily scalable to any frequency — for example, the response at 0.8 times the desired frequency is exactly that shown at 0.8 GHz.

Why not just use high-Q_L resonators? Unless the unloaded Q_U is much higher than Q_L, losses will be high, since R_o would be a significant part of the circuit. Some examples are shown in Figure 3 for a Q_L = 100, so that the bandwidth is only 1% of the operating frequency.

The loss increases rapidly as Q_U decreases. The loss may be calculated using Equation 2:[1]

$$\text{Insertion Loss} = 20\log\left(\frac{Q_U}{Q_U - Q_L}\right)\text{dB} \quad \text{[Eq 2]}$$

Figure 4 shows this relationship graphically: when the ratio of Q_U to Q_L is about 10, the loss is about 1 dB. The loss is lower with higher ratios, but loss increases rapidly with lower ratios: when $Q_L = Q_U$, the loss is 6 dB. Trying to make a sharp filter with low Q_U resonators will result in most of the power heating the filter.

At lower frequencies, unloaded Q_U may be improved by increasing physical size, like the large quarter-wave "cavities" used for repeater duplexers. At microwave frequencies, however, when large dimensions are a significant part of a wavelength, additional unwanted resonances will be created in the cavity.

[1]Notes appear on page 43.

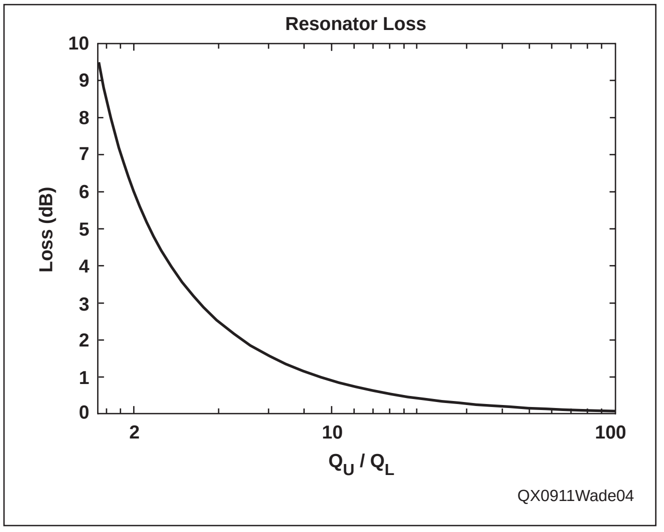


Figure 4 — Here is a graph showing resonator loss versus ratio of loaded *Q* to unloaded *Q*.

Multiple Resonators

A common rule of thumb is that a single capacitor or inductor in a circuit creates a 6 dB / octave rolloff. A simple resonator, with one C and one L, should roll off at 12 dB / octave, where an octave is doubling or halving the frequency relative to the bandwidth. To get faster rolloff, for better out-of-band rejection, therefore, you will need additional resonators.

Simply connecting resonators together produces interactions that distort the response. A traditional technique — dating back to tuned radio frequency (TRF) receivers, before the superheterodyne — is to separate resonators with amplifiers to limit interaction between the resonators. At the lower microwave frequencies, where MMIC amplifiers provide cheap gain, we often use two or three simple "pipe-cap" resonators separated by MMIC amplifiers. This com-

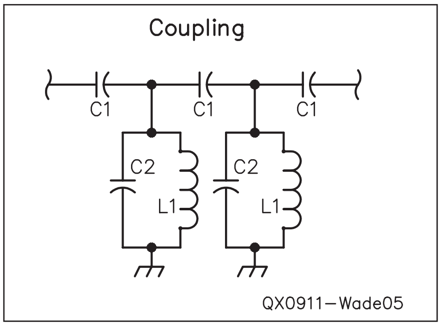


Figure 5 — This schematic diagram shows a simple double-tuned filter with coupled resonators.

bination can provide enough selectivity for good local oscillator (LO) and image rejection. The resonators may be synchronously tuned, all at the same frequency, for narrowest bandwidth. Alternately, they may be stagger-tuned, to slightly different frequencies, for a wider passband while still providing fast rolloff.

Modern filter design techniques use multiple resonators, or sections, coupled to control the interactions and achieve a desired response. By varying the coupling between resonators, the response may be controlled. A simple double-tuned circuit, Figure 5, with two coupled resonators, is a good example. Figure 6 shows the effect of coupling: the optimum coupling achieves a flat response; over coupling increases the bandwidth but creates some ripple in the passband, while under coupling decreases the bandwidth at the cost of increased loss. Note that all the responses roll off at the same rate outside the passband — only additional resonators will improve the rolloff.

The coupling between resonators may be capacitive, as shown in Figure 5, or inductive, or magnetic, with no physical connection. The input and output connections may also be capacitive, as shown, or inductive, either tapped down on the coils like Figure 1 or as a separate winding.

Adding additional resonator sections makes the filtering action much sharper, as shown in Figure 7. A filter may be designed for narrow or wide bandwidth with skirts as sharp as desired. The dimensions and tolerances become more critical, however, and tuning the filter can be much more difficult. I have a six-resonator filter that looks great in the computer design but has proven impossible to tune.

Many commercial applications have stringent filter requirements, to separate channels or to block adjacent bands. These may require a broad passband with low insertion loss, steep skirts that roll off quickly, and high stopband rejection. Figure 8 illustrates these terms. This surplus filter has close to 0 dB loss over a broad passband and more

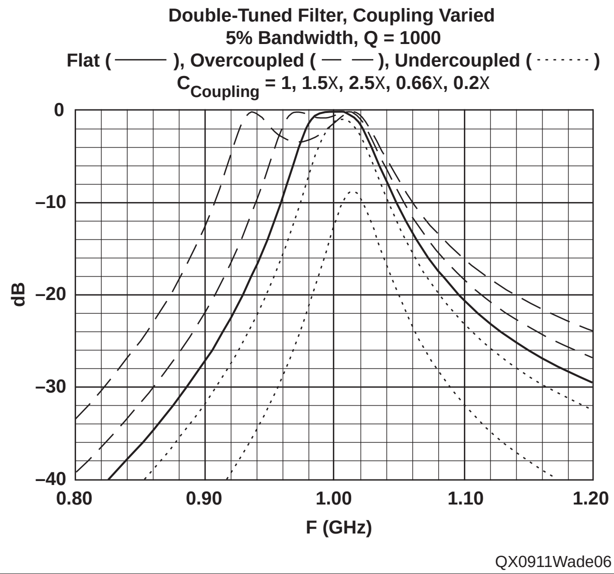


Figure 6 — The responses of a double-tuned filter with under-coupling, critical coupling, and over-coupling are shown in this graph.

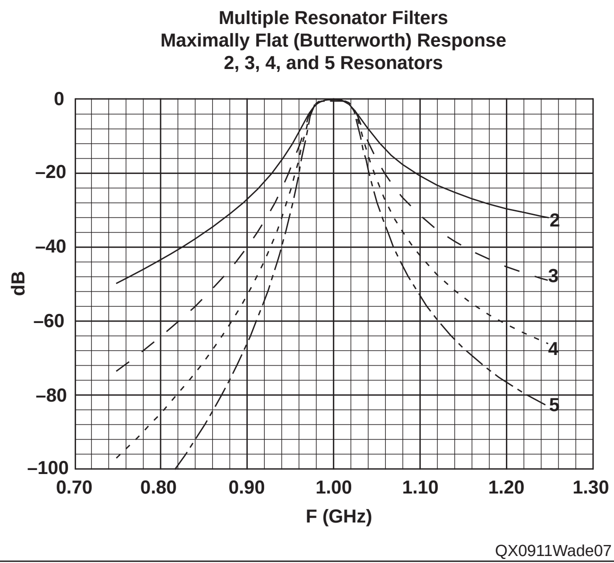


Figure 7 — This graph shows the response of filters with 2, 3, 4, and 5 coupled resonators.

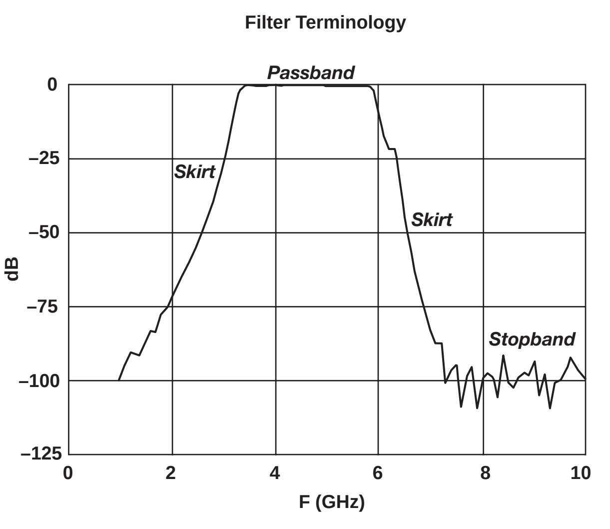


Figure 8 — This graph shows the response of a multiple-resonator surplus filter, and also illustrates the common filter terminology.

than 100 dB of stopband rejection — a good filter can be better than we can measure.

Advanced filter design techniques have been developed to meet these requirements. For instance, a Chebyshev (Чебышёв — also sometimes spelled Chebychev, Chebyshov, Tchebycheff or Tschebyscheff) filter has steeper skirts at the cost of some ripple in the passband loss; the allowable ripple is part of the design procedure.[2, 3] Figure 9 compares a five-section Chebyshev filter to the maximally-flat, or Butterworth, design from Figure 7. More advanced filter design techniques, such as Cauer, elliptic-function, and cross-coupled filters, offer high performance at the expense of more complex design procedures and tuning difficulty. Today, filter design software eases the task; traditionally, the design parameters were tabulated in books.[4, 5, 6] Either way, some engineering is still needed to design a practical filter that can actually be built. Amateurs rarely need such a fancy filter, except for the crystal filters in our transceivers.

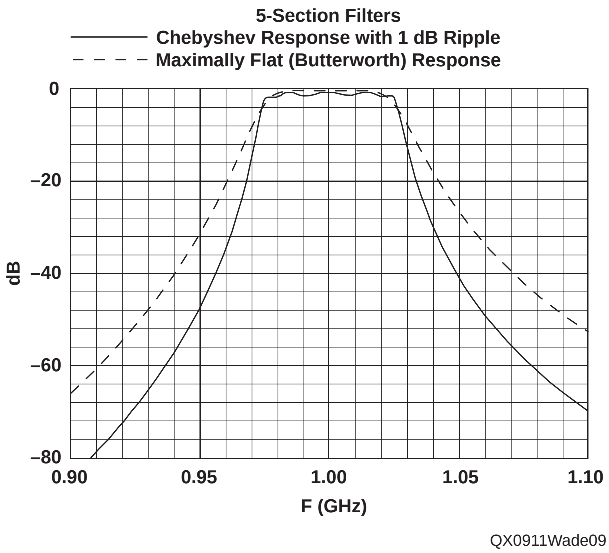


Figure 9 — This graph is for five section filters with a maximally flat Butterworth response and a Chebyshev response with 1 dB of passband ripple.

Most microwave operation is close to a standard calling frequency, so all that is required is a filter that passes the calling frequency and rejects the conversion image and any LO leakage from the mixer. For the common 144 MHz IF, the ratio of LO frequency to radio frequency is 0.89 at 1296 MHz and 0.937 at 2304 MHz. These ratios may be scaled to 0.89 GHz and 0.937 GHz on Figure 7. For at least 20 dB of LO rejection, a two section filter is adequate for 1296 MHz, while a three section filter may be needed for 2304 MHz. For higher bands, we need either a sharper filter or a higher IF, such as 432 MHz.

A sharp filter may be either very narrow or have more sections. Several factors limit how narrow a filter we can use. The first is the unloaded *Q*, or Q_U, of the resonators — we can't make a narrow, high *Q* filter with low *Q* resonators. The response curve of a lossy filter with low *Q* resonators tends to be more rounded and less sharp.[7]

A more practical limit is whether we can tune a narrow filter *and* have it stay tuned over temperature and vibration, particularly for rover operation. The alternative, adding more sections, also has problems. Unlike the ideal filters in Figures 7 and 9, each section adds additional loss, so that the filter loss is proportional to the number of sections. Also, filters with more than three or four sections are very difficult to tune properly without sophisticated test equipment.

With high *Q* resonators, like those found in waveguide filters, a narrow double-tuned circuit, or two-section filter, should be satisfactory for many amateur applications. We can see from Figure 5 that the skirt selectivity is not affected by the coupling, so the maximally flat, or Butterworth type of filter is a good choice. The bandwidth of a two section Butterworth filter is $\sqrt{2}$, or 1.414 times the bandwidth of a each single resonator. Thus, to find the desired Q_0, the loaded *Q* of each resonator, we simply calculate the value using Equation 3.

$$Q_0 = \frac{Frequency}{BW_3} \times \sqrt{2} \qquad \text{[Eq 3]}$$

For circuits with discrete L and C values, the coupling components are easily calculated.[8] For direct-coupled resonators like those in waveguide filters, however, we must rely on tables (such as those given in Note 6) or programs like *WGFIL*.[9] More importantly, we can estimate loss using Figure 4 if we have an idea of the unloaded *Q*, Q_U, of the resonators.

Different types of filter construction are available, each with advantages and disadvantages. Waveguide filters can have extremely high *Q*, so that narrow filters are possible with very low loss. At lower frequencies, however, they become very large. Printed circuit filters have low *Q* resonators, but are cheap and repeatable, requiring no tuning, so they may be preferred at lower frequencies where gain is cheap. Other possible choices include helical, interdigital, and combline filters.[10] Each type offers different tradeoffs in loss, size, and difficulty in design and construction. It is a matter of choosing an adequate filter for each application.

Part 2 of this series will describe practical "Waveguide Post Filters," and Part 3 will present "Evanescent Mode Waveguide Filters."

Notes

[1]Harlan H. Howe, *Stripline Circuit Design*, Artech, 1974, p 215.

[2]See the Wikipedia entry at **http://en.wikipedia.org/wiki/Pafnuty_Chebyshev**.

[3]See the Wikipedia entry at **http://en.wikipedia.org/wiki/Chebyshev_filter**.

[4]Anatol I. Zverev, *Handbook of Filter Synthesis,* Wiley-Interscience, 2005, available from **Amazon.com** and other book sellers.

[5]Philip R. Geffe, *Simplified Modern Filter Design,* Rider, 1964.

[6]G. Matthaei, L. Young, and E.M.T. Jones, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures*, McGraw-Hill, 1964.

[7]Hong, Jia-Sheng, and Lancaster, M.J., *Microstrip Filters for RF/Microwave Applications*, Wiley, 2001, p 71.

[8]Wes Hayward, W7ZOI, Rick Campbell, KK7B, and Bob Larkin, W7PUA, *Experimental Methods in RF Design,* ARRL, 2009, p 3.14. ARRL 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**.

[9]Dennis G. Sweeney, WA4LPR, "Design and Construction of Waveguide Bandpass Filters," *Proceedings of Microwave Update 1989*, ARRL, 1989, pp 124-132. The *WGFIL* program may be downloaded from **www.w1ghz.org/filter/WGFIL.COM**.

[10]Paul Wade, W1GHZ, "Waveguide Interdigital Filters," *QEX*, January 1999, p 3. A PDF file of this article is available at **www.w1ghz.org/10g/QEX_articles.htm**.

QEX

Ray Mack, W5IFS

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

SDR: Simplified

Mea Culpa

I made a serious mistake in preparing the text for the July/August issue. I believed that the tools and documentation were sufficient to actually allow us to produce a working program that would implement a software defined radio. I have spent much time attempting to find the necessary *C* program materials, but with little success. I am still working on the process and hope to have an updated zip file for the July/August issue on the ARRL Web site by the time you read this. As I sort out what to do and why, I will update the Web site with instructions and clarifications.

Michel Barbeau, VE3EMB, has created a new *Linux* kernel that can be loaded into the board.[1] The new kernel includes a proper SPI driver that is not in the board as shipped.

I have received feedback that some of the three letter initials that I thought were in common use are foreign to some readers. I have provided a short Glossary as a sidebar to this installment. I find the use of two and three letter initials to be generally antisocial behavior by engineers, but some have been used for so long that they are a reasonable part of our language and replace their English language equivalents. I will avoid initials as much as possible in future columns. Some initials have such wide use, however, that the original English has actually been replaced. NASA, FBI, and IBM are examples where no one uses the English, and most folks probably do not know the origin of the initials. DSP, SDR, DFT, and FFT are ones from this installment that fall in the common use category. (See the Glossary for definitions.)

Hardware Update

I have been devoting significant time (when not fighting the software tools) to developing reference boards that we can use for our lab experiments. I have enough requests that Dave Pollum and I developed a combination DAC08/AD7476 board. We have Gerber files if you want to buy your own board from a circuit board manufacturer, or you can contact me directly to buy a board.[2, 3] The board uses through hole parts except for the AD7476A and AD8027A, and has a silk screen for component placement. The design of the AD7476 portion is different from the Analog Devices version with respect to power and voltage reference. Our board is for demonstration purposes rather than a reference design for a precision component, so we removed some of the power supply components to reduce cost and complexity. Figure 1 is the schematic that Dave created for this board.

Resource Update

I do a sanity check on all of the topics I write for you, using the books I have in my library and other resources. I have a new book to recommend, based on my research for this column. *Multirate Digital Signal Processing* has a fair amount of math, but the descriptions are understandable even if you totally ignore the equations.[4] The authors worked at what used to be Bell Labs. The focus of the book is on telecommunication applications, so it fits well with our interest in software defined radios.

If you haven't discovered Wikipedia, you should give it a whirl. It is one of those other resources I use. For example, type "wiki frequency leakage" in the search bar of your Web browser, and you will see a link to a description of spectral leakage as described below. It is a great place to look if I describe something you don't understand.

In digging through the documentation for the *C* compiler, I found that it implements DSP specific functions and data types that are a part of the *Visual DSP* product from Analog Devices. The help files on the CD for the Gnu *C* compiler point you to the documentation for *Visual DSP*, but the link is very old. Instead, you will want to go to the Analog Devices Web site to get the manual.[5] You will also likely want to look at the site for additional manuals for the Blackfin processors.[6] I also recommend downloading the manual for the BF537 EZ-Kit Lite.[7] Our Stamp boards only implement a subset of the EZ Kit hardware. Even so, the manual is a good source of information on our hardware.

Sampling, Nyquist and Spectrum

We need to dig a little deeper into the math at this point. The sequence of digital numbers that we get from the ADC in our real hardware is a process called sampling. The math that is involved in sampling multiplies a sampling function times the input analog signal. This multiplication changes the continuous analog input signal into a sequence of individual samples that are equally spaced in time.

The math sampling function that we use doesn't really exist. It is a mathematical trick, but its properties are important for conversions we can do in the real world. The basis of the sample waveform is called an impulse. It is a signal that has infinite height, an area of one, and zero width. The sampling waveform is actually a sequence of impulses with the time between impulses equal to the period of the sampling frequency. The math that deals with the implications of an infinite-height pulse gets really messy, and, thankfully, we can ignore it. The most important consequence is that an infinite sequence of impulses in the time domain produces a Fourier Transform that is an infinite sequence of impulses in the frequency domain. This is the basis for all of the discrete transformations. The Fourier Transform of a sample function of 1 Volt-second at 500 kHz has a spectrum containing 1 V at dc plus a 1 V sine wave at 500 kHz and a 1 V sine wave at every positive and negative harmonic of 500 kHz, from negative infinity to positive infinity. In the RF world, we call such a time domain signal a comb generator, since the output spectrum looks like a comb with equal height tines. It is the zero width of the time domain signal that causes the "comb" to have equal heights at all frequencies.

Our hardware generates the sampling sequence and multiplies it times the input analog signal. The result is the sequence of digital numbers corresponding to the input analog signal. If you could actually produce the sampled sequence in the real world and then look at it on a spectrum analyzer, you would see the spectrum of the input around dc and a double sideband set of the input spectrum around each harmonic of the sample frequency. This is shown in Figure 2.

Even though we cannot really produce this exact spectrum in the analog world, it does exist in the DSP world, and is very real when we do the math. The spectrum between dc and one half the sample frequency is called the first Nyquist zone. It holds the exact representation of our input signal. The next Nyquist region contains the same spectrum, but it is inverted since it is the lower sideband. We can easily apply a band pass filter to the spectrum of any of the Nyquist zones to produce a frequency translated version of our baseband signal. We can also use the equivalent of a product detector to take an inverted spectrum version and invert it a second time to get back to a true version of the original signal.

Discrete Fourier Transform and Filters

We talked a little about the discrete Fourier transform (DFT) in the July/August issue. We were interested in using the DFT to generate the coefficients for our filters. We need to look at the relationship between the number of samples and the

[1]Notes appear on page 48.

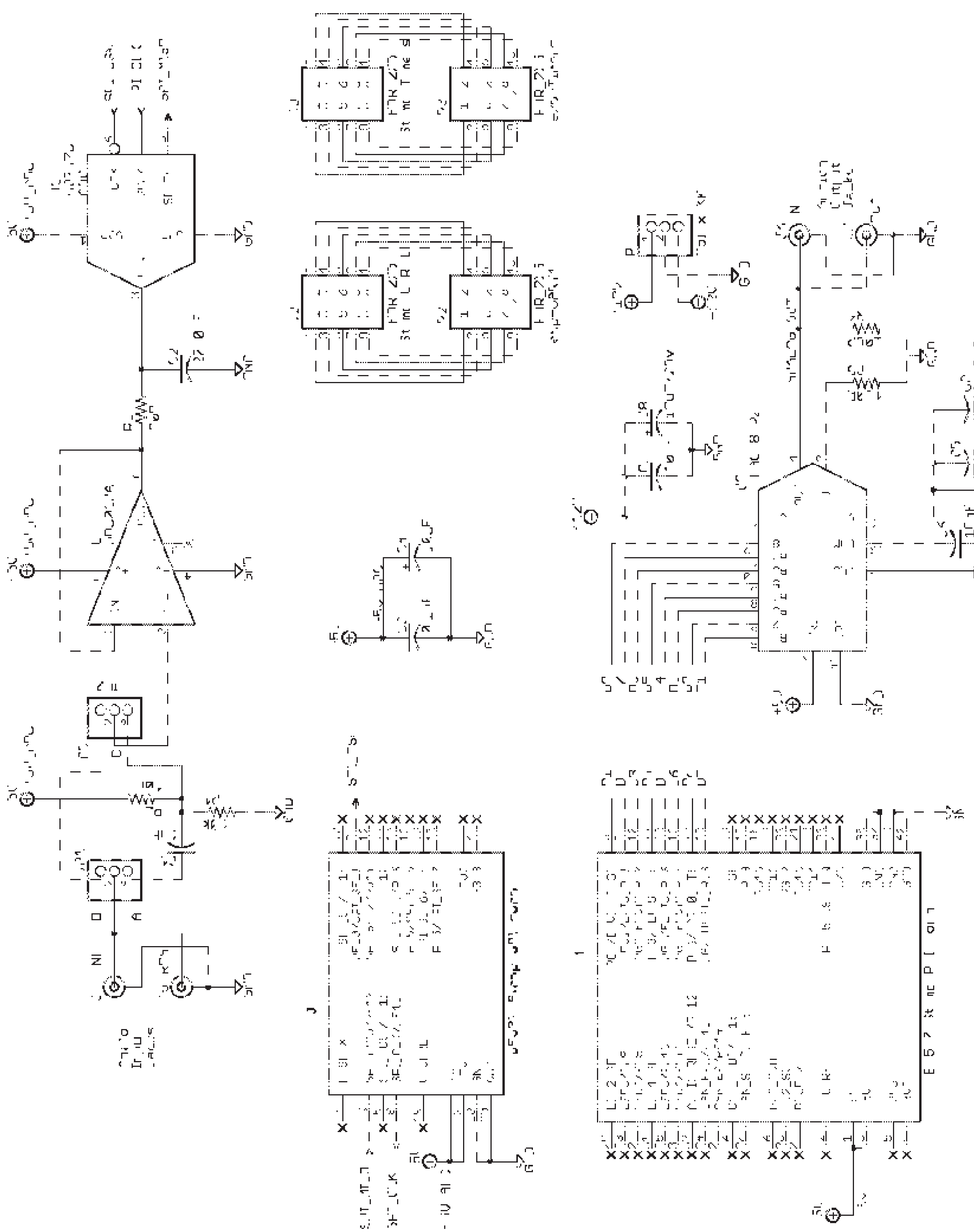


Figure 1 — Schematic of the combined AD7476 ADC and DAC08 DAC board.

sample rate and how that affects the representation of the data.

Mathematicians call the time domain signals that we use a "real" signal. Such a signal requires two wires: ground and signal. The signal from your microphone or your speaker is a "real" signal. This signal inherently contains all of the amplitude and phase information for a human to use it. A Fourier transform converts this signal into what mathematicians call a "complex" signal. All that really means is that the Fourier transform takes the single dimension real signal (voltage versus time) and creates two new signals that contain the phase and amplitude information in the frequency domain. We call the complex data I and Q.

A discrete Fourier transform starts with the conversion of our real signal into a sequence of evenly spaced samples in time. The Fourier transform action converts the time sequence into two sequences of samples in the frequency domain. One is the I sequence and the second is the Q sequence.

Our example samples the input at 50 kHz and the input data is bandwidth limited to 22 kHz (to meet the Nyquist criterion). We will do our discrete Fourier transform on a sequence of 1000 samples. The output of the transform always produces an I sequence and a Q sequence, where the input energy is split between the two sequences. The result is that the I sequence holds 500 frequency samples and the Q sequence holds the other 500 frequency samples. There is a small subset of real signals, where either the I sequence or the Q sequence will contain only zeroes, but the general case always has some energy in each sequence.

The center of the data is at time zero, with half of the samples before our arbitrary zero point and half after, so that the math works correctly. The symmetry about zero causes the frequencies of the transformed data to be centered around zero hertz, where half the samples represent negative frequencies and half are positive frequencies. Our input samples are spaced exactly 20 microseconds apart and they represent only the voltage at each instant that is measured; no values in between exist. Each transformed frequency sample also represents energy at ***exactly*** one frequency; no frequencies in between exist. Throwing away the information in between is the essence of the "discrete" in the discrete transform. The frequencies of each output transform sample cover the frequencies from –25 kHz through +25 kHz for a total frequency span of 50 kHz (the same as our sample rate). We have 500 I samples that cover 50 kHz. The math is pretty easy: 50 kHz / 500 samples = 100 Hz per sample. So the first I sample is –24900 Hz. The second sample is –24800 Hz and so on to +24900 Hz. The same frequency sample positions occur for the Q sequence. The DSP term for each of these sample positions is a ***bin***. ***Bin*** comes from the concept that the energy is stored in a container just as if it were an actual container on the wall. In our case we would need 1000 real containers. All of the input energy gets stored in one or more of the bins. There is energy, however, between the discrete 100 Hz locations of the bins. Another example can show us what happens to the energy that is not exactly at a bin frequency.

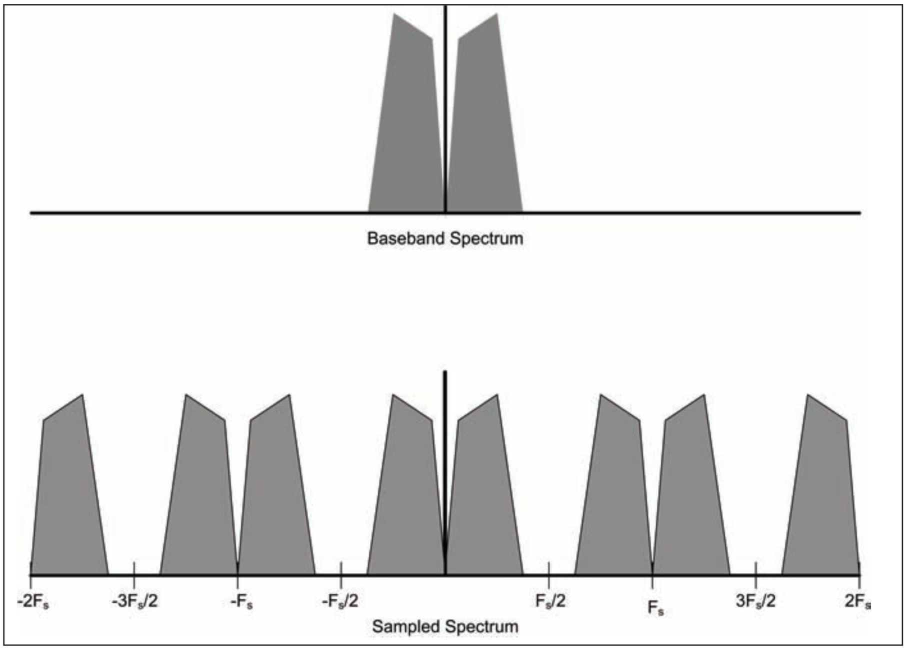


Figure 2 — The baseband spectrum of an analog signal before sampling is given at A, showing both positive and negative frequencies. Part B shows the spectrum of the baseband signal after it is sampled by an ideal sampling waveform. Note that there is an upper and lower sideband version of the baseband for each harmonic of the sampling frequency, just as if it were a double sideband suppressed carrier signal.

Let's set up a system with three crystal oscillators spaced 500 Hz apart. We choose frequencies of 100.0 kHz, 100.5 kHz, and 101.0 kHz with amplitudes of 1 V RMS.

Modern synthesized transceivers (like the ICOM IC-706) can be set up to look very much like a discrete spectrum analyzer. A discrete Fourier transform also implements a discrete spectrum analyzer. Let's set up the radio for CW with 100 Hz filter bandwidth. Let's also set up the tuning steps to 200 Hz and start tuning at 99.8 kHz. We will measure very little energy since the 100.0 kHz signal is quite far down on the filter skirt. The next step will tune the radio to exactly 100.0 kHz and we will measure the full strength of that signal. The next step will produce almost no energy. The next step tunes the radio to 100.4 kHz which puts the 100.5 kHz signal significantly up the slope of the filter skirt. The next step tunes the radio to 100.6 kHz, and we will measure the same value we read at 100.4 kHz due to energy on the other side of the filter skirt. Neither of those values will show the exact value that is at 100.5 kHz. We will once again get an exact value when the radio tunes to 101.0 kHz. The discrete tuning has created an output that is inexact for one of the signals in the input spectrum. It shows up at two frequencies and does not have an accurate amplitude. This is exactly what happens with a discrete Fourier transform if we do not use enough samples.

The energy that shows up in "wrong" bins is called spectral leakage. Spectral leakage is the frequency domain equivalent of quantization error in the time domain. Spectral leakage and quantization error are the math consequences of converting an infinite and continuous signal into a discrete signal with limited sample size and resolution. We'll look at quantization noise and other ADC and DAC errors in another column.

When we created a band pass filter in our AM radio experiment, we only used a 10 tap filter. Since the 10 taps come from doing a DFT of the filter response to an impulse, that means we only have filter bins every 500 kHz / 10, or 50 kHz per bin. That is not a very good filter. Since we mostly wanted to get rid of the dc component and energy at 180 kHz, the filter will do a moderate job. If we needed to resolve frequencies with better precision, we would need more taps to narrow the size of the bins. More taps means that we need to do a lot of math in each 2 microsecond interval. We will likely run out of CPU cycles very quickly. We only have to deal with energy over about 40 kHz in our AM radio example because it was filtered pretty well by our input filter. The signal is oversampled by a factor of 500 kHz / 40 kHz or 12.5 / 1. We could do all of the work we need and stay within the Nyquist criterion by sampling at 80 kHz if we manage the sample rate correctly. A lower sample rate would also allow us to have much smaller bins for our filtering.

Multirate Signal Processing

The sound of it is intimidating, but like everything we have looked at so far, it is not nearly as complicated as the mathematicians make it sound. The topic we are interested in now is sample-rate down conversion (decimation). Sample-rate up conversion is another multirate signal processing topic that we will cover when we look at transmitters.

If we want to operate directly at the 40 meter band for instance, we will need to sample above 14.6 MHz in order to satisfy the Nyquist criterion. We do not need a sample rate that fast to actually manipulate the information on receive, though, since the bandwidth of the widest signal will be on the order of 7 kHz or less. Even if we want to look at the entire band and generate a spectrum display, we only need about 650 kHz for the sample rate. Any higher sample rate on receive is a waste of processor resources.

There are two main reasons why we would want to match the sample rate closely to our intended bandwidth. The first reason to lower the sample rate is that the transition band and ripple of our filters are dependent on the ratio of the filter length (the number of taps) to the sample rate. The second reason is to allow more CPU cycles for processing each sample.

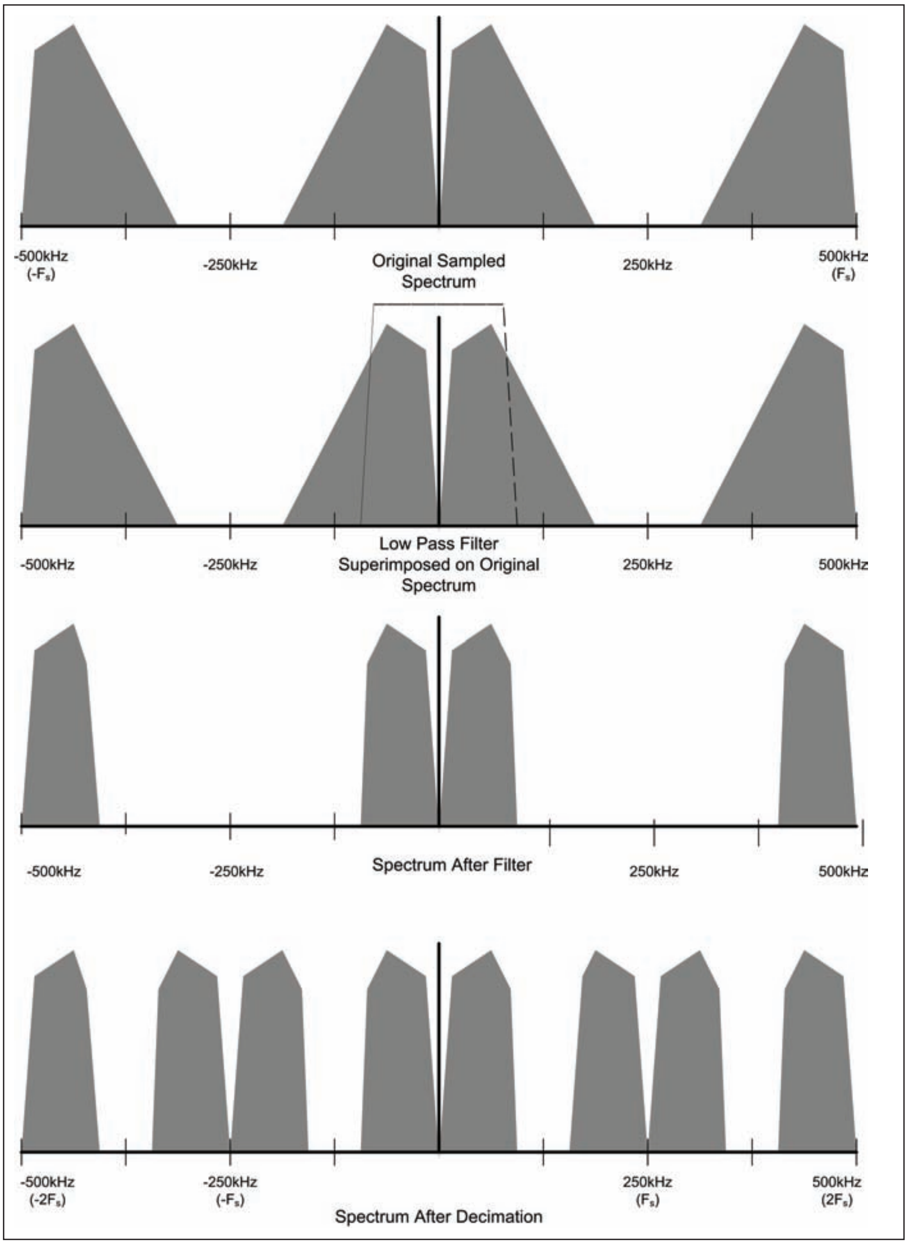


Figure 3 — Part A shows the spectrum of our sampled signal. Note that the frequency only extends from –500 kHz to +500 kHz since that is the extent that the math will manage. Part B shows the digital low pass filter response superimposed on the sampled spectrum. The Spectrum of the filtered signal, which is still sampled at 500 kHz, is shown at C. Part D shows the spectrum of the resulting sampled waveform after decimation by 2 (every other sample discarded). This is the spectrum of the signal when it is sampled at 250 kHz.

Sample-Rate Down Conversion

The super heterodyne version of our AM radio is a good candidate for sample rate conversion. Once we under sample the input and create the 90 kHz signal, the energy is all within the range of dc to 125 kHz. The input data is oversampled by 2 times the required amount. If it were possible to reduce the sample rate by 2, we would have two times as much CPU power to filter and demodulate the signal. This is low pass filter decimating.

The first step in the decimation process is to filter the input signal so that there is no energy above one fourth the sample frequency. A DSP low pass filter with cutoff at one fourth of the sample frequency has a very easy set of coefficients and can be implemented with a small number of taps. We sampled our AM radio at 500 kHz, so we need 125 kHz cutoff for the low pass filter. We now have a signal that has no energy above 125 kHz and is sampled at 500 kHz. We can throw away every other sample at this point to create a signal that is sampled at 250 kHz, with energy up to 125 kHz. The Nyquist criterion is satisfied with this new signal. Figure 3 shows the spectrum of the process.

The signal at 90 kHz is only about 40 kHz wide (70 kHz to 110 kHz), so it would be nice if we could drop the sample frequency even further. It turns out that our signal will fit into the band from dc to 62.5 kHz if it were translated down in frequency and would require a 125 kHz sample rate. This is integer band pass decimating. If we throw away every 3 samples of the original data, we can accomplish both frequency translation and sample rate reduction. The energy now spans from 15 kHz to 55 kHz. The signal spectrum is also inverted. In our AM radio case, we had band pass filtered the signal before sampling. It is possible to do the band pass filtering digitally with a small number of taps, and then do the decimation.

We succeeded in reducing the sample rate by a factor of 4. We cannot further reduce the sample rate using straight decimation. Each integer sample rate reduction requires that the band limited data fit completely within one of the Nyquist regions for the new sample rates. The next integer sample rate would reduce the sample rate by 5. The new sample rate would be 100 kHz. The data is contained in both the k = 1 (50 kHz to 100 kHz) and k = 2 (100 kHz to 150 kHz) Nyquist bands. Figure 4 shows the overlap. The overlap into the third Nyquist zone prevents further rate reduction. The requirement that all of the energy fits into one Nyquist zone limits the usefulness of this technique.

Integer sample rate reduction is a very useful tool because the filters are easily realized in hardware such as a field programmable gate array (FPGA) or other dedicated hardware. Texas Instruments (TI), Intersil (formerly Harris) and Analog Devices make dedicated ICs that do digital sample rate conversion using these techniques. My understanding is that the TAPR software defined radio project uses an FPGA for decimation. I recommend taking a look at the SDR resources on the TAPR site for additional information.

Next Issue

My plan for the next issue is to return to our work with the software on the Stamp board and do some more experiments. I should have sorted out the location of the necessary files and tools to be able to create *Linux* programs using a *Windows* system. We will also go through the process of loading a new *Linux* operating system on the Stamp, so you can take advantage of newer device drivers for our work.

Support Group

I get e-mail from time to time from folks with corrections, offers of help, and questions. You have seen the fruits of two contributors in this issue. For now I am putting folks into a group for my e-mail and sending out updates between issues as well as trying to keep the ARRL Web site up to date.

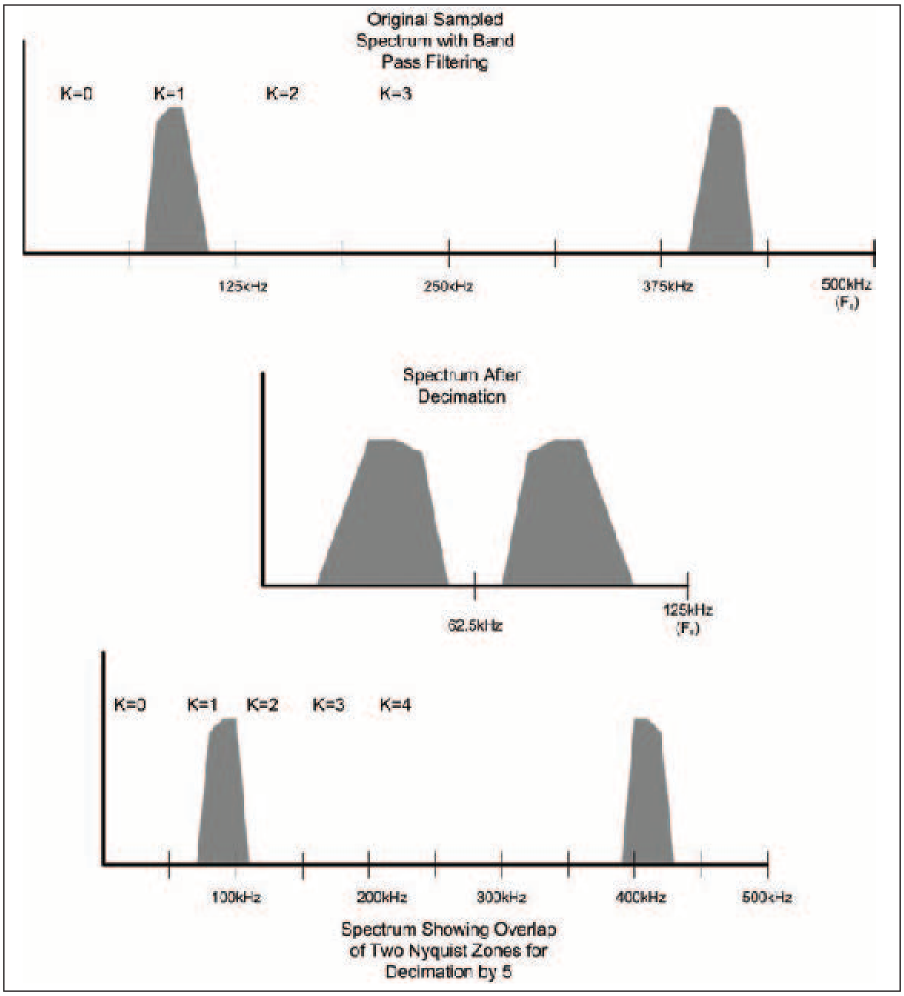


Figure 4 — Part A shows the spectrum of a 90 kHz signal that was band pass filtered and then sampled at 500 kHz. (only ½ of the spectrum is shown for clarity). The resulting spectrum when the signal is decimated by 4 (3 of every 4 samples discarded) is shown in Part B. Notice that this causes the signal to be aliased and the signal in the first Nyquist zone (k = 0) has its frequencies inverted just as a lower sideband signal is inverted. The signal is now sampled at 125 kHz. The initial sampled signal, showing the Nyquist zones that would occur if a decimation by 5 were attempted, is shown at C. Note that input energy exists in both the second and third Nyquist zones. This prevents proper decimation by 5.

Glossary

ADC (analog to digital converter) — An electronic circuit that converts an analog signal into a digital number with discrete values.

Bin — The DSP term for each of the discrete positions in a set of samples. Bin comes from the concept that the energy is stored in a discrete container.

CPU (central processing unit) — The part of a computer that does the actual math. Each operation is an instruction. The speed of the CPU is measured in millions of instructions per second (MIPS). It is normal for computers designed for digital signal processing to execute more than one instruction per clock pulse. Our Stamp board runs the CPU at approximately 400 MHz for the clock pulse rate, so we get at least 400 million instructions per second.

DAC (digital to analog converter) — An electronic circuit that converts a digital number into a corresponding voltage or current.

DFT (discrete Fourier transform) — The general case of a discrete transform from the time domain to the frequency domain.

DSP (digital signal processing) — The field of study in which signals are a sequence of data samples. Those samples are processed individually to perform operations such as filtering and conversion from time domain to frequency domain. The signals are not required to represent electrical signals.

FIR (finite impulse response) — A DSP operation that implements a transform on a sequence of samples with a fixed number of multiplications and additions. For example, an FIR transform with a length of 7 will do 7 multiplications and 7 additions on each input sample as it produces each output sample. Each output sample depends on only the last 7 input samples. There is no feedback from the output to the input.

FPGA (field programmable gate array) — An integrated circuit that can be programmed to glue multiple logic gates into new hardware functions.

FFT (fast Fourier transform) — A special case of the DFT. It requires a set of samples that is a power of 2 (2, 4, 8,… 1024, and so on). It takes advantage of the exact symmetry of the transform when all samples can be processed as pairs.

IIR (infinite impulse response) — A DSP operation that implements a transform on a sequence of samples using all of the preceding samples. Each output sample is presented to the input at the same time as each input sample. This feedback causes the output to be a function of all preceding input samples.

MAC (multiply/accumulate) — A special DSP instruction that implements a combined math operation that multiplies two numbers and adds the result of the multiplication to one of the original numbers.

SDR (software defined radio) — A radio where most (as close to 100% as possible) of the processing (filtering, gain, modulation, and demodulation) is done using digital hardware and software.

Notes

[1]Go to Michel Barbeau's Web site for a new Linux Kernel that includes SPI support: **www.scs.carleton.ca/~barbeau/SDR/**

[2]The Gerber circuit board files are available for download from the ARRL *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **11x09_Mack_SDR_Gerber.zip**

[3]The author had a limited quantity of the circuit boards made, and is offering them at his cost to interested readers. Contact him at the address listed at the beginning of this article for details.

[4]Crochiere and Rabiner, *Multirate Digital Signal Processing*, ISBN 0-13-605162-6. Available from Amazon, **www.amazon.com**

[5]Go to the Analog Devices Web site to get the *VisualDSP* manual: **www.analog.com/static/imported-files/software_manuals/50_blackfin_cc.rev5.1.pdf**

[6]The Analog Devices Web site includes additional manuals for the Blackfin processors: **www.analog.com/en/embedded-processing-dsp/blackfin/processors/manuals/resources/index.html**

[7]Downloading the manual for the Blackfin BF537 EZ-Kit Lite from the Analog Devices Web site: **www.analog.com/static/imported-files/eval_kit_manuals/ADSP-BF537_EZ-KIT_Lite_Manual_Rev_2_4.pdf**

QEX