$5

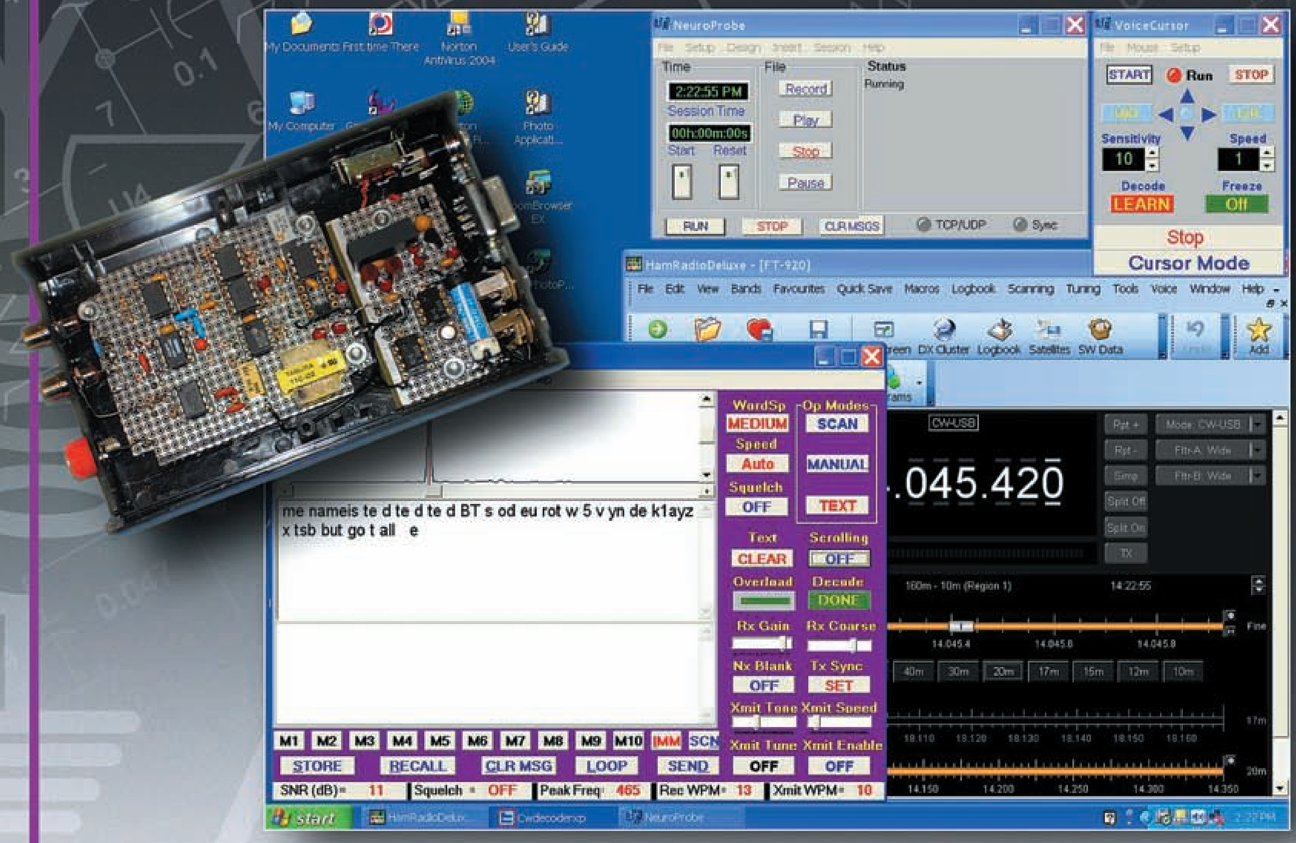
# QEX

November/December 2008

www.arrl.org

## A Forum for Communications Experimenters

Issue No. 251




**WD6CNF** presents an interface to use electromyography (EMG) signals produced by flexing his jaw muscles to move a computer mouse cursor, input text and operate a transceiver control program—even sending CW.

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

# TOKYO HY-POWER

# Two of the LIGHTEST and MOST COMPACT Amplifiers in the Industry!

## HL-1.5KFX

**HF/50MHz Linear Power**

### Features

- Solid State.
- The amplifier's decoder changes bands automatically with most ICOM, Kenwood, Yaesu.
- The amp utilizes an advanced 16 bit MPU (microprocessor) to run the various high speed protection circuits such as overdrive, high antenna SWR, DC overvoltage, band miss-set etc.
- Built in power supply.
- AC (200/220/235/240V) and (100/110/115/120V) selectable.
- Equipped with a control cable connection socket, for the HC-1.5KAT, auto antenna tuner by Tokyo Hy-Power Labs.

### Specifications

**Frequency:**
1.8 ~ 28MHz all amateur bands including WARC bands and 50MHz
**Mode:**
SSB, CW, RTTY
**RF Drive:**
85W typ. (100W max.)
**Output Power:**
HF 1kW PEP max.
50MHz 650W PEP max.
**Circuit:**
Class AB parallel push-pull
**Cooling Method:**
Forced Air Cooling
**AC Power:**
AC 240V default (200/220/235) – 10 A max.
AC 120V (100/110/115) – 20 A max.
**Dimensions:**
10.7 x 5.6 x 14.3 inches (WxHxD)/272 x 142 x 363 mm
**Weight:**
Approx. 20kgs. or 45.5lbs.
**Optional Items:**
Auto Antenna Tuner (HC-1.5KAT)
External Cooling Fan (HXT-1.5KF for high duty cycle RTTY)
**Accessories Included:**
Band Decoder Cables included for Kenwood, ICOM and some Yaesu


## HL-1.2KFX

**750W PEP Desktop Linear**

### Features

- Solid State.
- This world-class compact 750W HF amplifier is the easiest to handle and operate.
- The amplifier's broadband characteristics require no further tuning once the operating band is selected.
- The amplifier allows operation in full break-in CW mode due to the use of the amplifier's high speed antenna relays
- Quiet operation allows for even the weakest DX signals
- The amp utilizes a sophisticated circuit to run the various high speed protection circuits.

### Specifications

**Frequency:**
1.8 - 28MHz all amateur bands including WARC bands
**Mode:**
SSB, CW, RTTY
**RF Drive:**
75 - 90W
**Output Power:**
SSB 750W PEP max.,
CW 650W, RTTY 400W
**Circuit:**
Class AB parallel push-pull
**Cooling Method:**
Forced Air Cooling
**AC Power:**
1.4kVA max. when TX
AC 100/110/115/120V,
AC 200/220/230/240V
**Dimensions:**
9.1 x 5.6 x 14.3 inches (WxHxD)
**Weight:**
Approx. 33lbs.

## More Fine Products from TOKYO HY-POWER

**HC-1.5KAT**
**HF 1.5KW Auto Tuner**

**HL-350VDX**
**VHF 330W Amplifier**

**HC-200AT**
**HF/6m 200W Auto Tuner**
Lightning Tuning Speed

# TOKYO HY-POWER

TOKYO HY-POWER LABS., INC. – USA
Technical Support
28301 Tomball Parkway, Suite #500-210
Tomball, TX 77375
Phone 713-818-4544
e-mail: thpsupport@airmail.net

TOKYO HY-POWER LABS., INC. – JAPAN
1-1 Hatanaka 3chome, Niiza Saitama 352-0012
Phone: +81 (48) 481-1211 FAX: +81 (48) 479-6949
e-mail: info@thp.co.jp
**Web: http://www.thp.co.jp**

HAM RADIO OUTLET
WORLDWIDE DISTRIBUTION

*Exclusively from Ham Radio Outlet!*

**www.hamradio.com**

| | | |
|---|---|---|
| Western US/Canada 1-800-854-6046 | Southeast 1-800-444-7927 | Northeast 1-800-644-4476 |
| Mountain/Central 1-800-444-9476 | Mid-Atlantic 1-800-444-4799 | New England/Eastern Canada 1-800-444-0047 |

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 251

Harold Kramer, WJ1B
*Publisher*

Larry Wolfgang, WR1B
*Editor*

Lori Weinberg, KB1EIB
*Assistant Editor*

Zack Lau, W1VT
Ray Mack, W5IFS
*Contributing Editors*

**Production Department**

Steve Ford, WB8IMY
*Publications Manager*

Michelle Bloom, WB1ENT
*Production Supervisor*

Sue Fagan, KB1OKW
*Graphic Design Supervisor*

David Pingree, N1NAS
*Senior Technical Illustrator*

**Advertising Information Contact:**

Janet L. Rocco, W1JLR
*Business Services*
860-594-0203 – Direct
800-243-7768 – ARRL
860-594-4285 – Fax

**Circulation Department**

Cathy Stepina, *QEX Circulation*

**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.

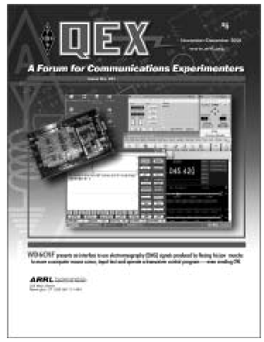
In order to ensure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.



**November/December 2008**

## About the Cover

Grant Connell, **WD6CNF**, presents an interface to use electromyography (EMG) signals produced by flexing his jaw muscles to move a computer mouse cursor, input text and operate a transceiver control program. His sytem involves generating Morse code characters for the computer control and text input, so he can even operate CW—all hands free!




**In This Issue**

# Features



## Index of Advertisers

## The American Radio Relay League

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interest in Amateur Radio communication and experimenta-tion, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of the radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every three years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur," ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A *bona fide* interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general corres-pondence should be addressed to the administrative headquarters:

ARRL
225 Main Street
Newington, CT 06111 USA
Telephone: 860-594-0200
FAX: 860-594-0259 (24-hour direct line)

**Officers**

***President:*** JOEL HARRISON, W5ZN
528 Miller Rd, Judsonia, AR 72081

***Chief Executive Officer:*** DAVID SUMNER, K1ZZ

**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 attended the 27th ARRL/TAPR Digital Communications Conference in Chicago on September 26-28. This was my second DCC, and it is definitely an event I want to attend in the future!

I met and talked with many *QEX* readers, authors and potential authors at the conference, and I enjoyed every minute of it. I even got to present a talk of my own this year: Writing for Publication — It's Not Rocket Science (Even if You are Writing About Rocket Science!) A number of attendees told me that I had encouraged them to write about a project they've been working on. Now if they follow through…

Of course, the real point of the DCC is all of the technical presentations, along with the projects on display in the Demonstration Room, or "play room," as it is called. TAPR had a D-STAR repeater system operating in the Demo Room, and Icom was also there with a display of D-STAR equipment for attendees to browse and learn about. (No equipment was for sale, though. The DCC is not a "Hamfest" in the typical sense of dealers and flea market sales.) Kenwood was also in attendance with a display of their TM-D710 VHF/UHF transceiver, along with the AvMap G5 GPS unit set up for APRS operation, and a few other radios. With the completion of the last pre-production Mercury direct-sampling receiver circuit board, TAPR was able to show a complete radio package for the High Performance Software Defined Radio (HPSDR) project. This setup included the "Mercury" board, the "Ozy" USB computer interface and the "Penelope" ½ W direct up-conversion transmitter all plugged in to the "Atlas" six-slot backplane.

Other "live" demos included a complete Winlink 2000 station in a box (or on a wagon for easy transport away from a car), an APRS demonstration station and DVDongle, a device that plugs into a USB port on your computer and allows you to access the D-STAR networks through their Internet gateways! There were other displays and equipment in the room, too.

Technical presentations at the DCC this year covered a range of topics. Friday and Saturday were full days of technical presentations, from 9 AM to 4 PM both days, with an hour lunch break. Steve Bible, N7HPR, gave an update on the SuitSat-2 project, Tom Clarke, K3IO, talked about other AMSAT programs. Scott Cowling, WA2DFI, talked about the current status of the HPSDR project, including the Mercury board. Victor Poor, W5SMM, gave us a Winlink 2000 update and Rick Muething, KN6KB, presented a much-anticipated talk about WINMOR — a Sound Card ARQ Mode for Winlink HF Digital Messaging. There were talks about various uses for the Amateur packet radio network as well as the APRS system.

In addition to all of the technical presentations, there was also a full day of "Introductory Sessions" on Saturday. Most of these had to do with various aspects of D-STAR networks and data modes.

D-STAR was a hot topic, with Friday evening presentations lasting from 7 until after 9 PM. Dan Smith, KK7DS described his D-RATS system for sending text and file transfers over the D-STAR system. Dan wrote this software in response to a need, expressed by his local ARES group, to transfer data during emergency operations. This became even more impressive when Dan showed how he could use his software to design a look-alike form for just about any served-agency document, then transfer the text information over D-STAR and "assemble" the document for printing at the receiving station. This must impress the officials at the served agency!

Pete Loveall, AE5PL, explained that Icom D-STAR radios can put GPS information into the digital data stream that is transmitted along with the digital voice. D-PRS is a program that converts the Icom GPS information into TNC2 formatted APRS strings, which can then be interfaced with the APRS packet network.

If you were unable to attend the DCC this year, you might want to mark your calendar now for the 2009 DCC. Plans are underway to return to the Holiday Inn Hotel, Elk Grove Village, Illinois, near Chicago's O'Hare International Airport for the 28th DCC on the weekend of September 25-27, 2009.

There are a number of other excellent technical conferences that are well worth your time. We list many of them in the Upcoming Conferences section of QEX, with calls for papers and information about dates and locations. Microwave Update, the AMSAT Space Symposium and the several VHF and VHF/UHF Society Conferences held around the country come immediately to mind. I hope you can attend one or more of these conferences in the next year.

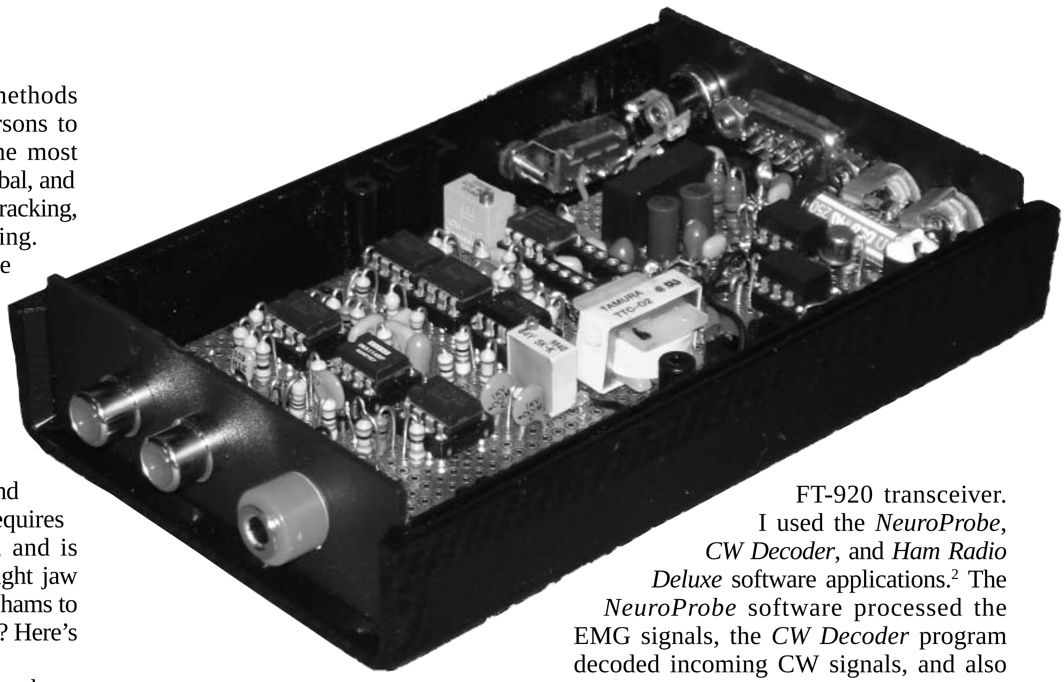
To share some of the "flavor" of the DCC with our readers who were unable to attend this year, we are reprinting a couple of the technical papers in this issue of *QEX*. From time to time we may also reprint some of the papers published in other Conference Proceedings. These articles make great reading, but they don't convey the excitement of actually being there, and hearing the presentation directly from the author.

**Grant Connell, WD6CNF**

1556 Elk Ravine Way, Roseville, CA 95661; wd6cnf@aol.com

# Using Morse Code for Computer and Ham Station Control

*A fast and efficient hands-off rig control system.*



There have been numerous methods developed that allow disabled persons to communicate with computers. The most popular of these focus on visual, verbal, and head oriented formats such as eye tracking, voice recognition, and head tracking. The hardware and software for these methodologies are sophisticated, and in many cases, expensive. Depending on the system design, communication with the computer can be inefficient, difficult, and tiring. The goal of this project is to overcome these limitations by developing a simple and efficient computer interface that requires minimal learning, is easy to use, and is inexpensive. Did you know that slight jaw movements can be used by disabled hams to control transceivers and operate CW? Here's how.

During the past several years, I have been using human electroencephalograph (EEG), commonly known as brain waves, to interface the brain to a computer (BCI — brain to computer interfaces). I have built several different types of interface units using the radio techniques of audio modulation and frequency modulation to convert these signals to a higher frequency band (such as 8192 Hz), usable by the computer via the sound card. A side benefit is that the amplitude modulated EEG units are able to pick up the higher frequency electromyography (EMG) signals. These signals are generated when we use muscles to perform a task. Based on some research at NASA AMES, I also investigated using vocal and sub-vocal EMG signals for speech recognition and computer control. During that investigation, I found that EMG signals from the jaw muscles proved to be reliable for moving the cursor on the screen.

The easiest encoding for cursor movement was to use Morse code.[1] Additional coding was needed, however, to allow someone to submit passwords and generate text quickly. In the past, you would normally generate text by moving the cursor over a "soft keyboard" and then transferring the text into a document. This turns out to be a slow process. With Morse encoding, the solution is simple — just switch between cursor movement commands and text generation. A non-Morse code sequence (dah dah dah dah) allows switching between the cursor and text input modes without being printed or affecting the cursor on the screen.

## Rig Control

My first attempt to control the rig used a sound card amplitude modulated EEG unit, an MFJ 1275 computer interface unit, and an FT-920 transceiver.

I used the *NeuroProbe*, *CW Decoder*, and *Ham Radio Deluxe* software applications.[2] The *NeuroProbe* software processed the EMG signals, the *CW Decoder* program decoded incoming CW signals, and also keyed the transmitter, while *Ham Radio Deluxe* tuned and controled the transceiver. While this combination of hardware and software worked successfully, the maximum Morse code speed was only around 10 words per minute. There was a noticeable delay between characters being sent, because the computer had to process the characters before sending them. While the characters were being sent, there was no processing of the EMG signals for the next character.

To improve the communication speed, I decided to build a "ham radio version" of the EMG cursor control unit. This version would pre-process the EMG signal and key the transmitter directly. I split the computer input between the audio signal from the radio and the EMG signal from the operator. I used the line input to accept both audio signals simultaneously. Processing of EMG commands and text can occur simultaneously while keying the transmitter.

[1]Notes appear on page 9.

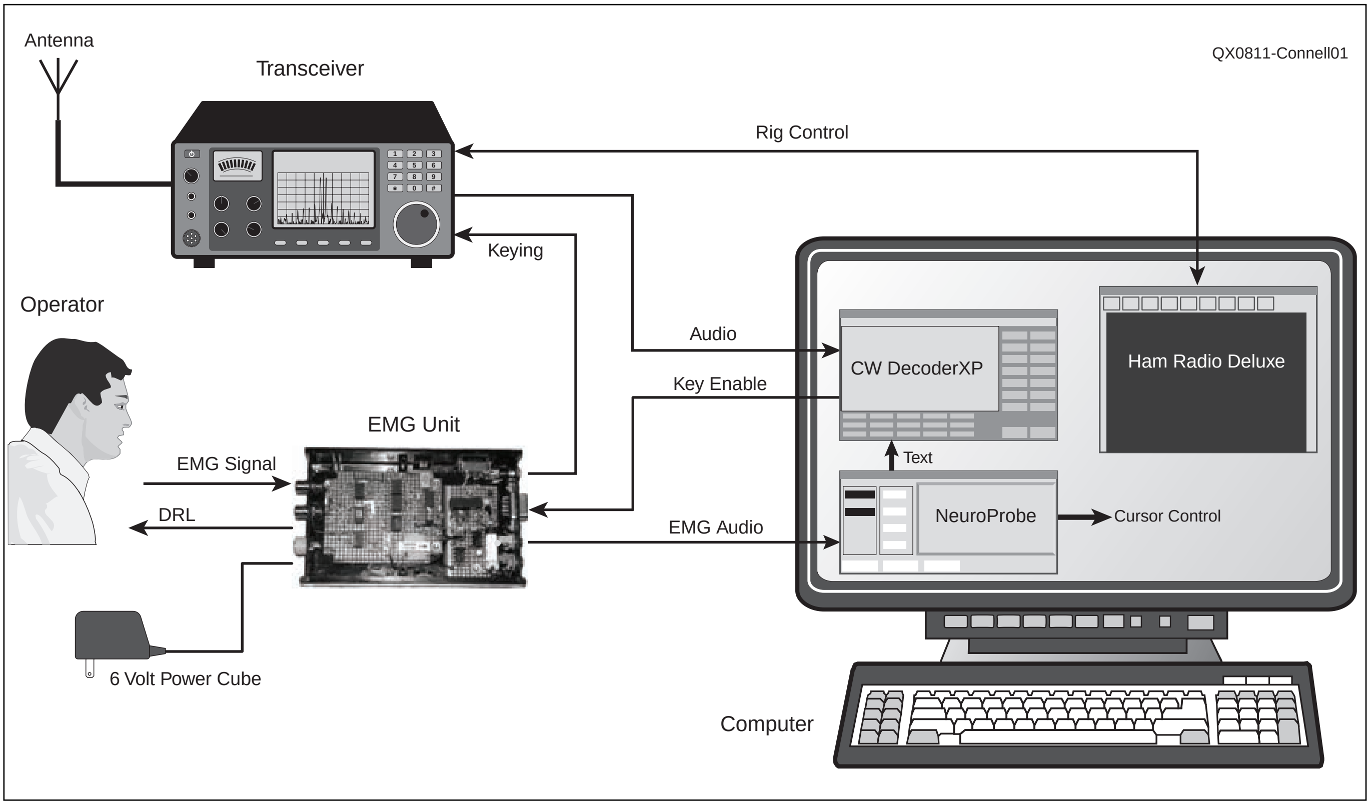
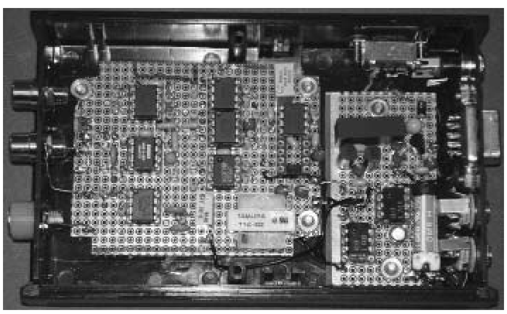



**Figure 1 — The EMG rig control system block diagram.**



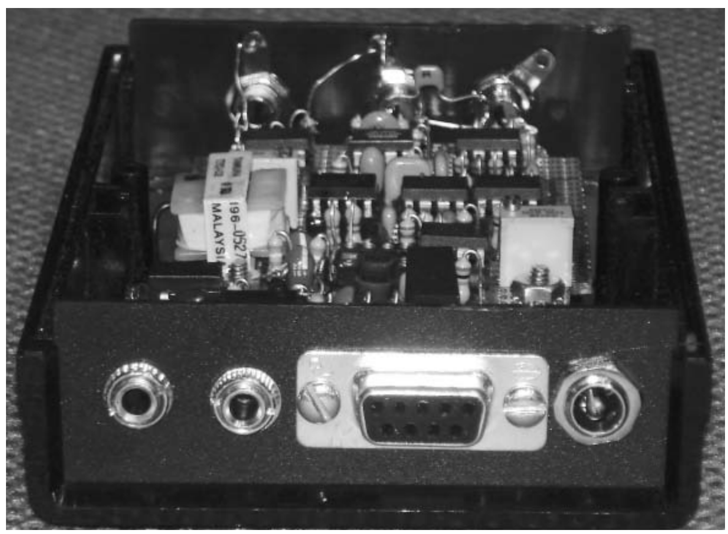


**Figure 2 — Part A shows the EMG unit hardware configuration. Notice the two circuit boards inside the box, as explained in the text. Part B shows the rear panel, with connectors. (The lead photo shows the front panel connectors.)**

This system used two RS-232 ports; one to control the radio and the other to enable keying. You can use USB-to-Serial converters if there are no serial ports available on your computer. A dc-to-dc converter isolates the power to the unit, while an opto-isolator interfaces the radio and computer. Figure 1 shows a block diagram of the system configuration.

## Hardware Design and Construction

The hardware for the EMG unit uses through-hole technology for ease of fabrication. Future versions could use surface mount to reduce the size of the unit. There are two printed circuit boards inside a 5¼ by 3¼ inch plastic enclosure. (See the lead photo.) The main circuit board contains the EMG signal processing and audio generation circuits, while the second board contains the dc-to-dc converter and transmitter keying circuits. The two boards are isolated to protect the operator from stray voltages. Figure 2A shows a picture of the circuit boards inside the box, and Part B shows the rear panel, with connectors. The interface to the operator is through RCA audio connectors and shielded cables, while the interface to the computer uses an opto-isolator-to-RS-232 converter for the key enablement, and transformer isolation for the EMG audio. The transmitter keying is isolated through a miniature relay.

I mounted the silver chloride (AgCl) electrodes (using strips of hook and loop fastener material) on a vertically oriented headband. The electrodes are located on each side of the jaw. Gold plated or silver plated electrodes will work as well. I use Spectra 360 electrode gel to ensure a good connection to the operator. The type of electrode gel is not critical for EMG signals. Shielded audio cables connect the electrodes to the EMG unit. Figure 3 is a picture of the electrodes and cables.

### Signal Flow

Figure 4 is a block diagram of how the EMG signal flows through the system. The EMG signal from the operator is amplified (gain of 40) by a low-noise differential instrumentation amplifier. The common mode signals on the electrodes are buffered, amplified, and fed back to the driven right leg (DRL) line. This reduces (through active feedback) external stray signals such as 60 Hz by about 30 dB. The output signal from the input amplifier is then high pass filtered at 70 Hz to remove the lower EEG and EKG signals. The signal is further amplified by a factor of 1000, and limited by diode clippers. This provides more dynamic range for the signal path. The amplified signal is then converted from ac to dc (rectified) and averaged. There are two averaging paths, one for the signal and one to generate a noise-riding threshold. The noise-riding threshold makes the unit more tolerant to external noise sources by adjusting the noise floor. The two averaged signals are then compared to each other and, depending on whether the signal is greater than the noise threshold, an audio oscillator is gated on. The audio oscillator is set to a 500 Hz (adjustable) tone and is buffered by an 1100 V isolation transformer. The transformer is a very important component in the system, because it protects the operator from stray line voltages within the computer. The threshold comparitor also drives an opto-isolator that gates a transistor and relay for keying of the transmitter. The transistor can be disabled by an opto-isolated signal from the computer. This prevents the transmitter from being keyed while the cursor is being moved on the screen.

An external 6 V power cube provides dc power to the unit. (It actually provides poorly regulated 7.5 V dc power.) This voltage is then regulated by an LDO regulator to 5 V dc. The 5 V dc is converted to ± 5 V for the +V and –V voltages by a dc-to-dc converter. RF filters limit the input EMI (radiated back to the power cube) and filters the output. Again, the dc-to-dc converter protects the operator from stray line voltages and is a very important design component. The output of the LDO regulator also drives the opto-isolators and the transistor relay driver.

Figure 5 shows the schematic diagram of the control unit.

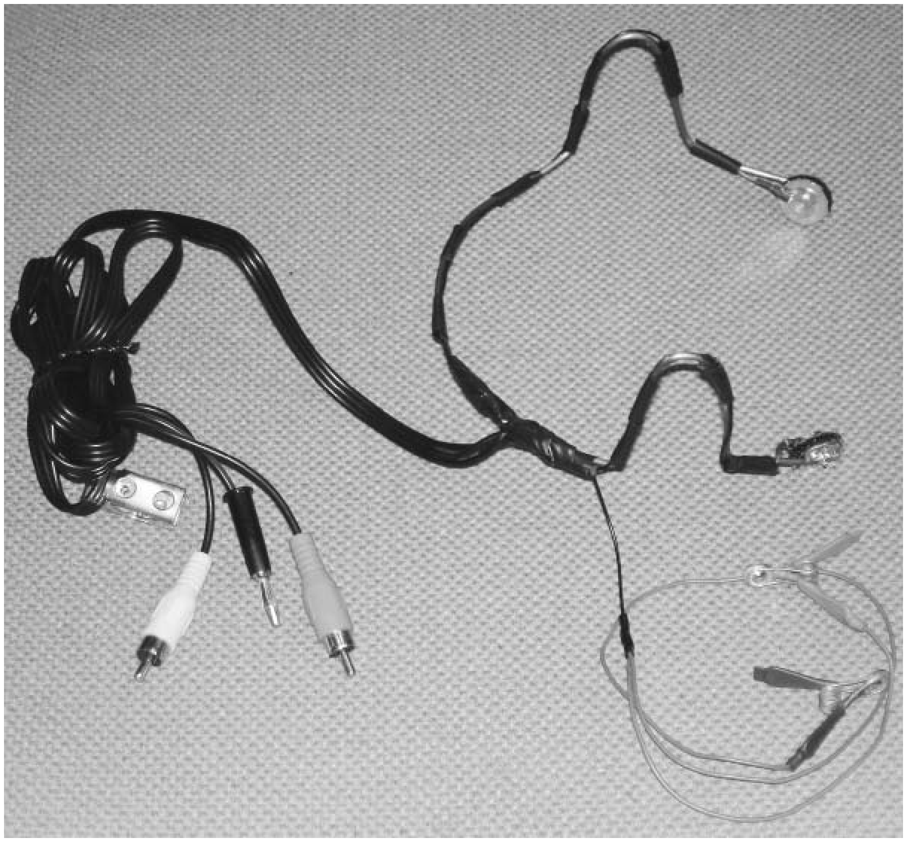


**Figure 3 — This photo shows the electrodes, which are held in place on the user's jaw using a headband and strips of hook-and-loop fastener material. The loops in the electrode frame fit over the user's ears to help hold the unit in position.**

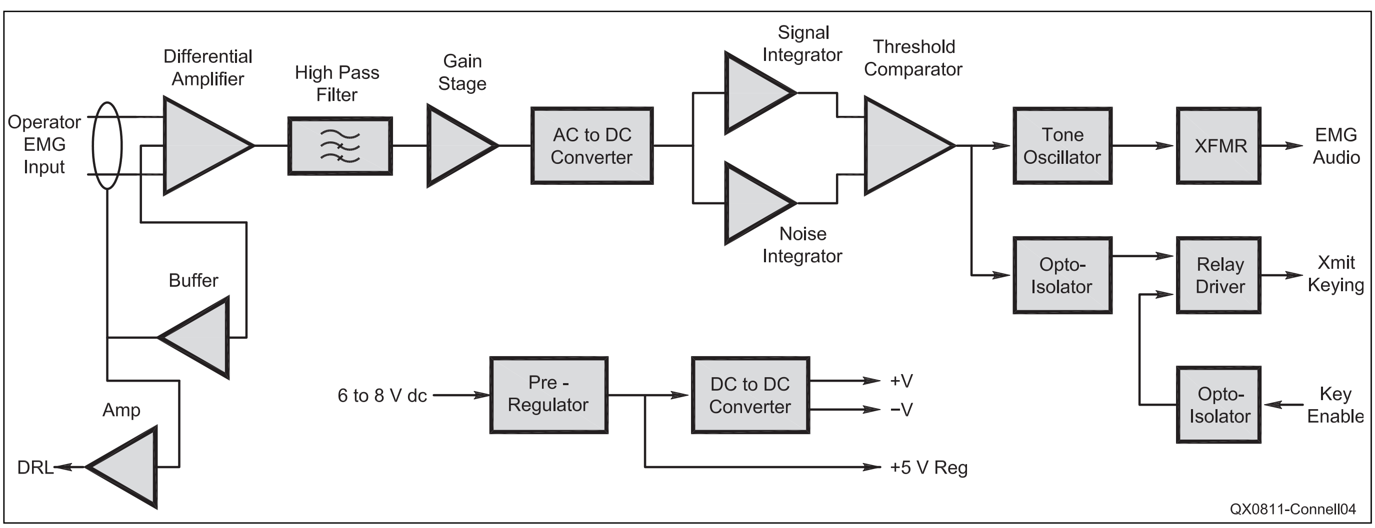



**Figure 4 — The EMG signal flow diagram.**

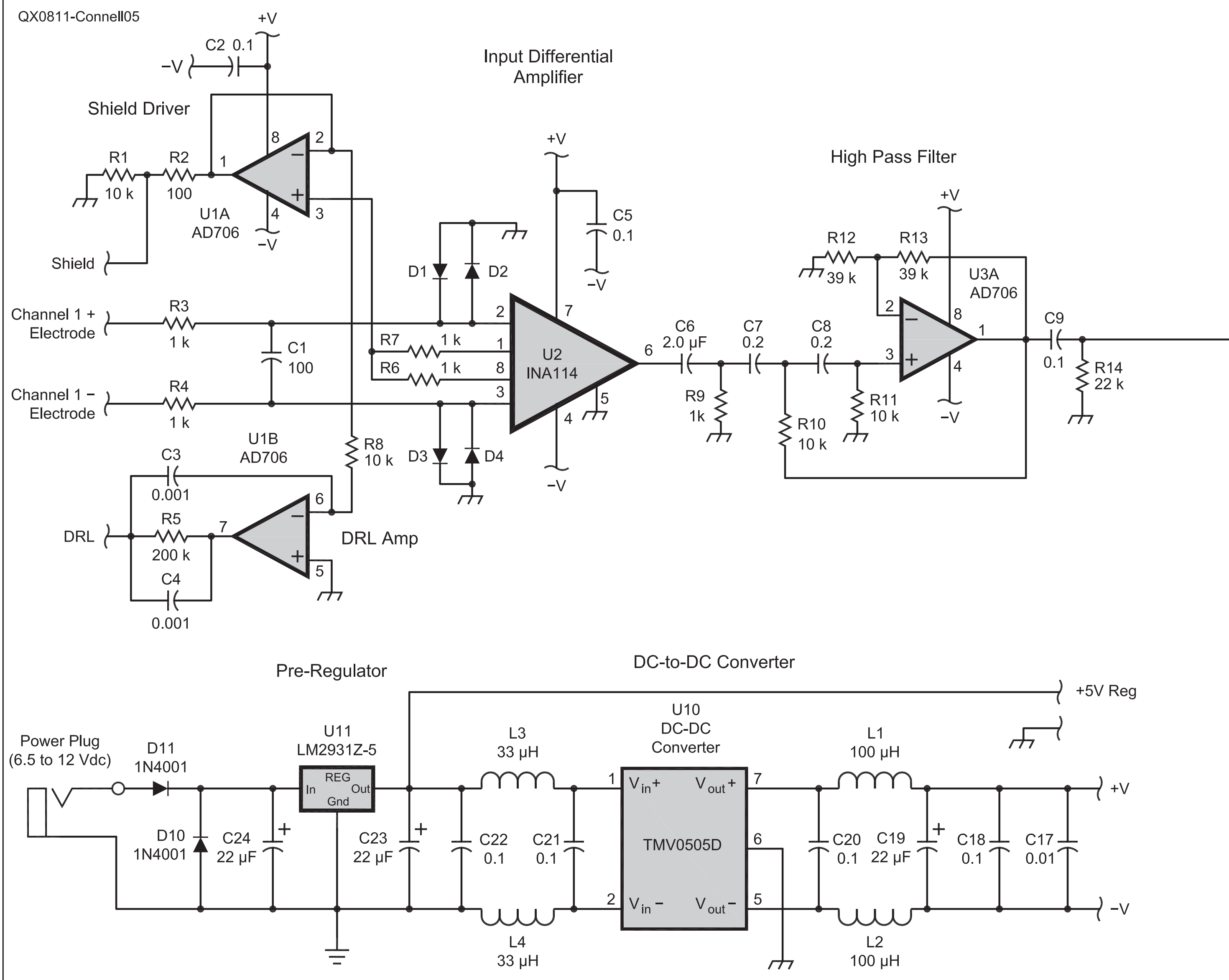
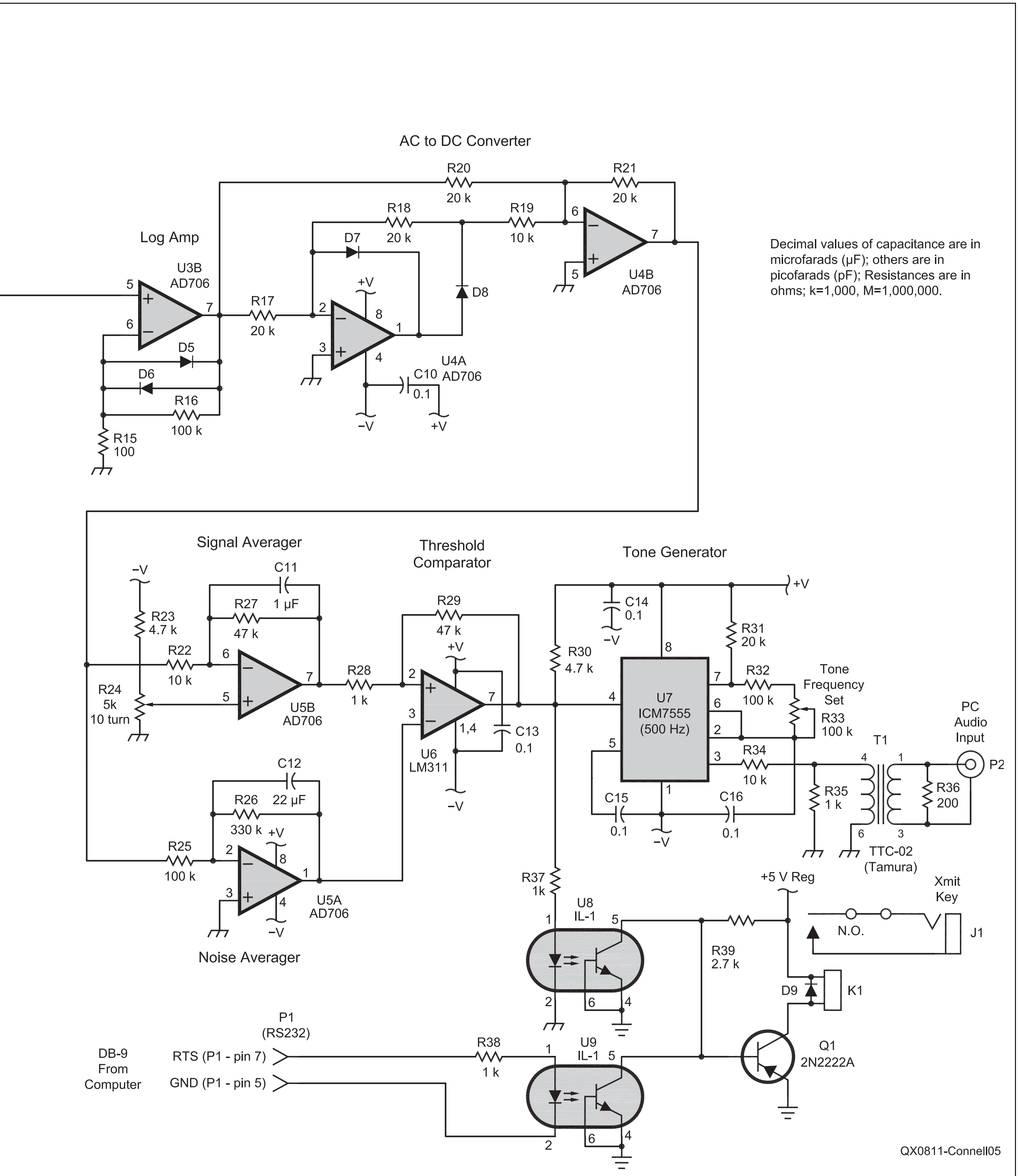



**Figure 5 — The schematic diagram of the Ham Radio EMG rig control system. (Parts List — See end of file.)**


AC to DC Converter
Log Amp
U3B
AD706
R15
100
R16
100 k
D5
D6
R17
20 k
D7
R18
20 k
D8
U4A
AD706
C10
0.1
R19
10 k
R20
20 k
R21
20 k
U4B
AD706
Decimal values of capacitance are in microfarads (μF); others are in picofarads (pF); Resistances are in ohms; k=1,000, M=1,000,000.
Signal Averager
R23
4.7 k
R22
10 k
R24
5k
10 turn
R27
47 k
C11
1 μF
U5B
AD706
Threshold Comparator
R28
1 k
R29
47 k
U6
LM311
C13
0.1
Noise Averager
R25
100 k
R26
330 k
C12
22 μF
U5A
AD706
Tone Generator
C14
0.1
R30
4.7 k
R31
20 k
U7
ICM7555
(500 Hz)
R32
100 k
Tone Frequency Set
R33
100 k
R34
10 k
C15
0.1
C16
0.1
R35
1 k
T1
TTC-02
(Tamura)
R36
200
PC Audio Input
P2
R37
1k
U8
IL-1
+5 V Reg
Xmit Key
N.O.
J1
R39
2.7 k
D9
K1
Q1
2N2222A
DB-9
From
Computer
P1
(RS232)
RTS (P1 - pin 7)
GND (P1 - pin 5)
R38
1 k
U9
IL-1
QX0811-Connell05

## Software Design

There are a minimum of three software applications used simultaneously to interface the operator to the transceiver. I use *Ham Radio Deluxe* for band selection and tuning the rig.[3] *NeuroProbe* processes the EMG signals, and *CW DecoderXP* enables keying and decodes incoming audio from the transceiver.[4, 5] All of the applications are set up on the screen so that they can be seen simultaneously, and are enabled (given focus) by moving the mouse cursor over them, and then clicking the mouse. You could also run other programs such as propagation and DX Cluster software, at the expense of complicating the screen switchover.

I modified the *NeuroProbe* application by adding an input element that processes tones when there is EMG activity. Previous software processed the EMG signals directly using the amplitude modulated EEG hardware. Because the sample rate was set at 1024 Hz, the maximum tone frequency is limited to 512 Hz. I created a design configuration that routes data from the EMG input element to the EMG Cursor output element, with 200 to 512 Hz bandpass filtering. A timeline screen is available to observe the input audio and the decoded waveform. I imported software from the *CW DecoderXP* application and modified it to allow for more delay between characters, to decode the EMG tone. Figure 6 shows the design configuration, and Figure 7 shows the EMG (and voice) cursor control screens of the *NeuroProbe* software.

The *CW DecoderXP* application did not require any modification but I configured it to operate in three different capacities. The received audio from the transceiver provides visual decoding of the received CW signal. This is convenient and prevents the operator from having to memorize call signs and messages. The output transmit screen is used to display the transmitted CW text and serves as storage for the output messages. I used the "Tune" button to enable the transmitter keying function. I disable the Tune button when the cursor is moving on the screen. The "Transmit Enable" remains in the off position.

The *Ham Radio Deluxe* application is sized down to show only a single band. I configured it for the FT-920, and the appropriate communications port. Figure 8 shows the three applications on the computer screen during operation.

### Morse Code Implementation

The EMG cursor software is designed to operate in one of two modes: a cursor control mode, and a text mode. The cursor control mode uses short Morse code sequences for the command codes. The shortest sequences are used to stop the cursor movement. In

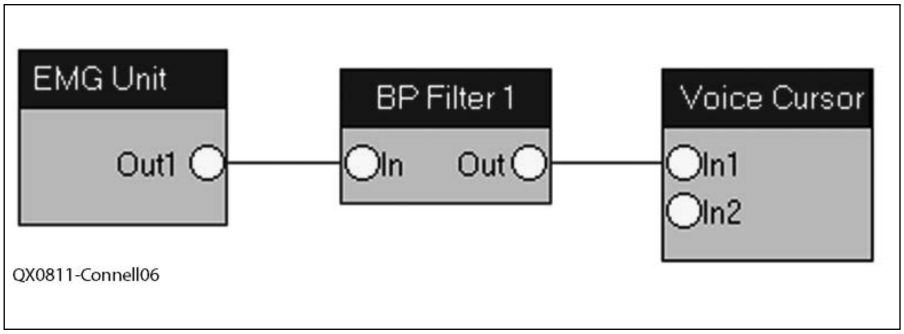



**Figure 6 — The EMG cursor design configuration screen from the *NeuroProbe* software.**


**Figure 7 — This is the Voice/EMG cursor screen portion of the *NeuroProbe* software.**

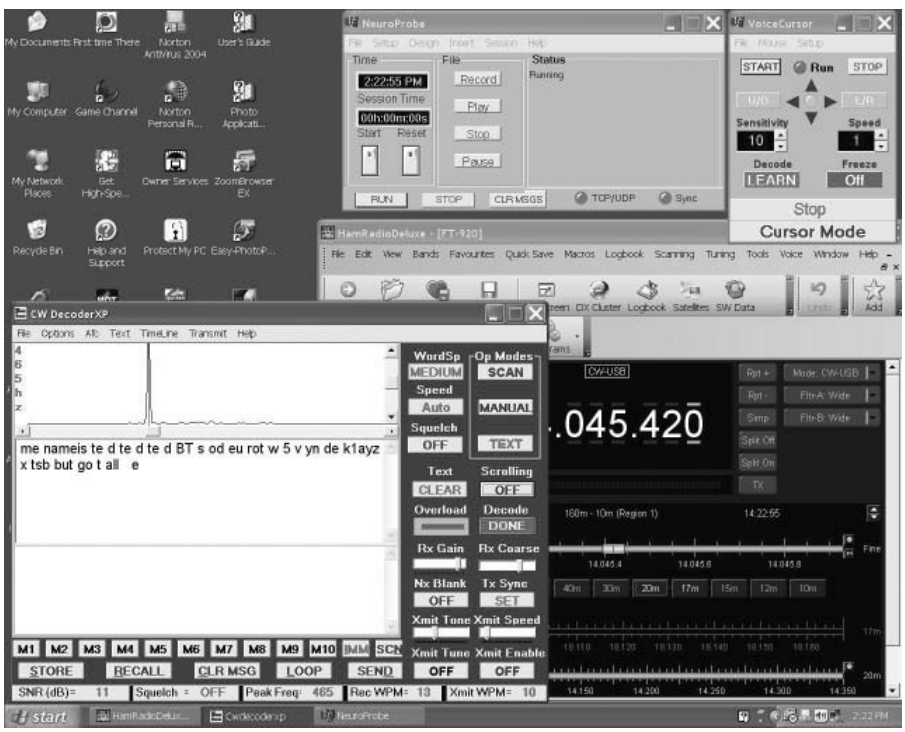



**Figure 8 — The various EMG cursor applications run simultaneously.**

the cursor mode of operation, the following codes are used:

1) Up command = di dit, letter "I"

2) Down command = dah dah, letter "M"

3) Left command = di dah, letter "A"

4) Right command = dah dit, letter "N"

5) Stop command = dit, letter "E" or dah, the letter "T"

6) Click command = di di dit, letter "S"

7) Double-click command = dah dah dah, letter "O"

8) Switch modes = dah dah dah dah, no printable letter, switching between cursor and text modes

9) Freeze command = di dah dit, letter"R", used to freeze the decoding for 3 seconds (to allow for swallowing)

In the text mode of operation, several text functions such as spaces, backspaces, and caps toggling are needed for efficient text input to other applications. The command codes for these functions are:

1) Space character = di di dah dah, non-printable Morse character

2) Caps toggling = di dah di dah, non-printable Morse character

3) Backspace = di di dah di dit, non-printable Morse character

4) Switch modes = dah dah dah dah, no printable letter, switching between cursor and text modes

To overcome the fast/slow cursor speed dilemma, I implemented a progressive speed control. When a command to move the cursor in a given direction is generated, the cursor speed is set to the slowest setting. A command to move the cursor in the same direction increments the cursor speed. This sequence can be repeated to increase the speed up to 7 times the slowest speed setting. A stop command resets the speed to the slow setting again. Close in cursor movement consists of a command sequence such as up — stop — down — stop — left — stop, and so on. This allows precise movement of the cursor to a target location.

## On the Air

To get on the air with the EMG cursor, the computer must be on and booted up, the *NeuroProbe* application must be running, the EMG unit must be on, and the transceiver must be on. The EMG unit takes about 20 seconds for the noise threshold level to settle. At this point the operator can bring up the *Ham Radio Deluxe* and *CW DecoderXP* applications. Tuning the radio and then switching to transmitting takes the most time because of the "transmit enable" function. Figure 9 shows the control system connected to my station equipment, ready to operate.

Once on frequency, the QSO is equivalent to normal hand key operation. I have operated with Morse code speeds of around 15 to 20 words per minute. With increased practice, higher code speeds are possible. I went through the same learning curve as when I first used a hand key. Prosigns are visible on the EMG cursor screen but not on the *CW DecoderXP* transmit screen. They are normally transmitted using the function keys. Since the jaw movement is very slight to produce the EMG signals, I can operate for several hours without tiring.

During a QSO there was an interesting exchange when I mentioned that I was using EMG signals for keying. The common response was that there was QSB on the air and that they did not copy everything. Operating CW with the EMG cursor requires skills similar to operating with a hand key. Good operating skills go a long way to making contacts.

One of the main problems I encountered was RF feedback. My antenna was located in the attic. The first time I keyed the transceiver, the system locked up in the transmit condition. The EMG electrodes and the operator have a high impedance connection. The operator acts like an antenna, and the input circuitry acts to rectify the RF. Adding an outside antenna and sitting next to a metal (grounded) desk solved the problem. The further you are away from the antenna, the less likely the RF energy will feed into the EMG unit. You also have to ensure that there is no RF feedback because of a high SWR on the feed lines running into the ham shack.

## Summary

Operating the EMG cursor is a bit like "flying by wire." You are performing all computer and radio control functions on the screen, without touching the mouse, the keyboard, or the radio. At first, this was uncomfortable since I'm still able to interact with those components. Eventually it became easy for me to rely on the applications for all of the needed functions. It would be even easier to get used to the system for those individuals who have limited capability to interact physically with their surroundings. With future software defined radios (SDRs), almost all of the radio functions will be controlled by the computer through the cursor and keyboard.

Although I built a "ham radio version" of the EMG unit, it still provides very good cursor control of the computer and fast, efficient text generation. The unit has proven very useful in that regard. In fact, the ham radio version could be used for just keying the radio, and that would leave both hands free for station control and logging operations.

The total cost for the project was about $200, including electrodes. My Web page (**www.hotamateurprograms.com/eeg.htm**) also has information on the sound card technology used in processing EEG and EMG signals. The bottom of the page has articles on eye tracking, voice recognition using EMG signals, and other topics.

### Notes

[1]Grant Connell, WD6CNF, "Cursor Control Using EMG Signals and Morse Code." This article can be found at **www.hotamateur-programs.com/eeg.htm**.

[2]*NeuroProbe* software can be found at **www.hotamateurprograms.com/eeg.htm**.

[3]*Ham Radio Deluxe* software can be found at **www.ham-radio-deluxe.com**.

[4]*CW DecoderXP* software can be found at **www.hotamaterprograms.com**.

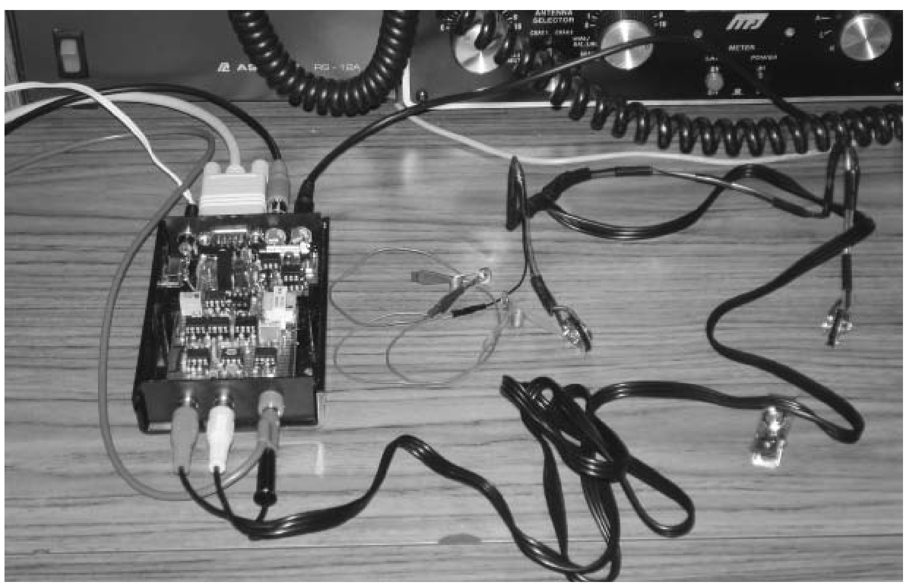


**Figure 9 — Here, the EMG rig control hardware is connected to my station, and ready to operate.**

## *Building Electrodes*

Good quality electrodes are important to receiving the low level EMG or EEG signals from the operator. Commercial electrodes are usually the disposable types with a sticky silver chloride surface, and they are expensive. You can make other electrodes from disks plated with gold, silver, or sliver chloride materials. You can obtain reasonably priced electrodes from Web based sites such as "The Electrode Store." The tricky part is in having a reliable mounting system, while keeping the price down.

I built two types of electrode configurations for this project; a headband with silver chloride electrodes and an ear mounted wire band with gold plated electrodes. Figure A shows the preferred location of the electrodes for the EMG signals to be picked up from the jaw muscles.

The headband configuration with the silver chloride electrodes used the center of the jaw for the DRL reference. The other two electrodes touched each side

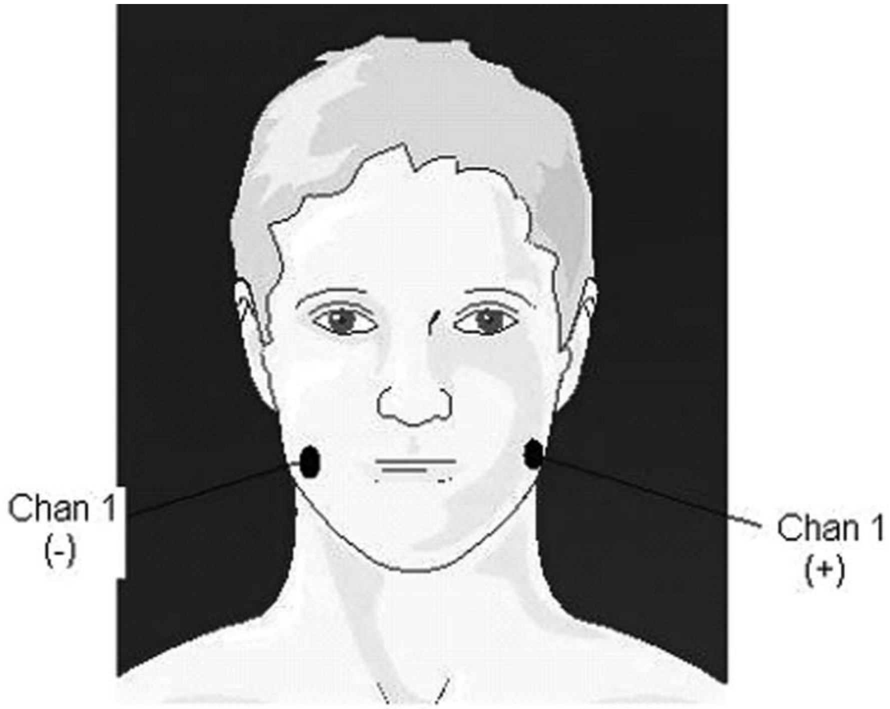



**Figure A — This drawing shows the location of the electrodes on the jaw.**

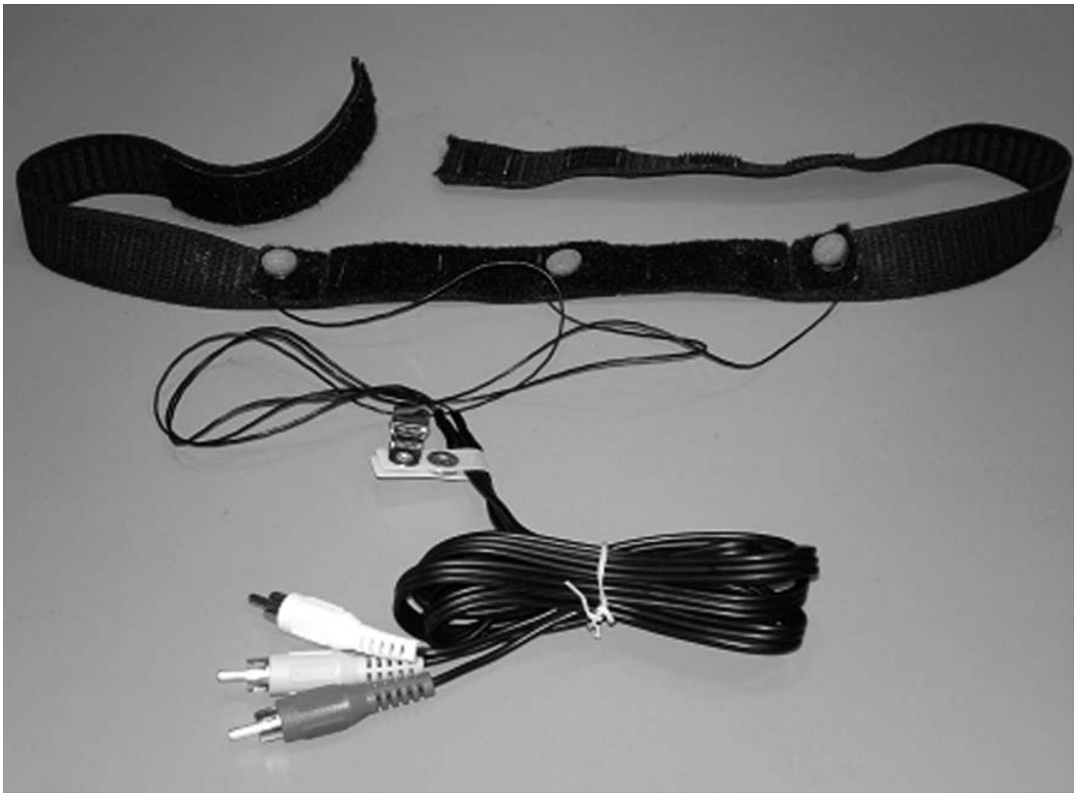


**Figure B — This is a photo of an EMG headband electrode configuration.**

of the jaw. I used a badge clip to hold the wires and prevent strain on the electrode wires. The electrodes are held to the headband with sticky-back Velcro strips. I also used Velcro at the ends of the headband to close the loop when it is mounted on the head. Figure B shows the completed headband assembly.

I fabricated the ear-mounted wire band from a coat hanger. I soldered a circular jewelry mount at each end of the cut and bent coat hanger. I then soldered the gold plated electrodes to the jewelry mounts. The wires from the electrodes connect to the audio cable, and are taped to the wire frame. The DRL line connects to the ear clips. Figures C, D, E, and F show the different stages in the wire band construction.

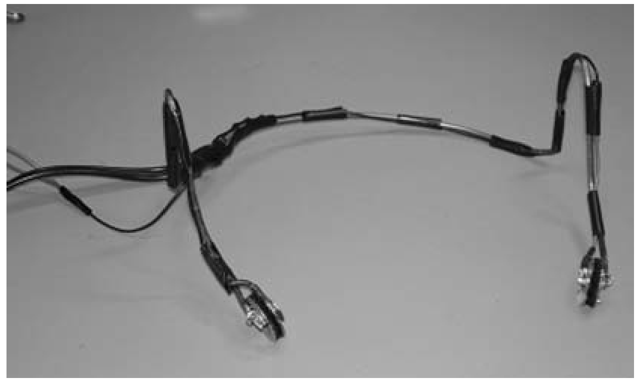


**Figure E — Here, the electrodes are mounted on the ends of the wire-band.**

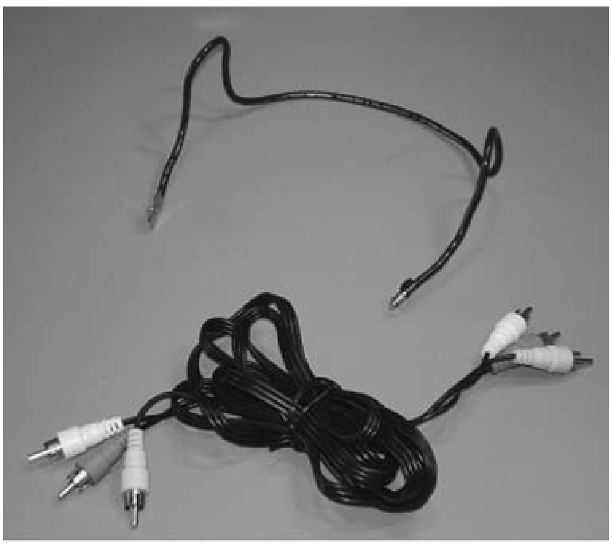


**Figure C — This photo shows the wire band and audio cable.**

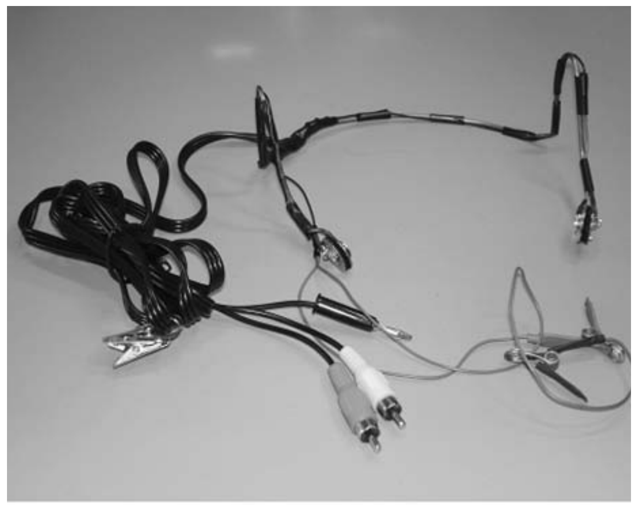


**Figure F — Here is a photo of the completed wire band electrode configuration.**

**Figure D — This photo shows the wire band ends.**

*Grant G. Connell, WD6CNF, is a communications Systems Engineer at Northrop Grumman in Sacramento, California. An ARRL member, he has been a licensed ham since 1959. Grant holds an MSEE from San Jose State University. He has been interested in software for the amateur radio operator and has written many programs that facilitate the operator's station, including CW decoding. Recently he has been interested in hardware and software that interfaces the computer to the brain (brain-to-computer interface, BCI) and related research. Related articles, hardware designs, and software can be found at **www.hotamateurprograms.com** and at **www.hotamateurprograms.com/eeg.htm**.*

**Parts List**

| *Quantity* | *Value* | *Tolerance* | *Voltage Rating* | *Package* | *Part* | *MFG/Comments* |
|---|---|---|---|---|---|---|
| *Integrated Circuits* | | | | | | |
| 4 | AD706 | NA | 10 V | DIL-08 | U1, U3 ,U4, U5 | Analog Devices |
| 1 | INA114 | NA | 10 V | DIL-08 | U2 | Analog Devices |
| 1 | LM311 | NA | 5-15 V | DIL-08 | U6 | Various Vendors |
| 2 | IL-1 | NA | | DIL-08 | U8, U9 | Various Vendors |
| 1 | ICM7555 | NA | 10 V | DIL-08 | U7 | Various Vendors |
| 1 | TMV0505D | NA | 5 V | CUSTOM | U10 | TRACO Electronics AG |
| 1 | LM2931Z-5 | NA | 20 V | TO-92 | U11 | National Semiconductor |
| *Discrete Semiconductors* | | | | | | |
| 12 | 1N4148 | NA | 100 V | DO-35 | D1-D8 | Various Vendors |
| 2 | 1N4001 | NA | 1 kV | DO-35 | D10, D11 | Various Vendors |
| 1 | 2N2222A | NA | 15 V | TO-92 | Q1 | Various Vendors |
| *Resisitors* | | | | | | |
| 1 | 100 Ω | 5% | | R-10 | R2, R15 | |
| 1 | 200 Ω | 5% | | R-10 | R36 | |
| 2 | 1 k Ω | 1% | | R-10 | R6, R7 | |
| 7 | 1 k Ω | 5% | | R-10 | R3, R4, R9, R28, R35, R37, R38 | |
| 1 | 2.7 k Ω | 5% | | R-10 | R39 | |
| 2 | 4.7 k Ω | 5% | | R-10 | R23, R30 | |
| 7 | 10 k Ω | 5% | | R-10 | R1, R8, R10, R11, R19, R22, R34 | |
| 5 | 20 k Ω | 5% | | R-10 | R17, R18, R20, R21, R31 | |
| 2 | 22k Ω | 5% | | R-10 | R14 | |
| 2 | 39 k Ω | 5% | | R-10 | R12, R13 | |
| 2 | 47 k Ω | 5% | | R-10 | R27, R29 | |
| 3 | 100 k Ω | 5% | | R-10 | R16, R25, R32 | |
| 1 | 200 k Ω | 5% | | R-10 | R5 | |
| 1 | 330 k Ω | 5% | | R-10 | R26 | |
| 1 | 5 k Ω | NA | | | R24 | 10 Turn Potentiometer |
| 1 | 100 k Ω | NA | | | R33 | 10 Turn Potentiometer |
| *Capacitors* | | | | | | |
| 1 | 100 pF | 10% | 100 V | C-5 | C1 | Ceramic |
| 2 | 0.001 µF | 10% | 100 V | C-5 | C3, C4 | Ceramic |
| 1 | 0.01 µF | 10% | 100 V | C-5 | C17 | |
| 2 | 0.02 µF | 10% | 50 V | C-5 | C7, C8 | Ceramic |
| 1 | 0 0.1 µF | 20% | 50 V | C-5 | C2, C5, C9, C10, C13, C14, C16, C18, C20, C21, C22 | Ceramic |
| 1 | 2.0 µF | 20% | 15 V | EL25B | C6 | Tantalum |
| 3 | 22 µF | 20% | 15 V | EL25B | C19, C23, C24 | Tantalum |
| *Inductors* | | | | | | |
| 2 | 100 µH | 10% | | | L1, L2 | |
| 2 | 33 µH | 10% | | | L3, L4 | |
| *Miscellaneous Parts* | | | | | | |
| 1 | Conn | | | P1 | 9-Pin RS-232 Conn. | |
| 2 | Jack | | | J1, P2 | Mono, miniature phone jack | |
| 1 | Jack | | | J2 | Power Connector | |
| 1 | TTC-02 | | | T1 | (Tamura) 1100 V isolation transformer | |
| 9 | 8-Pin IC Socket | | | | | |
| 1 | Relay | | | K1 | 5 V dc Relay Radio Shack, 275-232 | |
| 1 | PCB | | | | PCB with ground plane | |
| 1 | Enclosure | Plastic enclosure, 3.3 × 5.25 × 1.25 inch PAC-TEC, Model CM3-100 | | | | |
| 2 | RCA Phono Female jacks for electrodes | | | | | |
| 1 | Ground Plug For DRL | | | | | |
| 2 | Electrodes | AgCl | | | | |
| 1 | Cable | Stereo Cable with RCA Male Phono Plugs | | | | |

Note:
All resistors are metal film, 1/4 W, to reduce noise.

**Dr. Sam Green, WØPCE**

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

# Fully Automated DDS Sweep Generator Measurement System

*A DDS kit, a few simple circuits, and the author's software produce a 100 Hz to 30 MHz or 60 MHz sweep measurement system.*

The New Jersey QRP Club's Direct Digital Synthesis Daughtercard Kit provides direct digital synthesis on a 1 × 2 inch pluggable card.[1] See Figure 1. The new version synthesizes 0-60 MHz, and the original version that I have synthesizes 0-30 MHz. Several popular projects based on a variety of microprocessors incorporate the NJQRP DDS daughtercards, and there is free downloadable software available online to drive the DDS as a variable frequency oscillator or as a sweep generator when interfaced to a computer parallel port, as shown in Figure 2.

I wanted to use the DDS as a sweep generator but was disappointed to find that the available sweep generator application sweeps both up and down, and then repeats, while providing no way to display a stable, observable waveform on an oscilloscope. I e-mailed the author asking for source code so I could modify his application, but got no response.

This article describes the changes I made and the features I added once I learned how to program the DDS from a computer parallel port.

I added a discrete output to trigger an oscilloscope sweep at the beginning of a frequency sweep, a digital-to-analog converter (DAC) to synthesize a horizontal sweep, an analog-to-digital converter (ADC) to measure response at each frequency and save the results to a text file, and a logarithmic detector to measure RF signal levels over a wide dynamic range. Unused pins of the same parallel port that controls the DDS control the DAC and the ADC. How cool is that?

Figure 3 shows the connections between the socket for the NJQRP DDS and a local DB-25M connector in this project that attaches by cable to the DB-25F parallel (printer) port connection on your computer. I maintain the same DDS connections to the

[1]Notes appear on page 22.

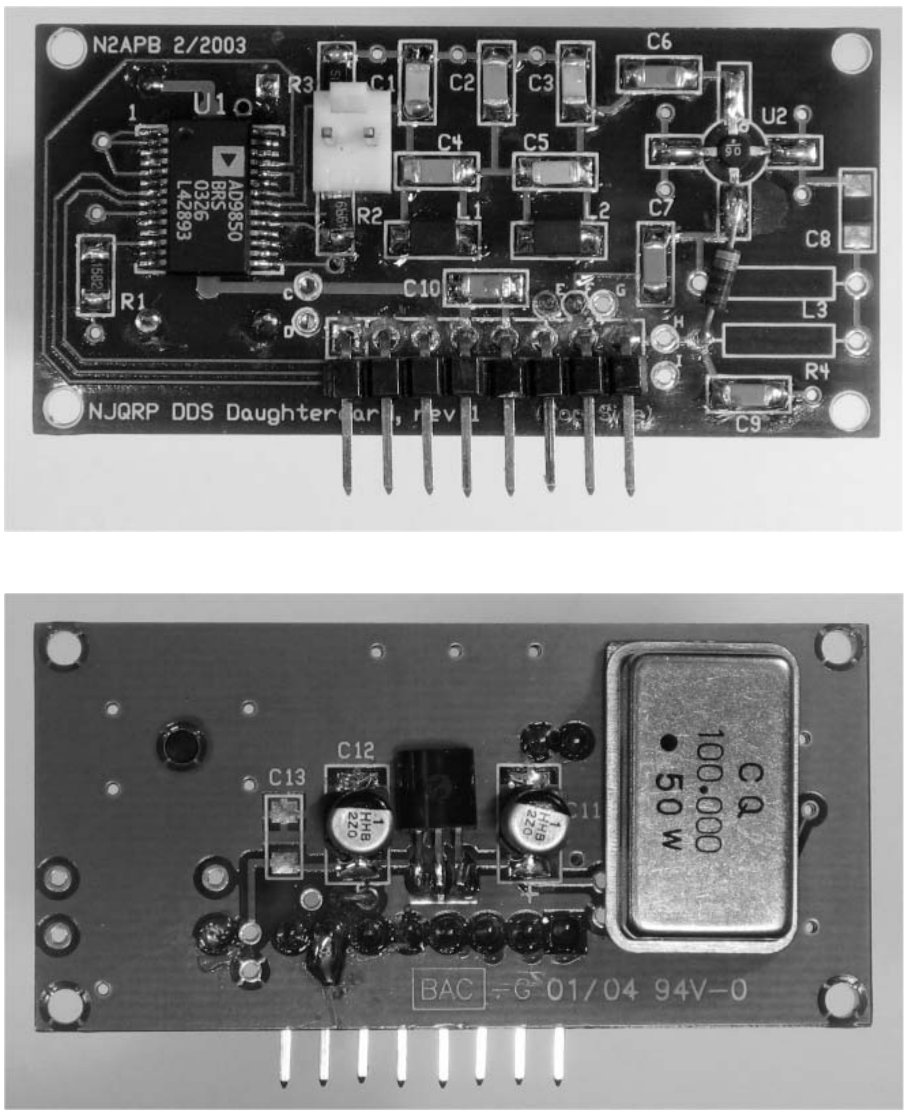



**Figure 1 — These photos show both sides of The NJQRP Club DDS daughtercard. Note the modification to derive a direct output for low frequency operation that bypasses the amplifier.**

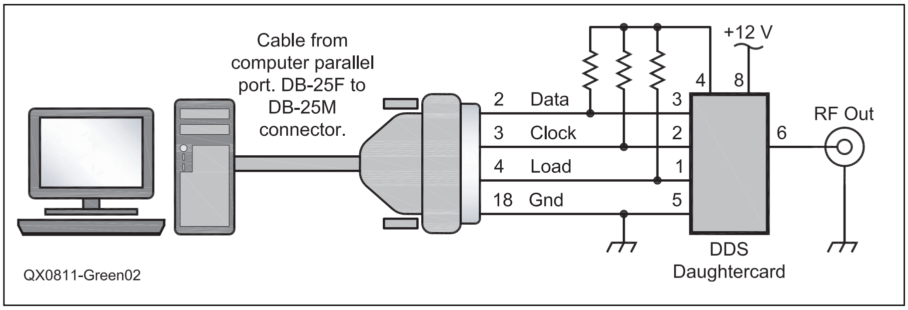



**Figure 2 — This diagram shows how the DDS connects to a computer parallel port.**

```
void write40bit2d0 (int freqword, int phaseword) {
        // with clock to d1 & frequency update strobe to d2
    int freqvar, i, writebit;

    freqvar = freqword ;
    for(i=1; i<33; i++) {
        writebit = freqvar & 1 ;
        _outp(base, writebit);          // write bit
        _outp(base, writebit | 2);      // writeclock high
        _outp(base, writebit);          // writeclock low
        freqvar >>= 1 ;
        }
    freqvar = phaseword ;               // all zeroes
    for(i=1; i<9; i++) {
        writebit = freqvar & 1 ;
        _outp(base, writebit);          // write bit
        _outp(base, writebit | 2);      // writeclock high
        _outp(base, writebit);          // writeclock low
        freqvar >>= 1 ;
        }
    _outp(base, 4);    // frequency update stobe pulse high;
    _outp(base, 0);    // frequency update stobe pulse low;
    }
```

**Figure 4 — This subroutine writes frequency and phase words to the DDS.**

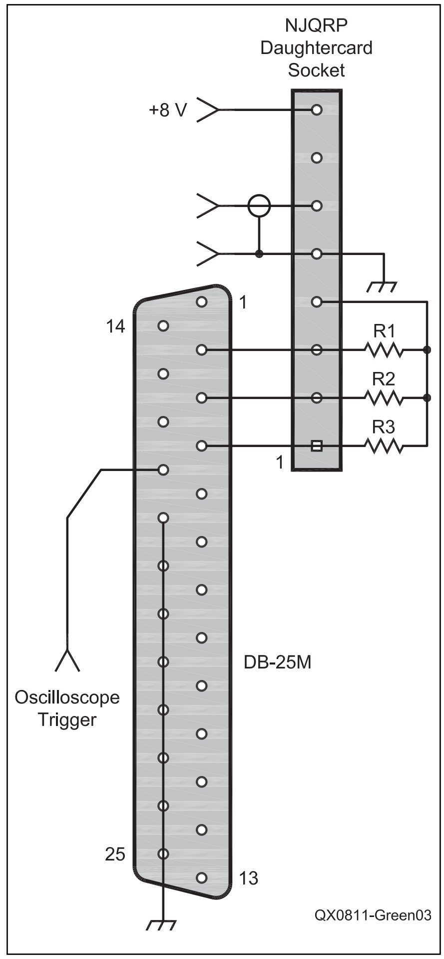



**Figure 3 — This pictorial layout shows the computer parallel port connections to the NJQRP Club DDS daughtercard.**

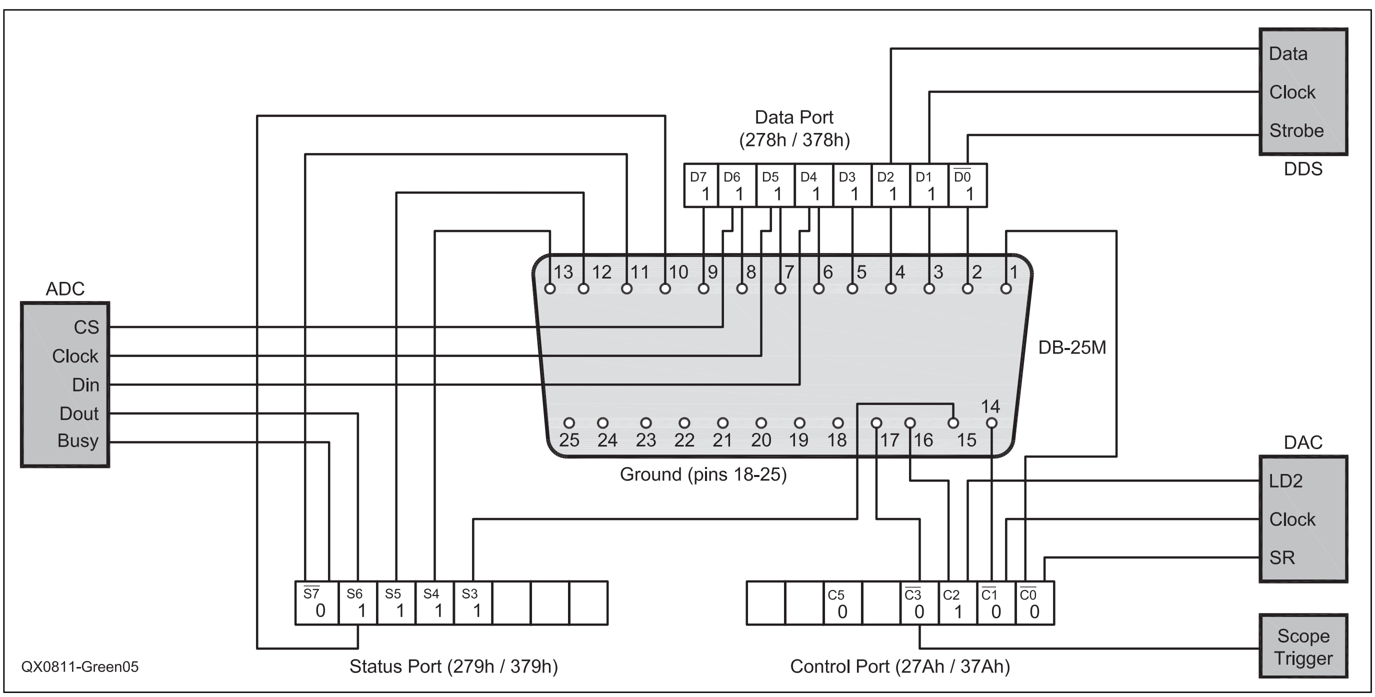



**Figure 5 — This diagram from the Parallel Port Viewer at www.soft-collection.com/images/ParallelPortViever.gif shows how the additional functions of the sweep generator system use the remaining pins on the parallel port.**

parallel port so that the original NJQRP software still works.

I chose an MX7543 for the DAC and a MAX110 for the ADC because they both communicate serially and thus each only requires a few of the remaining pins on the same parallel port as the DDS, and also because MAXIM sent me free evaluation samples.[2, 3] This is an honored tradition. The NJQRP Club tells you to obtain a free sample of the DDS chip from Analog Devices and does not include it in the daughtercard kit. I also honored this tradition in obtaining the Analog Devices logarithmic detector and the Texas Instruments reference diodes. TI no longer provides the free sample reference diodes, but Linear Technology does provide equivalents. I offer my thanks to the vendors for these samples.

This project provides a complete sweep-frequency measurement system that operates very well from 100 kHz to 30 MHz with only the DDS daughtercard, these three chips, a dual op-amp, the voltage reference diodes, and some voltage regulators. The new DDS daughtercard will operate to 60 MHz with appropriate adjustments to the data-word calculation.

I put an AD8307 logarithmic detector in the box so it could connect to one input of the ADC.[4] This device is so sensitive to pickup from the DDS that it might be better to package it separately. The AD8307 has an enormous dynamic range to provide a display that reaches orders of magnitude below what you can see with an ordinary diode detector.

Both the DDS and the logarithmic detector are direct coupled, so it is possible to work through the audio range. I will discuss difficulties that appear at low frequencies later.

You can download sample software programs from the ARRL QEX Web site.[5] I program in C in the Microsoft Visual Studio 6.0 environment, but you can translate my source code to your environment as needed. If you do not program, you can run my executable files from a command line in a DOS window, but you do not get to change anything. You will be stuck with the parallel port at 0x378, the 30 MHz DDS, and the calibration constants for the specific components in my prototype. To get to a DOS window in *Windows*, click on "Start" and then "Run" and then type "cmd" in the text box that appears. Finally, hit "Enter."

I expect a real programmer will step up and generate a fully general version of this prototype software with a *Windows* Graphical User Interface to replace my simple console applications.

*Windows* does not allow executable application programs running in a DOS window to alter the I/O ports, so install UserPort to enable these programs to access the parallel port.[6] Download and unzip UserPort. Copy UserPort.sys to C:\WINDOWS\system32\drivers. To run UserPort.exe, double click on its icon and then click "Start" in the UserPort dialog box. Now, all of these programs reach the parallel port.

## One Piece at a Time

I followed the programming instructions in the data sheet for the Analog Devices AD9850. The heart of the program is the subroutine shown in Figure 4. This routine writes 40 data bits sequentially to pin 2 of the parallel port (DØ) while toggling the clock pin 3 (D1) high and low between data bits, and finally loads the data into the frequency/phase data register by toggling the load pin (D2) high and low. The rest of the program decides the next frequency and computes the value of the variable freqword that constitutes the sequential data bit stream.

I built and debugged prototypes of the DAC and the ADC separately using a solderless breadboard before trying to make them work with the DDS. There is nothing critical here. The clock and data lines of the DAC operate just like the DDS, so the software routines are similar. The ADC is more complex because it has a second data line, a chip select, and a busy output to indicate that data is ready to read, but the principles are the same. Once individual programs work separately, you must take care in programming so that you do not mess up one device while attempting to write to another. Follow the examples in the downloadable source code that I provide.

Figure 5, adapted from **www.softcollection.com/images/ParallelPortViever.gif** shows the allocation of pins on the parallel port to each device. A parallel port has status and control registers at consecutive addresses following the data register that provide more inputs and outputs at the DB-25F connector on your computer. I use six of eight general-purpose pins on the data port that can be inputs or outputs. I use two of five input-only pins read by the status port, and I use

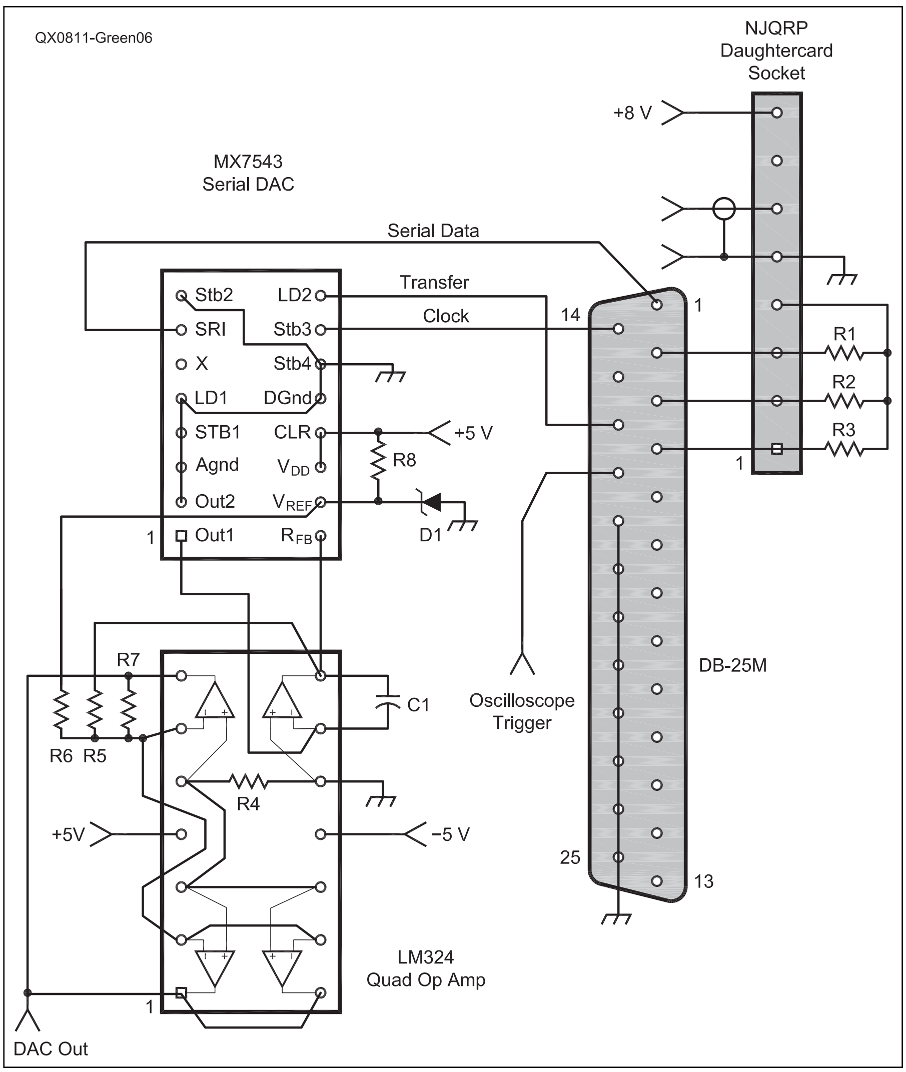



**Figure 6 — This pictorial layout shows how to build the digital to analog converter (DAC) around a DB-25M connector that attaches to the computer parallel port DB-25F connector. The pictorial diagram shows "bottom of the socket" wiring views of the ICs, so pin numbers increase clockwise from Pin 1.**

all four output pins written to by the control port, including one I provide as a signal to the external trigger input of an oscilloscope time base, when I don't use the horizontal sweep waveform that I generate with the DAC.

In case of any conflict or confusion, I provide references to the manufacturer's data sheets for the critical components.

### The MX7543 Digital-to-Analog Converter

Figure 6 shows the DAC circuitry and closely follows Figure 5 in the MAXIM data sheet for bipolar operation. The circuit requires two general-purpose op-amps. You can use a dual op-amp such as an LM358, but I used an LM324 quad op-amp, with three sections in parallel to triple the available output drive current. Unipolar operation requires only one op-amp, but the ability to generate both positive and negative voltages proves worthwhile as you will see below. Prototype this as I did to be sure you can make it work before building a final circuit.

The DAC interface on the control port is identical to that of the DDS on the data port, except the data and clock pins I selected have inverse polarity, requiring an appropriate software adjustment. The oscilloscope trigger discrete (as we in the aerospace community call individual digital on-off signals) is on the same port, so be sure to toggle only the correct bits when writing to the control port.

The reference diode, D1, sets and stabilizes the output voltage range, but the actual voltage is not critical unless you want to do precision work. Calibration constants in the executable software depend on the reference diode value and the offset of the op-amps. These will be close without revising the software if you follow the parts list.

### The MAX110 Analog-to-Digital Converter

Figure 7 shows the ADC circuitry. The MAX110 is a bipolar input ADC. Because the AD8307 always has a positive output, it makes sense to consider using the unipolar MAX111 ADC in this application. This would save one reference diode and increase the measurement resolution by one bit, or a factor of two. I preferred a more general-purpose instrument and selected the bipolar version. Again, the ability to measure both positive and negative input signals proves worthwhile as you will see later.

Software for the ADC is more complex than for the DDS or DAC, because there are more lines to write and read. The DDS and DAC only write to data, clock and transfer inputs. The ADC also requires writes to the chip select input and reads from the data and busy outputs. The main subroutine sequentially reads an output data bit after writing a corresponding input data bit. In addition, there are many options to select in the control word written to the ADC, such as the resolution, which affects the measurement time. The calibration procedure involves a series of write and read sequences.

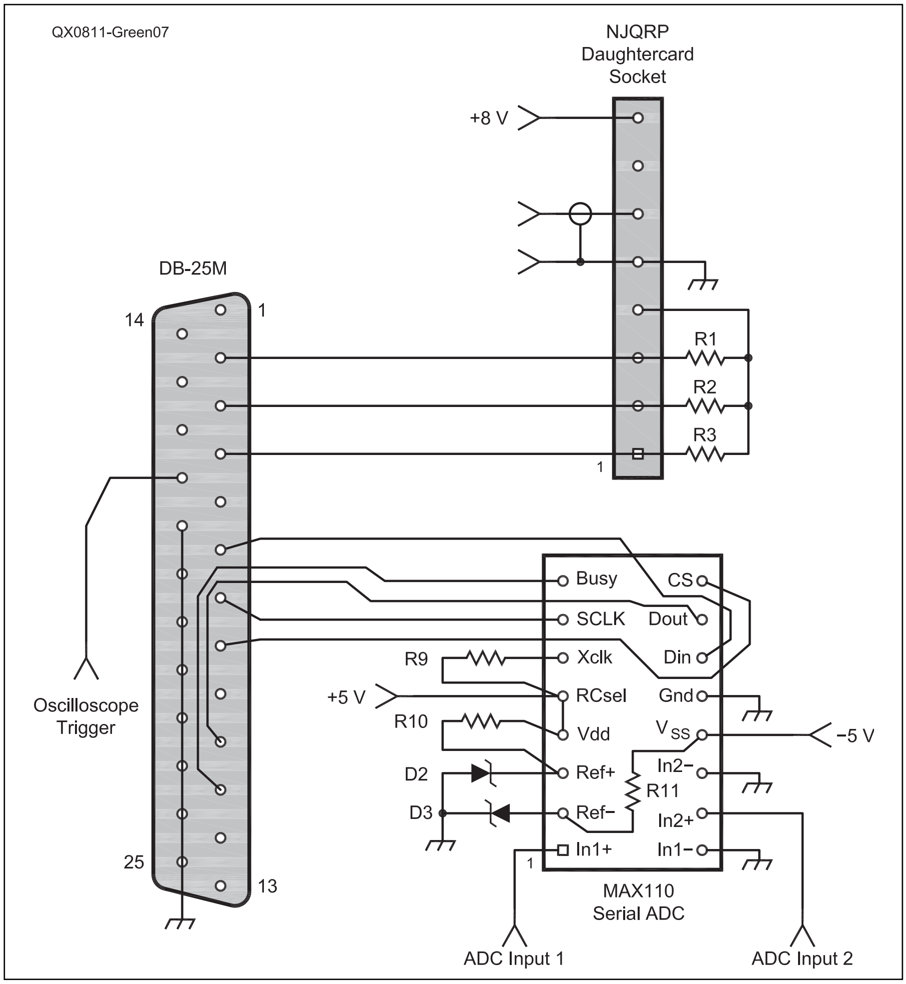



**Figure 7 — This pictorial layout shows how the analog to digital converter (ADC) connects to a DB-25M connector that attaches to the computer parallel port.**

Incredibly, these MAXIM ADC chips provide very accurate measurements and consistently agree with my Fluke 77 Multimeter. Thermal stability of the reference diodes is critical. After the internal three-step calibration procedure, the input range extends from negative $V_{ref}$ to positive $V_{ref}$, where $V_{ref}$ equals the sum of both reference diode voltages. It is difficult to gain an understanding of these important points from the data sheet, where the reference diode voltages are REF– and REF+. I, for one, consistently confuse REF+ with +Vref, which leads to a factor-of-two error. Before the calibration procedure, the input range is, strangely, half the calibrated value or $\pm V_{ref} / 2$. There's that factor of two again. Calibration constants in the executable software depend on the reference diodes, which will be close without revising the software if you follow the parts list.

### The AD8307 Logarithmic Detector

I previously used the AD8307 in a few projects, including a recent patent application. I can't tell you about that now, but it's really useful, so go to the **uspto.gov** Web site after mid-2009, search on my name, and sort among the results to see it.[7] The Patent and Trademark Office publishes patent applications 18 months after the filing date, whether or not a patent has issued or will ever issue.

I first learned of the AD8307 in a *QST* article.[8] I obtained the data sheet and a couple of samples. Studying the data sheet showed me that I could build a direct-reading RF power meter by subtracting 2.10 V from the output, multiplying by 0.4, and applying the result to a ±200 mV digital panel meter with the decimal point moved two positions to the right. I designed and built this in a small box and use it frequently. It is essentially a direct-reading digital version of the instrument in the Hayward *QST* article. This sweep measurement project needs the same functionality, but after measuring the logarithmic-detector output with the ADC, it is simpler to subtract and multiply in software in place of the op-amps and trim-pots I used.

Figure 8 shows the AD8307 circuitry, along with the DDS and the ADC and DAC.

The logarithmic detector has no connections to the parallel port. For observation of rapid scans on an oscilloscope, less output filtering is desirable. For slower measurement with an ADC, more filtering is desirable. Consider adjusting these values for your particular usage. Input 2 of the ADC connects to the logarithmic detector output. I retain Input 1 for general-purpose use, but you might want to connect it to a different output filter after the logarithmic detector.

With nothing else running, measure the output of the logarithmic detector. With no input, the output should be below 0.25 V. With no shielding, local RF emitters will provide enough signal level to exceed this level. Mount the detector in a shielded case to minimize pickup. If the logarithmic detec-

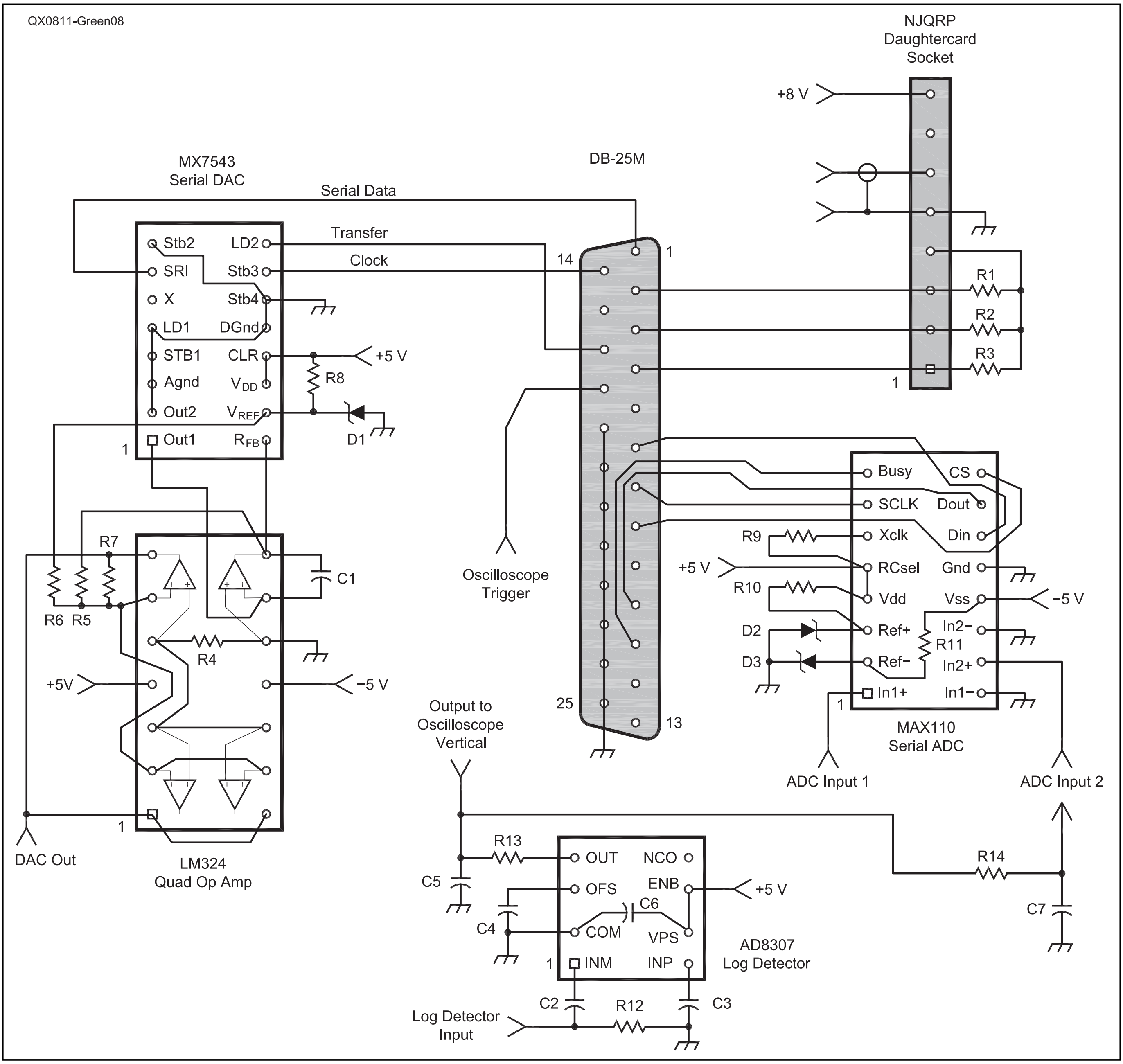



**Figure 8 — This complete pictorial layout shows the addition of the AD8307 logarithmic detector to the other elements of the DDS measurement system.**

| | |
|---|---|
| **C1** | **22 pF** |
| **C2, C3** | **1 µF or 100 µF for low frequency operation—see text** |
| **C4** | **1 µF** |
| **C5** | **0.001 µF** |
| **C6, C7** | **0.100 µF** |
| **D1** | **LM385-2.5 or LT1004-2.5** |
| **D2, D3** | **LM385-1.2 or LT1004-1.25** |

| | |
|---|---|
| **R1, R2, R3, R4** | **10 kΩ** |
| **R5** | **20 kΩ** |
| **R6, R7** | **40 kΩ** |
| **R8, R10, R11** | **3.9 kΩ** |
| **R9** | **1 MΩ** |
| **R12** | **51 Ω** |
| **R13** | **22 kΩ** |
| **R14** | **100 kΩ** |

**Active Components**
**MX7543 DAC**
**MAX110 ADC**
**LM324 Quad Operational Amplifier**
**AD8307 Logarithmic Detector**
**NJQRP DDS kit**

**Standard ham tolerance of half to double is acceptable for all components except R12, which should be near 50 Ω, and R4, R5, R6, and R7, which are not critical in value but should be very close to a ratio of 1:2:4:4.**

tor is in the same case as the DDS, power up the DDS to see if there is any interaction. The DDS does not even require programming to emit enough electromagnetic interference to obscure several orders of magnitude of the logarithmic detector's dynamic range. Isolate as required. Use a separate shielding case, or a case within a case, if you must. This exercise reveals that the logarithmic detector is incredibly sensitive. I originally placed the logarithmic detector on the board near the DDS and could not prevent pickup. I moved the logarithmic detector to the input connector and mounted related components directly onto conductive adhesive backed copper tape serving as a ground plane. Separate packaging external to the case would have been easier and less problematic.

## Circuit Isolation Hints

It is good practice to decouple the power supply leads at all microcircuits with capacitors small and large to minimize the coupling of signals between different functions along the power lines. All power pins have bypass capacitors with short leads to ground even though not explicitly shown in the schematics. The DDS and the logarithmic detector have many bypass capacitors, typically 0.1 µF ceramic capacitors with very short leads and 10 µF teardrop electrolytic capacitors. The logarithmic detector also has some six-hole ferrites in its power lead. The DDS has its own 8 V regulator, and the other circuits share +5 V and –5 V regulators. Next time I would provide a separate voltage regulator for the logarithmic detector.

I used an external ±12V supply and fed it into the box through some serious feed-through filters to keep external RF fields away from the logarithmic detector. Again, it would be simpler to package the AD8307 in its own separate shielded box, preferably with a battery power supply.

## Software

The download package contains programs called ADC, DAC, LSW, SSW, SWCENTER, SWDDS, and TRANSFER. The executable DOS console applications have a .EXE suffix, and the source files have a .CPP suffix.

### Analog to Digital Converter

The application program ADC.EXE digitizes the analog signal at input 1 and writes the value to the screen. This is a simple test program to demonstrate functionality. It is useful for making simple measurements, comparing your DVM to the results, and determining whether you need to adjust the calibration constants. Do this first, so you can use the ADC to check and calibrate the DAC. ADC will start from a command line or by double clicking the file name in *Windows Explorer*. In either case, a usage

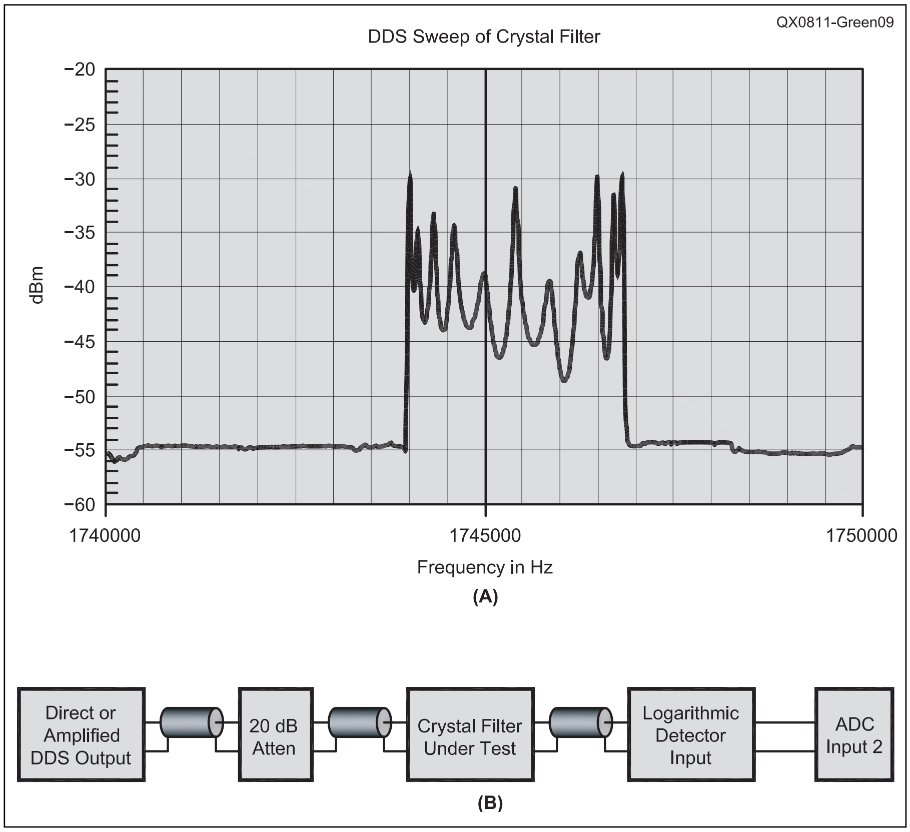



**Figure 9 — Part A shows the linear sweep frequency plot of a crystal filter made with the DDS measurement system. Part B shows the measurement connections, including a 20 dB attenuator.**

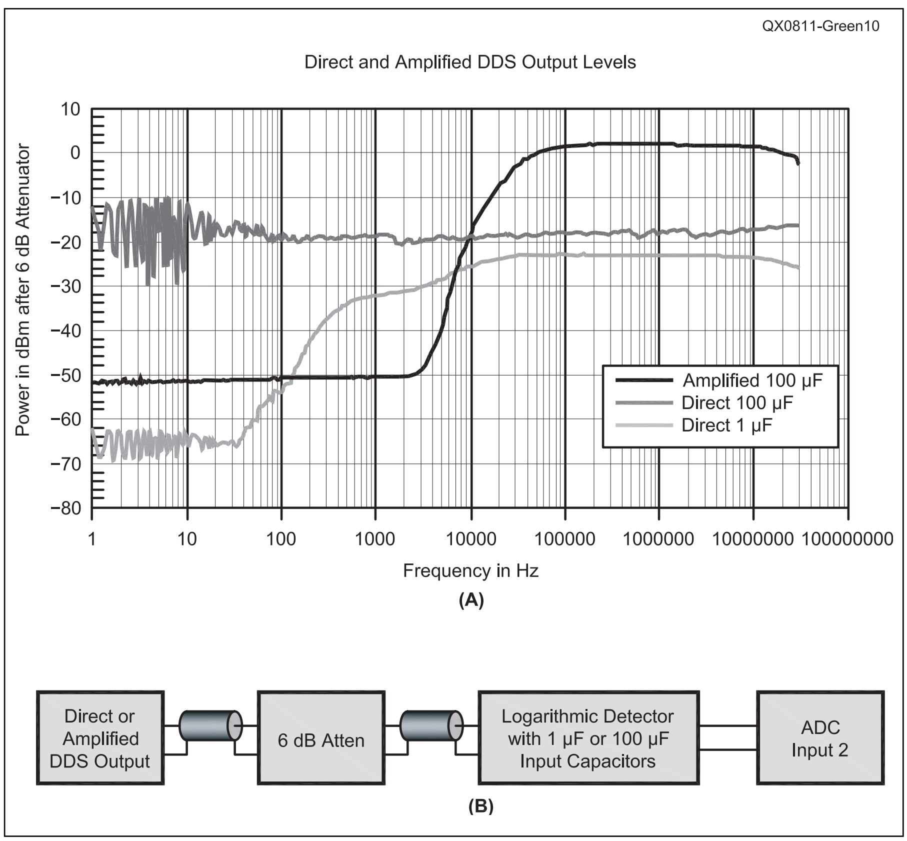



**Figure 10 — Part A shows the logarithmic sweep frequency plot of the direct and amplified DDS outputs. Part B shows the measurement connections, including a 6 dB attenuator.**

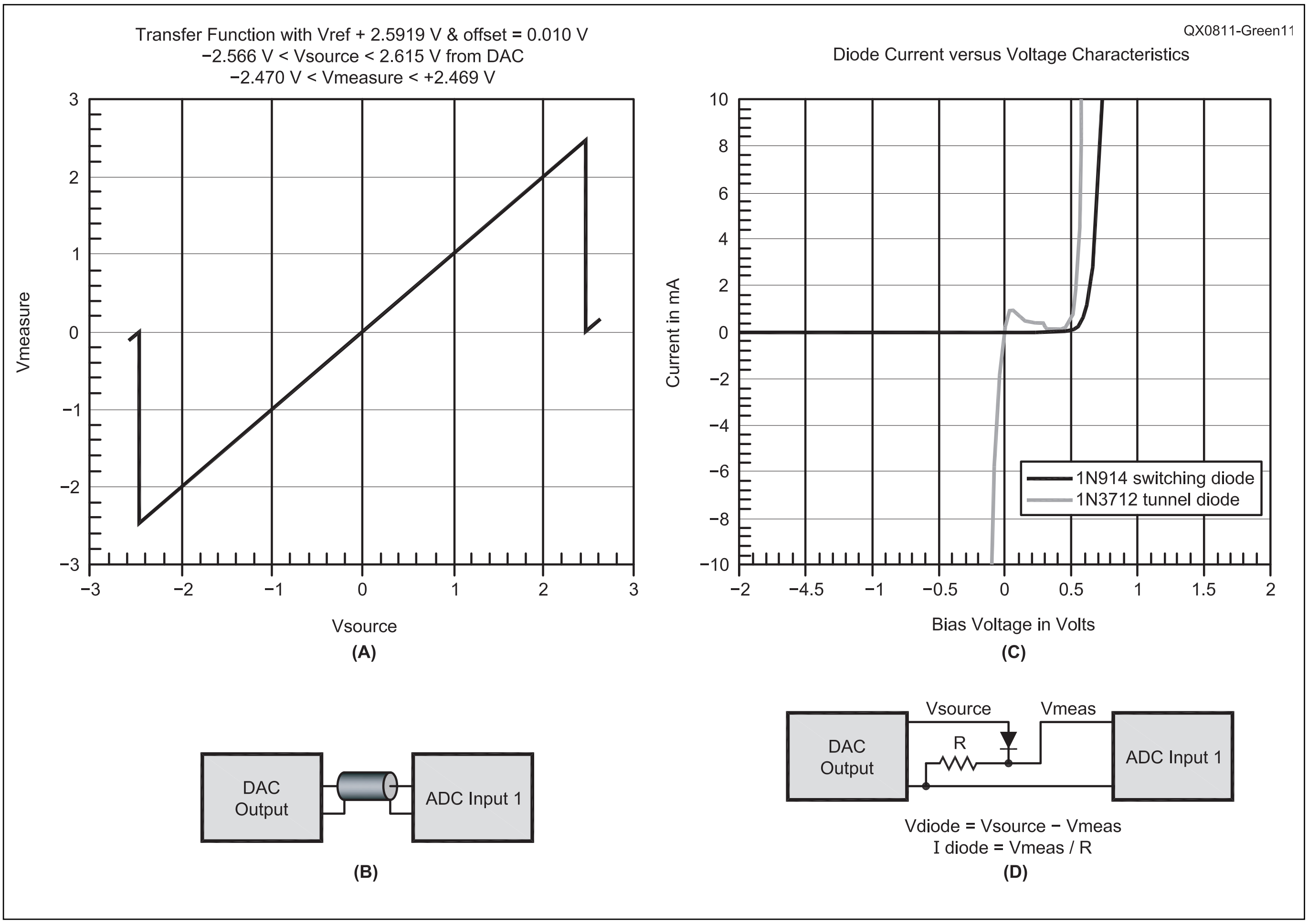



**Figure 11 — Part A shows the calibration waveform with the ADC measuring the DAC, and Part B shows the test set-up. Part C shows a diode current versus voltage-characteristic measurement made by the system. Part D shows the test set-up for the diode measurements.**

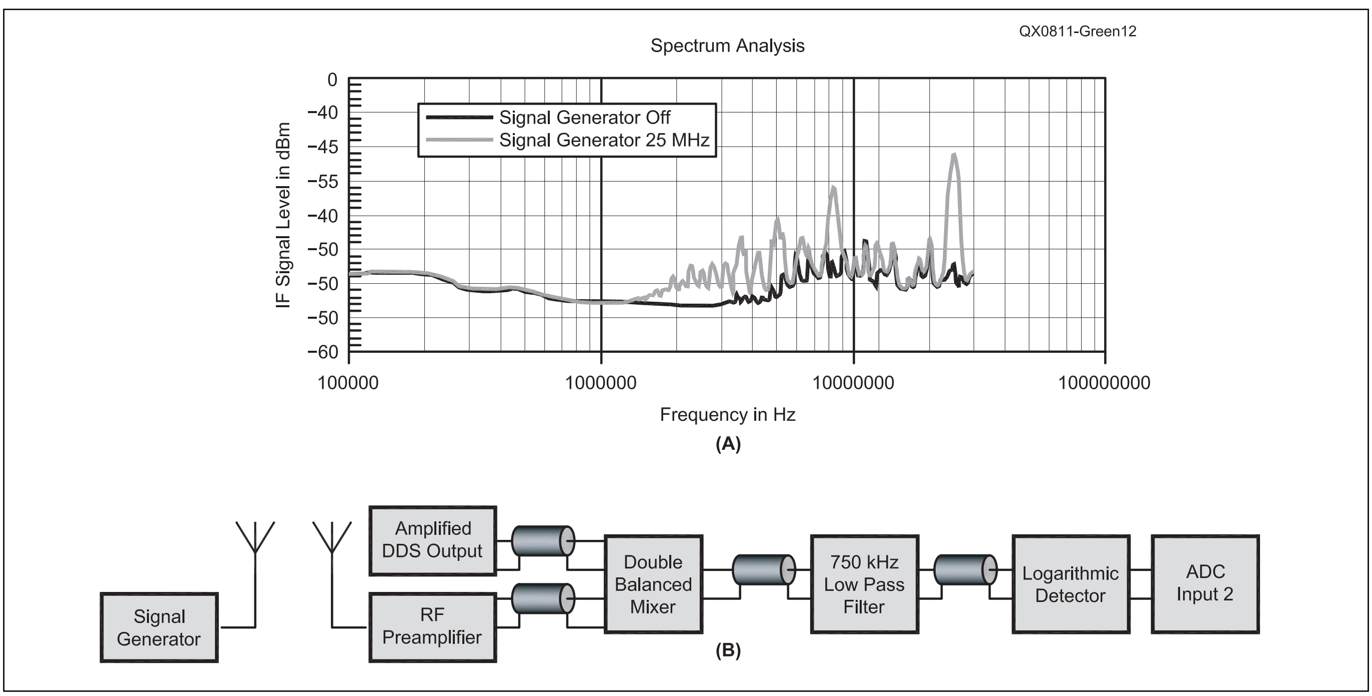



**Figure 12 — Part A shows the output display of a sweep using the DDS as the local oscillator of a spectrum analyzer. Part B is the test equipment set-up.**

statement appears that tells you the command line parameters to enter to decrease the resolution to 12 bits for faster measurements or increase it to 14 bits for more precise, but less frequent, measurements. The MAX110 data sheet explains the detailed operation and the tradeoffs involved. The calibration procedure built into ADC.EXE provides extremely good accuracy. The program scales the internal result by a calibration factor Vref that depends on the individual reference diodes in your implementation. Again, the result will be close without revising the software if you use the LM385-1.2 reference diodes called out for D2 and D3 in the parts list.

### Digital to Analog Converter

The application program DAC.EXE writes numbers to the DAC that appear as an output voltage. Again, this is a simple test program to demonstrate functionality. DAC will only start from a command line. With no parameters following the command, the program displays a usage statement that tells you proper syntax. With a single parameter, the output is a constant voltage. With two parameters, the output voltage sweeps from the first entry, which must be the most negative value to the second entry, which is the most positive value, and repeats. The possible range is about –2.5 V to +2.5 V. Enter start and end values in integer millivolts as parameters following the command to select values within this range. This sawtooth waveform generator is useful to demonstrate that the DAC functions. In principle, you can generate an arbitrary waveform if you change the program to read the contents of a lookup table and write the values to the DAC.

### RF Sweep Generator with Synthesized Sweep

The application program SWDDS.EXE provides a repetitive linear sawtooth sweep between a lower and an upper frequency and synthesizes a sawtooth voltage waveform from the DAC to drive a horizontal oscilloscope input to display the trace. SWDDS will start from a command line or by double clicking the file name in *Windows Explorer*. With no parameters, SWDDS comes up running default settings and sweeps from 1 MHz to 10 MHz in 4096 steps. Set frequency start, end, and step number parameters in the command line. The numeric keypad allows coarse increases and decreases of start and end frequencies and number of steps. The maximum number of steps is 4096, the maximum resolution of the DAC, which cannot operate below a step count of 1. The DDS is able to provide resolution to a small fraction of a hertz if you eliminate the DAC analog sweep.

The application program SWCENTER.EXE provides the same functionality as SWDDS with entry in the form of center frequency and sweep width. Default settings are a center frequency of 1 MHz and a sweep width of 100 kHz, so the default sweep range is between 950 kHz and 1050 kHz. The numeric keypad allows fine as well as coarse changes of center frequency, sweep width and number of steps. Enter numeric values and observe the effects on the displayed parameters as well as what you'll see on an oscilloscope.

The **11x08_Green.zip** file available for download from the ARRL Web site includes a file called "Notes on W0PCE Programs.doc" that has explicit instructions for using all the console applications, and explains the effects of numeric entries. See Note 5.

### RF Sweep Generator with Data Log

The application program SSW.EXE provides a single linear sweep between lower and upper frequencies with an adjustable number of equal frequency steps. It instructs the ADC to measure the voltage into input 2 from the logarithmic detector at each frequency, and writes the frequency and power measurements in Hz and dBm to a text file. When SSW completes, I import the data text file into a spreadsheet and plot it, as shown in Figure 9. SSW has evenly spaced frequency steps and is especially useful for narrow or moderate sweep widths with very high resolution.

The application program LSW.EXE provides a single logarithmic sweep between a lower and upper frequency. It instructs the ADC to measure the voltage out of the logarithmic detector at each frequency, and writes the data measurement to a text file. When LSW completes, I import the data text file into a spreadsheet and plot it, as shown in Figure 10. LSW has increasingly spaced frequency steps and is especially useful for sweep widths that extend over most or all of the full range. If a logarithmic frequency display best suits your measurement application, choose LSW, because it samples many fewer points and completes sooner. This provides the highest resolution where needed at low frequencies, and larger frequency steps at high frequencies.

Both SSW and LSW will start from a command line only. With no parameters, the programs print usage statements to explain the parameters they require. You must enter an output filename as the first parameter. If you enter anything else, a text file will appear on your drive with that as the filename. It is most convenient to append .TXT to the filename, so you can open the file with *WordPad* or an equivalent text editor. The output file appears in the same folder as the executable unless you redirect it using the old DOS filename rules. For example, if you enter "..\" before the filename, it will appear in the next higher folder in the pathname.

Note that the usage statement for LSW indicates that the fourth parameter is something called "step fraction denominator." This integer number allows you to control the frequency increment of the logarithmic sweep. For a value of 100, each frequency will be 1/100 or 1% higher than the previous

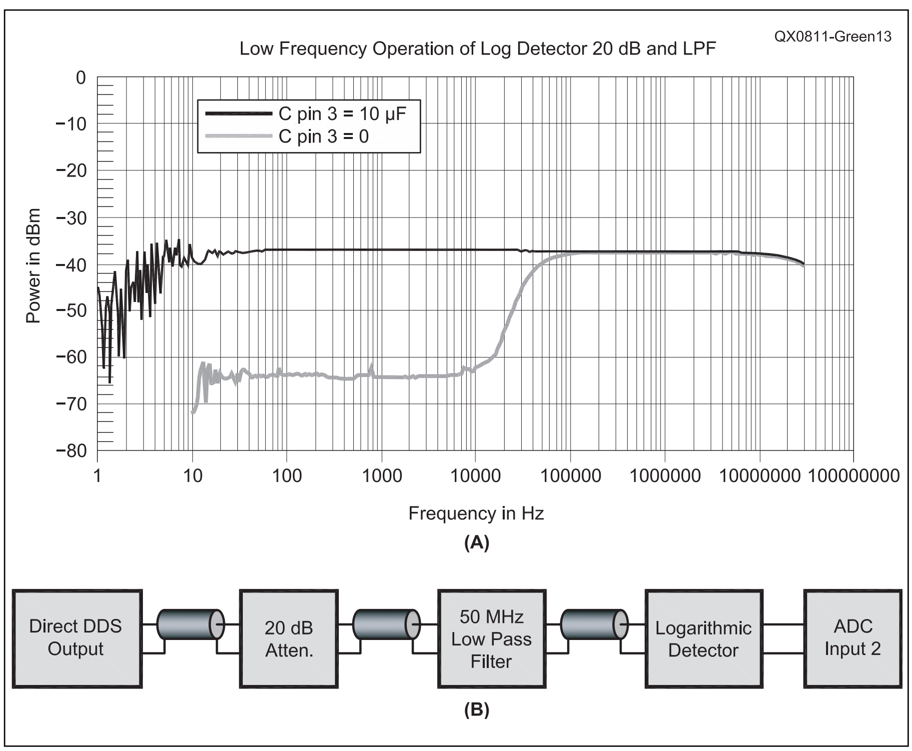
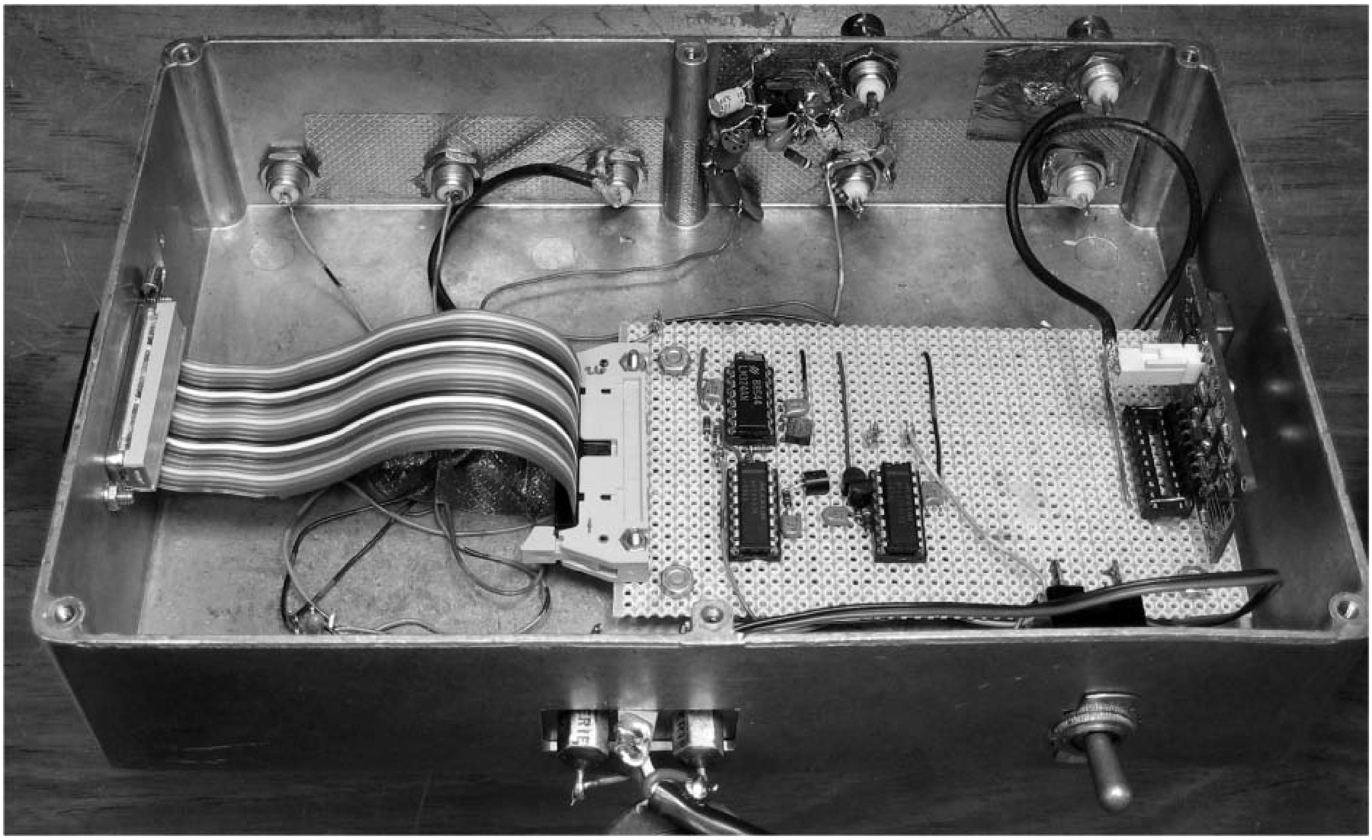



**Figure 13 — Part A shows a plot that illustrates a measurement of the low frequency performance of the log detector. Part B shows the test equipment set-up.**



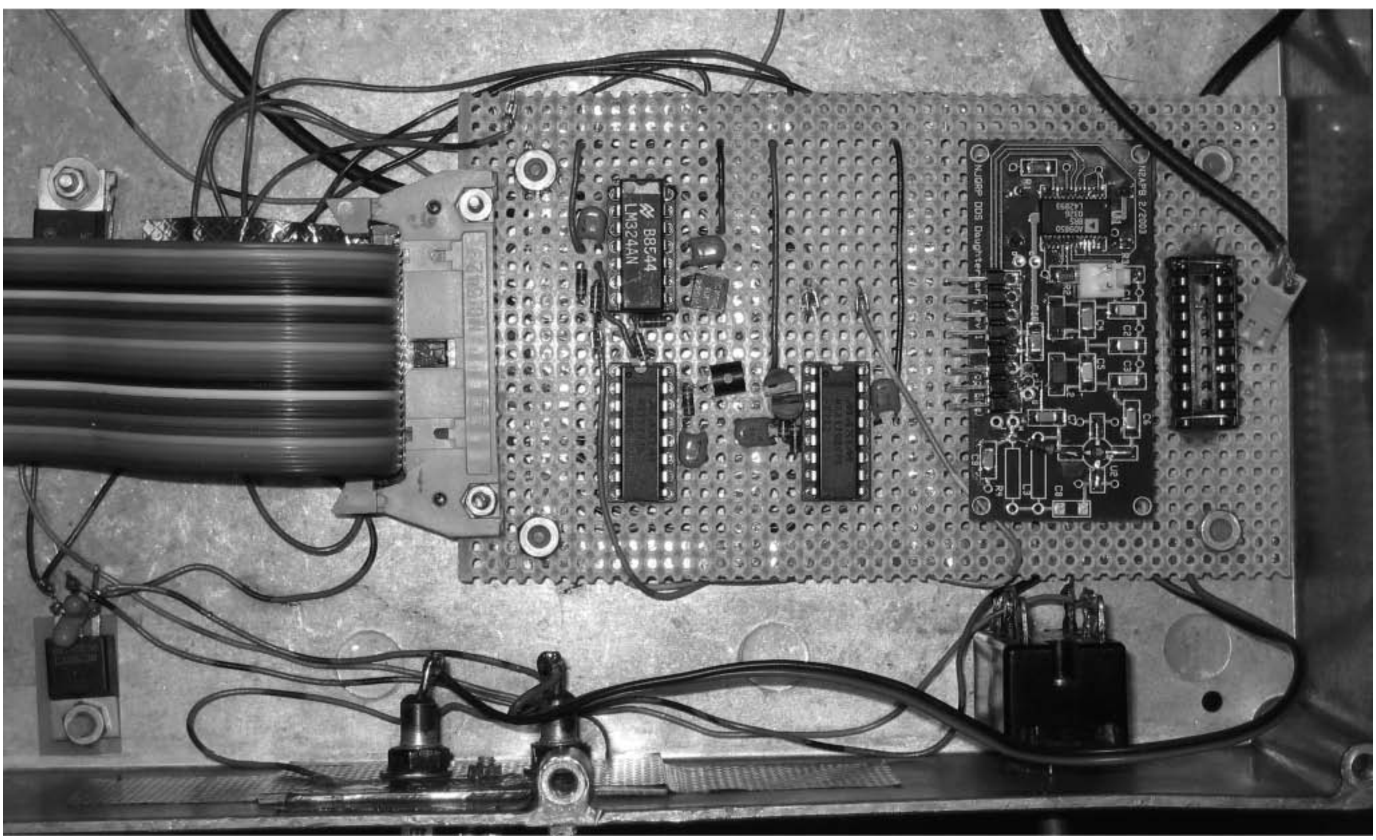



**Figure 14 — These photos show the prototype DDS sweep measurement system.**

value. For a value of 10000, each frequency will be 1/10000 or 0.01% higher than the previous one. LSW allows the step fraction denominator to be as large as you want, but the program still advances the frequency by a minimum of one count.

### Voltage Transfer Function

The application program TRANSFER.EXE drives the DAC rather than the DDS, and provides a single linear sweep between a lower and an upper voltage to apply to an external circuit under test. It instructs the ADC to measure the voltage into input number 1 at each output voltage, and writes the data measurement to a text file. When TRANSFER completes, I import the data text file into a spreadsheet and plot it, as shown in Figure 11 to display the transfer function. The plot on the left shows a calibration waveform with the DAC output connected to the ADC input. Note that the DAC output voltage range is wider than the ADC can accommodate for the particular reference diodes I chose. The plot on the right shows the current versus voltage characteristic curves of a switching diode and a tunnel diode.

### Spectrum Analysis

Figure 12 shows the result of a crude attempt at spectrum analysis. I drove the L-port of a double balanced mixer with the amplified output of the DDS and the R-port with a short antenna and a wideband preamplifier. Signals from the I-port passed through a low pass filter and to the logarithmic detector input. LSW.EXE measured power from the I-port into the logarithmic detector and yielded a plot of IF signal power versus local oscillator frequency.

## DDS Modifications

If you observe the waveform of the older 30 MHz DDS daughtercard with a fast oscilloscope, you may see some serious distortion caused by improper bias of the MAR-1A RF output amplifier. I provided a regulated 8 V supply for the amplifier, omitted the output inductor, and selected a lower resistor value that gave the best sine wave output over the entire frequency range. The optimum resistor value varies with power supply voltage and from amplifier sample to sample. Newer NJQRP DDS-60 daughtercards do not use the MAR-1A and probably do not have this problem.

Decouple the power lead to the daughtercard socket with several capacitors with short leads and perhaps a few ferrite beads if the logarithmic detector is in the same box. Keep the DDS and the logarithmic detector separated, isolated, and shielded to get the best measurement sensitivity.

### Low Frequency DDS Operation

Figure 10 shows that the capacitively coupled amplifier limits the low frequency response of the DDS to about 100 kHz. I accessed an unused second DDS output signal IOUTB from pin 20 of the Analog Devices AD9850 or AD9851 DDS chip. The second output terminates, unused, in a 24 Ω chip resistor labeled R2 in Figure 1 and labeled R11 on the DDS-60. Remove the 24 Ω chip resistor and replace it with a 50 Ω chip resistor. On the 30 MHz daughtercard you can add a couple of Berg pins in the unused holes nearby. This provides a direct output that works down to a fraction of a hertz, and up to 30 MHz. With no amplifier or filter, the level is lower, and there are some glitches to remove with an external lowpass filter. Bring this signal out through some miniature RG-174/U or equivalent coaxial cable to the front panel. It expects a 50 Ω termination in operation. Remember to design the external lowpass filter for operation out of, and into, 50 Ω. I use a 50 MHz five-pole elliptical low-pass filter. The DDS-60 provides no convenient means to bring out this complementary output directly. You might affix a short length of miniature coaxial cable across chip resistor R11 after you change it from 24 Ω to 50 Ω.

### Low Frequency Logarithmic Detector Operation

The DDS amplifier is not the only problem at low frequencies, but it is the easiest to resolve. The AD8307 logarithmic detector also has some issues.

The logarithmic detector has direct coupling and should work to very low frequencies. The data sheet suggests split supplies to operate the COM pin 2 V below the input. I have not tried this yet because results with large input capacitors are satisfactory to 20 Hz.

Note Figure 10 again. With 1 µF input capacitors, the logarithmic detector works well down to 20 kHz. With 100 µF input capacitors, it works well to below 100 Hz.

The RC lowpass filter on the logarithmic detector output requires a short time constant for rapid sweeps observed on an oscilloscope, and a long time constant for lower noise at all frequencies and steadier response at very low frequencies when measured with the ADC. Experiment with larger resistor and capacitor values for very slow sweeps and smaller values for rapid sweeps.

A frequency dependence that is most notable at low signal levels occurs because the AD8307 offset control-loop corner-frequency is too high. For operation down to 10 Hz, add a 1 to 10 µF capacitor from OFS pin 3 to ground.[9] Figure 13 shows that this capacitor is unnecessary for operation above 100 kHz. My thanks go to Jim Bedrosian at Analog Devices for this explanation.[10]

### Low Frequency Software Adjustment

At high frequencies, many cycles occur in a period corresponding to the delay between the frequency step and the time when the ADC reads the logarithmic detector output. At audio frequencies, only a small fraction of a cycle occurs before the ADC can sample the logarithmic detector. The sweeper driver LSW.EXE provides a variable delay to allow several cycles to occur before measurement at low frequencies, so very low frequencies take a long time to sweep. This feature does not appear in SSW.EXE.

## In Conclusion

Try the simple setup in Figure 14 to automate your own measurements! There isn't much to it besides the DDS daughterboard, four chips, three reference diodes, and three regulators to provide ±5 V and +8 V. The prototype isn't pretty, but this hardware offers vast capability. Note that I moved the logarithmic detector from the perforated board to a position next to the input connector and that I expended much effort to filter interference conducted by the power leads.

My thanks also go to my friends Matt Kastigar, WØXEU, and Ken Gianino, WBØQNA, for providing NJQRP kits.

## Notes

[1]**www.njqrp.org/dds/index.html**

[2]**www.maxim-ic.com/quick_view2.cfm/qv_pk/1580**

[3]**www.maxim-ic.com/quick_view2.cfm/qv_pk/1018**

[4]**www.analog.com/en/prod/0,,759_847_AD8307,00.html**

[5]The software files associated with this article are available for download from the ARRL *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **11x08_Green.zip**.

[6]**www.embeddedtronics.com/design&ideas.html**

[7]**appft1.uspto.gov/netahtml/PTO/search-bool.html**

[8]Wes Hayward, W7ZOI, "Simple RF Power Measurement," *QST*, June 2001, pp 38-43.

[9]**www.analog.com/UploadedFiles/Application_Notes/896652515AN_691_0.pdf**

[10]Private communication 17 June 2008.

*Dr. Sam Green, WØPCE, is an aerospace engineer in Saint Louis, Missouri, with degrees in Electronic Engineering from Northwestern University and the University of Illinois at Urbana. Sam specializes in optical data communications and photonics. Sam became KN9KEQ and K9KEQ in 1957 while a high school freshman in Skokie, Illinois, and held a Technician class license for 36 years before finally upgrading to Amateur Extra Class in 1993. He is an ARRL Member. Sam is a Registered Professional Engineer in Missouri and a life senior member of IEEE. Sam holds fourteen patents, with more pending.*

**Frederick B. Hubler**

9422 Deer Ridge Dr, Cedar Rapids, IA 52411; fhubler@msn.com

# Broadband Impedance Matching

*Improve Your Antenna's Performance.*

Procedures for the design of band-pass impedance matching networks, where the load resistance is constant and the load includes a single series or shunt reactive element, are well established. These methods have limited utility, however, for simple monopole or dipole antennas, where the antenna resistance varies and the antenna impedance continually transitions between series and shunt resonances as the wavelength decreases. This paper illustrates that the classical band-pass design techniques can be used as a starting point to design antenna matching networks of moderately wide bandwidth. The design is completed with the aid of a circuit analysis program containing a built-in optimizer that can minimize the difference between the desired Standing Wave Ratio (SWR) and the actual SWR response of the circuit.

An antenna simulator is presented that simulates the input impedance of a Shakespeare Model 390-1 MG 23 foot whip marine HF antenna. The impedance data used to develop this simulator was obtained from Shakespeare's Web site. It is used as the terminating load of the impedance matching networks in the circuit analysis program when optimizing the impedance matching networks.

With an antenna simulator, the load can be characterized over any desired bandwidth for the purpose of designing an impedance matching network over that band. An added benefit is that it makes the task of determining the network response outside the match band a straight forward circuit analysis task.

The circuit analysis program used is a modified version of the U C Berkeley *SPICE* program, version 3f5. The modified program includes a reduced gradient optimizer that can optimize the linear (AC) circuit response for voltage, current, impedance, reflection coefficient, VSWR, or power goals. The gradient can be calculated either numerically, or by the adjoint method. Methods for evaluating the gradient using the adjoint method are provided in "Current Trends in Network Optimization," *IEEE Transactions on Microwave Theory and Techniques*.[1]

[1]Notes appear on page 29.

## Broadband Matching Fundamentals

**Definitions**

$f_1$ = lower band edge

$f_2$ = upper band edge

$R$ = average series load resistance over the match band

$G$ = average parallel load conductance over the match band

$BW$ = bandwidth in Hertz used to calculate $\Delta X$ or $\Delta B$

$\Delta X$ = change in load series reactance over band BW

$\Delta B$ = change in load parallel susceptance over band $BW$

Note: $BW$ does not represent the match bandwidth, but rather, a narrow bandwidth around the band center that is used to calculate the reactance slope parameter discussed below. The band center is given by

$$f_0 = \sqrt{f_1 \times f_2} \quad \text{[Eq 1]}$$

and

$$\omega_0 = 2\,\pi \times f_0 \quad \text{[Eq 2]}$$

The fractional match bandwidth is defined as:

$$\omega_m = \frac{f_2 - f_1}{f_0} \quad \text{[Eq 3]}$$

For a series equivalent circuit the reactance slope parameter is defined in Reference 2 as:

$$\chi = \frac{\omega_0}{2}\frac{dx}{d\omega} \quad \text{[Eq 4]}$$

evaluated at $\omega_0$.

For a load consisting of a resistor in series with a single reactive element, the reactance slope parameter is simply the reactance of the reactive element at $f_0$.

For more complex loads this parameter can be approximated numerically by determining the change in reactance over a narrow band (BW) around $f_0$, while including the reactance required to resonant the load at $f_0$. This change in reactance over a small band will represent the change in both capacitive and inductive reactance, and thus must be divided by two to obtain the reactance of a kind. Therefore, the reactance slope parameter used here for series equivalent loads is defined as:

$$\chi = \frac{f_0}{2}\frac{\Delta X}{BW} \quad \text{[Eq 5]}$$

Similarly, the susceptance slope parameter used for parallel equivalent loads is defined as:

$$\beta = \frac{f_0}{2}\frac{\Delta B}{BW} \quad \text{[Eq 6]}$$

The above parameters are required to calculate the load decrement defined in Reference 2 as:

$$\delta = \frac{R}{\chi\omega_m} \quad \text{[Eq 7]}$$

for series equivalent loads, and

$$\delta = \frac{G}{\beta\omega_m} \quad \text{[Eq 8]}$$

for parallel equivalent loads.

In the above equations, $R$ is the average series resistance, and $G$ is the average parallel conductance. Note that the smaller the load decrement, the higher the antenna Q, since δ is just the inverse of antenna Q scaled by the fractional bandwidth.

In the discussion that follows, unless otherwise noted, frequencies are in MHz, resistance is in ohms, inductance is in microhenrys, and capacitance is in picofarads. For all figures containing SWR or reflection coefficient, the curves are with respect to the applicable source impedance, which may or may not be 50 Ω.

## Series Load 25 to 30 MHz

The load used to acquire insight into broadband matching is a 10 Ω resistor in series with a 53.05 pF capacitor. For a match band of 25 to 30 MHz, this load yields a load decrement of 0.5, which makes it convenient to read the *g* parameters from Figures. 4.09-5 through 4.09-7 (pp 126-128) of *Microwave Filters, Impedance Matching Networks and Coupling Structures*.[2] The series load characterization parameters are listed in Table 1.

Once the load decrement has been determined it is necessary to select the order of the matching network. The first, second and third order matching network $g$ parameters for $\delta = 0.5$ are listed in Table 2.

The $g$ parameters represent the impedance or admittance of the load, source, and resonators for a 1 Ω prototype. The next step is to impedance scale all $g$ parameters to the load impedance. If $g_1$ is a series resistance, $g_0$ represents the load conductance. If $g_1$ is a parallel (or shunt) reactance, $g_0$ is the load resistance. The $g$ parameters then alternate between series and parallel up to and including the source impedance ($g_{n+1}$). If $g_n$ is a series resonator, then $g_{n+1}$ represents the source conductance, otherwise it is the source resistance. Impedance scaling is accomplished by dividing the parallel $g$ parameters by the load resistance value, and multiplying the series $g$ parameters by the load resistance value. Table 3 lists the series load impedance scaled $g$ parameters.

Given the impedance scaled $g$ parameters, the resonator L and C values can be calculated as follows: For series resonators the reactance is:

$$x_i = \omega_0 L_i = \frac{1}{\omega_0 C_i} = \frac{g_i}{\omega_m} \quad \text{[Eq 9]}$$

For parallel resonators the susceptance is:

$$b_i = \omega_0 C_i = \frac{1}{\omega_0 L_i} = \frac{g_i}{\omega_m} \quad \text{[Eq 10]}$$

When we apply Equations 9 and 10 to the impedance scaled $g$ parameters listed in Table 3, we obtain the matching network element values given in Table 4.

Figure 1 shows a graph of the complex reflection coefficients for the first and second order networks. For the n = 1 case, the best possible broadband impedance match requires a series inductor to be added to achieve series resonance at $f_0$. Then the source impedance is the resistance required to make the real part of the reflection coefficient zero at the band edges.

For the n = 2 case, the first resonator remains unchanged. The second resonator is chosen such that the imaginary part of the reflection coefficient is zero at the band edges, and the source impedance is the resistance required to equalize the band edge SWR and the SWR at $f_0$.

The reflection coefficient gets more complicated for third and higher order networks. Although the SWR at the band edges remains equal to the high points of the in-band SWR ripple, in general neither the real nor the imaginary parts of the reflection coefficient are zero at the band edges.

R. M. Fano has shown that there are theoretical limitations on the broadband matching of arbitrary impedances.[3] Namely, the integral of the reflection coefficient across the match band increases as the Q of the load increases, and this integral must be equal to or greater than a certain minimum, even for infinite order networks.

Figure 2 shows a graph of SWR for the first and second order networks. In order to achieve the least possible maximum SWR you don't get to choose the source impedance. Impedance inverters, discussed later, can be used to alleviate this problem for third and higher order band-pass networks, and may be feasible for second order networks if the negative element of the inverter can be absorbed by the load resonating element.

Reflection coefficient and SWR for the

**Table 1**
**Series Load Characterization**

$f_0$ = 27.4 MHz
$\omega_m$ = 0.183
$R$ = 10 Ω
$\chi$ = 109.5 Ω
$\delta$ = 0.5

**Table 2**
**Series Load *g* Parameters**

| *n=1* | *n = 2* | *n = 3* |
|---|---|---|
| $g_0$ = 1 | $g_0$ = 1 | $g_0$ = 1 |
| $g_1$ = 2 | $g_1$ = 2 | $g_1$ = 2 |
| $g_2$ = 0.45 | $g_2$ = 0.40 | $g_2$ = 0.76 |
| | $g_3$ = 2.25 | $g_3$ = 1.35 |
| | | $g_4$ = 0.6 |

**Table 3**
**Series Load Impedance Scaled g Parameters**

| *n = 1* | *n = 2* | *n = 3* |
|---|---|---|
| $g_0$ = 0.1 | $g_0$ = 0.1 | $g_0$ = 0.1 |
| $g_1$ = 20 | $g_1$ = 20 | $g_1$ = 20 |
| $g_2$ = 0.045 | $g_2$ = 0.04 | $g_2$ = 0.076 |
| | $g_3$ = 22.5 | $g_3$ = 13.5 |
| | | $g_4$ = 0.06 |

**Table 4**
**Matching Network Element Values**

| *n = 1* | *n = 2* | *n = 3* |
|---|---|---|
| $R_0$ = 10.0 | $R_0$ = 10.0 | $R_0$ = 10.0 |
| $L_1$ = 0.6366 | $L_1$ = 0.6366 | $L_1$ = 0.6366 |
| $R_2$ = 22.22 | $L_2$ = 0.0265 | $L_2$ = 0.01396 |
| | $C_2$ = 1273 | $C_2$ = 2419 |
| | $R_3$ = 22. | $L_3$ = 0.4297 |
| | | $C_3$ = 78.6 |
| | | $R_4$ = 16.7 |

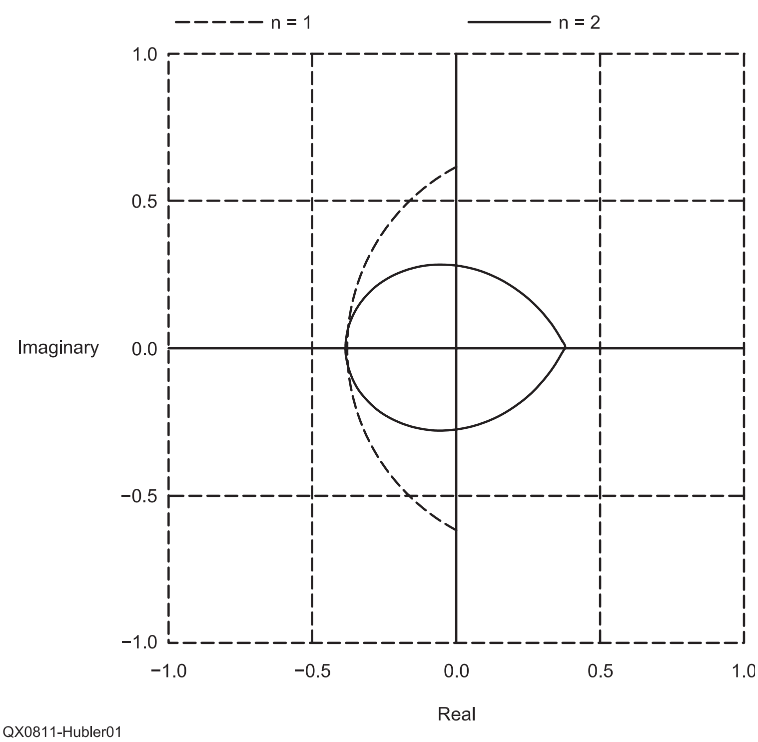



**Figure 1 — This graph shows the complex reflection coefficients for first and second order networks.**

third order network are shown in Figures 3 and 4 respectively. The third order matching network is shown in Figure 5, where CL and RL represent the load.

The improvement in SWR achievable with networks beyond third order rapidly reaches a point of diminishing returns, and Matthaei, Young and Jones do not even provide *g* parameter charts for networks beyond fourth order. (See Note 2.) It may be desirable, however, to use fourth order networks for other reasons, such as providing more flexibility in the type of impedance inverter used to achieve the desired source impedance, a desire to have a series resonant tank at the input to assure that the out-of-band impedance goes high instead of low, or to decrease passband ripple.

## Antenna Simulator

The 23 ft whip simulator consists of a series resistor, a series capacitor, and two parallel tank circuits in series, as shown in Figure 6. The element values used for the simulator were obtained with the aid of the *SPICE* program by optimizing the input impedance so as to minimize the squared error between the simulator input impedance and the measured impedance data provided by Shakespeare over the 8 to 30 MHz band. Starting values for tank 1 were based on the in-band parallel resonance of the antenna. A second out-of-band tank circuit was added to provide a means for the optimizer to better match the impedance at the high end of the band. Starting values of Ra and Ca were determined from the measured impedance at the low end of the band. Graphs comparing the measured and simulated input impedance are shown in Figures 7 and 8. The optimized antenna simulator element values are listed in Table 5.

This network is used as the load for all of the matching networks that follow. These element values are not allowed to be varied by the optimizer when optimizing a matching network. If one were to actually build antenna matching networks based on the simulator impedance, it might be preferable to have simulators for each match band of interest in order to obtain a better fit between measured and simulated impedance. It would also be desirable to measure the antenna impedance in the actual environment in which it is used, including duplication of the ground plane.

## Antenna Matching Networks

The first step in designing a matching network is to determine the frequency band to be matched. The next step is to determine whether the load is more nearly series or parallel equivalent. This is accomplished simply

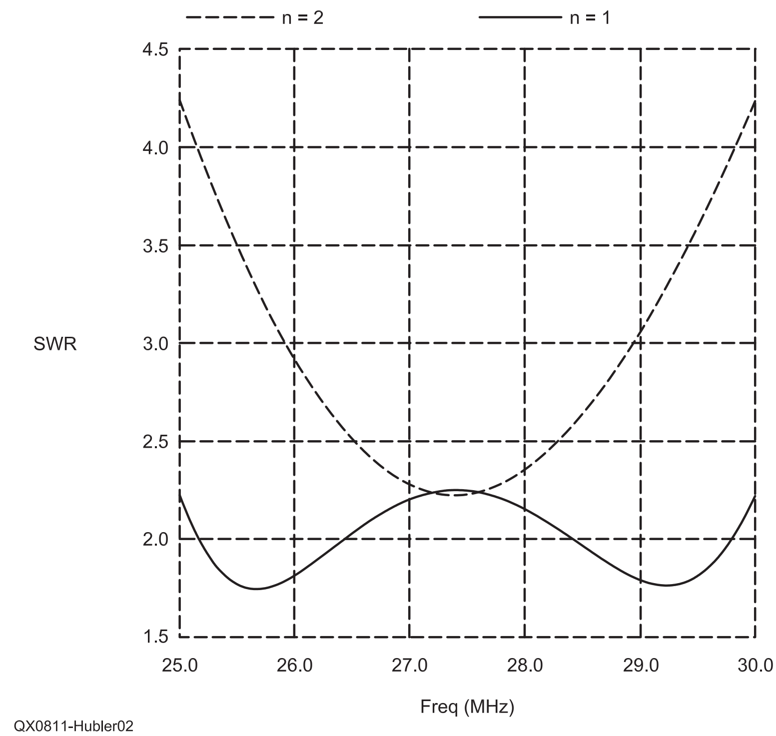



**Figure 2 — SWR versus frequency graph for first and second order networks.**

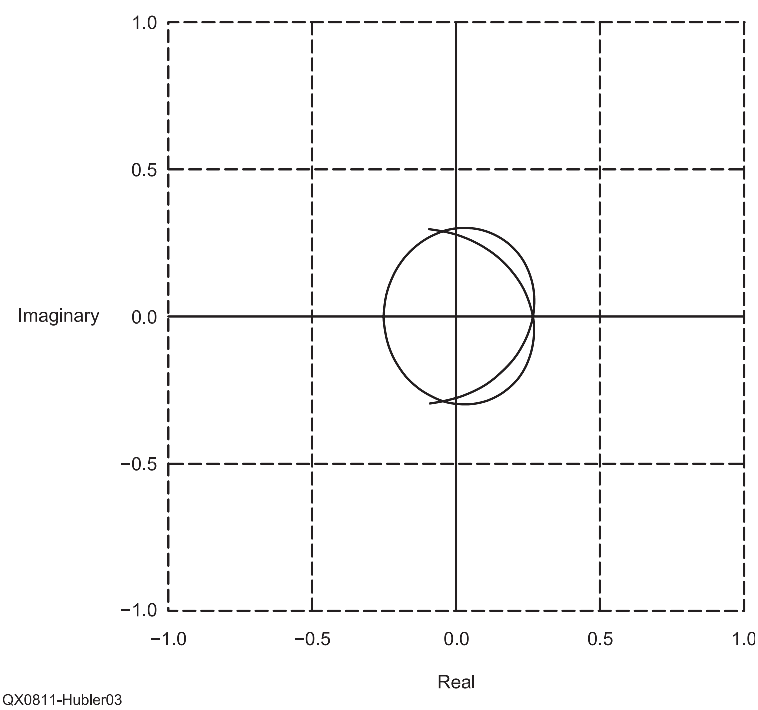



**Figure 3 — Reflection coefficient for the third order matching network, n = 3.**

**Table 5**
**Antenna Simulator Element Values**

| | |
|---|---|
| Ra = 17.6 Ω | Ca = 73.51 pF |
| R1 = 1021 Ω | R2 = 4675 Ω |
| C1 = 45.58 pF | C2 = 32.30 pF |
| L1 = 1.545 µH | L2 = 0.6177 µH |

**Table 6**
**Antenna 8 – 10 MHz Load Characterization**

$f_0$ = 8.94 MHz
$\omega_m$ = 0.224
$R$ = 31.7 Ω
$L_1$ = 1.675 µH
$BW$ = 0.2 MHz
$\Delta X$ = 12.3 Ω
$\chi$ = 275 Ω
$\delta$ = 0.498

**Table 7**
**N = 4 *g* Parameters for an 8-10 MHz Matching Network**

| *1 Ω Prototype* | *Impedance Scaled* |
|---|---|
| $g_0$ = 1 | $g_0$ = 0.0326 |
| $g_1$ = 2 | $g_1$ = 61 |
| $g_2$ = 0.9 | $g_2$ = 0.0293 |
| $g_3$ = 2.35 | $g_3$ = 72 |
| $g_4$ = 0.425 | $g_4$ = 0.0138 |
| $g_5$ = 1.75 | $g_5$ = 54 |

**Table 8**
**Antenna 16 – 20 MHz Load Characterization**

$f_0$ = 17.89 MHz
$\omega_m$ = 0.224
G = 0.00105 siemens
$C_1$ = 3.01 pF
BW = 0.2 MHz
$\Delta B$ = 0.0001253 siemens
$\beta$ = 0.0056 siemens
$\delta$ = 0.837

**Table 9**
**N = 4 *g* Parameters for a 16-20 MHz Matching Network**

| *1 Ω Prototype* | *Impedance Scaled* |
|---|---|
| $g_0$ = 1 | $g_0$ = 952 |
| $g_1$ = 1.20 | $g_1$ = 0.00126 |
| $g_2$ = 1.15 | $g_2$ = 1095 |
| $g_3$ = 1.60 | $g_3$ = 0.00168 |
| $g_4$ = 0.518 | $g_4$ = 493 |
| $g_5$ = 1.27 | $g_5$ = 0.00133 |

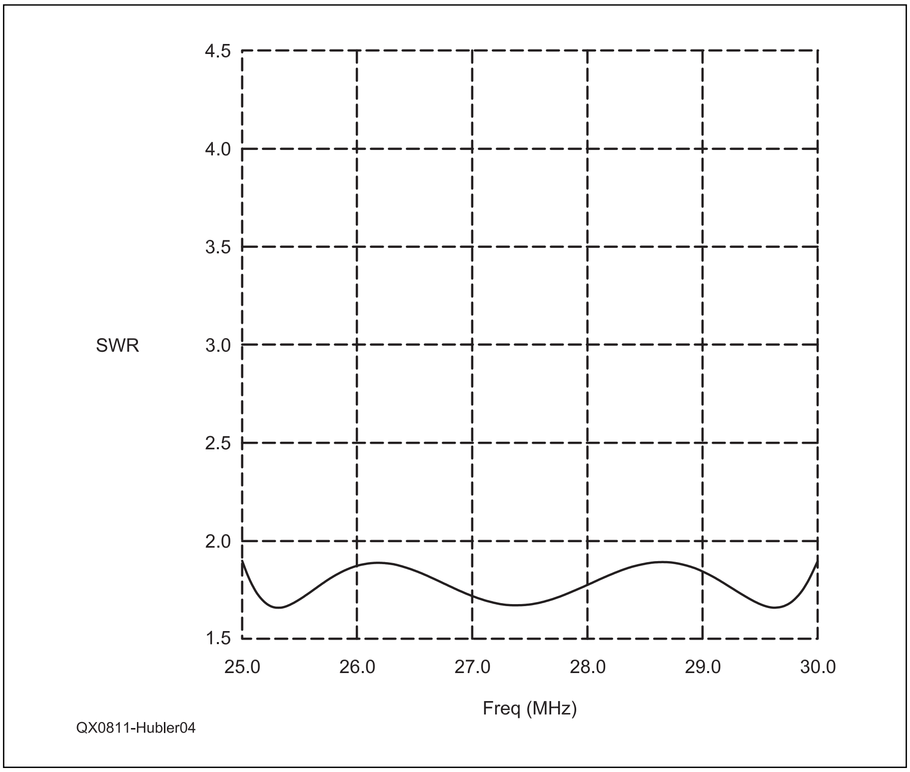



**Figure 4 — SWR versus frequency for the third order matching network, n = 3.**

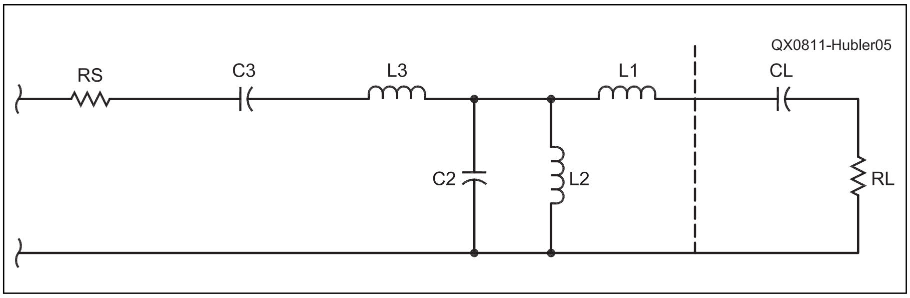



**Figure 5 — The third order matching network circuit.**

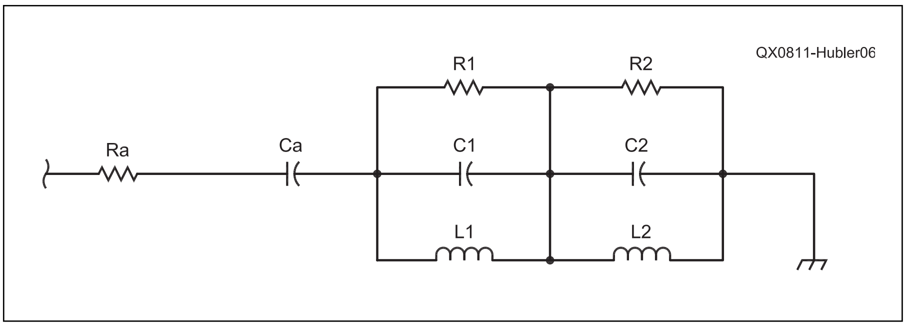



**Figure 6 — This schematic diagram represents the antenna simulator circuit.**

by determining whether the series equivalent resistance has a smaller percentage variation over the match band than the parallel equivalent conductance. If it does the load is more nearly series equivalent, otherwise it is more nearly parallel equivalent. It is also necessary to determine the equivalent load resistance (or conductance) over the match band. In the following examples, the average resistance (or conductance) over the match band was used. In some cases it may be preferable to simply use the series resistance or parallel conductance at $f_0$. Doing so will affect the network starting values and may make it somewhat easier or somewhat more difficult for the optimizer to determine the optimum network values.

The next step is to determine the reactance (or susceptance) slope parameter. This is accomplished by determining the series (or parallel) reactance required to resonate the load at $f_0$. The starting value of the first resonator element is then calculated to match this reactance. Then, with the aid of a circuit analysis program, the change in reactance, $\Delta X$ (or susceptance, $\Delta B$) over a small band including $f_0$ is determined. A 200 kHz bandwidth (BW) was used for this purpose. Once this has been determined, the reactance slope parameter is calculated using Equation 5 for series equivalent loads, or Equation 6 for parallel equivalent loads. The load decrement is then calculated using Equation 7 for series equivalent, or Equation 8 for parallel equivalent loads.

### 8 to 10 MHz Band

The above steps will now be applied to the antenna simulator load for the 8 to 10 MHz band for the design of a fourth order matching network. An inductor of 1.675 µH is required to resonate the load at $f_0$, and a bandwidth of 0.2 MHz was used to numerically calculate the reactance slope parameter, yielding the results shown in Table 6.

Given the load decrement, the $g$ parameters for a 1 Ω prototype matching network can be read from Figure 4.09-8 (p 129) of the text listed in Note 2, and the impedance scaled by the load resistance. This gives the results shown in Table 7.

The network of Figure 9 is obtained from the impedance scaled $g$ parameters using Equations 9 and 10. Inductor values are in µH and capacitor values are in pF.

Next, a capacitive K impedance inverter is added between resonators 3 and 4 to obtain a source impedance of 50 Ω. Note that regardless of what method is used to determine the equivalent source impedance above, adding an impedance inverter to obtain the desired source impedance gives the optimizer another degree of freedom to obtain the optimum network that will not depend on how the equivalent source impedance

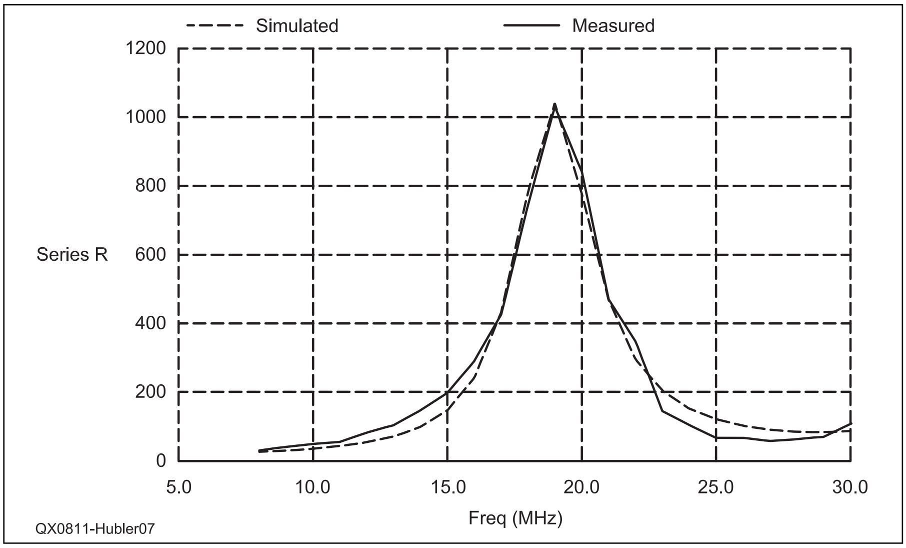



Figure 7 — Antenna series resistance versus frequency.

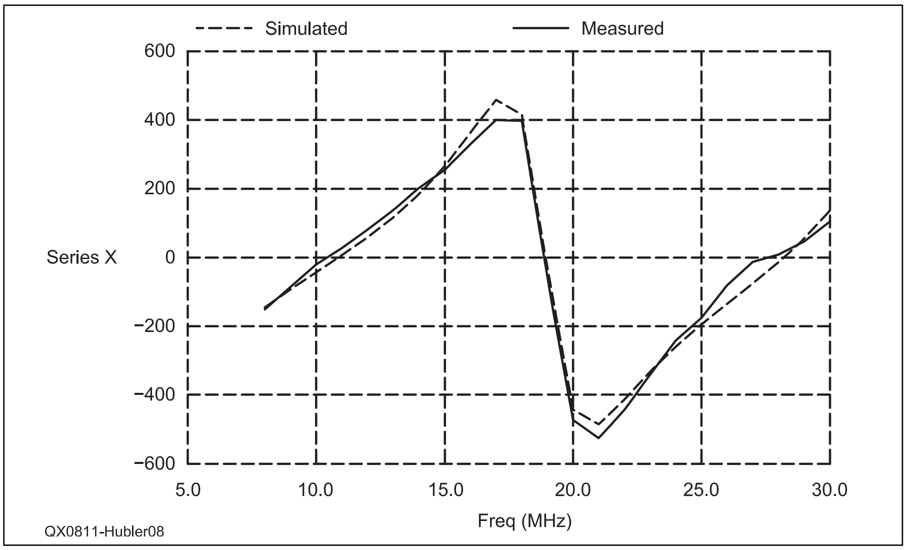



Figure 8 — Antenna series reactance versus frequency.

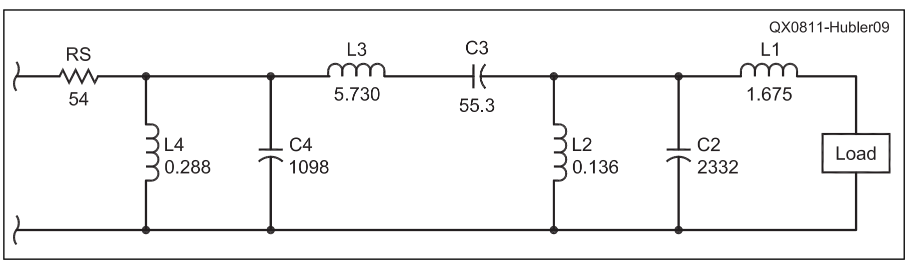



Figure 9 — Matching network for 8 to 10 MHz.

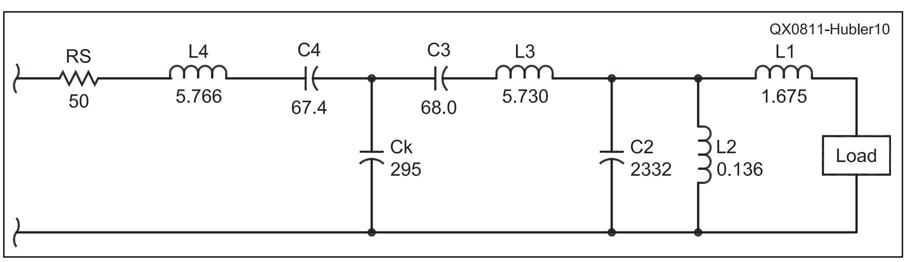



Figure 10 — Matching network for 8 to 10 MHz, with K inverter.

was calculated. The value of the K inverter is obtained from:

$$K_{i,i+1} = \sqrt{\frac{\chi_i \chi_{i+1} \omega_m}{g_i g_{i+1}}} \quad \text{(Eq 11]}$$

Applying Equation 11 to resonators 3 and 4 with a reactance slope parameter of 50 Ω for resonator 4 yields a K-inverter value of:

$K_{3,4} = 60.2\ \Omega$

Inserting an impedance inverter between resonators 3 and 4 changes resonator 4 from parallel to series, and $g_4$ becomes:

$g_4' = 1 / g_4 = 72.5\ \Omega$

yielding the network shown in Figure 10.

The SWR responses of the networks in Figures 9 and 10 are shown in Figure 11. Optimizing the network of Figure 10 to achieve minimum SWR in the 8 to 10 MHz band yields the network shown in Figure 12. The SWR responses of the optimized network terminated by the antenna simulator, and the antenna with no matching network are shown in Figure 13.

### 16 to 20 MHz Band

Repeating the above steps to design a 4th order matching network for the 16 to 20 MHz band yields the parameters given in Tables 8 and 9.

The network of Figure 14 is obtained from the impedance scaled g parameters using Equations 9 and 10.

A J impedance inverter could be added between resonators 3 and 4 to obtain a 50 Ω source impedance. This would, however, result in having a shunt resonator at the input. Instead, two K inverters will be added; one between resonators 2 and 3 with no impedance transformation, and the other between resonators 3 and 4 to obtain a 50 Ω source impedance and series input resonator. Applying Equation 11 to resonators 2 and 3 yields:

$K_{2,3} = 281\ \Omega$

$g_3' = 1 / g_3 = 595\ \Omega$

$g_4' = 1 / g_4 = 0.00203$ siemens

Applying Equation 11 to resonators 3 and 4 with $\chi_4$ equal to 50 Ω yields:

$K_{3,4} = 157\ \Omega$

Adding K-inverters derived from the values calculated above to the network in Figure 14 yields the network in Figure 15. The SWR responses of the networks in Figures 14 and 15 are shown in Figure 16.

Optimizing the network of Figure 15 to achieve minimum SWR in the 16 to 20 MHz band yields the network in Figure 17.

The SWR responses of the optimized network terminated by the antenna simulator, and the antenna with no matching network are shown in Figure 18.

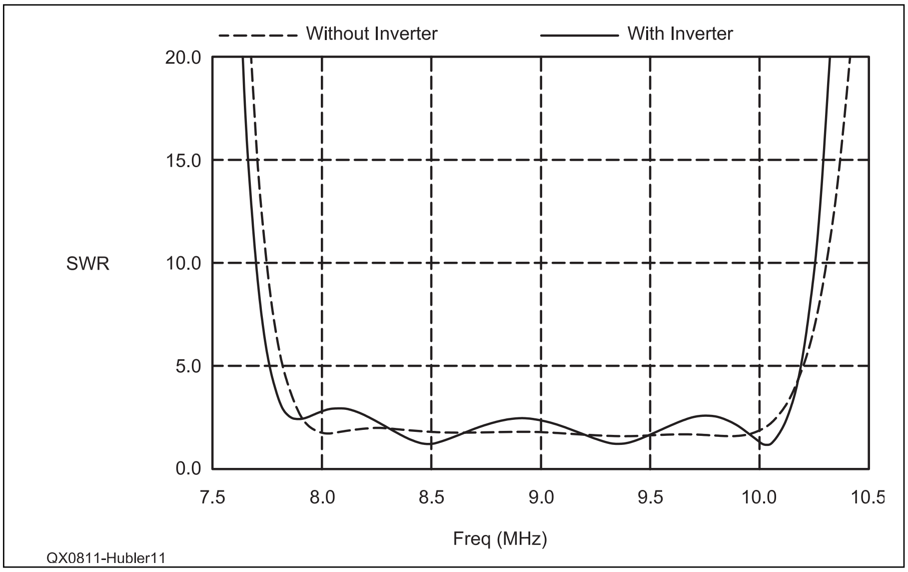



**Figure 11 — SWR Response of unoptimized networks.**

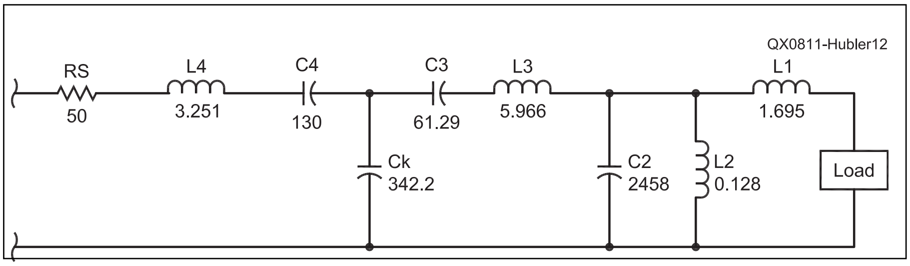



**Figure 12 — Optimized matching network for 8 to 10 MHz.**

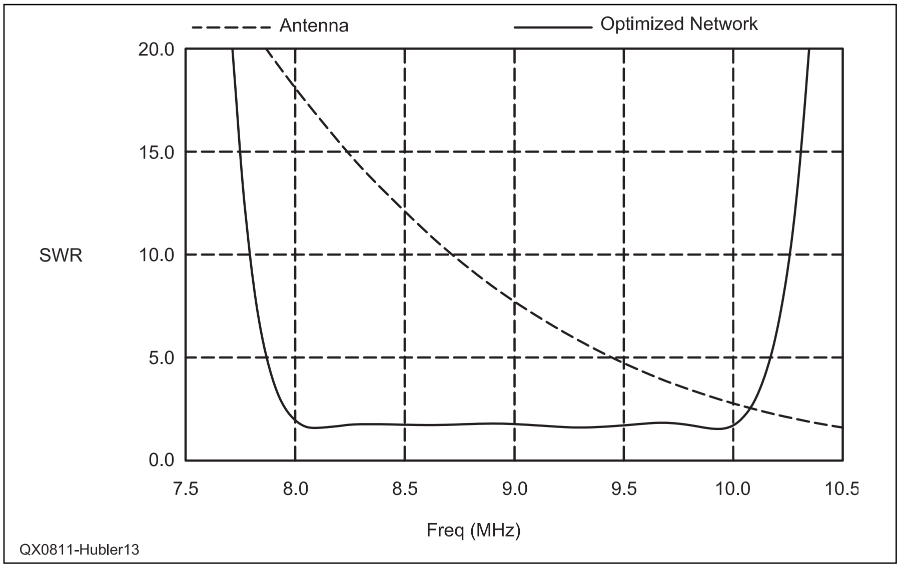



**Figure 13 — SWR Response of the optimized network and antenna.**

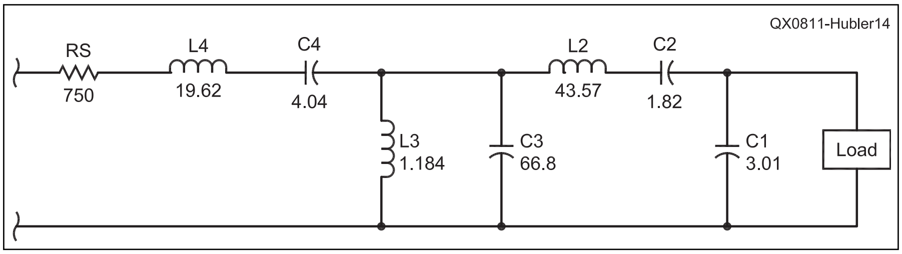



**Figure 14 — Matching network for 16 to 20 MHz.**

## Notes

[1]J. W. Bandler and R. E. Seviora, "Current Trends in Network Optimization," *IEEE Transactions On Microwave Theory and Techniques*, Vol MIT-18, December 1970.

[2]G. Matthaei, L. Young, and E.M.T. Jones, *Microwave Filters, Impedance Matching Networks, and Coupling Structures*, Artech House, 1980.

[3]R. M. Fano, "Theoretical Limitations on the Broadband Matching of Arbitrary Impedances," *Journal of the Franklin Institute*, Jan and Feb 1950.

*Fred Hubler earned a BSEE degree from Penn State University in 1969, and an MEIE degree from Texas A&M University in 1971. He was employed by the Department of the Army from 1969 to 1973, and by Rockwell Collins from 1973 until his retirement in 2002. He has many years of experience in the design of HF antenna couplers, and in performing communications system analysis, particularly dealing with mutual coupling between closely spaced antennas, or simultaneous operation (SIMOP) analysis.*

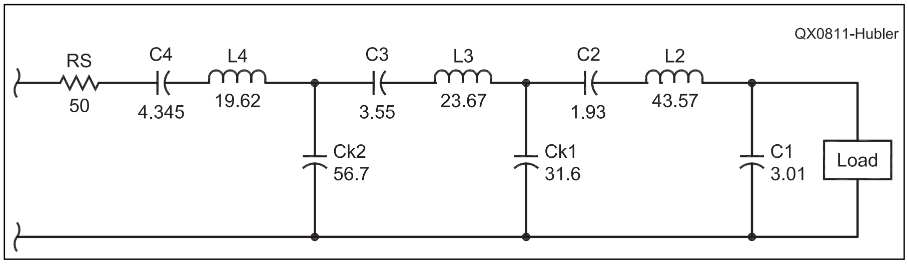



**Figure 15 — Matching network for 16 to 20 MHz, with Inverters.**

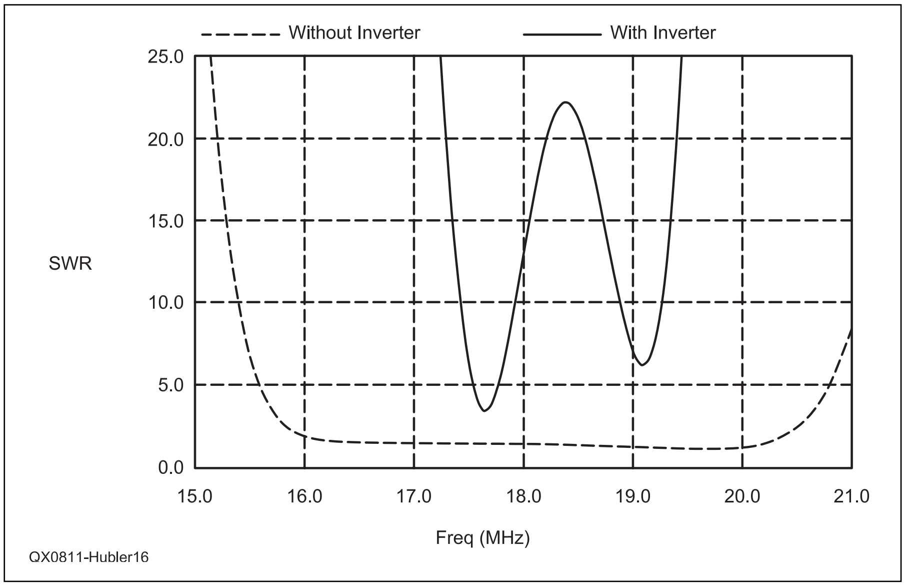



**Figure 16 — SWR Response of unoptimized matching networks**

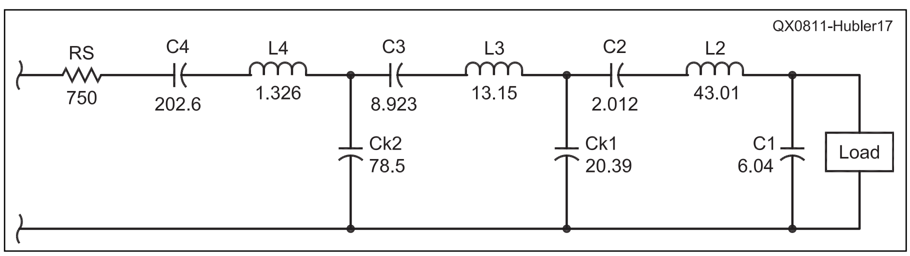



**Figure 17 — Optimized matching network for 16 to 20 MHz.**

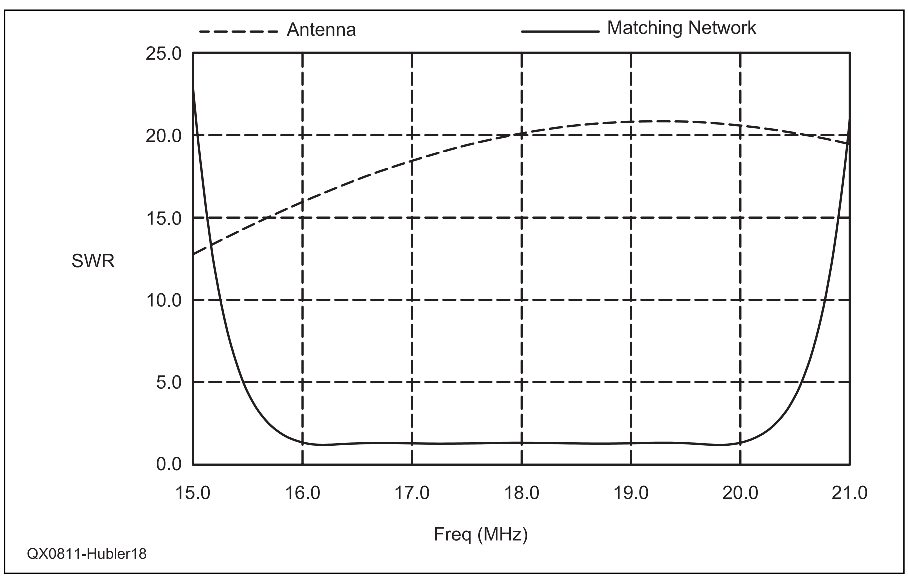



**Figure 18 — SWR response of optimized network and antenna.**

**George Murphy, VE3ERP**
77 McKenzie St, Orillia, ON L3V 6A6 Canada; ve3erp@rac.ca

**Robert J. DeHoney**
4602 Palm Blvd, Isle of Palms, SC 29451; rdehoney@comcast.net

# Dual Output Power Supply

*Need a positive and negative power supply for your next op amp or other project? Here is the answer.*

The unique circuit shown in Figure 1 provides very nearly equal positive and negative dc voltages with a common ground from an ac power transformer. Plus and minus outputs from an untapped transformer secondary normally require plus and minus half wave rectifiers, but unless the loads are balanced there will be a net field in the core, which might cause it to saturate. This double bridge design prevents that from happening. D1 to D8 form two bridge rectifiers. All capacitors are electrolytic. If C2 and C3 are considerably larger than C4, then V1 and V2 will be very close to the same at any load.

This circuit can be used for op amps and many other ICs, such as comparators and function generators. It was designed by Robert Dehoney, of IEEE, and verified by extensive testing and measurements of test-bench prototypes.

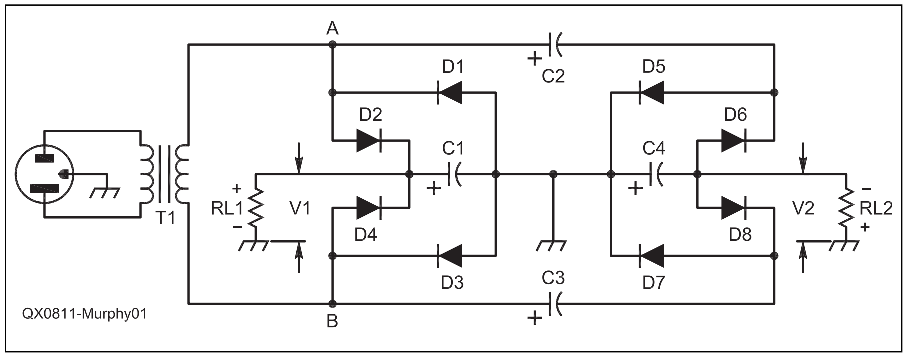



**Figure 1 — This schematic diagram shows the double bridge rectifier circuit.**

## How it Works

In Figure 2A, when point A is more positive than point B by a large enough margin, C1 charges through D2 and D3, C2 charges through D5 and D3, and C4 charges through C2, D5, D8 and C3. During this half cycle, voltage $V_{C4}$ follows $V_{C3}$ since $V_{C4} = V_{D3} + V_{C3} - V_{D8}$.

In Figure 2B, when point B is enough more positive than point A, C1 charges through D1 and D4, C3 charges through D7 and D1, and C4 charges through C3, D7, D6 and C2. During this half cycle, voltage $V_{C4}$ follows $V_{C2}$ since $V_{C4} = V_{D1} + V_{C2} - V_{D6}$.

## Design a Dual Output Power Supply

For computerized design using these equations, download *HamCalc* (version 103 or later) and run the "Power Supply — Dual Output" program.[1] If you prefer to do the math yourself, proceed as follows:

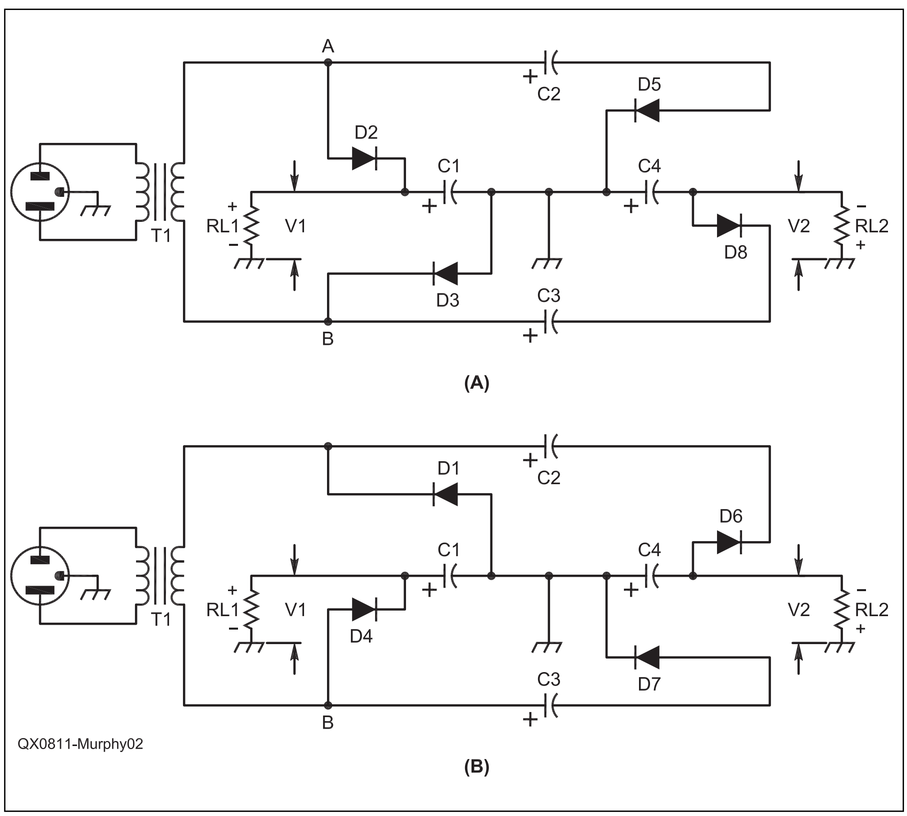



**Figure 2 — Part A shows the current path through the circuit when the top of the transformer is positive. Part B shows the current path when the bottom of the transformer is positive.**

[1]George Murphy, VE3ERP, *HamCalc* "Painless Math for Radio Amateurs." This free software is available for download at **www.cq-amateur-radio.com**.

a) Specify desired voltage (E) to RL1 and RL2 and current I1 and I2 through each, where:

E is in volts
RL1 and RL2 are in ohms
I1 and I2 are in amps.

b) Calculate the values of RL1 and R2:

RL1 = E / I1 [Eq 1]

RL2 = E / I2 [Eq 2]

where:

RL1 and RL2 are in ohms
E is in volts dc
I1 and I2 are in amps.

d) Specify the ac line frequency (F) in hertz.

e) Specify allowable peak-to-peak ripple voltage of V1 (across R1) and V2 (across R2).

f) Calculate the values of C1 and C4:

C1 = 375 × I1 × $10^3$ / F / R1 [Eq 4]

C4 = 375 × I2 × $10^3$ / F / R2 [Eq 5]

where:

C1 and C4 are in µF
I1 and I2 are in amps
F is in hertz
R1 and R2 are peak-to-peak volts.

g) Calculate the values of C2 and C3:

C2 = C3 = 3 × C4 (minimum) or 5 × C4 (recommended) [Eq 6]

where:

C2 and C3 are in µF.

h) Estimate the required transformer secondary voltage (TE) and current (TI):

TE = 2 + E / 1.41 [Eq 7]

TI = 1.8 × (I1 + I2) [Eq 8]

where:

TE is in volts
TI is in amps.

## Design Notes

1. Rectifier diodes should have a rating not less than 1.4 × TE volts at TI amperes.

2. All capacitors are electrolytic.

3. C1 can be any value not less than the calculated value of C1.

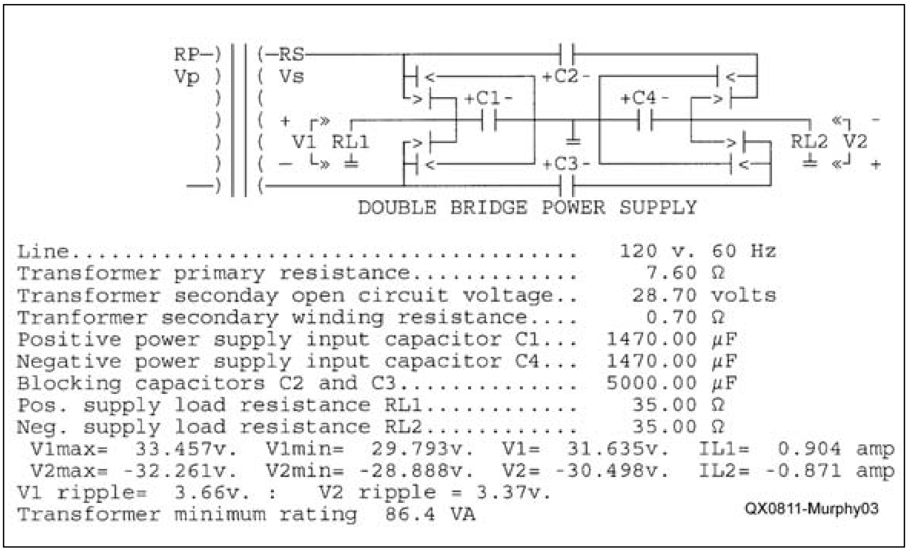



**Figure 3 — Here is the *HamCalc* printout for prototype number 1.**

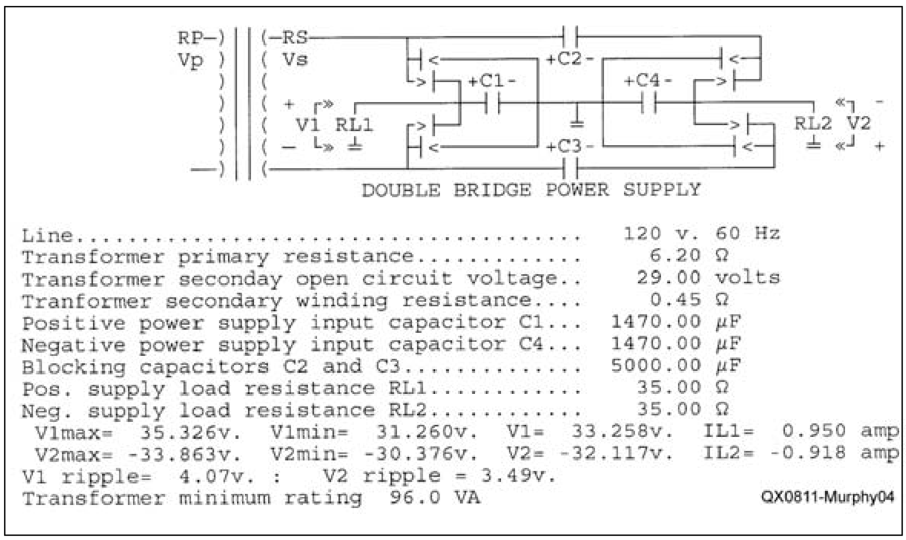



**Figure 4 — Here is the *HamCalc* printout for prototype number 2.**

**Table 1**
**Prototype Bench Test Values**

| PROTOTYPE | TRANSFORMER - 120 V. 60 Hz | | | | | C1 & C4 µF | C2 & C3 µF | CALCULATED VALUES | | | | MEASURED VALUES | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PRIMARY WINDING OHMS | SECONDARY WINDING OHMS | SECONDARY NO-LOAD VOLTS | LOAD RL1 OHMS | LOAD RL2 OHMS | | | V1 Volts | V1 Ripple volts | V2 Volts | V2 Ripple volts | V1 Volts | V1 Ripple volts | V2 Volts | V2 Ripple volts |
| # 1 | 7.60 | 0.70 | 28.70 | 35 | 35 | 1470 | 5000 | 31.7 | 3.7 | -30.5 | 3.4 | 32.2 | 2.7 | -30.9 | 2.9 |
| # 2 | 6.20 | 0.45 | 29.00 | 35 | 35 | 1470 | 5000 | 33.3 | 4.1 | -32.1 | 3.5 | 32.8 | 2.5 | -31.5 | 2.6 |
| # 3 | 40.50 | 1.00 | 13.76 | 150 | 150 | 1000 | 5000 | 16.4 | 0.70 | -16.0 | 0.69 | 16.1 | 1.7 | 15.7 | 1.7 |

4. C4 can be any value not less than the calculated value of C4.

5. C2 and C3 can be any value not less than 3 times the value selected for C4. (We recommend 5 or more times the value of C4, for minimum ripple.)

6. Select a transformer with a secondary voltage somewhat higher than TE and a current rating not less than TI.

7. When a suitable transformer has been found, you may verify the design and predict the actual V1 and V2 voltages using the *HamCalc* “Power Supply - Double Bridge” program or an equivalent program.

## Prototype Bench Test Results

Three prototypes were built and tested using salvaged junk box transformers and capacitors.

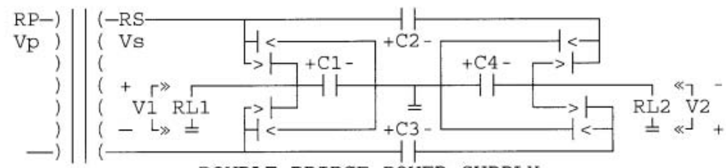
Figures 3 and 4 are identical except for the transformers. Unfortunately, there were no capacitors on hand to allow C2 and C3 to be 5 times the value of C4, so the V2 value is less than optimum. Figure 5 indicates a situation in which the capacitor values have a 5:1 ratio, and where V2 is within a few percent of V1.

Table 1 shows the component values of each prototype, the output values predicted by the *HamCalc* program, and the actual measured output values, assuming that the values of all capacitors are exactly as marked on the capacitor which, of course, never happens in the real world. Figure 6 shows how an oscilloscope might show the output in a perfect world.

This design is very flexible, limited only by your tolerance for precision and the contents of your junk box!

*George Murphy is a retired industrial designer and professional musician with no vocational RF engineering experience. Licensed as VE3ERP in 1960, his Amateur Radio hobby has led to his writing many articles for international Amateur Radio publications since 1985 and worldwide distribution of his HamCalc software since 1993. George is an ARRL member.*

*Robert Dehoney is a retired professional electrical engineer who worked for a defense company developing HF, VHF and UHF systems, enjoys analyzing and constructing useful circuits and has authored papers for* RF Design Magazine.




DOUBLE BRIDGE POWER SUPPLY

```
Line.....................................  120 v. 60 Hz
Transformer primary resistance...........   40.50 Ω
Transformer seconday open circuit voltage..  13.76 volts
Tranformer secondary winding resistance....   1.00 Ω
Positive power supply input capacitor C1...  1000.00 µF
Negative power supply input capacitor C4...  1000.00 µF
Blocking capacitors C2 and C3.............  5000.00 µF
Pos. supply load resistance RL1...........   150.00 Ω
Neg. supply load resistance RL2...........   150.00 Ω
 V1max=  16.433v.  V1min=  15.733v.  V1=  16.080v.  IL1=  0.107 amp
 V2max= -16.350v.  V2min= -15.657v.  V2= -15.981v.  IL2= -0.107 amp
V1 ripple=  0.70v. :   V2 ripple = 0.69v.
Transformer minimum rating   6.1 VA
```

QX0811-Murphy05

**Figure 5 — This *HamCalc* printout is for prototype number 3.**

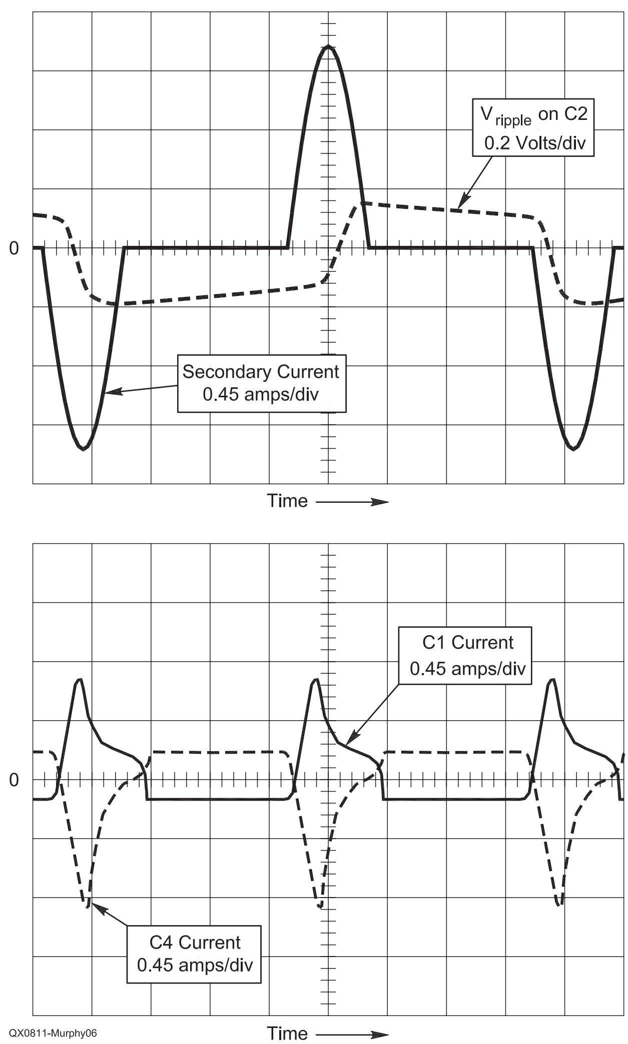



**Figure 6 — Part A shows the transformer secondary current and the ripple voltage on C2. During the positive current pulse, the voltage on C2 increases a bit, and during the negative current pulse the voltage decreases. This ripple is riding on the 40 or so volts dc across C2. Part B shows the current into and out of C1 and C4. The positive going pulse is across C1. You can see the capacitor charging for about 25% of the time and discharging for about 75% of the time. In both cases, RL1 = RL2 = 220 Ω.**

QEX

**Bob Raffaele, W2XM**

5 Gadsen Court, Albany, NY 12205; raffaele@nycap.rr.com

# The Mystery of the Q-section... Revealed by a Science Teacher

*Antenna gurus often explain how a coaxial impedance transformer works by referring to the venerable Smith Chart. Wonderful predictor that it is, this invention from 1939 can tell you what to expect, but it may leave you wondering how the impedance transformation is actually accomplished.*

As we play with our antenna systems, we conscientiously try to prevent reflections and the resulting standing waves. The SWR bridge is installed at the transmitter, and the only reflections that we typically observe are those that travel back to that meter. If we use matching sections, open-wire feeders with tuners, or stubs, we actually *cause* helpful reflections that cancel undesired ones! By design, we have standing waves along the matching section, along the stub, or between the tuning device (stub or tuner) and the antenna feed point.

Any transmission line that is not terminated with a resistance equal to its own characteristic impedance ($Z_0$) will serve as an impedance transformer. Design parameters can be found by using the Smith Chart. The Smith Chart can offer specifications because it can reveal the relationships among four variables of a piece of cable: characteristic impedance, length, and the two impedances (generally complex) at its ends.

I'd like to explain the operation of a very particular transmission-line match, the Q-section. The principle herein described — matching by canceling reflections — is universal. I chose a Q-section for my example because it is simple and popular, and because I needn't refer to complex impedances while I make an explanation.

The design of the most common Q-section is well known: a quarter wavelength of cable inserted between a 50 Ω feed line and an approximately 100 Ω antenna (perhaps a loop). The inserted cable has a characteristic impedance of 73 Ω, near the geometric mean of the values of impedance at its two ends. Expressed as a formula:

$$Z_0 = \sqrt{Z_1 \bullet Z_2} \qquad \text{[Eq 1]}$$

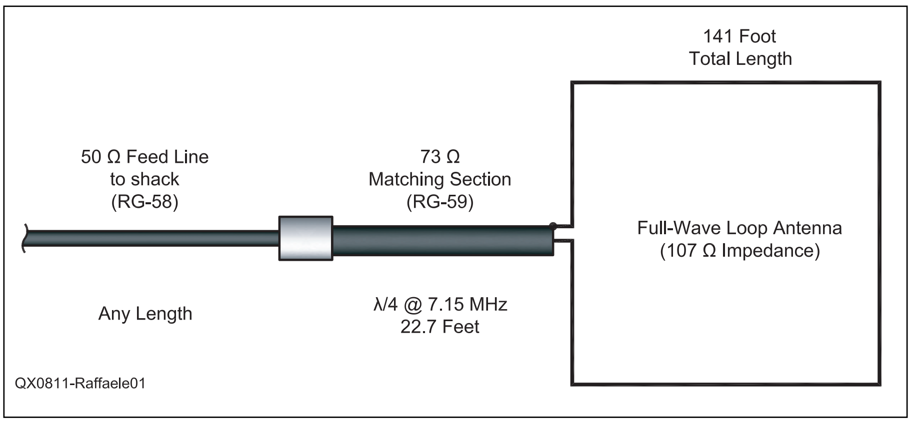



**Figure 1 — Antenna feed line matching system as described in the text. The length of the ¼ λ matching section assumes a velocity factor of 0.66 for the cable.**

where $Z_0$ = the characteristic impedance of the matching section
$Z_1$ = the characteristic impedance of the feed line
$Z_2$ = the antenna impedance.

This applies to the special case in which all $Z$ values are pure resistances. This requirement is met when the antenna is resonant and the transmitter output circuitry is properly adjusted.

We will be talking about Figure 1, which shows a loop antenna fed by RG-58 feed line through a ¼ λ matching section of RG-59 coax. Let's suppose that this particular loop has a feed point impedance of 107 Ω. That value is reasonable, and it satisfies Equation 1. The dimensions shown are for a 40 m loop, cut to 7.15 MHz.

Visualize a waveform riding toward the right on the RG-58 coax until it encounters the RG-59; then it is partially reflected and partially transmitted. ("Transmitted" here means "propagated across the boundary." Keep this definition in mind!) The reflected part goes back toward the source, and the nonreflected portion continues along the RG-59 for ¼ λ until it meets the antenna, from where part of it continues and part of it reverses direction. This returning part travels ¼ λ back toward the source, and is, for the third time, partially reflected and partially transmitted, this time at the boundary with the RG-58. This signal going back toward the transmitter is weaker than the first and opposite in phase. There is some cancellation.

The waveform goes — continuously and "forever" — back and forth along the matching section (RG-59). Each time, it is partially reflected and partially allowed through at each end. Theoretically, an infinite number of interfering wave disturbances comes back to the junction of the feed line and matching section. The wave-like disturbances are sequentially smaller, but they are continuously produced and constantly superimposed, and all of these wave portions are moving simultaneously. The process may be difficult to visualize, but the result is very practical.

With a matching section of the right length, the innumerable wave segments going back toward the transmitter have an essentially zero sum, and all of those going toward the antenna add constructively. "No net signal going back down the feed line" is a necessary and sufficient condition for a match. As a consequence of having coherent portions of our signal moving in opposite directions within the matching section, there is a standing wave and there is some loss; however, I can't imagine the case in which this would be unacceptable.

The preceding can easily be understood more quantitatively. See Figure 2, which looks like something I would put on a classroom board. There are two vertical lines, which divide the illustration into three impedance environments. The left-most one is the RG-58 feed line, with a 50 Ω impedance; the middle one is the 73 Ω matching section (RG-59); and the one on the right is the 107 Ω antenna.

The signal path is indicated by the zig-zag line. The signal is not really localized into lines, nor does it zig-zag. It may be useful to follow the line, however, and to think of the events encountered along the way as being sequential. As you read what is happening to our initial incident wavefront, keep in mind that it's not really the first of anything. After a few cycles, you might say that the system has achieved its steady state; then you could say that a wave's propagation in the cable, its splitting into two at the interfaces, and the emergence from the Q-section of its infinite number of portions are all simultaneous.

Each time the wave encounters an interface, such as at points A, B, C, D, E, F, G..., it splits into two portions. The points in the diagram denote boundaries between different impedances; they are actually more like cross-sections than points, and I will call them interfaces. With each reflection, voltage decreases by a factor of ρ, the reflection coefficient. With each transmission (energy going through the boundary), the signal's amplitude (voltage) becomes either smaller or larger. [The power of the transmitted wave is always smaller than that of the incident wave by the factor $(1 - \rho^2)$. The corresponding voltage depends on the new values of both power and impedance.] All of the wave portions in the matching section are coherent and they can interfere with each other. That is what makes the whole system function.

As seen in Figure 2, the uppermost straight line, representing the incident wave going through the matching section to the antenna, is both transmitted and reflected at each interface. We say that the wave portion incoming at A has an amplitude of 1. We'll call its reflection, which has amplitude ρ, the "first reflected wave." There is no phase change upon reflection because $Z_0$ is greater than $Z_1$. Immediately to the left of the interface, in the diagram, both the incident wave and the "first reflected wave" have a phase that we will call "+."

As I have already stated, a zero sum for the reflected wave portions indicates an impedance match. Simultaneously with its reflection, the incident wave propagates across the boundary into the matching section. It travels ¼ λ to the junction with the antenna, where it is reflected (and also sent across the boundary to the antenna). The reflected portion travels ¼ λ back toward the feed line; and it is both reflected and transmitted at the junction with the feed line. The part that is transmitted emerges from the 73 Ω environment traveling to the left. This is the first of countless reflected wave portions that tend to cancel the first reflected wave.

You should probably glance at Figure 2 frequently as you finish reading this. Taking it from the top, the incident wave front has an amplitude of ρ upon reflection. The portion that travels from A to B to C, where it emerges, is decreased in amplitude at three places. At the entrance to the RG-59, the amplitude goes down by a factor of $\sqrt{(1 - \rho^2)}$; where the wave portion is reflected from the antenna, it is reduced by a factor of ρ; and where it crosses over into the feed line, it is again made smaller by a factor of $\sqrt{(1 - \rho^2)}$. Because it travels ½ λ farther than the first reflected wave (making the round trip along the ¼ λ matching section), it has "–" phase compared to the first reflected wave. You can see that it is a wave portion of amplitude $-\rho(1 - \rho^2)$ moving toward the shack, along with the wave portion of amplitude ρ. The explicit minus and plus signs in front of $\rho(1 - \rho^2)$ and ρ indicate that the two are always opposite in phase. There is partial cancellation.

As a signal leaves the RG-59 at C, it is both reflected and transmitted there; it is reflected (and transmitted) at D, and emerges at E with amplitude $\rho^3(1 - \rho^2)$ and with the correct phase to partially cancel our "first

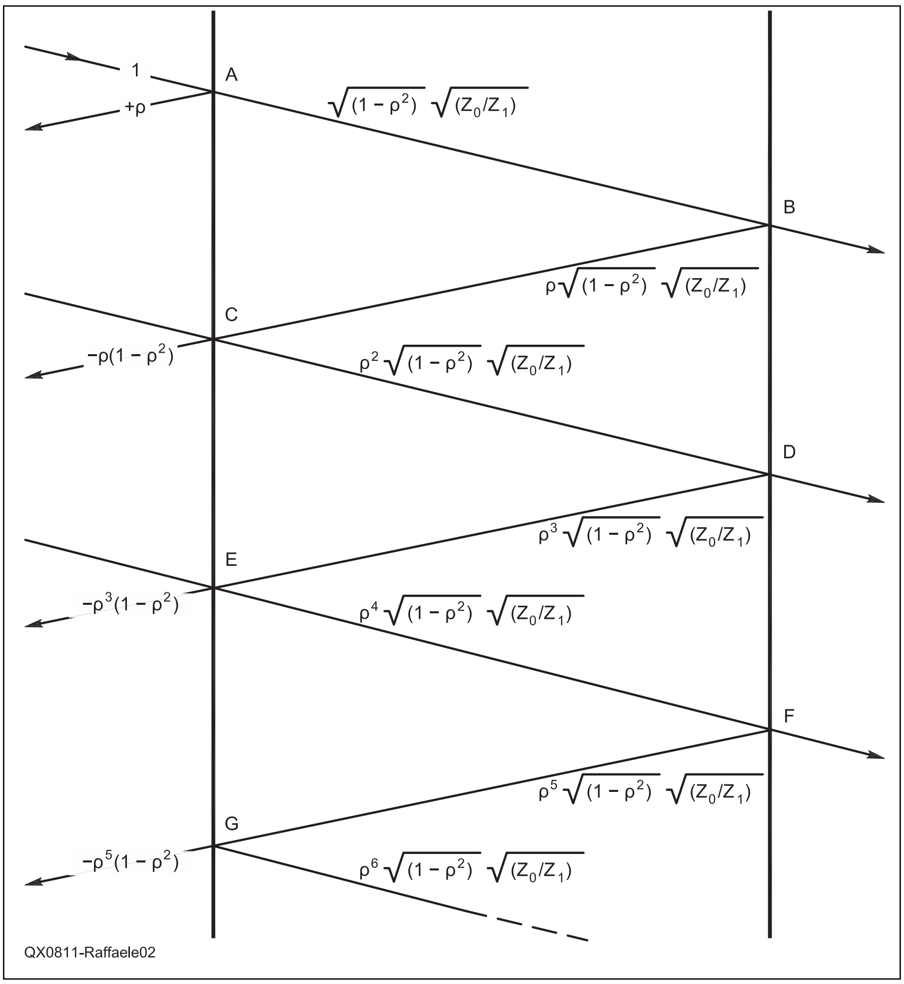



**Figure 2 — A mind's eye view of the inner workings of a ¼ λ matching section. The labels adjacent to the path segments indicate the amplitudes. The factor $\sqrt{Z_0/Z_1}$ is required because the signal coming from the $Z_1$ environment has been assigned an amplitude of 1.**

reflected wave." Then there is another portion with the same phase to help cancel the first reflected wave; its amplitude is $\rho^5(1 - \rho^2)$, and another, with amplitude $\rho^7(1 - \rho^2)\ldots$

And so it goes. The sum of the amplitudes of all the waves traveling toward the transmitter is:

$$+\rho - \rho(1 - \rho^2) - \rho^3(1 - \rho^2) - \rho^5(1 - \rho^2) - \rho^7(1 - \rho^2) \ldots = 0 \qquad \text{[Eq. 2]}$$

Simple expansion shows Equation 2 to be true as long as $\rho$ is less than 1. There is no net reflected signal! … Therefore we have a perfect match, thanks to the magic of the Q-section.

I appreciate the help of Professor John Browne, WA2VDH, who offered suggestions and encouragement as he conscientiously reviewed several versions of this paper.

*Bob Raffaele, first licensed while in seventh grade in 1962, holds BS and MS degrees in physics. He was working as a broadcast-station chief engineer until suffering disabling injuries in 1977. After some rehabilitation, Bob began teaching physics at a community college. Now, unable to work full-time, he substitute teaches science and math at Niskayuna (NY) High School. Bob has been an ARRL member for the last 45 years.* QEX

**William E. Sabin, WØIYH**

1400 Harold Dr SE, Cedar Rapids IA 52403; w.sabin@mchsi.com

# A Modern Discrete-Method for Signal Analysis and Design

*This article shows how the Mathcad program can be used to analyze a circuit response to various input signals.*

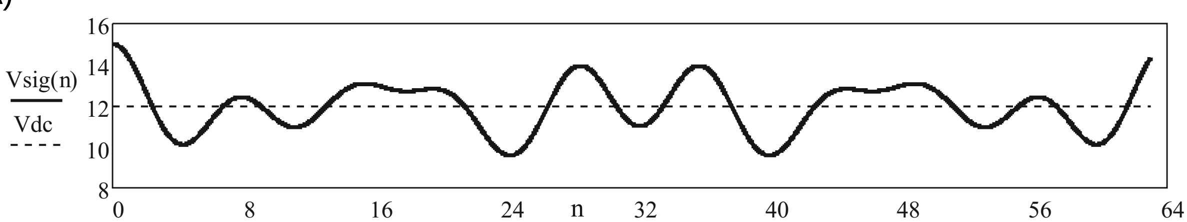
The modern personal home computer, in conjunction with an elegant and sophisticated mathematics calculation program — *Mathcad* Version 14.0 — is employed in this brief article at an introductory level that is very user-friendly.[1] This program is used to calculate, process and graph-plot a variety of signal-processing and very many other kinds of math problems. The so-called *symbolic* math methods — see the *Mathcad* User Guide, Chapter 13 — are also used in modern math computing. This software is an outstanding tutorial tool for the advancement

[1]Notes appear on page 38.

$N := 64 \qquad n := 0, .01 .. N-1 \qquad k := 0, 1 .. N-1 \qquad Vdc := 12 \qquad Gv := 2$

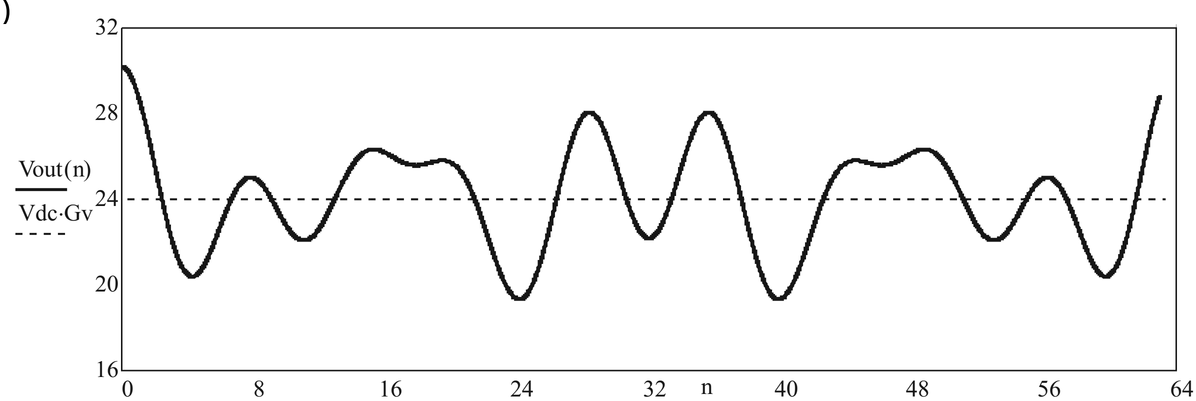
$$Vsig(n) := Vdc + \cos\left(2\cdot\pi\cdot\frac{n}{N}\cdot 4.0\right) + \cos\left(2\cdot\pi\cdot\frac{n}{N}\cdot 7.0\right) + \cos\left(2\cdot\pi\cdot\frac{n}{N}\cdot 9.0\right) + .001\cdot\cos\left(2\cdot\pi\cdot\frac{n}{N}\cdot 29.0\right)$$

(A)




$$Vout(n) := Gv\cdot Vsig(n)$$

(B)



of the engineer's and student's math skills, which is considered to be an important goal in today's advanced technology environment.

The signals and their analyses can be in the time domain or the frequency domain, and they can be linear or nonlinear in nature. Signals, before and after processing, can be switched back and forth easily between time domain and frequency domain. A complete *Mathcad* 14.0 program, with no time limits on its use, is included with my new book *Discrete-Signal Analysis and Design*.[2] This is a special and very generous promotional offer by PTC, the company that publishes *Mathcad*. The User Guide that is on the included CD can be placed on your computer Desktop as an icon.

## Example: A Three-Tone Distortion Simulation

A typical discrete-signal processing example will help to illustrate just a few of the basic ideas for using *Mathcad* in the practical world of electronic design. This and similar examples can also be further explored with circuit simulation programs such as *Multisim* (student edition) by National Instruments (**www.ni.com/academic/multisimse.htm**) using accurate models of ICs, transistors, diodes, tubes and many kinds of passive components.

In our example, we will analyze a circuit response to three desired tones. We will also add in a large undesired "out-of-band" signal that contributes to distortion products.

Time domain information can be converted to spectrum results. After modifying the frequency spectrum, for example by introducing a filter, an interfering signal, or random noise (additive or multiplicative), the modified time domain results can then be obtained. We can also *begin* with various spectrum shapes that achieve certain desired time-domain results.

Certain communications waveforms use methods similar to this example, and laboratory test equipment (**www.ni.com/analysis/**) is used to perform discrete-time and discrete-frequency tests automatically. Pseudorandom data error rates can be evaluated.

## Analysis of the Example

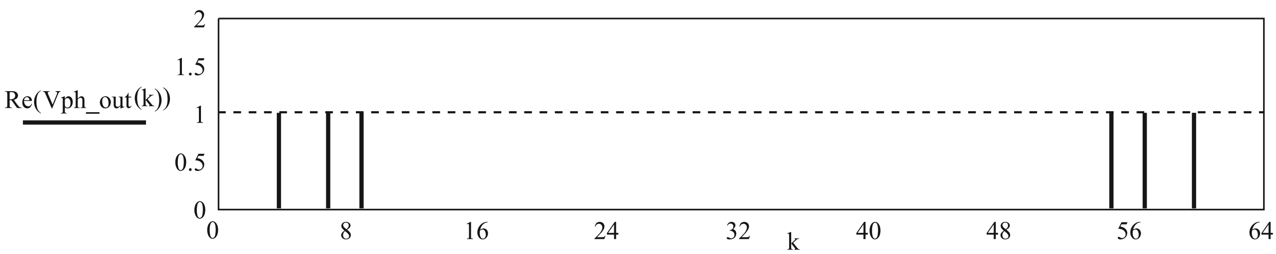
Part (A) shows the time-domain input Vsig(n) and part (B) the time-domain output Vout(n) of an amplifying device that is perfectly linear. The device delivers only output signals that are proportional to those at the input, for example, $Vout(n) = Vsig(n)^{1.0}$ times a voltage gain constant Gv. The signals in Part (A) are at frequencies 4, 7 and 9. Another much smaller input at frequency 29 will be considered a little later. (Notice that we haven't assigned any units to this scale. It could be Hz, kHz, MHz or any other arbitrary unit.) Observe also the dc bias voltage on the device. This is called the "operating point."

Part (C) is the two-sided (positive-frequency and negative-frequency) *phasor* spectrum of the output as calculated by the Discrete Fourier Transform (DFT) of the time-domain output (part B) of the device. Part (D) shows the same actual positive-frequency *signals* as the input. We get these positive-frequency signals by combining the positive-frequency phasors and the negative-frequency phasors. Chapter 2 of my book explains how this is done. Also, Chapter 1 discusses *phasors* versus *signals*.

Part (E) introduces a nonlinearity, the exponent 1.5, which is often used in text books [see Sabin Chapter 2], in the transfer function of the device. This creates distortion products in the form of new frequencies at the output, Vout(n), that are not present in the input, Vsig(n). In the signal spectrum, Vout(k) of part (F), many, but not all, of the nonlinear output products are identified. For example, the term at (k) = 3 is due to the term at (k) = 4 interacting with the term at (k) = 7. The 1.5 exponent also increases the voltage

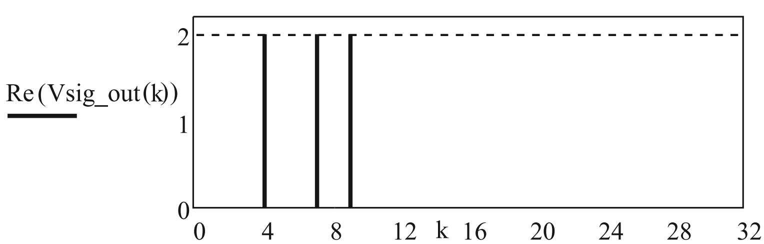
(C)

$$Vph\_out(k) := \frac{1}{N} \cdot \sum_{n=0}^{N-1} \left( Vout(n) \cdot \exp\left( -j \cdot 2 \cdot \pi \cdot \frac{n}{N} \cdot k \right) \right)$$




(D)

$$Vsig\_out(k) := Vph\_out(k) + Vph\_out(N - k)$$

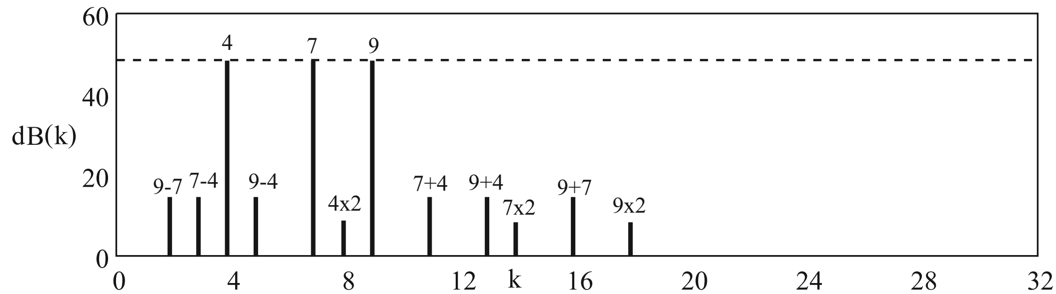


(E)

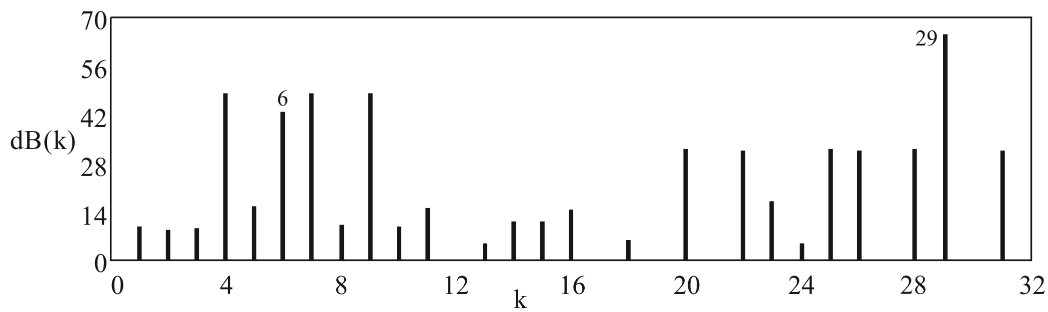
$$\mathrm{Vout}(n) := (\mathrm{Gv}\cdot\mathrm{Vsig}(n))^{1.5} \qquad \mathrm{Vph\_out}(k) := \frac{1}{N}\cdot\sum_{n=0}^{N-1}\left(\mathrm{Vout}(n)\cdot\exp\left(-j\cdot 2\cdot\pi\cdot\frac{n}{N}\cdot k\right)\right)$$

$$\mathrm{Vsig\_out}(k) := \mathrm{Vph\_out}(k) + \mathrm{Vph\_out}(N-k) \qquad \mathrm{dB}(k) := 20\cdot\log(|\mathrm{Vsig\_out}(k)|) + 25$$

(F)




(G)




gain, in this particular example, to $\mathrm{Gv}^{1.5}$. Gv can also be a more complicated time-varying function of (n) such as $[\mathrm{Gv}(n)]^{1.5}$.

Another consequence of the nonlinearity is that the spectrum *phasors*, and therefore also the *signals*, are "complex" and may possibly display items that have a real (Re) part, an imaginary ($\pm j$ Im) part, a magnitude (| |) and phase angle ∡ with respect to some "reference" phase, such as zero. The graphs of parts (F) and (G) would identify these. *Mathcad* can be instructed to calculate and plot all of these results.

In part (G) the interfering "out-of-band" signal at (k) = 29 is greatly increased in amplitude so that distortion terms are emphasized. The ability of a strong out-of-band interferer to corrupt a desired signal spectrum is illustrated. In particular, the spurious product at (k) = 6 stands out. In addition to this large distortion term, there are many smaller distortion products that can degrade the desired weaker signals. Also, the original input signals at 4, 7 and 9 are degraded slightly. This is very serious interference that can be difficult and expensive to repair, especially in wideband high-level systems where filtering of strong interfering signals is expensive. A tunable notch filter is sometimes possible for constant interferers, but costly high dynamic range equipment is often indicated. We see this effect in practical environments, for example in HF/VHF radio. We now see it also mathematically.

Beyond this simplified introduction, a much more complete study would be well-justified. For a closer look at the various related technologies for this and many other topics, visit the new ARRL website **www.wedothat-radio.org**.

**Notes**

[1]*Mathcad*, version 14.0, PTC company (**www.ptc.com/products/mathcad/**), Needham MA. The *Mathcad* program is very mature and has a history of fourteen versions in more than 25 years.

[2]William E. Sabin, *Discrete-Signal Analysis and Design*, 2008, Wiley, (**www.wiley.com**) Interscience Division, available from your local ARRL dealer, or from the ARRL Bookstore, Order no. 0140. 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**.


*Bill Sabin received the call sign W9YFA in 1941 in Covington, KY at age 15. This changed to W4YFA in 1946, and to W0IYH in Iowa (Collins Radio) in 1964. He holds BSEE and MSEE degrees from the University of Iowa. He retired from the Rockwell Collins Company in Cedar Rapids IA in 1990. He is co-editor and contributor to, with E.O. Schoenike, three books on Single-Sideband and HF Radio, and author of more than 40 articles on technical topics, including portions of The ARRL Handbook, 1995 to 2009 editions. In 1983 he received the annual ARRL Technical Excellence Award. Bill is an ARRL Member and a Life Senior Member of the IEEE, and he expects soon to become a member of the ARRL DXCC Honor Roll.*


QEX

**Robert Zimmerman, VE3RKZ**

54 Radford St, Hamilton, ON L8S 3E6, Canada; rkzimmerman@gmail.com.

# Offset Matching Of Yagi-Uda Beams

*Here is a 2 meter Yagi antenna that uses an offset-fed driven element.*

Everyone knows the impedance of a half-wave dipole in free-space is 73 Ω. When a dipole is used as the driven element in a Yagi beam, the impedance is much lower, depending on:

1) the number of parasitic elements,
2) the spacing between elements, and
3) the diameter of the elements.

For this reason, some form of matching is always used for 50 Ω coaxial cable — usually a Gamma match. With this article, I would like to suggest what I believe is an alternative, simpler matching technique for VHF/UHF beams: offset matching.

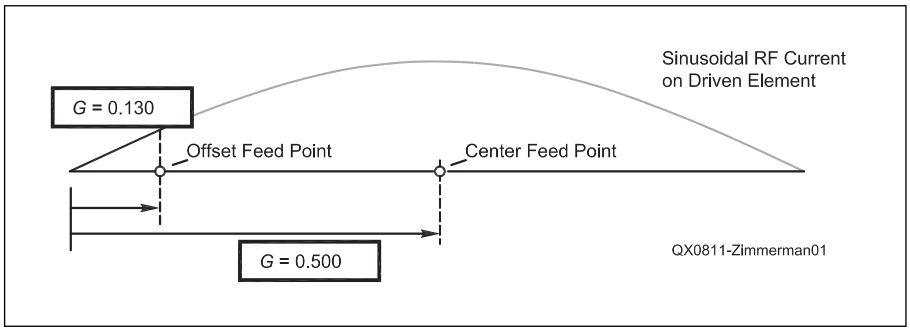



**Figure 1 — The offset parameter, *G*, is specified as a fraction of the entire length of the driven element. The design in this article uses *G* = 0.130, while a center fed driven element would use *G* = 0.500.**

## How it Works

Usually, even without thinking, we reference the impedance of a dipole at the center of the dipole. This is where the sinusoidal current is highest, resulting in the lowest possible impedance. If we place the feed point somewhat off center, then for the same applied power, the current will be less, meaning a higher impedance.

Suppose $Z_{center}$ is the center impedance. If we specify the offset parameter $G$ as a fraction of the entire length, as in Figure 1, then the approximate impedance at position $G$ is:

$$Z_{offset} = \frac{Z_{center}}{\left[\sin\left(G \times 180°\right)\right]^2} \quad \text{[Eq 1]}$$

Note that for the center feed position with $G = 0.5$, the formula yields simply $Z_{center}$, as it must.

## Modeling with *EZNEC*

As a demonstration of this method, I modeled the K1KO 2-meter beam dimensions for a 144.2 MHz, five-element Yagi beam using *EZNEC*.[1, 2, 3] The first five element lengths and spacings are given in Table 1. [The author provided all length measurements in mm. We converted those measurements to inches, using decimal fractions of an inch. Readers can use the Metric system measurements, as provided by the author or the US Customary units with an engineering scale. — *Ed.*] I used ½ inch (12.7 mm) OD diameter elements. *EZNEC* gave the center impedance of the driven element as $Z_{center}$ = 7.13 – $j$24.11 Ω. The negative reactance indicates that we are below resonance, and the driven element must be lengthened for operation at 144.2 MHz. With a little experimentation, I found the zero-reactance length is 39.3 inches (998 mm), and $Z_{center}$ = 9.467 Ω.

[1]Notes appear on page 41.

Then I moved the feed point from $G$ = 0.500 to $G$ = 0.130, where the offset impedance is $Z_{center}$ = 49.56 – $j$1.2 Ω. Rather than engaging in further experimentation to eliminate the –$j$1.2 Ω reactance, I decided to proceed. We can check *EZNEC* by using Equation 1. Entering $Z_{center}$ and $G$, we have:

$$Z_{offset} = \frac{9.467\,\Omega}{\left[\sin\left(0.130 \times 180°\right)\right]^2} = 60\,\Omega \quad \text{[Eq 2]}$$

**Table 1**
**Offset Feed Dimensions for 5 Element 144.2 MHz Yagi**

| *Element Name* | *Element Position (Inches)* | *Element Position (mm)* | *Element Length (Inches)* | *Element Length (mm)* |
|---|---|---|---|---|
| Reflector | 0 | 0 | 41.10 | 1044 |
| Driven Element | 12.28 | 312 | 37.60 | 955 |
| Director 1 | 17.60 | 447 | 37.87 | 962 |
| Director 2 | 27.52 | 699 | 36.93 | 938 |
| Director 3 | 41.34 | 1050 | 36.30 | 922 |

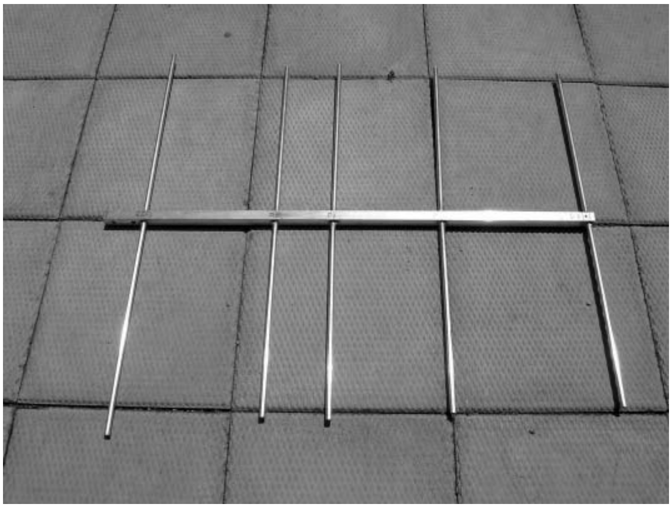


**Figure 2 — This photo shows the construction of the five element Yagi. The reflector element is to the left of the photo. All of the elements have been heli-arc welded to the boom.**

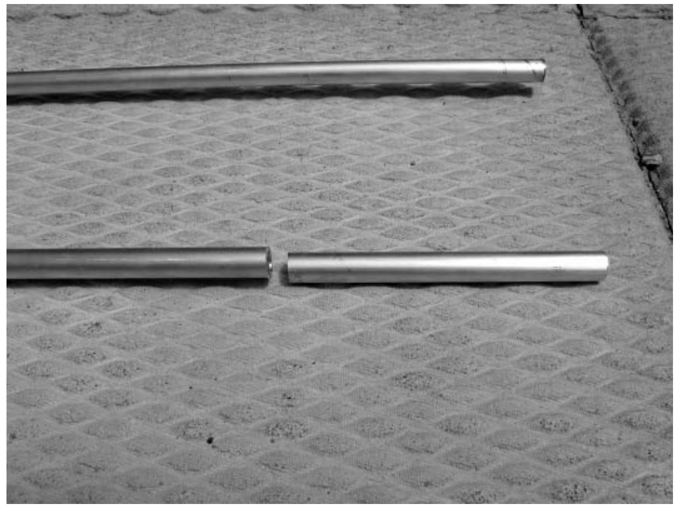


**Figure 3 — A 5.1 inch (130 mm) length of the driven element is cut from one end in preparation for forming the feed point.**

**Figure 4 — This photo shows the 1.2 inch (30 mm) nipple cut from a piece of ¼ inch aluminum tubing, which will be used to join the two sections of the driven element. Each side of the driven element tubing has a lengthwise slit in the tubing, so the hose clamps can compress the tubing and secure it to the nipple.**

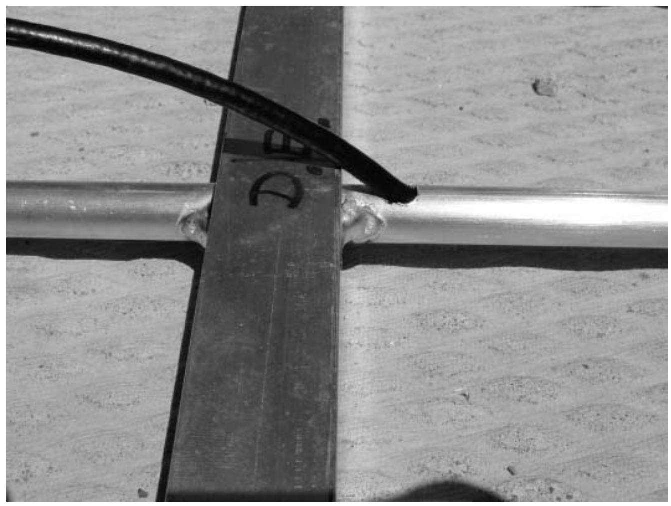


**Figure 5 — A hole in the driven element next to the boom allows the coaxial feed line to enter the tubing to reach the feed point. Drill the hole at an angle so the coaxial cable doesn't have to make a sharp bend as it goes through the tubing. Use a file to remove any burrs from the edges of the tubing, so you don't cut the coax outer sheath. Use silicone sealer to seal around the hole to reduce the amount of water entering the tubing.**

which is somewhat of a confirmation. Best to trust *EZNEC*! The modeled gain is 9.46 dBi.

## Building the Beam

I had 1 inch (25.4 mm) square aluminum tubing, which I used as a boom. I center-punched the five element positions, and carefully bored five ½ inch diameter holes with a hand drill.

I cut the ½ inch round aluminum elements to length as in Table 1, making no allowance for the boom. A local welding shop heli-arc welded the elements to the boom, placing a "bead" on both sides of the boom on each element. The beam is shown in Figure 2.

I cut off the driven element 5.1 inches (130 mm) from one end, as shown in Figure 3. Then the driven element and the 5.1 inch stub must be axially slotted with a hacksaw to accommodate hose clamps. Cut a 1.2 inch (30 mm) nipple from ¼ inch OD aluminum tubing. This will provide the mechanical coupling between the driven element and the stub, as shown in Figure 4.

Drill a hole in the driven element adjacent to the boom to feed the coaxial cable into the driven element. See Figure 5. It helps to drill this hole at an angle to make insertion of the coaxial cable easy.

Figure 6 shows the nipple on the end of the coaxial cable, with Teflon plumbing tape used for insulation. Be sure the Teflon tape fully insulates the end of the nipple where the braid is located. Stretch that tape over the end!

Figure 7 shows the insulated nipple ready for insertion into the end of the driven element. Note that: (1) the braid has been trimmed so as to fit only over the Teflon, and (2) the center conductor extends 3/8 inch (10 mm) beyond the nipple.

It may be necessary to slightly "open" the end of the driven element, so the insulated nipple can be inserted. Watch out for loose braid fibers, which could short out the Teflon insulation. Push the insulated nipple and braid into the driven element and secure with a hose clamp as Figure 8 shows.

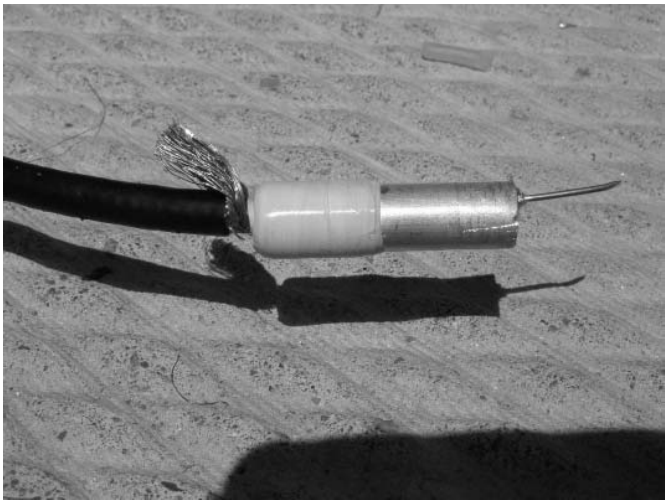


**Figure 6 — Wrap a heavy layer of Teflon plumbing tape over the nipple to insulate it from the inner length of the driven element tubing. Be sure to stretch the tape over the end of the nipple so it will be insulated from the coax braid. The braid has been trimmed so it won't extend beyond the Teflon tape when the assembly is inserted into the end of the driven element. There is about ⅜ inch (10 mm) of the center conductor extending beyond the open end of the nipple.**

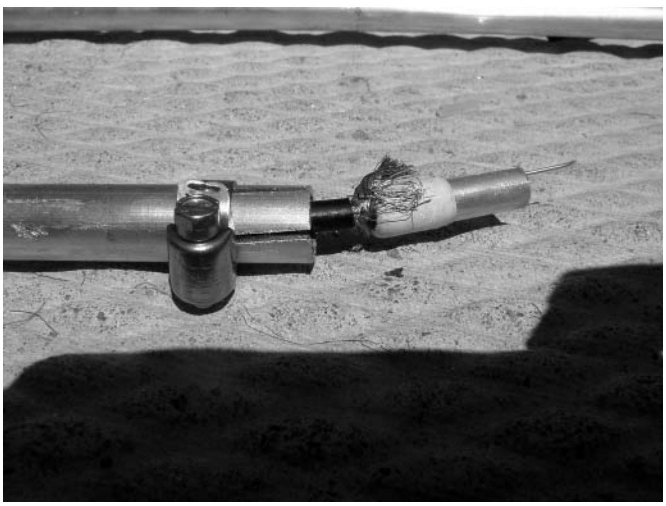


**Figure 7 — This photo shows the coax and nipple assembly ready to be inserted into the driven element tubing.**

**Figure 8 — The inner section of the driven element is now ready for the short section of tubing to be added. Before sliding the short section of tubing onto the nipple, fold the coax center conductor over the edge of the nipple, so it will make good contact with the outer length of the driven element.**

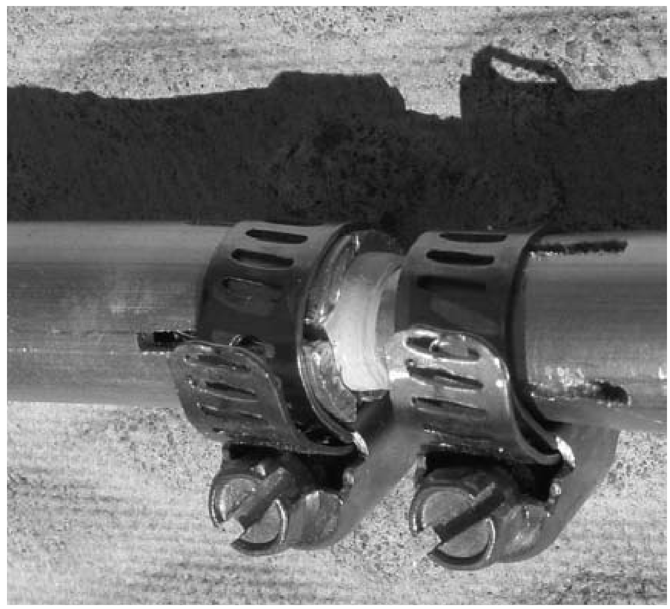


**Figure 9 — This photo shows a close-up of the completed driven element feed point.**

Finally, slide the 5.1 inch (130 mm) stub onto the bare nipple, folding over the center conductor so as to make good contact with the stub. Secure with another hose clamp as shown in Figure 9. The driven element is now fully assembled and may be checked with an ohmmeter for accidental shorts.

## Testing the Beam

I have an AEA "VIA Bravo" antenna analyzer, which I used to tune my antenna. Upon sweeping the beam, the SWR was 1.2:1 at 145.0 MHz. I had to add 0.55 inch (14 mm) to the driven element on the opposite end from the driven stub. This lowered the center frequency to 144.2 MHz with the Yagi pointed skyward. See the graph in Figure 10.

## Summary

I feel this method is simple theoretically, and not difficult to implement and optimize. The far end of the driven element can be lengthened or shortened to provide easy centering of the SWR curve, while the fractional parameter *G* allows for the proper transformation ratio from the center impedance. If you already have *EZNEC* the process is easy. I hope that this technique will become commonplace for building VHF/UHF beams.

## Notes

[1]Mark Wilson, K1RO, Editor, *The ARRL Radio Amateurs Handbook*, 2008 Edition, ARRL, (ARRL Order No. 1018) or R. Dean Straw, N6BV, Editor, *The ARRL Antenna Book*, 22nd Edition, ARRL (ARRL Order No. 9876). ARRL publications are available from your local ARRL dealer, or from the ARRL Bookstore. Telephone toll free in the US 888-277-5289 or call 860-594-0355, fax 860-594-0303; **www.arrl.org/shop**; **pubsales@arrl.org**.

[2]Roy W. Lewallen, W7EL, *EZNEC*, available for download from **www.eznec.com**.

[3]The author's *EZNEC* model file is available for download from the *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **11x08_Zimmerman.zip**.

*Robert K. Zimmerman, Jr. was born in 1951 in Dupo, Illinois. He graduated from Southern Illinois University, Edwardsville, with BS and MS degrees in physics (1973, 1975) and then attended the University of Illinois, Urbana-Champaign, where he was awarded the MSEE degree in 1980. He has spent his entire career in radio science, working for Cornell University (Arecibo Observatory), NASA Goddard Spaceflight Center, Los Alamos National Laboratory (accelerator division), and most recently as a radar engineer on Kwajalein Atoll. He is presently involved in microwave antenna research at McMaster University, Hamilton, Ontario. He has been licensed as WN9PXG (1965), WA9ZSF, NP4B, V73BZ, and now as VE3RKZ. Zimmerman is an ARRL Life Member, and is active on 40 m and 23 cm.*

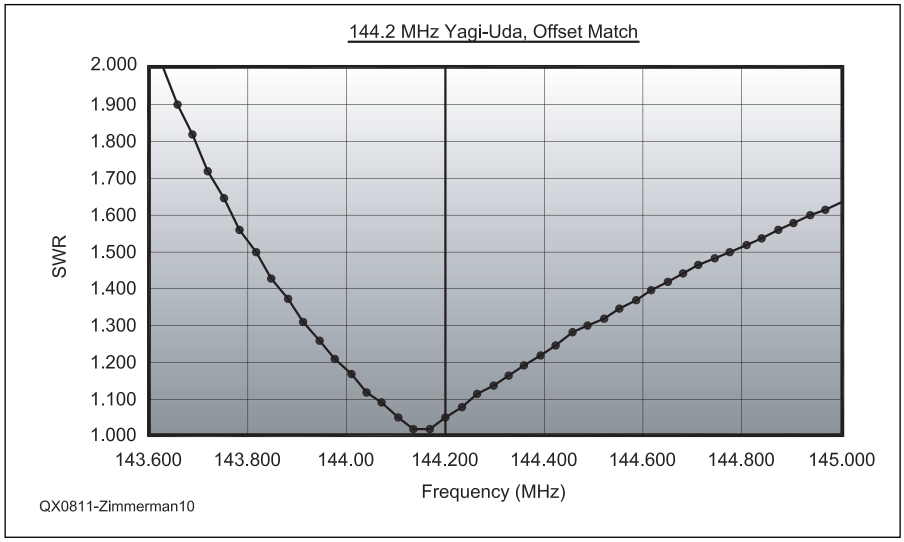



**Figure 10 — This graph shows the SWR of the completed antenna, as measured with an AEA VIA Bravo antenna analyzer.**

QEX

**Paul D. Wiedemeier, Ph.D., KE5LKY**

700 University Avenue, Monroe, Louisiana 71209; ke5lky@arrl.net

# Using *Udpcast* to IP Multicast Data over Amateur Packet Radio Networks

*This article is reprinted with the author's permission from the* 27th ARRL and TAPR Digital Communications Conference Proceedings. *The conference was held in Chicago, Illinois, September 26-28, 2008.*

During disasters and crises, the ability to transmit data to multiple destinations is paramount and amateur packet radio networks are often used to provide emergency data communication services when these situations occur. Most data transmission software, including Winlink 2000, *AirMail*, FTP, and SCP, unicast data. Thus, using these, many point-to-point data transmissions are needed to transmit data to multiple computers. Conversely, multicast software can transmit data to multiple destinations using a single point-to-multipoint data transmission. Unfortunately, available multicast software designed specifically for data transmission over amateur packet radio networks (1) require a dedicated computer to act as a multicast server, (2) are written only for the Microsoft *Windows* operating system, and (3) are not all actively supported.

In this article, we discuss how to IP multicast data over amateur packet radio networks using the *Udpcast* file transfer tool (e.g. data transmission software). Compared to other multicast software, we have identified three advantages to using the *Udpcast* file transfer tool. First, *Udpcast* was written for both the *UNIX/Linux* and Microsoft *Windows* operating systems, so you have the freedom to choose between operating systems. Second, *Udpcast* multicasts data without using a dedicated computer. This advantage will be highlighted later in this article. Last, *Udpcast* is actively supported and widely used within the *UNIX*/*Linux* community. For these three reasons, we advocate using *Udpcast* when data must be transmitted simultaneously to multiple computers during emergencies.

## IP Addressing and Data Transmissions

Regardless of whether data is transmitted using unicast or multicast, all computers attached to IP based networks must be assigned an IP address of the form A.B.C.D, where A, B, C, and D are 8 bit numbers in the range 0 through 255. For a thorough discussion of IP addressing the reader should refer to (Comer, 2000) and (Tanenbaum, 2003).*

Once a computer receives an IP address, data can be transmitted using either IP unicast, IP broadcast, or IP multicast (Comer, 2007). Each data transmission method is discussed in the following paragraphs, but for a thorough discussion of these data transmission methods the reader should refer to (Comer, 2000) and (Tanenbaum, 2003). For the remainder of this article, we will refer to the terms IP address, IP packet, IP unicast, IP broadcast, and IP multicast as address, packet, unicast, broadcast, and multicast respectively.

### Unicast

Unicast, often referred to as point-to-point data communication, transmits packets from a source computer to a destination computer. Each packet includes the addresses of the source and destination computers. Many of the four traditional groups of Internet applications (e.g. tools) (Comer, 2007), including electronic mail, news, remote login, and file transfer, use unicast to transmit data.

*All references appear on page 47.

### Broadcast

If unicast is referred to as point-to-point data communication, then broadcast is point-to-multipoint data communication, in that all packets transmitted by a source computer are delivered to all computers attached to the same network (Mogul, 1984-A) (Mogul, 1984-B). The address resolution protocol (Plummer, 1982), the dynamic host configuration protocol (Drom, 1997), and routing table updates are examples of Internet applications that are known to use broadcast.

Actually, two forms of broadcast exist: limited and direct. The address for limited broadcast is 255.255.255.255, and packets transmitted using limited broadcast are only delivered to those computers attached to the same network as the transmitting computer.

Direct broadcast addresses are of the form A.B.C.255, A.B.255.255, or A.255.255.255. The specific broadcast address used by a computer depends upon the underlying computer network to which it is attached. Packets transmitted using direct broadcast are usually only delivered to those computer attached to the same network as the transmitting computer, but routers may be configured to transmit direct broadcast packets across other networks. The *UNIX* ifconfig command can be used to obtain the broadcast address for a specific *UNIX* computer's network interface. Figure A1 in the Appendix shows the output generated by the *UNIX* ifconfig command, which queries the computer dcrl1's AX.25 ax0 network interface. The computer dcrl1 is discussed later in the *Using Udpcast to Multicast Data* section.

### Multicast

Multicast is similar to broadcast in that packets are transmitted to multiple computers attached to the same network as the transmitting computer. However, multicast packets are only accepted by a subset of all attached computers. That is, only those computers that agree to accept packets from a specific multicast address do so. Multicast addresses are in the range 224.0.0.0 through 239.255.255.255 and those Internet applications that use multicast, such as the University of California, Berkeley's *Streaming Media Toolkit* and *VideoCharger* by International Business Machines (IBM) Corporation, often transmit audio and/or video data.

### Multicast Software for Amateur Packet Radio Networks

As of July 2008, two software applications have been identified that were specifically developed to multicast data over amateur packet radio networks. They are *RadioMirror* and *AltCast*, and each are discussed in the following paragraphs.

*RadioMirror*, written by John Hansen, PhD, W2FS, for the Microsoft *Windows 95* operating system, was introduced to the amateur packet radio community during the mid 1990s. Dr Hansen wrote *RadioMirror* to provide "a general purpose mechanism for moving data from a central server to a lot of client computers, all at the same time. … In emergencies, a *RadioMirror* server would provide a mechanism to distribute large quantities of information (including graphics and other multimedia files) over a wide region" (Hansen, 1997-A).

Regarding functionality, a dedicated *RadioMirror* server continuously multicasts data over a very high frequency (VHF) or ultra high frequency (UHF) amateur packet radio network for some duration of time, usually 24 hours, to guarantee that all *RadioMirror* clients have an opportunity to receive the data.

While the *RadioMirror* software is still available for download (see Table A1 in the Appendix), little Web-based documentation exists. The best *RadioMirror* documentation found were those files included with the actual software distribution. Specifically, the files RADIOMIR.TXT (Hansen, 1997-A), RMTECH.txt (Hansen, 1997-B), RMSERVER.TXT (Hansen, 1997-C), and RMCLIENT.TXT (Hansen, 1997-D) were read. Unfortunately, it appears that Dr Hansen is not actively supporting the *RadioMirror* software as versions for the Microsoft *Windows XP* and *Vista* operating systems do not exist.

*AltCast*, written by Walt Fair, Jr., W5ALT, for the Microsoft *Windows XP* operating system, was introduced to the amateur packet radio community during the early 2000s. Similar to *RadioMirror*, a dedicated *AltCast* server continuously multicasts data over a high frequency (HF) amateur packet radio network, again for a given duration of time, using one of several phase-shift keying (PSK) modes: PSK31, QPSK31 or PSK63 (Brabham, 2005). Any *AltCast* client can receive the data multicasted by the dedicated *AltCast* server. The *AltCast* software can be downloaded from the Web site shown in Table A1 in the Appendix.

### The *Udpcast* File Transfer Tool

The *Udpcast* file transfer tool was written by Alain Knaff for the *UNIX*/*Linux* and Microsoft *Windows XP* operating systems and released in the early 2000s under the GNU General Public License 2.0. Widely used within the *UNIX*/*Linux* community, *Udpcast*'s primary strength lies in its ability to multicast operating system images from one computer to multiple computers attached to the same network. Functionally, *Udpcast* is primarily used to multicast data from one computer to multiple computers, but it is capable of unicast data transmissions.

Installation of the *Udpcast* software on a *UNIX*/*Linux* computer is accomplished by (1) using *UNIX*/*Linux* package manager software or (2) downloading the software (See Table A1 in the Appendix) and performing a manual install. Once installed, the *Udpcast* file transfer tool is comprised of two command line programs; udp-sender and udp-receiver. A complete list of *Udpcast* command line arguments can be obtained from (Knaff 2005).

### Udp-sender Command Structure

The structure of the udp-sender command is shown in Figure 1. The **--file** argument specifies which file to multicast to the udp-receivers. If this argument is absent, input is read from standard input, which permits UNIX piping and redirection. The **--interface** argument is used to specify which of the computer's network interfaces will be used to transmit data. If this argument is absent, the default network interface used is eth0. The **-b** argument (note, that the argument is **-b**, not **--b**) specifies the packet size to use when the sender transmits data. If this argument is absent, the default packet size used is 1456.

The **--async**, **--fec** and **--max-bitrate** arguments are used when the sender does not expect request confirmations from the receiver(s). This is referred to as unidirectional mode and is beneficial when a high latency and/or low bandwidth (e.g. amateur packet radio networks) channel is used. Specifically, the **--async** argument defines asynchronous mode and is used without confirmation. The **--fec** argument adds forward error correction (FEC) to the transmitted data to provide reliability when the data communication channel is noisy. The amount of FEC added is based on the factors interleave (I), redundancy (R) and stripsize (S). For detailed information about the **--fec** argument the reader should refer to (Knaff, 2006). The argument **--max-bitrate** is used to limit how fast the sender transmits data to the receiver(s). It essentially prevents the sender from "drowning" one or more receivers with data.

The **--log** and **--bw-period** arguments permit the udp-sender to log instantaneous bandwidth data to a file. The **--log** argument specifies the name of the file in which to write the data. The **--bw-period** argument defines the number of seconds between which instantaneous bandwidth is written to

```
udp-sender --file FILETOSEND --interface INTERFACE -b BLOCKSIZE
--async --fec IxR/S --max-bitrate BPS --log LOGFILE
--bw-period SECONDS
```

**Figure 1 —The udpcast-sender command.**

```
udp-receiver --file FILETORECEIVE --nosync --interface INTERFACE
```

**Figure 2--The udpcast-receiver command.**

the log file. That is, "every so many seconds, log instantaneous bandwidth seen during that period" (Knaff, 2005). While Udpcast only generates instantaneous bandwidth, file transmission time can be computed by dividing the size of the file transmitted by the average instantaneous bandwidth written to the log file (Wiedemeier, 2007).

### Udp-receiver Command Structure

The structure of the udp-receiver command is shown in Figure 2. Similar to udp-sender, the **--file** argument specifies the name of the file used to store the data received from the udp-sender. If this argument is absent, input is written to standard output. Again, this permits *UNIX* piping and redirection. The **--nosync** argument must be used when output is written to a file or a pipe. The **--interface** argument is used to specify which of the computer's network interfaces will be used to transmit data. As with the udp-sender command, if this argument is absent, the default network interface used is eth0.

## Using *Udpcast* to Multicast Data

The computers, transceivers, terminal node controllers, and antennas used in this research are owned and maintained by The University of Louisiana at Monroe (ULM) Digital Communication Research Laboratory (DCRL) (see Figure 3). As shown, we operate three similarly configured "rigs", each comprised of a Dell GX240 or GX270 computer, a Kantronics KPC-3+ terminal node controller, a Kenwood TM-271A transceiver and a Diamond X30A or X50A antenna.

Our computers' AX.25 network inter-

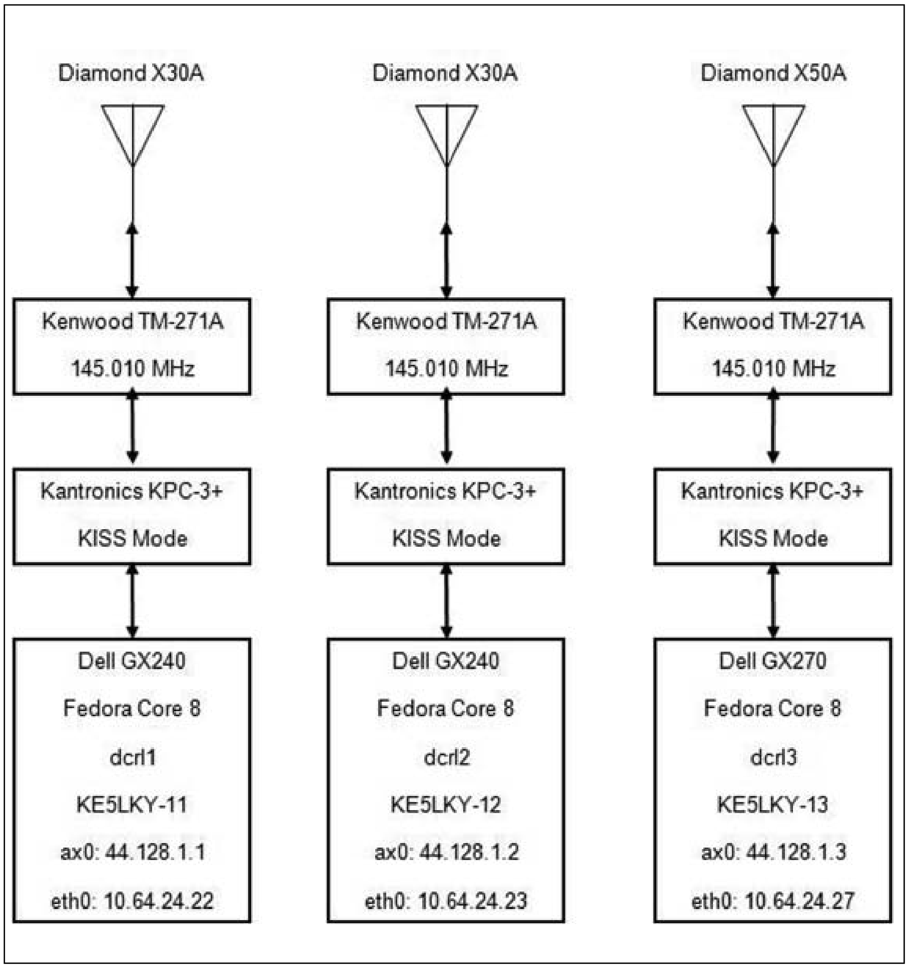



**Figure 3--The physical configuration of computers, transceivers, terminal node controllers, and antennas owned and maintained by The University of Louisiana at Monroe Digital Communication Research Laboratory.**

```
$ udp-sender --file file4KB.txt --interface ax0 -b 256 --async
--fec 4x1/1 --max-bitrate 1200 --log logfile.txt --bw-period 5
stripes=4 redund=1 stripesize=1
Udp-sender 2007-12-28
Using mcast address 236.128.1.1
UDP sender for file4KB.txt at 44.128.1.1 on ax0
Broadcasting control to 44.128.1.255
Ready. Press any key to start sending data.
Starting transfer: 00000029
bytes=4 096  re-xmits=0000000  (  0.0%)  slice=0004  4 096 -   0
Transfer complete.
$
```

**Figure 4--Transmission of data using the udpcast-sender command on the computer dcrl1.**

faces (e.g. ax0) (Tranter, 2001) (Jones, 1996) have been assigned amateur packet radio addresses of the form 44.128.C.D because these addresses are designated for testing purposes (Amateur Packet Radio, n.d.). However, we could have assigned any un-routable addresses to our computers (Rekhter, 1996). Likewise, our computers' AX.25 network interfaces have been configured with call signs of the form KE5LKY-##, where KE5LKY is the author's call sign. Call signs are assigned based on information found within our computers' /etc/ax25/ax25d and /etc/ax25/axports files (see Figures A2 and A3 respectively in the Appendix). Our computers' Ethernet network interfaces (e.g. eth0) have been assigned static addresses associated with ULM's local intranet.

Before we initiate data transmission using *Udpcast*, all terminal node controllers are placed into KISS mode and the transceivers are tuned to the frequency 145.010 Megahertz (MHz). A Bourne shell script that will place a Kantronics KPC-3+ into KISS mode is shown in Figure A4 in the Appendix. A similar Bourne shell script that will take a Kantronics KPC-3+ out of KISS mode is shown in Figure A5 in the Appendix.

**Example udp-sender and udp-receiver Command Execution**

An example of the udp-sender command executed on the computer dcrl1 is shown in Figure 4. The file transmitted was comprised of 4 kilobytes (KB) of American Standard Code for Information Interchange (ASCII) text and was multicasted over the computer dcrl1's AX.25 network interface ax0 to computers dcrl2 and dcrl3. The packet size was defined to be 256 because this value represents the maximum transfer unit (MTU) for all computers' ax0 network interface (see the Bourne shell script shown in Figure A6 in the Appendix). All Kantronics KPC-3+s' PACLEN values are also set to 256. The **--max-bitrate** was set to 1200 bits per second (bps), which is the Kantronics KPC-3+s maximum data rate. The instantaneous bandwidth associated with this multicast data transmission was stored in the file logfile.txt.

An example of the udp-receiver command executed on the computer dcrl2 is shown in Figure 5. The size of the file received was 4 KB, so we named the file appropriately. We use the **--nosync** argument because we are writing data to a file. We also specify the use of the computer dcrl2's AX.25 network interface ax0 using the **--interface** argument.

Three tasks must be completed to initiate the transmission of data between the udp-sender and udp-receiver(s). First, execute the udp-sender command using the arguments shown in Figure 4. Once started, the output "Ready. Press any key to start sending data." is generated (see Figure 4). Second, execute the udp-receiver command using the arguments shown in Figure 5 on all computers that will receive the file transmitted by the computer executing the udp-sender. The udp-receiver command generates the output "Connected as ## to 44.128.1.1 Listening to multicast on 236.128.1.1", which implies that the udp-receiver is waiting to receive data from the udp-server (see Figure 5). Last, press any key within the udp-server terminal window to start the transmission. Both the udp-sender and udp-receiver(s) show the amount of data that have been transmitted. If the transmission was successful, both the udp-sender and udp-receiver(s) generate the phrase "Transfer complete" (see both Figures 4 and 5).

```
$ udp-receiver --file file4KB.txt --nosync --interface ax0
Udp-receiver 2007-12-28
UDP receiver for file4KB.txt at 44.128.1.2 on ax0
Connected as #0 to 44.128.1.1
Listening to multicast on 236.128.1.1
bytes=4 096  (  0.00 Mbps)  4 096
Transfer complete.
$
```

**Figure 5--Receipt of data using the udpcast-receiver command on the computer dcrl2.**

**Table 1**
**A comparison of the *RadioMirror*, *AltCast* and *Udpcast* software**

| *Software Traits* | *RadioMirror* | *AltCast* | *Udpcast* |
|---|---|---|---|
| Supported Operating Systems | Microsoft *Windows 95* | Microsoft *Windows XP* | Microsoft *Windows XP* *UNIX/Linux* |
| Dedicated Multicast Server | Yes | Yes | No |
| Primary Use | VHF/UHF Packet Radio | HF Packet Radio | Transmission of Operating System Images |
| Actively Supported | No | Yes | Yes |

## Conclusions

As discussed, the *RadioMirror* and *AltCast* software can be used to multicast data over VHF/UHF and HF amateur packet radio networks respectively. However, both are written to execute only on the Microsoft *Windows* operating system, and both require a dedicated computer to continuously transmit data. Additionally, while *AltCast* is actively supported, it appears *RadioMirror* is not. These three traits are shown in Table 1.

In this article, we introduce the *Udpcast* file transmission tool. In comparison to *RadioMirror* and *AltCast*, we find that *Udpcast* is written for both the Microsoft *Windows* and *UNIX/Linux* operating systems, and is actively supported. More importantly, *Udpcast* does not require a dedicated computer to multicast data. The benefit to using *Udpcast*, versus *RadioMirror* or *AltCast*, is any given computer attached to an amateur packet radio network can execute the udp-sender command and multicast data to multiple udp-receivers. Likewise, the same computer can execute the udp-receiver command and receive multicasted data from another computer executing the udp-sender command. For this reason, and those shown in Table 1, we advocate using *Udpcast* to multicast data over amateur packet radio networks during emergencies.

While our current research uses *Udpcast* to multicast data over VHF amateur packet radio networks, we believe *Udpcast* can be used to multicast data over both UHF and HF amateur packet radio networks. As such, in the near future, we plan to purchase Kenwood TM-V71A transceivers and Kantronics KCP-9612+ terminal node controllers to explore multicasting data over UHF amateur packet radio networks using *Udpcast*.

## Acknowledgements

The author graciously thanks E. Benson Scott II, MD, AE5V, for providing answers to my numerous packet radio questions, and Allison MD Wiedemeier, PhD, for reading several drafts of this article. The author's research is supported through funds provided by a ULM College of Business Administration Faculty Summer Research Grant, the Clarke M. Williams, Jr Endowed Professorship in Computer Science sponsored by the ULM Foundation, and the ULM DCRL.

## References

Amateur Packet Radio, net 44, and AMPR. ORG. (n.d.). Retrieved July 16, 2008, from **http://www.ampr.org/**.

Brabham, Charles (N5PVL). (2005). Utilizing Amateur Multicast Protocol. Retrieved July 16, 2008, from **http://www.uspacket.org/multicast.htm**.

Comer, Douglas E. (2007). *The Internet Book*, 4th Edition. Upper Saddle River, NJ: Pearson Prentice Hall Publisher.

Comer, Douglas E. (2000). *Internetworking with TCP/IP*, Volume 1, 4th Edition. Upper Saddle River, NJ: Pearson Prentice Hall Publisher.

Drom, R.. (1997, March). *Dynamic Host Configuration Protocol*. Request For Comments (RFC) 2131. Retrieved July 16, 2008 from **http://www.ietf.org/rfc/rfc2131.txt**.

Hansen, John. (1997-A, April 20). *RadioMirror: Rethinking Packet Radio*. Retrieved July 16, 2008 from **ftp://ftp.tapr.org/pub/wa0ptv/RadioMirror.exe**.

Hansen, John. (1997-B, April 5). *Technical Documentation for the RadioMirror Server*. Retrieved July 16, 2008 from **ftp://ftp.tapr.org/pub/wa0ptv/RadioMirror.exe**.

Hansen, John. (1997-C, April 20). *RadioMirror Server: Sysop Information*. Retrieved July 16, 2008 from **ftp://ftp.tapr.org/pub/wa0ptv/RadioMirror.exe**.

Hansen, John. (1997-D, April 20). *RadioMirror Client Documentation*. Retrieved July 16, 2008 from **ftp://ftp.tapr.org/pub/wa0ptv/RadioMirror.exe**.

Information Sciences Institute (ISI). (1981, September). Internet Protocol. Request For Comments (RFC) 791. Retrieved July 16, 2008 from **http://www.ietf.org/rfc/rfc791.txt**.

Jones, Greg (Ed.) (WD5IVD). (1996). *Packet Radio: What? Why? How? Articles and Information on General Packet Radio Topics*. Tucson, AZ: Tucson Amateur Packet Radio Corporation Publisher.

Knaff, Alain. (2006, March 20). Multicasting over satellite. Udpcast Forums. Retrieved July 16, 2008, from **http://udpcast.linux.lu/pipermail/udpcast/2006-March/000493.html**.

Knaff, Alain. (2005, May 14). *Udpcast Command line Options*. Retrieved July 16, 2008, from **http://udpcast.linux.lu/cmd.html**.

Mogul, Jeffery. (1984-A, October). Broadcasting Internet Datagrams. Request For Comments (RFC) 919. Retrieved July 16, 2008 from http://www.apps.ietf.org/rfc/rfc919.html.

Mogul, Jeffery. (1984-B, October). *Broadcasting Internet Datagrams in the Presence of Subnets*. Request For Comments (RFC) 922. Retrieved July 16, 2008 from **http://www.apps.ietf.org/rfc/rfc922.html**.

Plummer, David C.. (1982, November). *An Ethernet Address Resolution Protocol*. Request For Comments (RFC) 826. Retrieved July 16, 2008 from **http://www.ietf.org/rfc/rfc826.txt**.

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E.. (1996, February). *Address Allocation for Private Networks*. Request For Comments (RFC) 1918. Retrieved July 16, 2008 from **http://www.apps.ietf.org/rfc/rfc1918.html**.

Tanenbaum, Andrew S.. (2003). *Computer Networks, 4th Edition*. Upper Saddle River, NJ: Pearson Prentice Hall Publisher.

Tranter, Jeff (VE3ICH). (2001, September 19). *Linux Amateur Radio AX.25 HOWTO*. Retrieved July 16, 2008, from **http://tldp.org/HOWTO/AX25-HOWTO/**.

Wiedemeier, Paul (KE5LKY). (2007, September). *Performance Modeling of TCP and UDP over Amateur Packet Radio Networks using the ns-2 Network Simulator*. (pp. 141-154). *Proceedings of the 26th ARRL and TAPR Digital Communications Conference*.

*Dr Paul D. Wiedemeier, KE5LKY, is an Assistant Professor of Computer Science at The University of Louisiana at Monroe where he holds the Clarke Williams, Jr Endowed Professorship in Computer Science. He is also the Principle Investigator of The University of Louisiana at Monroe Digital Communication Research Laboratory. Dr Wiedemeier obtained a PhD in Computer Engineering and Computer Science from the University of Missouri – Columbia, a MS in Computer Science from Michigan Technological University, and a BS in Computer Science from Drake University. He is a member of the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery, the American Radio Relay League, the Tucson Amateur Packet Radio Corporation, the Consortium for Computing Sciences in Colleges, and the Louisiana Academy of Sciences. Dr Wiedemeier also holds a Technician Amateur Radio license.*

Appendix starts on next page

## Appendix

**Table A1**
**Multicast Software URLs**

| | |
|---|---|
| *RadioMirror* | **ftp://ftp.tapr.org/pub/wa0ptv/*RadioMirror*.exe**<br>**ftp://ftp.tapr.org/pub/wa0ptv/radiomirrorsource.zip** |
| *AltCast* | **http://www.comportco.com/hamsoft/*AltCast*.zip** |
| *Udpcast* | **http://udpcast.linux.lu/** |

```
ax0     Link encap:AMPR AX.25  HWaddr KE5LKY-11
        inet addr:44.128.1.1  Bcast:44.128.1.255
Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:256  Metric:1
        RX packets:986 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1037 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:10
        RX bytes:201012 (196.3 KiB)  TX bytes:85586 (83.5 KiB)
```

**Figure A1--Output returned by the "ifconfig ax0" command executed on the computer dcrl1.**

```
[KE5LKY-11 VIA radio]
NOCALL * * * * * * L
default * * * * * * - root /usr/sbin/node node
```

**Figure A2--The computer dcrl1's /etc/ax25/ax25d.conf file.**

```
radio  KE5LKY-11  9600  256  7  9600 Kantronics KPC-3+ 145.010
```

**Figure A3--The computer dcrl1's /etc/ax25/axports file.**

```
#!/bin/sh
########################################################################
# Script Name:   kiss-on.sh
########################################################################
/bin/stty 9600 < /dev/ttyS0
/bin/echo -e "intface kiss\rreset\r" > /dev/ttyS0
```

**Figure A4--A Bourne shell script that places a Kantronics KPC-3+ into KISS mode.**

```
#!/bin/sh
######################################################################
# Script Name:   kiss-off.sh
######################################################################
/bin/stty "9600" < /dev/ttyS0
/bin/echo -e "\r\xC0\xFF\xC0\r" > /dev/ttyS0
```

**Figure A5--A Bourne shell script that takes a Kantronics KPC-3+ out of KISS mode.**

```
#!/bin/sh
######################################################################
# Script Name:   ax0-on.sh
######################################################################
/usr/sbin/kissattach /dev/ttyS0 radio 44.128.1.1
/sbin/ifconfig ax0 44.128.1.1 broadcast 44.128.1.255 netmask
255.255.255.0 arp mtu 256 up multicast
/sbin/route del -net 44.128.1.0 netmask 255.255.255.0 ax0
/sbin/route add -net 44.128.1.0 netmask 255.255.255.0 irtt 12000
window 1792 ax0
/usr/sbin/ax25d &
/usr/sbin/mheardd -l -f
```

**Figure A6--A Bourne shell script that configures the computer dcrl1's network interface ax0, and starts the ax24d and mheardd daemons.**

```
#!/bin/sh
######################################################################
# Script Name:   ax0-off.sh
######################################################################
/bin/ps -ef | /bin/grep "/usr/sbin/ax25d" | /bin/awk '{print $2}' |
/usr/bin/xargs -i kill -9 {}
/bin/ps -ef | /bin/grep "/usr/sbin/kissattach" | /bin/awk '{print
$2}' | /usr/bin/xargs -i kill -9 {}
/bin/ps -ef | /bin/grep "/usr/sbin/mheardd" | /bin/awk '{print $2}'
| /usr/bin/xargs -i kill -9 {}
```

**Figure A7--A Bourne shell script that terminates the ax24d, kissattach and mheardd daemons.**

QEX

**Peter Loveall, AE5PL**

3624 County Road 413, Melissa, TX 75454-3014; ae5pl@arrl.net

# D-STAR® Uncovered

*This article is represented with the author's permission from* The 27th ARRL and TAPR Digital Communications Conference Proceedings. *The DCC was held in Chicago, Illinois, Sep 26-28, 2008.*

**Abstract**

D-STAR® is a digital streaming over-the-air protocol developed by the Japan Amateur Radio League, Inc (JARL) which supports Ethernet at 128 kbps (DD) and digital voice at 4800 bps (DV).[1] DV uses 3600 bps for voice (2400 AMBE encoding, 1200 bps FEC) and 1200 bps for synchronization and multiuse (approximately 900 bps is available for general use). DD provides an encapsulated Ethernet bridge for connecting two or more Ethernet clients over RF. We will explore this bit streaming protocol, its primary components, and current implementations. This paper explores the protocol from a technical, not subjective, viewpoint.

**Keywords**

D-STAR, D-PRS, Icom

## Introduction

Icom Incorporated has introduced a number of radios with D-STAR capability. D-STAR is comprised of the digital voice (DV) protocol (4800 bps) and the digital data (DD) protocol (128 kbps). D-STAR was developed by the Japan Amateur Radio League, Inc to support digital bit streaming modes of communication in the VHF, UHF, and higher Amateur bands. D-STAR is an open protocol. The voice codec (vocoder) used for digital voice (DV) is the proprietary AMBE algorithm developed by the same company that owns the P25 voice algorithm.

The DV and DD protocols are highly misunderstood primarily due to our preconceptions of a digital protocol and the lack of well translated documents. Also misunderstood is the RF framing information carried in both protocols. We will try to push through these misconceptions to expose a very powerful and forward-thinking protocol specification. We will also delineate the differences between the basic D-STAR specification and the Icom enhancements.

D-STAR uses the GMSK transmission medium and has no compatibility with existing amateur radio digital transmissions. For instance, D-STAR (bit streaming) is not directly compatible with AX.25 (packet) networks. As with any digital protocols, networks designed around one protocol are not directly compatible with another independently developed protocol.

This paper will first address the content portion of each protocol so the characteristics of the "data" is well understood before we address the D-STAR wrapper which is used for routing, identification, and other protocol-specific information. Finally, this paper addresses enhancements that Icom has built into their radios so it is understood what is part of the D-STAR specification and what are Icom enhancements.

[1]D-Star® is a registered trademark of Icom Incorporated Corporation Japan.

## Digital Voice (DV)

The Digital Voice (DV) specification is well-defined in the translated D-STAR specification. The content bit stream is defined as 72 bits of voice information followed by 24 bits of loosely defined "data". This pattern of 96 bits is repeated for the entirety of the transmission.

The RF header will be addressed later in this paper. The "voice frame" consists of encoded voice and forward error correction (FEC) defined as part of the AMBE algorithm. The AMBE encoding is defined as being the only content in the voice frames ensuring compatibility with all radios that use the D-STAR protocol.

The "data frame" is defined as (edited from shogen.pdf):

(3) The first data frame and then every 21st data frame in a repeating cycle, are used

| Radio Header | | | | | | | | | Data | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ID | | | | | | | | | | | | |
| Bit Sync | Frame Sync | Flag 1 | Flag 2 | Flag 3 | Destination Repeater Callsign | Departure Repeater Callsign | Compaion Callsign | Own Callsign | Own Callsign Ext | P_FCS | Voice Frame | Data Frame | Voice Frame | Data Frame | … | Voice Frame | Data Frame |
| 64 bit | 15 bit | 1 byte | 1 byte | 1 byte | 8 byte | 8 byte | 8 byte | 8 byte | 4 byte | 2 byte | 72 bit | 24 bit | 72 bit | 24 bit | | 72 bit | 48 bit |
| | | error correction 660 bit | | | | | | | | | | | | | | | |

**Figure 1 — Digital Voice Components.**

only for synchronizing data for each modulation type. Synchronization corrects for the lag between transmission and reception, including the transit time of communications.

(4) The data in a data frame is transmitted without modification from the input data. If the data is required as error correction or synchronization, these frames are processed before processing the data input.

(5) If the data signal length is greater than the length of the voice communication the transmitting switch is turned on until the completion of the data signal manually. The processing can be allowed automatically.

(6) The last data frame, which requires a means of terminating the transmission, is a unique synchronizing signal (32 bit + 15bit "000100110101111" + "0", making 48 bits) as defined by the modulation type.

Note that the "data frame" is not a data protocol as we normally think of one. There is no specification of data information separate from the total transmission. There is no error detection or correction. The first frame and every 21st frame thereafter is reserved for synchronization. The last frame is reserved as a terminator indicator. The voice frame is always present and may not be used for anything other than the AMBE encoding.

Also note that these "frames" are not separated by anything. They are not packets. They are definitions of bit positions in the continuous DV bit stream. This precludes interleaving of transmissions by different stations. Any interleaving of bits by a repeater would be modifying the data from one station without any ability to properly identify the data as being from a different source. Any attempt to imbed the "change of source" information in the data stream defeats the D-STAR routing information.

The JARL intentionally did not define the inter-sync data frame format. This leaves it open to manufacturers and radio designers to manipulate as they see fit. These bits make up 1/4th of the total 4800 bps data stream which gives a maximum transport rate of 1142 bps (1200 × 20/21).

Icom has made use of these data frames to repeat the RF header, repeatedly send a radio-to-radio 20 character message, and for carrying rudimentary data supplied by the user. These applications are described further in the "Icom Enhancements" section of this paper. It is mentioned here so designers are aware that there is a de facto standard use of these data frames already in place as implemented by Icom which must be taken into account with any new radio design for compatibility reasons. It is also important to note that the serial port on the Icom radios does not give full access to the bits in the data frame. This last point brings the maximum transport rate of serial data presented to the serial port of an Icom radio down to about 761 bps (1142 × 2/3). If the radios have a "message" programmed in, this number is even lower.

These are primary factors for anyone designing software or hardware for D-STAR DV:

1) Voice information is always ¾ of the transmission bandwidth.

2) Even "silent" voice information can be heard due to bit errors and other control issues.

3) DV transmissions are bit streams, not packets.

a) DV transmissions are not AX.25 compatible.

b) DV transmissions are not interleaved.

c) Only the bit stream is identified, not individual frames.

4) Data frames contain no forward error correction or error detection information.

5) Voice information can be accurately reconstructed because of FEC when the data frames are rendered virtually useless due to bit errors.

6) Efficiency dictates short, repetitive data containing error detection information (FCS, CRC, etc.) for accurate and reliable data reception.

## Digital Data (DD)

The Digital Data (DD) specification is well-defined in the translated D-STAR specification. The content portion of the bit stream is defined as a complete Ethernet packet, including FCS up to 1500 bytes long (MTU=1500). It is important to note that the content must be an Ethernet packet (not random data). The MTU of 1500 is industry standard. However, transport over the D-STAR IP network (discussed later in this paper) may cause multiple tunnel packets to be created for one MTU-size Ethernet packet.

DD transmissions are essentially Ethernet bridge transmissions. There are no predefined restrictions on the Ethernet packet content in the D-STAR specification. While IP packets have some support in current hardware and software implementations (see "Icom Enhancement" section), any valid Ethernet packets may be exchanged between stations and still be compliant with the D-STAR specification.

Each DD transmission encompasses one and only one Ethernet packet. Any error detection and retransmission is done by the Ethernet clients. The D-STAR DD protocol provides no forward error correction safety for the Ethernet packets.

The DD protocol provides a simple Ethernet bridge over amateur radio. Because of this flexibility, it is important operationally to limit as much as possible the over-the-air communications. Broadcast protocols such as NETBIOS must be disabled if successful communications are to occur.

## D-STAR Control

Every D-STAR transmission starts with synchronization bits followed by an RF header. The RF header is critical to proper interpretation of the bit stream. To ensure reliable reception of the RF header, the D-STAR specification interleaves forward error correction information and includes a 16 bit CRC for just the header information (Figure 1 and Figure 2.) The full text description of the flag bytes can be found in the D-STAR specification (Japan Amateur Radio League, Inc, 2005). The only portion of the flag bytes of interest here is the bit in Flag 1 that the content is either voice or data. The rest of the flag information is used to determine repeatability, urgent status, special use, etc. that are not significant to the user but are significant to the radio designer and the

| Radio Header | | | | | | | | | | | Data | | | | | FCS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ID | | | | | | | MAC Header | | | | |
| Bit Sync | Frame Sync | Flag 1 | Flag 2 | Flag 3 | Destination Repeater Callsign | Departure Repeater Callsign | Compaion Callsign | Own Callsign | Own Callsign Ext | P_FCS | E_Len | SA | DA | Type | Data Frame | CRC |
| 64 bit | 15 bit | 1 byte | 1 byte | 1 byte | 8 byte | 8 byte | 8 byte | 8 byte | 4 byte | 2 byte | 6 byte | 6 byte | 6 byte | 2 byte | 46-1500 byte | 4 byte |
| | | \|◄ error correction 660 bit ►\| | | | | | | | | | | | | | | |

**Figure 2 — Digital Data Components.**

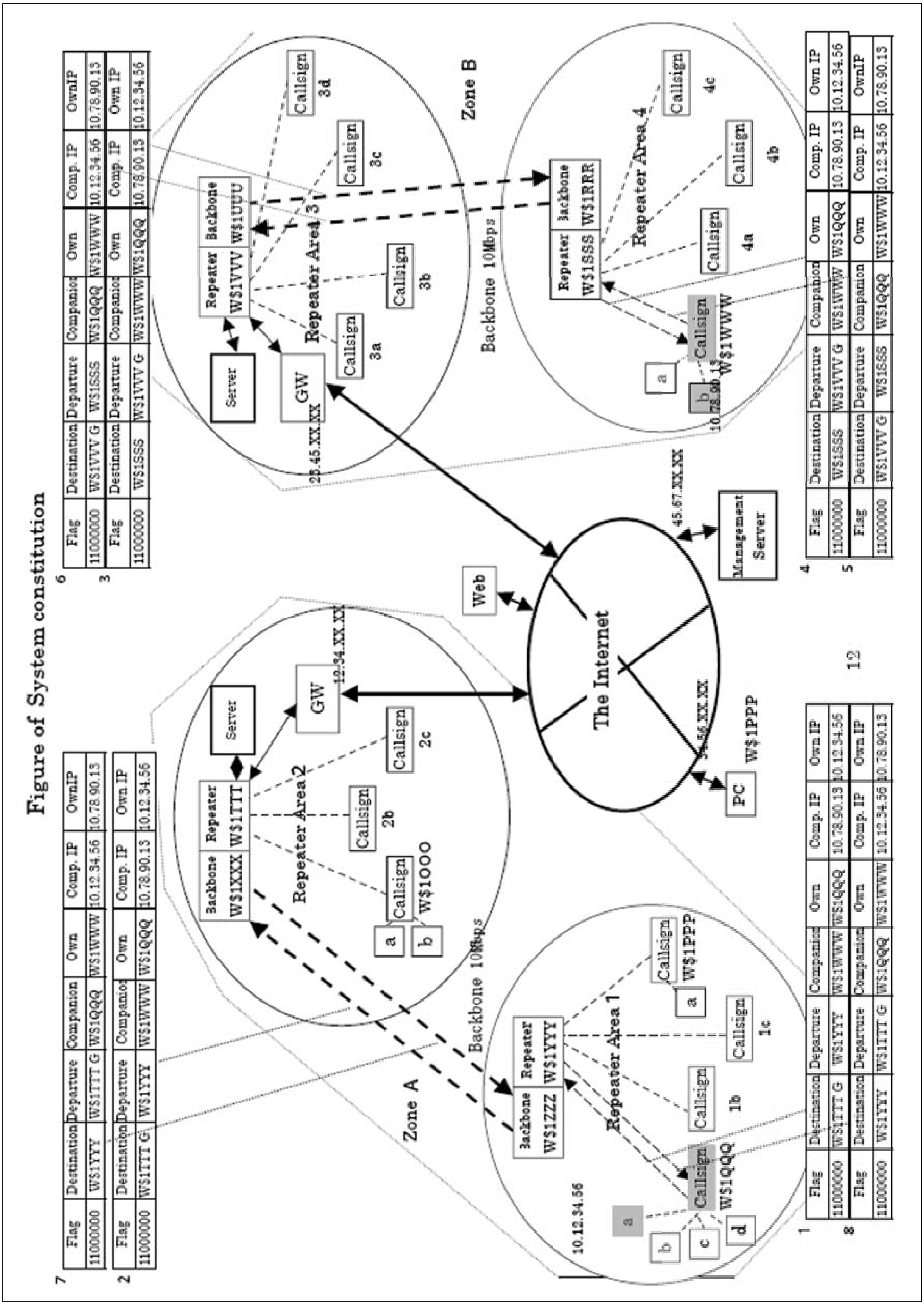



Figure 3 — A D-STAR system overview.

designer of gateway and repeater software. Designers are encouraged to closely review these flag bytes in actual use.

The rest of the header is dedicated to "ID". This is the heart of the D-STAR protocol. The D-STAR specification defines a callsign as a 7 character sequence of upper-case ASCII characters, numbers, or spaces. All callsigns are padded with spaces to fill the 7 characters. Repeater callsigns are restricted to 6 characters or numbers with a trailing space to accommodate zone communications described later in this section. The callsign fields are always 8 bytes long (not variable). The eighth byte is the ID character. This character can be an upper-case ASCII character, a number, or a space.

"Own Callsign" is the originating station's callsign and ID. This must never be modified at any point during the repeating of the signal. "Own Callsign Ext" (Extension, 2) is a 4 character extension that facilitates reciprocal licensing prefix identification requirements or can be used by the originating station for any other purpose. It is not used by the D-STAR protocol but must not be altered as it is defined as part of the originating station identification. "Own Callsign" corresponds to "MY" in the Icom radios.

"Companion Callsign" is the callsign and ID of the station being called. D-STAR defines "CQCQCQ" as the generic "broadcast" callsign and ID. This may be a specific station callsign and ID. This may be a "zone" or "area" callsign and ID. "Zone" and "area" callsigns and IDs are repeater callsigns and IDs prefixed with a forward slash '/'. For instance, to make a call to anyone listening to the K5TIT 440 MHz DV repeater (K5TIT B), an operator would program "UR" to "/K5TIT B" on an Icom radio. Note that the repeater callsign has two trailing spaces before the ID but only one trailing space when used as a "zone" callsign. This is why a repeater callsign may only be a maximum of 6 characters while a user's callsign may be up to 7 characters.

"Departure Repeater Callsign" is the repeater callsign local to the originating station "Own Callsign". This corresponds to "RPT1" in the Icom radios. This will be spaces if simplex operation is desired.

"Destination Repeater Callsign" is the target repeater callsign LOCAL to the originating station. This corresponds to "RPT2" in the Icom radios. This will be spaces for simplex or local repeater operation. This can be set to a local repeater callsign and ID for cross-band operation (MY="AE5PL I", UR="CQCQCQ", RPT1="K5TIT A", RPT2="K5TIT B"). This will be set to the local gateway callsign and ID (always 'G') to communicate throughout the D-STAR network (even locally) (MY="AE5PL I", UR="/K5TIT B", RPT1="K5TIT A", RPT2="K5TIT G").

The callsigns are used for all routing determination by D-STAR repeaters and gateways. D-STAR routing is designed on the premise that the operator knows who or where they want to talk to but have no need to know how their voice (or data) gets there (network-defined routing). It is also important to note that the routing is of the origin station's transmissions and does not affect the "companion" station's routing. For the "companion" to properly respond via D-STAR routing (non-local) so the origin station hears them, the "companion" station must either program UR with the origin station's callsign and ID or the zone callsign of the repeater that the origin station is using. As you will see, the preferred method is using the origin station's callsign and ID because that is all that is seen at the remote end of the transmission.

This methodology sets up the amateur radio equivalent of "Follow Me Roaming"® in the early days of cellular telephone networks. Every time a station transmits via a D-STAR repeater, that station's information is made available to the D-STAR network. This makes it possible for me to program my radio's UR with my wife's callsign and have my station callsign and ID programmed in her radio's UR, and we can communicate with each other anywhere in the world.

Key factors regarding the RF header information:

1) Flags indicate type of D-STAR transmission and miscellaneous control information.

2) "Own Callsign" 1 & 2 (extension) must never be modified

3) "Companion Callsign" is the station intended to hear the transmission.

a) Per the precepts of amateur radio, there are no private communications (everyone with a D-STAR radio in range of the transmitting station(s) can hear the conversation).

b) "Companion Callsign" is used by D-STAR gateways (routers) to determine the repeater to transmit on.

c) "Companion Callsign" is used solely for D-STAR routing. Other uses such as listening only for a specific station are possible but not covered in the D-STAR specification.

4) "Departure Repeater Callsign" is the repeater local to the transmitting station. This can be modified by D-STAR gateways at the "Companion" end to be the repeater local to the "Companion".

5) "Destination Repeater Callsign" is the "second" "repeater" the transmission is to pass to/through. This will normally be the local gateway or another local repeater module. This can be modified by D-STAR gateways at the "Companion" end to be the gateway local to the "Companion".

6) No other routing information is contained in the D-STAR protocol. None may

be added as any additions would break the existing networks.

7) No modification of the RF header is allowed on RF (repeaters, etc. must leave the header intact).

## D-STAR Network

The D-STAR specification defines the repeater controller/gateway communications and defines the general D-STAR network architecture. The following diagram is taken from the English translation of the D-STAR specification:

The Comp. IP and Own IP are shown for reference if this was a DD communications. As they do not change and are not passed as part of the D-STAR protocol, they can safely be ignored for the purposes of the following explanation.

Headers 1 through 4 are W$1QQQ calling W$1WWW. Headers 5 through 8 are W$1WWW calling W$1QQQ. Note that "Own Callsign" and "Companion Callsign" are never altered in either sequence. The "Destination Repeater Callsign" and the "Departure Repeater Callsign" are changed between the gateways. This is so the receiving gateway and repeater controller know which repeater to send the bit stream to. It also makes it easy to create a "One Touch" response as Icom has done by simply placing the received "Own Callsign" in the transmitted "Companion Callsign", the received "Destination Repeater Callsign" in the transmitted "Departure Repeater Callsign", and the received "Departure Repeater Callsign" in the transmitted "Destination Repeater Callsign".

Use of the "special" character '/' at the beginning of a callsign indicates that the transmission is to be routed to the repeater specified immediately following the slash. For instance, entering "/K5TIT B" in the "Companion Callsign" would cause the transmission to be routed to the "K5TIT B" repeater for broadcast. Using the above example, W$1QQQ would put "/W$1SSS" in the "Companion Callsign" for the same sequence 1 through 4 to occur. At the W$1VVV gateway, however, the "/ W$1SSS" in the "Companion Callsign" would be changed to "CQCQCQ". All stations within range of W$1SSS would see the transmission as originating from W$1QQQ and going to CQCQCQ just like that station was local (but the "Departure Repeater Callsign" would be "W$1VVV G" and the "Destination Repeater Callsign" would be "W$1SSS"). Replying would still be done the same way as before since the received "Companion Callsign" is ignored when programming the radio to reply.

Every "terminal" (station) has an IP address assigned to it for DD purposes. The address is assigned from the 10.0.0.0/8 address range. The D-STAR gateway is always 10.0.0.2. The router to the Internet is always 10.0.0.1. The addresses 10.0.0.3-31 are reserved for local-to-the-gateway (not routable) use. What this makes possible is the ability to send Ethernet packets to another station by only knowing that terminal's IP address and the remote station can directly respond based solely on IP address. This is because the gateway software can correlate IP address with callsign and ID. This makes it possible to route DD Ethernet packets based on the "Companion Callsign" or based on IP address with "Companion Callsign" set to "CQCQCQ".

Key points of D-STAR routing:

1) Simplex uses only "Own Callsign" and "Companion Callsign". The repeater callsigns are set to spaces. "CQCQCQ" is the universal "calling everyone" callsign and ID for the "Companion Callsign" only.

2) Local repeater use sets only the "Departure Repeater Callsign" repeater. "Destination Repeater Callsign" is set to spaces.

3) Accessing the gateway requires the local repeater in "Departure Repeater Callsign" and the local gateway in "Destination Repeater Callsign".

4) Routing outside of the local zone requires programming the radio to access the gateway.

5) Inter-zone and inter-area routing by station callsign and ID requires the radio to be programmed to access the gateway.

6) Inter-zone and inter-area routing using the slash with a repeater callsign and ID requires the radio to be programmed to access the gateway.

7) DD routing based on assigned 10.0.0.0/8 IP address requires the radio to be programmed to access the gateway.

8) Routing is unidirectional. Responding station(s) must program their radios correctly to reach the calling station.

a) D-STAR routing is not linking, it is routing of the bit stream from source to destination.

b) Responding requires proper configuration of the responding radio.

## Icom Enhancements

The D-STAR specification defines the protocols and general network architecture. It is left to the manufacturers (including independent developers) to refine this into an operational model. As Icom is the first manufacturer of D-STAR radios and software, they have had a significant amount of flexibility in their implementation.

On the RF side, the data frame content in the DV protocol is generally undefined (every 21st frame and the last frame are reserved). This is the only opportunity for customization of the RF signal in the specification that does not have to be forced compatible with other radios and software. Icom has made use of it for 3 different and distinct functions which are "hidden" from each other by a "control" byte occupying the first byte of the 3 byte data frame:

1) Control functions including repeating the RF header periodically to help receiving radios get a valid RF header for informational purposes only when operating under noisy conditions.

2) Repetitive transmission of a 20 character message for display on the receiving radio(s)' front panel. The message is displayed upon reception in a scrolling manner.

3) Passing 8 bit data between data ports on the radios.

Item 1 is solely used by the radios to recover sending callsign information on a noisy connection. Because it is not defined in the specification, it cannot be used to alter a header that has already been passed. It is very unlikely that these will be received more intact than the original RF header because the data frames do not have forward error correction.

Item 2 exists to allow a person to put a short announcement message in their transmission. This is not modifiable while transmitting and very difficult to change in-between transmissions.

Item 3 is usable by any application connected to the serial port on an Icom radio. As indicated, the data on the serial port is not placed into the data frames unaltered. It is transmitted 1 or 2 bytes at a time with the Icom "control" byte occupying the first byte of the data frame. Icom also implemented always-enabled xon/xoff flow control. This precludes the use of those characters in the data stream. Because of bit errors that can be received, it also precludes use of xon/xoff flow control by the attached device as there is no way to know if an xoff is received or issued by the radio. Some radios use the serial port to control the radio by using certain bytes (mostly 0xF0-0xFF) to denote control information. This means that these characters are also off-limits.

Icom makes use of item 3 to pass GPS information. Because this information is passed using the item 3-type control bytes, it is visible to the serial port on radios not running in GPS mode (GPS modes block the serial port from the data frames). The original GPS format (implemented on all radios that support GPS) is standard NMEA GPS strings with checksums followed by an ASCII line that contains the transmitting callsign, ID, and a duplicate of the 20 character message being sent to the front panel. This line is generated by the transmitting radio and is not interpreted by the receiving radio. GPS-A mode was introduced to provide a more concise and reliable position representation. GPS-A mode consists of a single APRS format line prefixed by a CRC.

It is important to remember that all uses of the data frames are not protected from bit errors nor do they have any indication of bit errors. In the case of item 1, the RF header contains a CRC for validation. Item 2 has no validation. Validation with item 3 is left to the connected applications.

The other area open for significant customization is the gateway operation and the gateway-to-gateway communications. The latter may be addressed in a future addition to the D-STAR specification. However, it was never the intention of the JARL to dictate how software or devices operate so gateway operation will be left to the manufacturers and independent developers.

The Icom gateway software is currently at version 2 released this year. Version 1 was based on a mesh network design where all gateways communicate directly with all other gateways. Specifically, when one gateway's table(s) gets updated, that gateway contacts all other gateways with the new information. While this works well with small gateway networks, it became obvious that it would not be supportable with the explosive growth experienced in the world-wide D-STAR network. To meet the growth requirements of the new D-STAR network, Icom introduced G2 (Gateway Version 2) earlier this year.

G2 is designed around a hub and spoke topology for information exchange but maintaining the basic D-STAR communications architecture where DV and DD communications occur directly between the gateways. Gateways exchange information with the central Trust Server regarding their callsign (zone call), Internet IP address, connected area repeaters, and recent activity on the area repeaters. This information is mentioned in generalities in the D-STAR specification but the actual data interchange format is not defined. To accomplish the allocation of 10.0.0.0/8 addresses for individual stations, the G2 software automatically reserves 8 addresses for each callsign registered through the gateway registration software. Each registered user can designate which ID is assigned to the specific IP address and also a DNS name in the dstar.local domain. This information is also exchanged with the central Trust Server. The IP address groups are obtained from the Trust Server so there is no duplication in the network. The central Trust Server then broadcasts the information received from individual gateways to all of the gateways so they can quickly determine which gateway a user or repeater is connected to and the Internet IP address of that gateway.

One final note on the G2 gateway: due to a programming restriction in station registration form, G2 restricts the ID character (8$^{th}$ character of callsign) to a space or uppercase alphabetic (no numbers). This restriction is specific to the G2 software, not to the D-STAR specification.

### Independent Enhancements

In addition to Icom, there have been other developers create applications for the D-STAR systems. Icom's adherence to the D-STAR specification has lead to DStarMonitor which monitors the repeater/gateway communications and parses the information into database entries and parses the Icom "serial port" data to a TCP port for use by an application just like the application would work connected to a radio monitoring the repeater frequency.

DPlus is a multifaceted application that runs on the gateway PC. It also monitors the repeater/gateway communications and provides enhanced features like echo, prerecorded announcements, DV reflectors, etc.

D-PRS is a translation specification for the Icom GPS information to APRS format and to use the Icom serial data port for reliable transport of APRS packets between APRS clients.

Other applications tend to concentrate on the Icom serial port. These include messaging, form completion, and small file transfer.

There are independent gateway projects underway at the time of writing of this paper. Further information may be available for presentation at DCC 2008.

### Conclusion

The D-STAR specification is a focused definition of 2 RF digital protocols: digital voice and digital data. It also defines the control information contained in every transmission and the network architecture that control information supports. Finally, it defines the repeater/gateway communications protocol to ensure compatibility across platforms between gateways and repeaters.

Digital voice (DV) is a protocol designed to convey audio information. Because of its design, there is limited bandwidth for very low speed data to accompany the audio information. Icom has made available a serial port to utilize a subset of that low speed data bandwidth by applications attached to the radios. Icom uses this same subset of low speed data bandwidth to convey GPS information to remote radios.

Digital data (DD) is a protocol designed to convey Ethernet packets. Implied D-STAR network design designates the 10.0.0.0/8 IP address range for use to connect network devices via the DD protocol. The DD protocol is not limited to IP packets although extra features are available to IP packets.

The D-STAR specification defines the repeater/gateway IP communications to facilitate multiple repeater or gateway platforms. This portion of the specification reinforces the overall network architecture mentioned elsewhere in the specification. The specification purposely does not define how any function such as the gateway operates. Only the communication between entities is defined in the specification. The gateway/gateway communication is yet to be included at the time of the writing of this paper.

The D-STAR network architecture is purely based on callsign and station ID. The callsign and ID applies to individual stations, repeaters, and gateways. All routing is handled within the gateways. The user only needs to know who they want to talk to and what the callsign and ID is of the local repeater.

D-STAR brings true world-wide station-to-station communication capabilities to amateur radio.

### References


Japan Amateur Radio League, Inc. (2005). *Translated D-STAR System Specification.* Retrieved 8 23, 2005, from JARL English: **www.jarl.com/d-star/shogen.pdf**.

TAPR APRS Working Group. (2000, 31 8). *APRS Protocol Specification Version 1.0.* Retrieved from TAPR: **www.tapr.org/aprs_working_group.html**.


QEX

### Next Issue in *QEX*

Rudy Severns, N6LF, begins a series of articles describing his "Experimental Determination of Ground System Performance for HF Verticals." In Part 1, Rudy describes the antenna test range, test equipment and test procedures that he used in his experiments. Rudy has tested both on-ground and above-ground radial systems, and compares his test results with NEC-based antenna modeling software predictions.

In the Jan/Feb 2009 issue, Rudy also describes his experiments with ¼ λ vertical antennas using a small number of ¼ λ and shorter on-ground radials. Antennas like this are often used for Field Day and other portable operations. The results may surprise you!

In future accounts of his experimental results, Rudy will compare several different systems of elevated radials, describe the effects of radial numbers on the characteristics of ¼ λ and shorter loaded vertical antennas and look at some problems of ground systems for multiband verticals, where a range of 7 to 30 MHz must be accommodated. Later in the series, Rudy will also report on some experiments with a full size 160 m ¼ λ vertical.

If you use or experiment with vertical antennas, don't miss a single issue in this series!





UNITED STATES POSTAL SERVICE

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

| 1. Publication Title | 2. Publication Number | 3. Filing Date |
|---|---|---|
| QEX | 0 8 8 6 - 8 0 9 3 | September 25, 2008 |
| 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: 225 Main Street, Newington, Hartford County, CT 06111-1494
Contact Person: Amy Hurtado
Telephone: 860-594-0257

8. Complete Mailing Address of Headquarters or General Business Office of Publisher: 225 Main Street, Newington, Ct 06111-1494

9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor

Publisher: Harold Kramer, 225 Main Street, Newington, CT 06111-1494

Editor: Larry Wolfgang, 225 Main Street, Newington, CT 06111-1494

Managing Editor: Larry Wolfgang, 225 Main Street, Newington, CT 06111-1494

10. Owner

| 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

12. Tax Status: ☒ Has Not Changed During Preceding 12 Months

PS Form 3526, September 2007 (Page 1 of 3) PRIVACY NOTICE: See our privacy policy on www.usps.com

| 13. Publication Title: QEX | 14. Issue Date for Circulation Data Below: Sept/Oct07-July/Aug08 | Sept/Oct 08 |
|---|---|---|
| 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) | 7064 | 8150 |
| b. Paid Circulation (1) Mailed Outside-County Paid Subscriptions Stated on PS Form 3541 | 4822 | 5529 |
| (2) Mailed In-County Paid Subscriptions Stated on PS Form 3541 | 0 | 0 |
| (3) Paid Distribution Outside the Mails Including Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Paid Distribution Outside USPS® | 1313 | 1435 |
| (4) Paid Distribution by Other Classes of Mail Through the USPS (e.g. First-Class Mail®) | 430 | 388 |
| c. Total Paid Distribution (Sum of 15b (1), (2), (3), and (4)) | 6565 | 7352 |
| d. Free or Nominal Rate Distribution (1) Free or Nominal Rate Outside-County Copies included on PS Form 3541 | 89 | 102 |
| (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) | 44 | 9 |
| (4) Free or Nominal Rate Distribution Outside the Mail (Carriers or other means) | 83 | 83 |
| e. Total Free or Nominal Rate Distribution (Sum of 15d (1), (2), (3) and (4)) | 216 | 194 |
| f. Total Distribution (Sum of 15c and 15e) | 6781 | 7546 |
| g. Copies not Distributed | 283 | 604 |
| h. Total (Sum of 15f and g) | 7064 | 8150 |
| i. Percent Paid (15c divided by 15f times 100) | 96.82% | 97.43% |

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 08 issue of this publication.

17. Signature and Title of Editor, Publisher, Business Manager, or Owner: HAROLD KRAMER, CEO — Date: September 25, 2008

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

KENWOOD
Listen to the Future

# Kenwood & AvMap
## The Ultimate APRS® Combination

AvMap G5

TM-D710A


## Two Great Companies...Providing One Outstanding Solution!

**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#01308

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

AvMap SATELLITE NAVIGATION
Tele Atlas Maps
www.avmap.us • e-mail: info@avmap.us
1.800.363.2627