$5

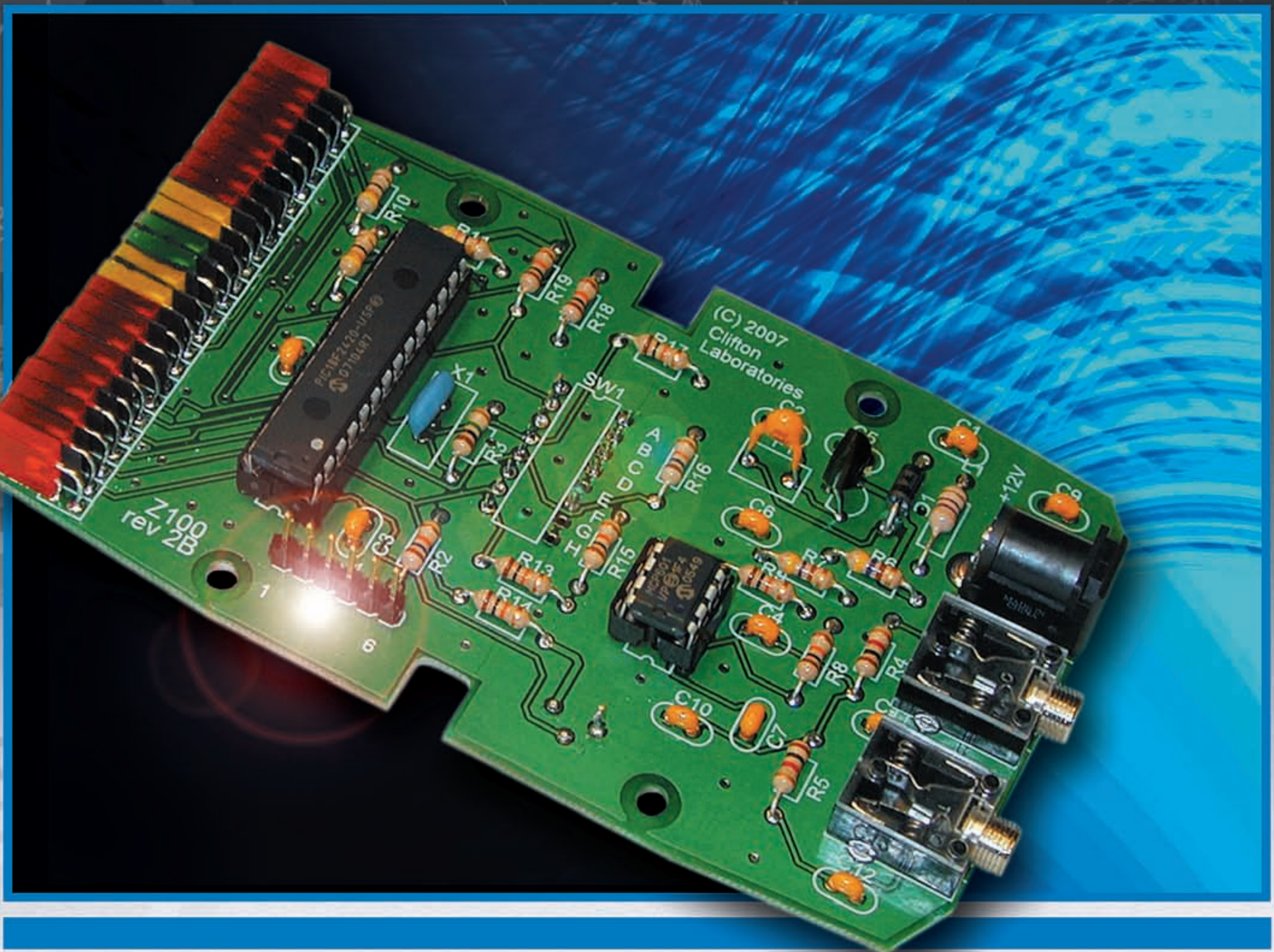
January/February 2008

www.arrl.org

QEX

A Forum for Communications Experimenters

Issue No. 246



K8ZOA describes the Z100, a tuning aid that will help you quickly zero beat a CW signal or tune RTTY signals.

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

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 243

Harold Kramer, WJ1B
Publisher

Larry Wolfgang, WR1B
Editor

Lori Weinberg, KB1EIB
Assistant Editor

L. B. Cebik, W4RNL
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

Devon Neal, KB1NSR
Technical Illustrator

Joe Shea
Production Assistant

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;

Elsewhere by Surface Mail (4-8 week delivery): ARRL member $31, nonmember $43;

Canada by Airmail: ARRL member $40, nonmember $52;

Elsewhere by Airmail: ARRL member $59, nonmember $71.

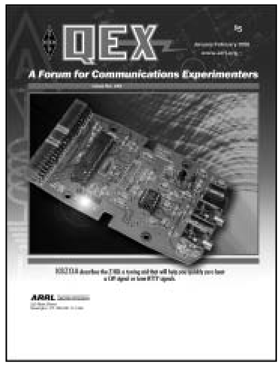
Members are asked to include their membership control number or a label from their QST when applying.

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.

January/February 2008

About the Cover

Jack Smith, K8ZOA, presents the Z100 tuning aid; a modern version of a station accessory that will help you quickly zero beat a CW signal or tune RTTY signals. A PIC microcontroller provides selectable center frequencies and tuning steps for the LED display.



In This Issue

Features

Columns

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

Many of you are looking for the second part of Cornell (KW7CD) Drentea's article about his *Star-10* transceiver. Unfortunately, you will have to wait for the Mar/Apr issue of *QEX* to read that article. While preparing the schematic diagrams for that article, our graphics art staff had to shift their focus on completing drawings for the February issue of *QST*. Rather than delay the publication and mailing of this issue of *QEX*, we decided to send the magazine to the printer without the *Star-10* article. I share your disappointment, but trust you will enjoy the articles in this issue while looking forward to reading more about the *Star-10* in the next issue.

Out and About

On September 28-30, 2007, I had the opportunity to attend the ARRL/TAPR Digital Communications Conference, held outside of Hartford, CT. This was a great opportunity to meet some *QEX* readers, as well as to learn about various projects on which some of the attendees have been working. In addition to a solid line-up of technical presentations all day Friday and Saturday, the conference included a demonstration room with both commercial products and homebrew projects of interest to those attending the conference. The conference concluded Sunday morning with an interesting presentation: "A Stroll Through Software Radio, Information Theory, and Some Applications" by Bob McGwier, N4HY.

There were many presentations about TAPR projects involving software defined radio. While I was aware of the concept of SDR, I really had not read much about what people were doing with this "new" radio concept. Terms like Gnu Radio, HPSDR, Janus, Ozy (or Ozymandias), Pinocchio, FPGA and more were flying around so fast my head was spinning! By the end of the conference, after talking with a few of those who obviously understood this "foreign" language, at least I had some idea what the terms meant, if not what they were actually talking about.

A few days before this conference, at a Technical Editorial Review meeting, we accepted an article from Milton Cram, W8NUE and George Heron, N1APB about a portable digital modem for PSK31 operation — no computer required. It's a very interesting project, and there it was at the DCC!

I spotted another interesting project in the demonstration room, and had the opportunity to talk with Jim Ahlstrom, N2ADR about his software defined radio SSB exciter project. Jim has since submitted his project for publication in *QEX*.

Have you thought about writing up your favorite project or technical investigations? As I told people at the DCC, your project doesn't have to be professionally packaged, and may even be a "work in progress." Perhaps you are uncertain about how to solve a particular problem or are looking for a better way to implement a circuit idea. By sharing the work you have done so far, other readers may come up with answers. *QEX* is about sharing ideas.

I also attended the AMSAT Space Symposium in Pittsburgh, PA on October 26-28. This was another first for me, and I was very interested to learn about the latest plans for Amateur Radio satellites. Of course there were many interesting presentations and quite a bit of news about the status of various AMSAT-NA projects.

I was especially intrigued by the number of presentations about educational projects and some of the ways that Amateur Radio and space concepts are being taught in school programs.

Bob Twiggs, KE6QMD, described his "PearlSat experiment." He cuts ping-pong balls in half and has students put various "experiments" inside, then tapes the balls back together. They string the balls like pearls, and hang the streamers below a weather balloon. The balloon reaches 100,000 feet before bursting and dropping the "PearlSats" back to Earth! The students had some very interesting results from their experiments, but the most interesting result was the students' interest in math and science, and their increased state standardized exam scores after this program!

Ivan Galysh described the "CanSat" competition. Various experiments and sensors are built into a soda can, and the cans are launched in a high power rocket.

Other presentations described contacts with the International Space Station as part of the ARISS or earlier SAREX program. The enthusiasm of the presenters was tremendous, and the impact the programs have had on young people is amazing. Patricia Palazzolo, KB3NMS, talked about the long-term impact of these programs on students from her school — PhDs, MDs, Engineers and so on. Quite a legacy.

What are you doing to learn, and to share your enthusiasm for Amateur Radio? We'd like to hear about it.

Jack R. Smith, K8ZOA

7236 Clifton Rd, Clifton, VA 20124, Jack.Smith@cliftonlaboratories.com

The Z100 CW Tuning Aid

Quickly zero beat that CW signal or tune RTTY signals perfectly with this project.

When separate transmitters and receivers were the norm, you quickly learned to "zero beat" CW stations before calling. With one hand holding the transmitter "spot" switch, the other adjusted the VFO frequency control until your frequency matched the station you were calling. When exactly matched, the difference tone — or "beat note" — vanished and you were ready to transmit.

For most of us, transceivers have long replaced separate transmitters and receivers, but matching your transmit frequency to the station you are working avoids leapfrogging up or down the band, as explained in the Sidebar. Transceivers offer various techniques for zero beating. For example, the Elecraft K2 transceiver injects a continuous audio tone via a "spot" switch. You then adjust the main tuning knob to match the spot tone with the incoming CW signal. This works because the injected tone matches the K2's CW offset frequency. Although effective, it's still inconvenient — you must press the spot button, tune, and then press the spot button again to turn the tone off. The process must be repeated during a QSO, if the other station drifts or if you hit the tuning knob accidentally.

Zero beating is nothing more than matching two audio frequencies — the incoming CW signal's pitch with your transceiver's sidetone frequency. The most elegant frequency indicating aid I've seen was in the TT/L and TT/L-2 radioteletype demodulators.[1] When correctly tuned, the right and left bars of a "magic eye" vacuum tube were equal length — tuning errors caused the bars to be unequal. This concept, with some improvements, and using LEDs instead of vacuum tube displays, has appeared in many Amateur Radio data devices over the years. This was the basis of a stand-alone tuning aid from HAL Communications 25 years ago, the SPT-1 Spectra-Tune.[2]

Although developed for RTTY and data tuning, with modest changes they make an excellent CW tuning indicator. I recently designed and built a microcontroller-based frequency matching aid, the "Z100 CW Tuning Aid," based on these concepts. Figure 1 shows a Z100 in use with my Elecraft K2 transceiver. (Of course, the Z100 also is useful for RTTY and data modes.) To make it even more flexible, I've provided the source code and programming connections to allow customization.

[1]Notes appear on page 10.

K8ZOA

Figure 1 — The Z100 tuning aid sits atop the author's Elecraft K2 transceiver.

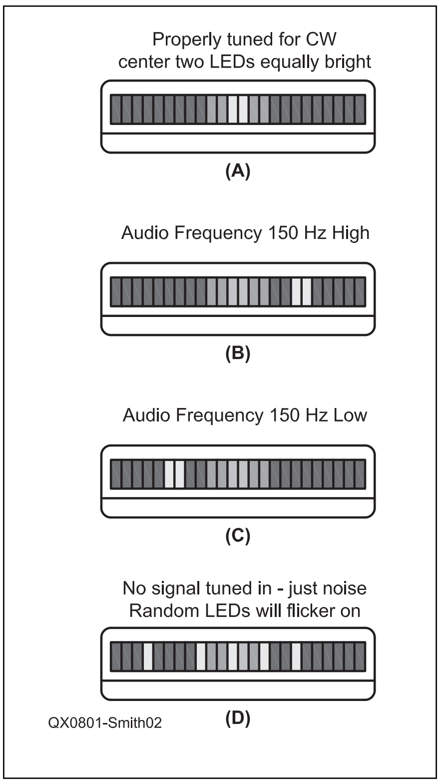


Figure 2 — Part A shows the center LEDs glowing, to indicate that the CW signal is properly tuned. Part B shows two LEDs to the right of center glowing, to indicate that the signal is about 150 Hz too high. Part C shows two LEDs to the left of center glowing, to indicate that the signal is about 150 Hz too low. Part D shows random LEDs flickering from noise when no signal is tuned.

Tuning Using the Z100

The Z100 display uses 24 LEDs. There are two green LEDs at the center, flanked by two yellow LEDs on either side, and nine red LEDs on each end. A signal is properly tuned when one or both of the green LEDs flash in sequence with the keying, as illustrated at Figure 2A. If tuned high, LEDs right of center will illuminate, as in Figure 2B; if low, the illuminated LEDs will be left of center, as in Figure 2C. During key-up periods, or when there is no signal, a random display will be seen as in Figure 2D. Each LED represents a

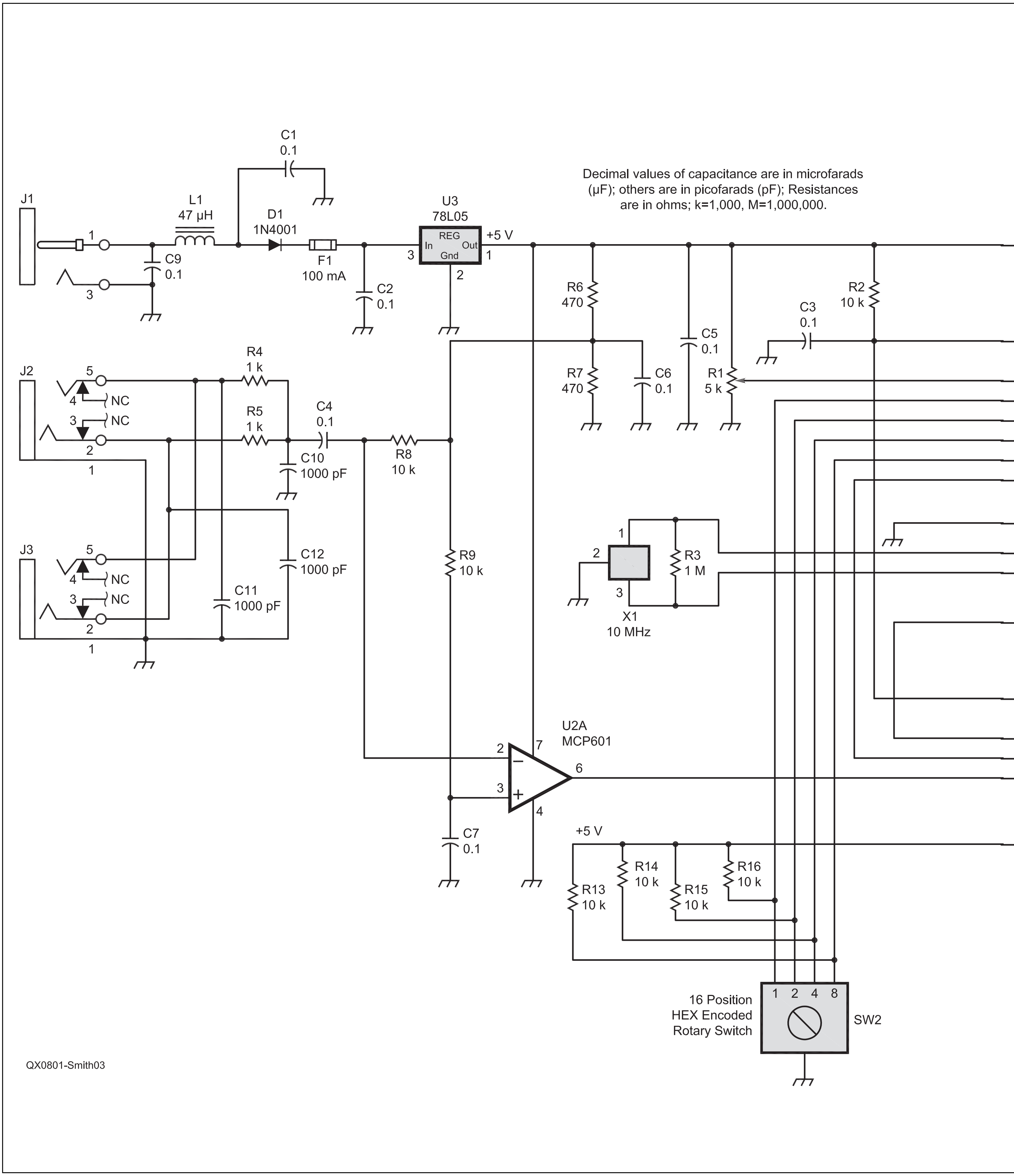
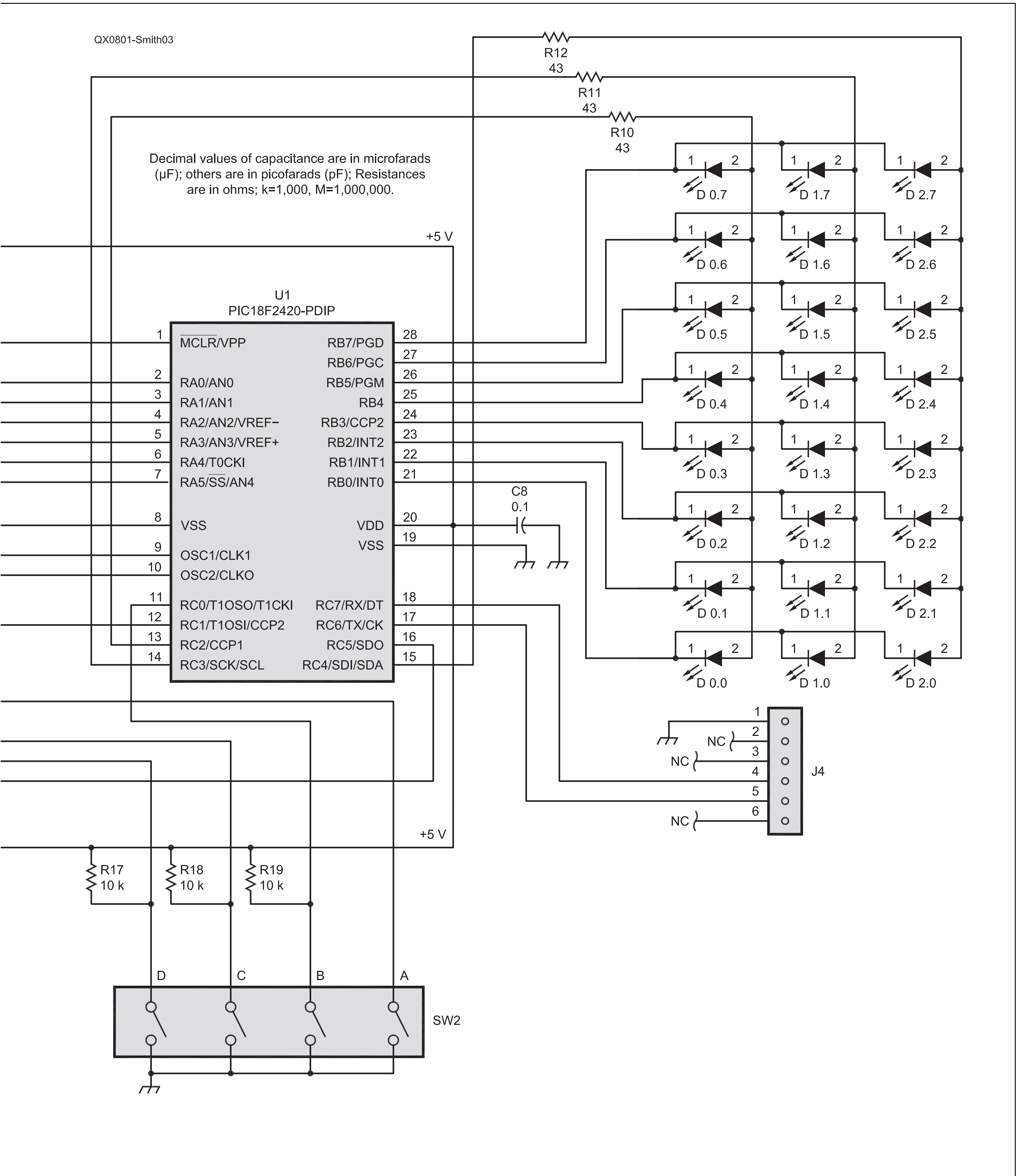


Figure 3 — This drawing shows the schematic diagram of the Z100 circuit. Parts list on next page


QX0801-Smith03
R12
43
R11
43
R10
43
Decimal values of capacitance are in microfarads (µF); others are in picofarads (pF); Resistances are in ohms; k=1,000, M=1,000,000.
+5 V
U1
PIC18F2420-PDIP
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF−
RA3/AN3/VREF+
RA4/T0CKI
RA5/SS/AN4
VSS
OSC1/CLK1
OSC2/CLKO
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RB7/PGD
RB6/PGC
RB5/PGM
RB4
RB3/CCP2
RB2/INT2
RB1/INT1
RB0/INT0
VDD
VSS
RC7/RX/DT
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
C8
0.1
D 0.7
D 1.7
D 2.7
D 0.6
D 1.6
D 2.6
D 0.5
D 1.5
D 2.5
D 0.4
D 1.4
D 2.4
D 0.3
D 1.3
D 2.3
D 0.2
D 1.2
D 2.2
D 0.1
D 1.1
D 2.1
D 0.0
D 1.0
D 2.0
J4
NC
NC
NC
+5 V
R17
10 k
R18
10 k
R19
10 k
D
C
B
A
SW2

Parts List for Figure 3

Ref Des	*Value*	*Type*	*DigiKey, Mouser or Manufacturer's Part Number*	*Quantity*	*Identifier*
C1, C2, C3, C4, C5, C6, C7, C8, C9	0.1 µF	Monolithic ceramic cap, 50 V or greater, 0.1 inch lead spacing	BC1084CT-ND	9	104
C10, C11, C12	1000 pF	Ceramic capacitor, 50 V or greater; 0.1 inch lead spacing	B37981M1102K000	3	102
D1		1N4001, or any similar power diode.	583-1N4001-B	1	
D0.0, D0.1, D0.2, D0.3, D0.4, D0.5, D0.6, D0.7, D1.0, D1.7, D2.0, D2.1, D2.2, D2.3, D2.4, D2.5, D2.6, D2.7		Rectangular LED, 0.091 inch wide	604-WP153HDT	18	Red rectangular LED
D1.1, D1.2, D1.5, D1.6		LED	604-WP153YDT	4	Yellow rectangular LED
D1.3, D1.4	LED	604-WP153GDT	2		Green rectangular LED
F1	100 mA	Fuse; resettable polyfuse type	RXEF010-ND	1	XF010
J1	DC Power	CP-002A	1		
J2, J3	PG203J	Kobiconn 161-3507	2		
J4	MA06-1	571-6404526	1		6 position male header
L1	47 µH	RF Choke, 200 mA or greater current rating	542-78F470J-RC	1	Yel-viol-blk
R1	5 kΩ	Trimpot	Bourns R3386P case	1	502
R2, R8, R9, R13, R14, R15, R16, R17, R18, R19	10 kΩ	¼ W axial carbon film resistor, 5%	CF1/4C103J	10	Bwn-blk-org
R3	1 MΩ	¼ W axial carbon film resistor, 5%	CF1/4C105J	1	Bwn-blk-grn
R4, R5	1.0 kΩ	¼ W axial carbon film resistor, 5%	CF1/4C102J	2	Bwn-blk-red
R6, R7	470 Ω	¼ W axial carbon film resistor, 5%	CF1/4C471J	2	Yel-viol-bwn
R10, R11, R12	43 Ω	¼ W axial carbon film resistor, 5%	CF1/4C430J	3	Yel-org-blk
SW1		Hex rotary DIP switch		1	Rotary, HEX Encoded. 0…F
SW2		4 position DIP switch		1	4 position on/off
U1		PIC18F2420-PDIP	579-PIC18F2420-I/SP	1	18F2420
U2		MCP601, DIP-8	579-MCP601IP	1	MCP601
U3	78L05	LM78L05ACZ, 5 V regulator, TO-92 case	511-L78L05ACZ	1	78L05
X1	10 MHz	10 MHz resonator	CSTLS10M0G53-B0	1	3 leads, oval shaped body; 10M or 10MZ, or similar ID
		Plastic enclosure with hardware Hammond 1553B with holes milled		1	
		Printed circuit board, Z100, Rev 2.B		1	
		Cardstock shim stock, approx 0.0085 inch thick, two lengths		1	
		Clear lens for enclosure		1	
		Strip gray optical filter film		1	
		8 pin DIP socket		1	
		28 pin DIP socket 0.3 inch spacing		1	
		Adhesive feet		4	
		Condensed instructions adhesive label		1	
		Power cable		1	
		Audio jumper cable; 3.5 mm male, stereo both ends		1	
		4-40 × 1/4 inch stainless steel machine screws		4	

Why Should I Care About My CW Frequency?

What happens if you are in a CW QSO and you don't match the other station's beat note to your transceiver sidetone offset?

Let's start with the case where you do match the two frequencies. As Figure S1A and B illustrate, your radiated transmit frequency matches the other station's, the desired result.

Figure S1C and D show the case where you do not match the received signal with your transceiver's sidetone offset. Your transmitted frequency differs from the other station's transmitted frequency by the amount of the tuning error. This means your QSO occupies more frequency space than necessary, not a good thing. However, this is not the only problem.

If the station you are working also uses a transceiver (highly probable) then your QSO partner may retune his transceiver to match your frequency. On the next exchange, you will find him at a different frequency and you will then retune. The result is a QSO that starts on one frequency (or pair of frequencies) and then leapfrogs up or down the band with each exchange, as illustrated at Figure S2.

Of course, one possible solution to the problem is to use your transceiver's "receiver incremental tuning" or RIT, to adjust the receiver section of your transceiver. In this case, your transmit frequency stays put and leapfrogging should not occur. Or, you can keep the RIT off and correctly tune the stations you work, perhaps using a Z100 to reduce guesswork.

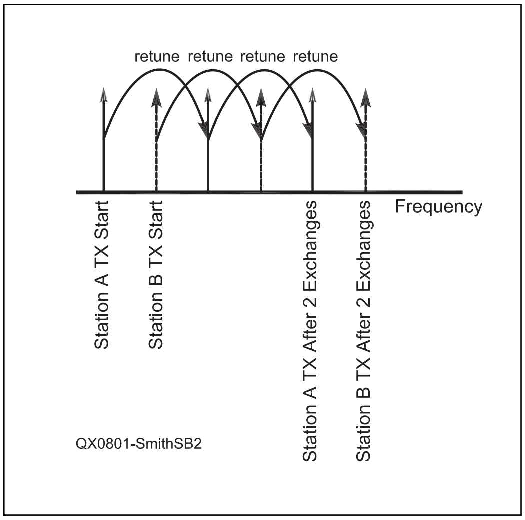


Figure S2 — This figure illustrates the effect of the two stations continuing to retune their receivers as they chase each other up the band. This can be frustrating for both stations.

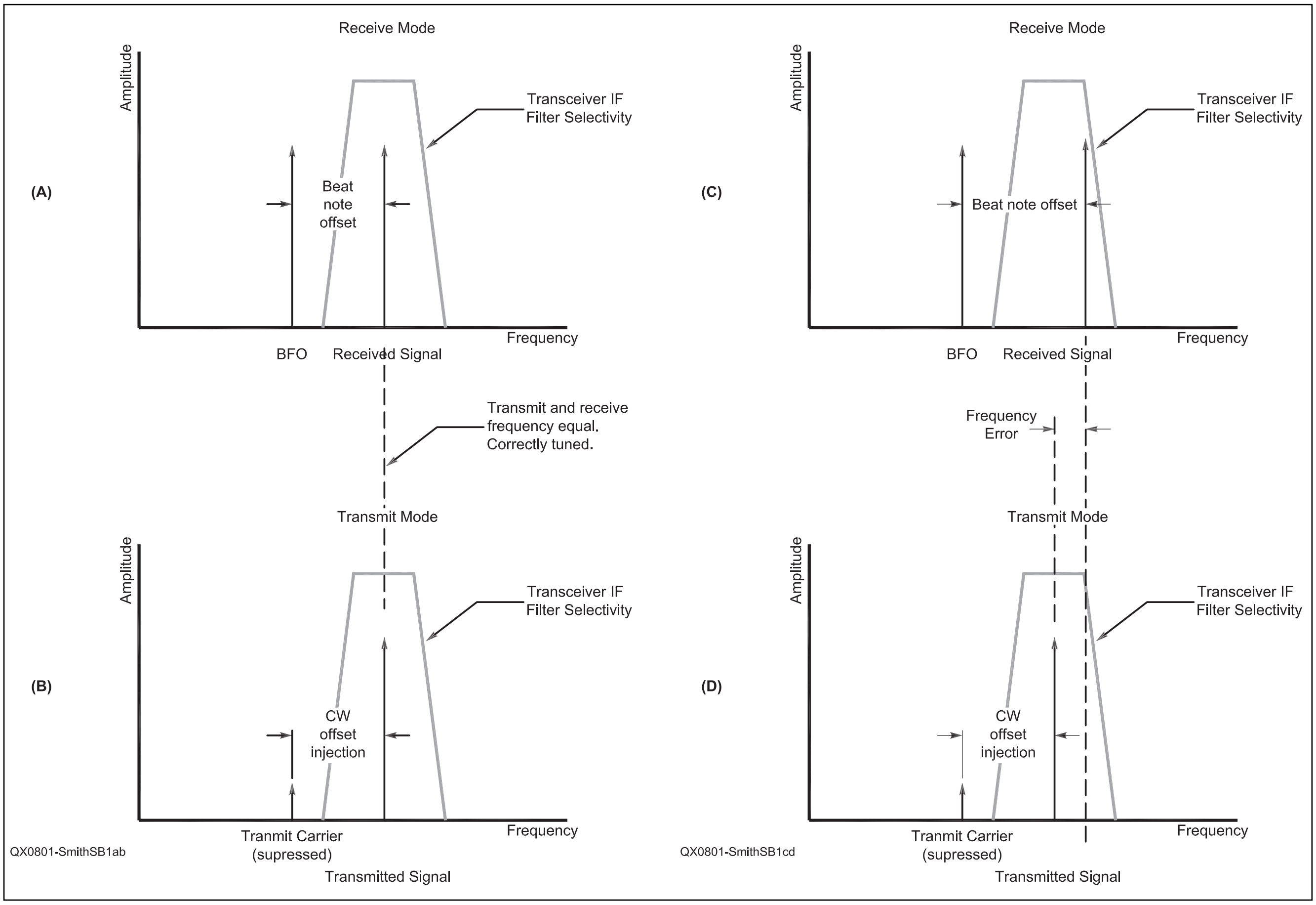


Figure S1 — Parts A and B show a receiving station that has zero beat the transmitting station's CW signal. When the first receiving station switches to transmit, the first transmitting station will receive the signals centered in their receive passband. Parts C and D show a receiving station that has tuned too high on the transmitted signal. When the two stations change transmit/receive conditions, the new receiving station will hear the transmitted signal as being too high in frequency. If the second receiving station retunes their receiver to correct this tuning error on the part of the first station, the two stations can chase each other up the band.

frequency error of 25 Hz (or 50 Hz, depending on an option switch setting) so the Z100 indicates both the direction to tune and the amount of error.

Circuit Description and Theory of Operation

As Figure 3 shows, the heart of the Z100 is a PIC18F2420 programmable microcontroller with 16 Kbytes of program memory, 768 bytes of RAM and 256 bytes of nonvolatile EEPROM. Audio arrives through either J2 or J3, standard three-circuit 3.5 mm (1/8 inch) stereo phone jacks. The tip and ring connections for J2 and J3 are bridged together by R4 and R5, 1 KΩ resistors, permitting either channel to drive the display. Since J2 and J3 are wired in parallel, one may be used as an input and the other as an output, simplifying transceiver connections. Capacitors C10, C11 and C12, (1000 pF) bypass RF that might otherwise enter the audio circuitry.

Incoming analog audio is clipped and converted to a 0/+5 V logic level signal in U2, a Microchip MCP601 rail-to-rail output op-amp, run open loop as a limiter or "slicer." As little as a 50 mV peak-to-peak analog input signal drives U2 to full saturation, permitting the Z100 to be used with either headphone or speaker audio levels. The limiting action of U2 is shown in Figure 4. Although running an op-amp open loop as a limiter is not considered optimum practice, at the low audio frequencies involved in the Z100 the MCP601 works more than adequately. Since U2 runs from a single +5 V supply, its inverting and noninverting inputs are biased at 2.5 V dc via a voltage divider (R6 and R7) and isolation resistors R8 and R9. Analog audio coupled to the inverting input via C4 thus causes the output from U2 to swing between 0 and +5 V, with amplitude variations wiped out.

The U2 logic level output connects to U1, the 18F2420 PIC. U1 determines the input audio frequency and sorts it into twenty-four 25-Hz-wide "bins," and turns on the associated LED until the next frequency measurement. Approximately 200 to 300 frequency measurements are made per second. LEDs D0.1 through D2.7 connect to U1 in a 3 × 8 matrix. To illuminate a particular LED, its anode bus (connected to pins RC2, RC3 and RC4 through current limiting resistors R10, R11 and R12) must be high, and its cathode bus (connected to Port B, pins RB0 through RB7) must be low. The matrix permits 11 pins to control 24 LEDs.

The U1 time base is provided by X1, a 10 MHz ceramic resonator.

The Z100 firmware provides for saving and recalling 16 stored center frequencies, direct center frequency setting and 25 and 50 Hz step selection through switches SW1 and SW2. Resistors R13 through R19 are pull-up resistors for these switches, whose settings are read by seven of the pins on U1.

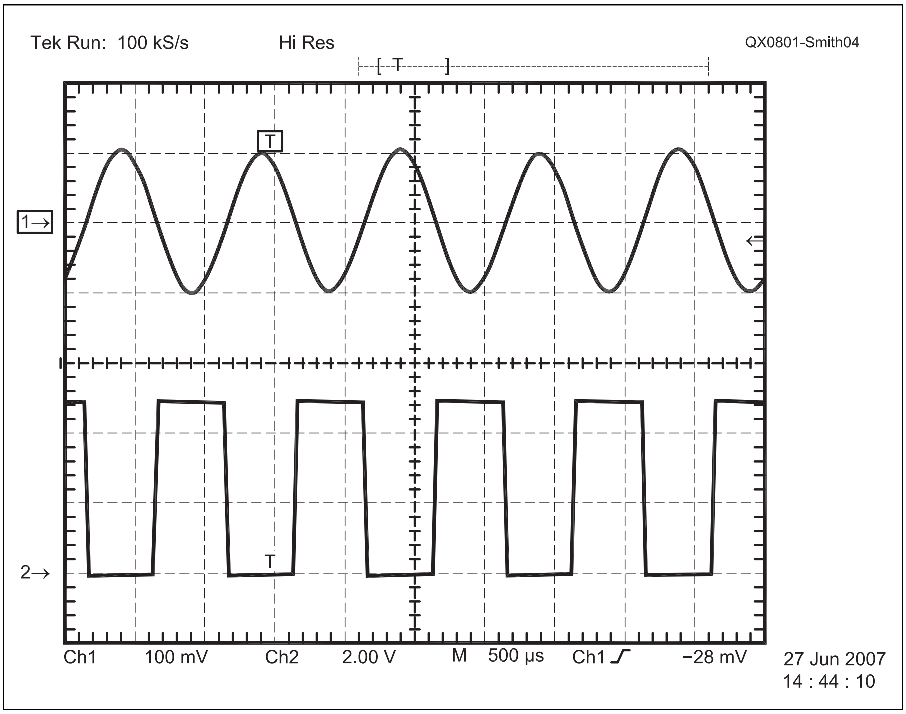


Figure 4 — This oscilloscope display shows the effect of the Z100 limiting action on the input signal. The top trace shows the sine wave audio input and the lower trace shows the clipped square wave of the U2 output.

Switch SW1 is a four section DIP slide switch controlling the following functions:

• SW1A resets U1 when closed.

• SW1B selects save to memory or to use saved value.

• SW1C sets frequency from potentiometer R1 or uses memory value.

• SW1D selects 25 Hz or 50 Hz frequency step per LED.

SW2 is a 16-position hex-encoded rotary switch used to select memory positions. Frequency settings are made via potentiometer R1, whose wiper voltage is read by the A/D converter built into U1, and translated to a target frequency via the Z100 firmware.

U1 is first programmed with a "bootloader" stub code, permitting the main firmware to be entered over a standard serial port, without using a specialized PIC programmer. To avoid the cost of an RS232 conversion IC — rarely, if ever, used once the code is developed — the Z100 brings the U1 logic level serial pins out to J4, a 6-position 0.1 inch header strip. The wiring for J4 is compatible with the TTL-232R USB-to-logic serial interface cable from Future Technology Devices International Ltd. The programming interface is provided to encourage Z100 owners to experiment with the stock firmware, and for possible "official" firmware updates I may provide.

Power to the Z100 is provided through a coaxial power jack, J1, and is regulated to +5 V by U4, a 78L05 three-pin voltage regulator. L1, C1, C2 and C9 reduce radiated interference via the power line. Diode D1 offers reverse voltage protection, and polyfuse F1 protects the power source against short circuits within the Z100. The Z100 will work with a voltage input between 8 V and 15 V dc. Current requirements are less than 50 mA.

Z100 Firmware

The Z100 firmware is written in BASIC, using Mechanique's Swordfish compiler.[3] Although I used a licensed version, the code is simple enough to fit within the free Swordfish SE compiler, should you wish to change it.

The firmware performs three main functions:

• Measure the incoming signal's frequency.

• Associate the frequency with the appropriate LED and illuminate that LED.

• Check for user commands and take appropriate action.

Let's look at each in more detail.

The classical method of measuring frequency is to count the number of cycles occurring in a specific gate period with one of the 32-bit counter modules internal to U1. Measuring frequency to within 10 Hz using this approach requires a 100 ms gate, limiting us to 10 frequency measurements per second, too slow for useful real-time display. The Z100 firmware reverses this approach, gating the U1 internal 2.5 MHz instruction cycle with the output from U2, as illustrated in Figure 5. This permits measuring the incoming signal period with 400 ns resolution. (Frequency is the reciprocal of period. Since the firmware measures only the logical low half period, the frequency in Hz equals 2 divided by the measured width.) Since a period measurement takes only as long as the logical low half period itself, a 2 kHz frequency can be determined in only 250 µs. The price paid for this test measurement is increased, error due to noise.

At 2 kHz, the U1 internal 400 ns step corresponds to a frequency error of 3.2 Hz, more than acceptable for 25 Hz frequency bins. In fact, period measurements are so fast that the firmware intentionally slows and smoothes data collection by computing the running average of the last eight readings, providing the software equivalent of low pass filtering. This digital signal processing filtering reduces the sample-to-sample jitter and significantly improves noise rejection.

Smoothed frequency measurements are assigned to a particular frequency range or "bin," numbered from –12 (lowest) to +11 (highest). The firmware computes the frequency limits of each bin based on the center frequency setting and the current bin width of 25 or 50 Hz. With 24 LEDs, the total frequency span is thus 600 Hz or 1200 Hz, respectively. Bin assignments use integer division, dividing the smoothed frequency value by the bin width, with a center frequency offset.

The bin index number is passed to a subroutine that illuminates the appropriate LED. As mentioned earlier, the 24 LEDs are arranged in a three anode by eight cathode matrix. The firmware efficiently handles this via a combination of integer division (anode selection) and a bit-array lookup table (cathode selection). The selected LED stays illuminated until the next frequency measurement.

The remaining firmware code deals with accepting user commands. The user command controls are read once every time a period measurement is taken.

The Z100 user controls are a compromise, driven by my desire for a compact, low-profile enclosure. I used a Hammond model 1553B soft-sided plastic box. This decision rules out conventional front and rear panel potentiometers, switches or rotary encoders. My solution is to mount the controls on the bottom surface of the printed circuit board, and access them through openings milled into lower enclosure half, as seen at Figure 6. The controls fall into three groups:

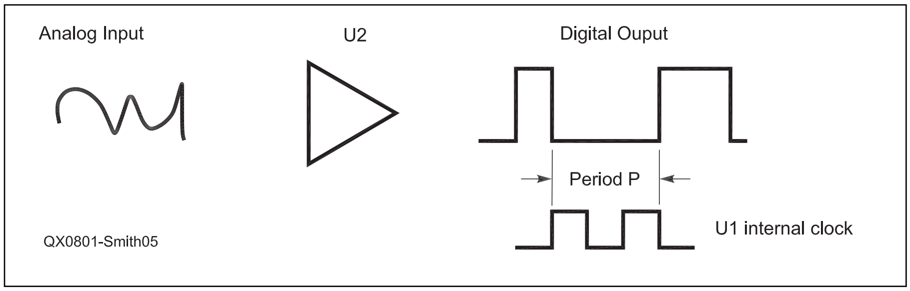


Figure 5 — An analog input signal is limited and squared in U2. The U2 output pulse is then used to gate the 2.5 MHz U1 internal clock. This permits measuring the incoming signal period with 400 ns resolution.

K8ZOA

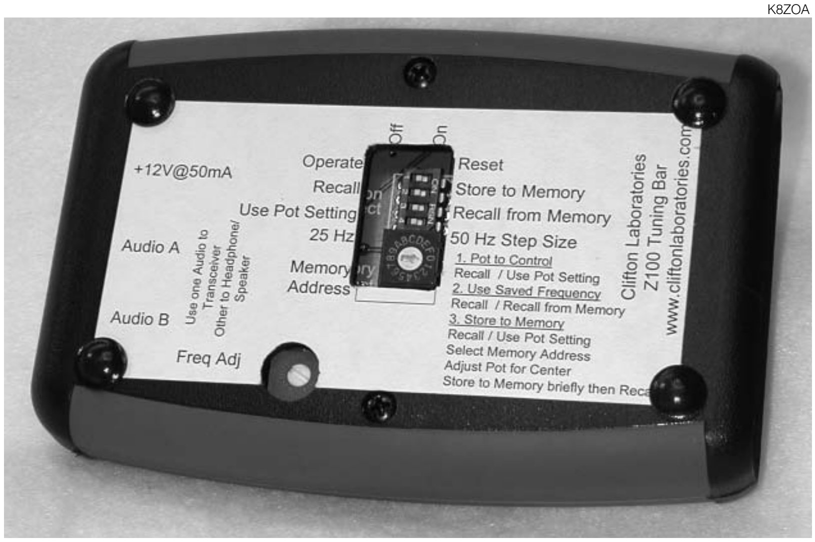


Figure 6 — To allow the use of a low profile enclosure, the operating controls are mounted to the bottom of the circuit board and access holes are cut in the bottom of the enclosure.

• Center frequency settings are accomplished by a cermet potentiometer, R1, configured as a voltage divider across the +5 V supply. The R1 wiper position is read by a 10-bit analog-to-digital converter module internal to U1. Given the internal minimum and maximum resistance of R1, the result is 950 to 1000 usable discrete A/D values from the 1024 theoretical 10-bit codes. The wiper position is read once every time the incoming frequency is measured, with the A/D value multiplied by 3 to convert to a center frequency setting. The usable center frequency range is approximately 200 Hz to 3000 Hz. Of course, with 25 Hz/step, the lowest center frequency permitting all LEDs left of center to illuminate is 300 Hz.

• Switches SW1B, C and D are also read every measurement cycle. (SW1A is tied to the U1 reset line and closing the switch immediately resets U1.) The operating modes are discussed elsewhere in this article.

• Switch SW2, the rotary memory position selector, is also read every measurement cycle. All switches are tied to +5 V via 10 kΩ pull-up resistors, and are read by U1 as either open (logic level 1) or closed (logic level 0).

Constructing the Z100

I built the first self-contained Z100 on a BusBoard Prototype Systems PR1553B prototyping board, a standard 0.1 inch perf board, but cut to fit the Hammond 1553B enclosure.

The Z100 shown in Figure 7 is built on a double sided, silk-screened and solder masked printed circuit board designed to fit in a Hammond 1553B enclosure. The stock 1553B enclosure has a black opaque end cap, so I had a replacement laser cut from 0.062 inch thick clear acrylic plastic. I found that applying a 35% transmission gray plastic film to the inside surface of the end lens improved the LED contrast, and makes the printed circuit board less visible when looking into the Z100.

If building a perf-board version of the Z100, the parts placement shown in this article should be followed. The most difficult wiring area is associated with U1 and the LEDs, where space is limited. I used a combination of bare no. 28 AWG wire with Teflon tubing and no. 30 AWG wire-wrap wire in building the prototype. It would not win a prize for the best looking construction job, but it works as well as the later PCB versions. You can make your own clear end lens from a scrap piece of acrylic plastic. When constructing the Z100 prototypes, I found it easier to score thin acrylic and snap it with a sharp blow than to cut it cleanly with a saw, finishing it to final dimensions with a file and sandpaper.

The input jacks and power connector should be attached to the perf-board with epoxy for mechanical support. One subtle point is that light leaks out the sides of the LEDs. You could use a piece of opaque paper or plastic as a baffle between adjacent LEDs. If you use the same LEDs as in the Z100 kit, you may wish to make the baffle from material approximately 0.008 inch thick so that it also spaces the LEDs uniformly when mounted on 0.1 inch centers.

Of course, there's nothing magic about the Hammond 1553B enclosure, and a Z100 design could be packaged in a variety of ways. I've put one into a homemade aluminum enclosure fashioned to resemble an Elecraft K2 transceiver, for example.

K8ZOA

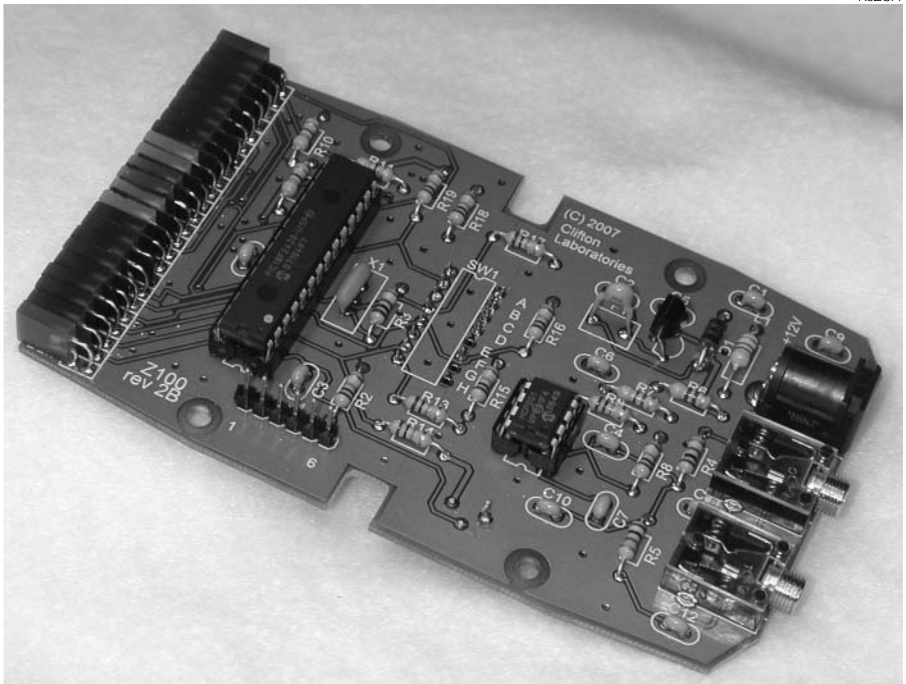


Figure 7 — Double sided, silk screened circuit boards are available from the author as part of a Z100 kit. The author also prototyped the circuit on perf board, and that is an alternative construction method.

Operating the Z100

Connect the Z100 to a source of +12 V dc power capable of supplying 50 mA. Audio from either the speaker or headphone output of your transceiver is connected to J2 or J3. The headphone or speaker itself can be connected to J3 or J2.

You select the operating modes with SW1B and SW1C. The Z100 supports three operating modes.

• Mode 1: Adjustable frequency operation; the center frequency is based upon the current potentiometer settings. (SW1B OFF, SW1C OFF.)

• Mode 2: The center frequency is based upon a previously stored non-volatile memory position, with 16 storage positions. (SW1B OFF, SW1C ON.)

• Mode 3: Program a center frequency into memory. (SW1B OFF, momentarily slide SW1C to ON to save the current center frequency to memory.)

Normal operation is in either Mode 1 or Mode 2; Mode 3 is used only to fill memory positions with your custom center frequency settings. My K2 is set for a 600 Hz sidetone, and that's where I keep the Z100 set. (The Z100 firmware includes preset memory values for common frequencies.)

• SW2, the rotary switch selects the storage or recall memory position.

• SW1D selects between 25 and 50 Hz per LED.

• SW1A resets U1, which is necessary if programming it via the serial interface connector, J4.

Final Observations

Although I've operated CW or data modes for many years, I've found the Z100 helps me quickly tune my K2 to exactly the correct frequency. I can get "close enough" by ear most of the time, but it's nice to know that I'm within 50 Hz of being perfectly tuned.

I offer a complete Z100 kit, including the printed circuit board, all parts, programmed PIC18F2420, a Hammond 1553 enclosure with machined openings, the clear acrylic end lens, gray film, power cable, 3.5 mm stereo jumper cable and all other parts required to duplicate the unit shown in the photographs, along with a detailed printed manual. The price is $59.95 plus $7.50 for Priority Mail shipping within the US (plus sales tax if you live in the Commonwealth of Virginia. international shipping is available upon request, as are fully assembled Z100s). If you decide to make your own Z100 without a PCB, a programmed PIC18F2420 is available from the author, or you may program it yourself if you have access to a suitable programmer. Further details on the Z100, including ordering information, a copy of the 40 plus page assembly and operating manual and the firmware source code, can be found at my Web page, **www.cliftonlaboratories.com**. Look for the Z100 page link.

Assembling a Z100 kit should take an average builder about 2.5 to 3 hours. All parts use through-hole mounting, and the only slightly challenging part is the mechanical skill required in installing 24 LEDs uniformly. Reports from the two dozen plus having already built a Z100 kit is that it is a project suitable for a beginner.

Notes

[1]The TT/L was originally published in *RTTY Magazine*, November 1964, with an improved version appearing the following year in QST. See I. Hoff, K8DKC, "The Mainline TT/L F.S.K. Demodulator," *QST*, Aug 1965, pp 27-36, and K. Petersen, "The Mainline TT/L-2 F.S.K. Demodulator, Part I," *QST*, May 1969, pp 28-34, and "The Mainline TTL/2 F.S.K. Demodulator, Part II," *QST*, Jun 1969, pp 24-28 (originally published as K. Petersen, "The Mainline TT/L-2 FSK Demodulator," RTTY Journal, Sept 1967, pp 4-12. See also, I. Hoff, "RTTY Indicator Systems," *QST*, Oct 1965, pp 21-26, presenting a rich variety of tuning concepts.

[2]The SPT-1 was briefly noted as a "new product" in P. Pagel, "New Products – HAL SPT-1 Tuning Indicator," *QST*, Aug 1985, p 16.

[3]More information, including downloading the free Swordfish SE compiler for 18F-series PICs, can be found at Mechanique's Swordfish Web site **www.sfcompiler.co.uk/swordfish/**.

Jack Smith, K8ZOA, has been licensed since 1961, first as KN8ZOA, and has held the Amateur Extra Class license since 1963. He received the BSEE degree from Wayne State University in Detroit in 1968 and a JD degree magna cum laude from Wayne State University School of Law in 1976. Presently retired, he has enjoyed a career involving both engineering and telecommunications law. He is a co-founder of the telecommunications consulting firm TeleworX and is the author of Programming the PIC Microcontroller with MBasic *(Newnes Publishing, 2005), as well as articles published in* QEX, *and* 73 Amateur Radio *magazines. His Web site is* **www.cliftonlaboratories.com**.

QEX

Professor Dr Thomas C. Baier, DG8SAQ

University of Applied Sciences, Prittwitzstrasse 10, 89075 Ulm, Germany, baier@hs-ulm.de

A Low-Cost, Flexible USB Interface

No parallel port on your laptop PC?
Use this interface and a USB port to control external devices.

After publication of my DDS-based, parallel-printer-port-controlled vector network analyzer, I have received a lot of requests asking to consider using the USB interface instead, since many modern notebook computers don't have a parallel printer port.[1] This was motivation enough for me to look into how a USB interface could be realized in a simple and low cost way.

If special tasks are required, such as controlling a VNA, a simple USB-to-parallel or USB-to-serial device won't suffice. A microcontroller will be necessary. There are many microcontrollers available on the market with built-in USB hardware, but generally these are more expensive than simple ones without USB support.

During an Internet search, I stumbled over a freeware firmware-only USB solution on the Objective Development Web site. This interface is realized with a low cost ATMEL AVR IC chip.[2] After a couple of weeks of learning how to program AVRs, I could easily modify the PowerSwitch reference example given on the Objective Development Web site, to control a DDS and do other things.

To me, there seem to be a multitude of possible Amateur Radio applications for this AVR-USB IC, but this solution doesn't seem to be widely known among hams. Therefore I have decided to describe it here in a simple way.

Since I know little about USB ports, I don't dare to write anything about USB theory. I would like to show, however, that even without knowing the intricacies of USB, such a device can easily be built and configured.

Hardware

The necessary hardware to control a DDS unit via USB is seen in Figure 1. The circuit consists of an ATMEL ATTiny2313

[1]Notes appear on page 13.

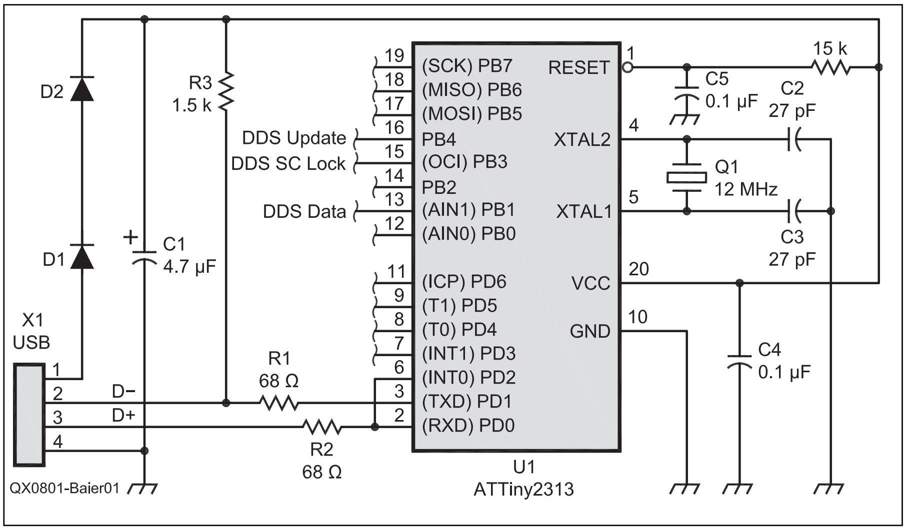


Figure 1 — The schematic diagram of the USB interface shows the computer USB connector and an ATMEL ATTiny2313 IC. Only a few additional components are required to complete the circuit.

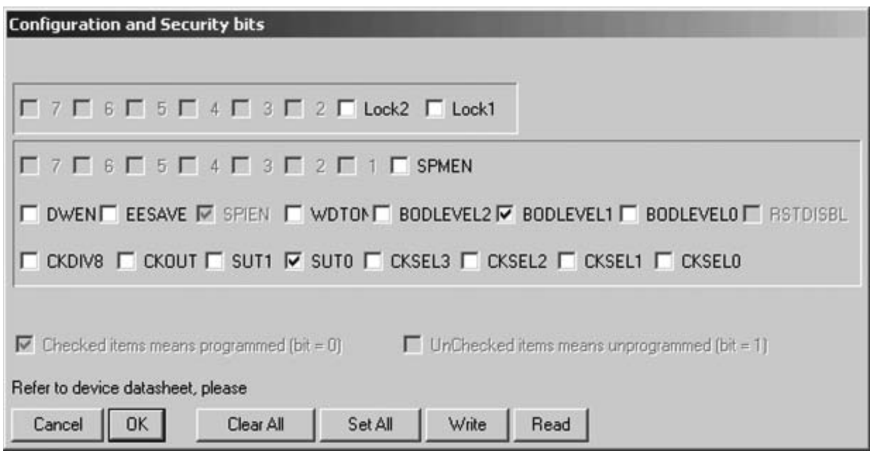


Figure 2 — This screen shot shows the *Pony Prog* menu with the correct "fuse" settings for the interface. Note the check marks in the BODLEVEL1 and SUT0 boxes.

Table 1
Data Structure of a USB Data Transfer

Data	*Size*	*Usage*	*Access in usbFunctionSetup*		
request type	1 Byte	driver usage	data[0]	equ.	rq->bmRequestType
request	1 Byte	user command	data[1]	equ.	rq->bRequest
value	2 Bytes	user data	data[2] (lo), data[3] (hi)	equ.	rq-> wValue
index	2 Bytes	user data	data[4] (lo), data[5] (hi)	equ.	rq->wIndex
length	2 Bytes	high byte ignored by driver	data[6] (lo), data[7] (hi)	equ.	rq-> wLength
optional data payload	length bytes	user data	see usbFunctionWrite		

MCU and a 12 MHz crystal. The USB driver requires an MCU clock frequency of 12 MHz. The latest driver version also allows a 16 MHz crystal clock, or on some devices, even 16.5 MHz derived from the internal RC oscillator. The MCU is powered from the USB hub. Diodes D1 and D2 are necessary to reduce the output voltage swing of the MCU at the USB D+ and D– lines to the USB specification limits. It is important to mention that the hardware inside the MCU can be configured by programming so-called "fuse bits," which define, for example, whether the internal RC oscillator or the external crystal is used. The AVR-USB device will not work with the factory fuse settings. The fuses need to be programmed as shown in Figure 2.

Figure 3 shows my USB interface with a DDS board and serial programming interface connected for *PonyProg*, described in the next section of the text.

DG8SAQ

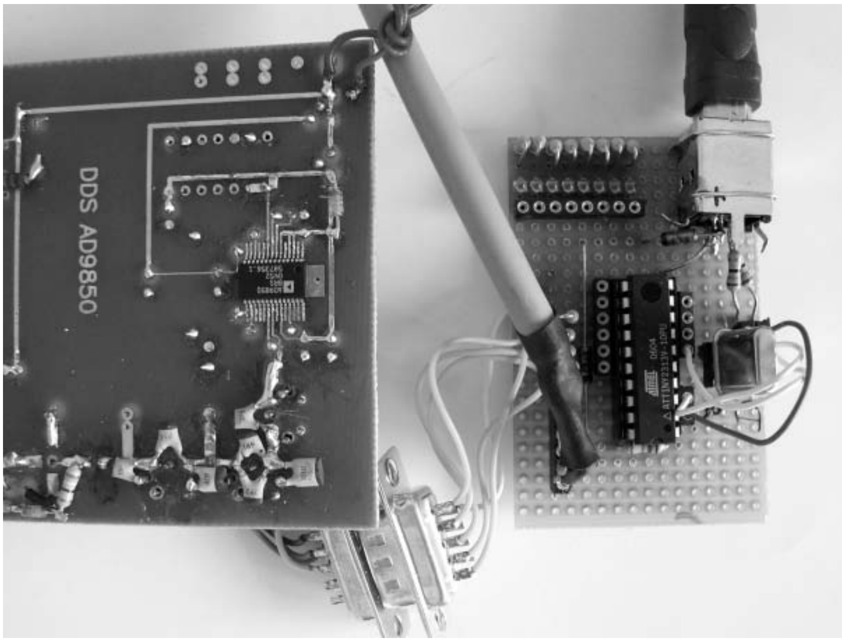


Figure 3 — This photo shows the USB interface connected to a DDS board and serial programming interface.

Development Tools

Before an MCU can do its job, software needs to be written and transferred into the device. For all these tasks there are excellent free software tools available. The software package *WinAVR* is a powerful yet free software development platform for ATMEL AVR MCUs based on the well-known GNU GCC compiler for C and C++.[3] I found that the older version, *WinAVR-20060421*, produces the smallest code size since the firmware was apparently optimized on that compiler version. *AVR Studio 4* from ATMEL is a free integrated development environment for AVRs, including Assembler and a code simulator. In its latest version, C-code can be programmed and debugged from within *AVR Studio* through the usage of *WinAVR* as a plug-in.[4] This makes the combination of *WinAVR* and *AVR Studio* very comfortable to use. Both are running on Microsoft's *Windows XP* operating system. All that is necessary to transfer the compiled code from the PC into the MCU flash memory is a cable connecting 4 lines of the parallel printer port with the MCU and the free software *PonyProg*, which can also be accessed as plug-in from within AVR Studio.[5, 6] *PonyProg* can also program the AVR fuse bits.

Firmware

The AVR USB firmware is partly written in Assembler for the time critical sections and in C for ease of interface to user code. The driver itself consumes about 1.4 kbytes of the 2 kbytes flash memory available in the selected MCU. This leaves about 600 bytes of space for user code. For larger user programs, it is helpful that all 8-bit AVR MCUs share the same CPU core, thus the driver can easily be adapted to any other AVR type, such as those with bigger flash memory.

Even though the USB specification and the firmware allow for several modes of data transfer, I have used only the simplest one, which is the so-called USB control transfer. It allows you to send or receive up to 254 bytes of data in one shot, and it offers high priority on the host side (this is the PC side, as described later).

Table 1 shows the structure of the transmitted data in a single USB control transfer. If only a few data bytes need to be transferred from the host to the USB device, the value and index words (2 bytes) can be used, and no data payload is needed, which means a length of 0.

Program Listing 1 shows my user interface in the main.c program to the firmware driver.

As can be seen from the listing, depending on the request value, data[1], different user tasks are performed. The simplest task is the first one (ECHO value), which simply sends the two value bytes back to the PC. This function is useful for diagnostics only. The return value of the discussed function specifies the number of bytes (stored in replyBuf) to be answered back to the host machine on the control transfer. Obviously, the USB device implemented here can do much more than just control a DDS. It can write or read any of the MCU port pins, widening its usage even further, to other switching and controlling applications without firmware changes.

Request 5 is a special request, with return value 0xff, which instructs the driver that a data payload is available and the user function usbFunctionWrite is to be called. That function is shown in Program Listing 2.

Here, the data payload is received and sent without modification to the DDS chip. If the value is nonzero, a DDS data update pulse is issued. No DDS type specific code is implemented in the firmware, but the firmware supports any DDS type. For different types, only the data payload has to be adapted on the PC side. Also, the data doesn't need to be sent in one chunk, but the DDS control word and data can be sent in separate control transfers.

Host Software

In order to enable a Windows PC to access the USB hardware, a device driver is necessary. Just like Objective Development's reference example, I use the freeware LibUSB driver.[7] It is also possible to use the HID driver integrated in Microsoft *Windows XP* if the USB device is configured as a Human Interface Device. Once the driver is installed and the device is plugged in, it can be accessed by means of any programming language on the host PC side. Since my personal preference is *Pascal*, I have written the host software in *Delphi*. In order to access the interface from within *Delphi*, a LibUSB to Pascal headerfile is necessary. This describes the driver interface in a *Pascal* way.[8] A considerable amount of *Delphi* coding is necessary to establish a connection to our USB device and to diagnose it. This part of the code is identical for any application, though, and can be reused. The only thing that needs to be adapted is the data transmitted in the USB control transfer call. Program Listing 3 shows this section of the *Pascal* host code.

It displays the function call, which transmits "len" bytes of data stored in "buffer" from the host PC to the USB device. The variables "request," "value" and "index" have the same meaning as discussed in Firmware section of this article. Varying these before the control transfer call will let the USB device do all kinds of desired jobs. The return value of the usb_control_msg function is the number of bytes answered back from the USB device in "buffer." The control transfer has high priority on the Windows host system, and requires about 5 ms in order to reach the USB device.

Summary

A simple, flexible and low cost USB hardware interface based on an Atmel AVR microcontroller and on Objective Development's free firmware has been introduced in a hopefully instructive way. I hope this will enable readers who are not MCU programming specialists to customize the solution to their own needs. The source codes, binaries and schematics can be downloaded from the author's Web site.[9] For those who prefer to download the files from the ARRL Web site, they are also available at **www.arrl.org/qexfiles**.[10]

Thanks to the following Lars Kvenild of Atmel Norway for excellent software support and to Christian Starkjohann of Objective Development for great forum support and for reviewing this article.

Notes

[1]Professor Dr Thomas C. Baier, DG8SAQ, "A Low Budget Vector Network Analyzer for AF to UHF," *QEX*, Mar/Apr 2007, ARRL, pp 46-54. See also **www.mydarc.de/DG8SAQ/VNWA/**

[2]**www.obdev.at/products/avrusb/**

[3]**http://sourceforge.net/projects/winavr/**

[4]AVR Studio 4.13, build 528 (Release) from **www.atmel.no/beta_ware/**

[5]**s-huehn.de/elektronik/avr-prog/avr-prog.htm.**

[6]**www.lancos.com/**

[7]**libusb-win32.sourceforge.net/**

[8]The LibUSB.pas header file was written and provided by Yvo Nelemans through private communication. It can be downloaded from Objective Development's Web page in the PowerSwitch reference example. See Note 2.

[9]**www.mydarc.de/dg8saq/AVR-USB/**

[10]The program files associated with this article are available for download from the ARRL Web site. Go to **www.arrl.org/qexfiles** and look for the file **1x08_Baier.zip**. Be aware that the author and manufacturer's Web sites may have updated listings available for download.

Professor Dr. Thomas Baier, MA, teaches physics, mathematics and electronics at the University of Applied Sciences in Ulm, Germany. Before his teaching assignment, he spent 10 years of work on research and development of surface acoustic wave filters for mobile communication with Siemens and EPCOS. He holds 10 patents.

Tom, DG8SAQ, has been a licensed radio amateur since 1980. He prefers the soldering iron to the microphone, though. His interests span from microwave technology to microcontrollers. Lately, he has started Windows *programming with* Delphi. *Tom spent one year in Oregon USA rock climbing and working on his master's degree.*

See Program Listings starting on next page.

Program Listing 1

The C function usbFunctionSetup in the main.c program listing is the user interface to the firmware driver.

```
USB_PUBLIC uchar usbFunctionSetup(uchar data[8])
{
usbRequest_t *rq = (void *)data;
static uchar    replyBuf[3];
   usbMsgPtr = replyBuf;
   if(rq->bRequest == 0){                          // ECHO value
      replyBuf[0] = data[2];              // rq->bRequest identical data[1]!
      replyBuf[1] = data[3];
      return 2;
   }
   if(rq->bRequest == 1){                          // set port directions
      DDRA = data[2];
      DDRB = data[3];
      DDRD = data[4] & (~USBMASK & ~(1 << 2));// protect USB interface
      return 0;
   }
   if(rq->bRequest == 2){                          // read ports
      replyBuf[0] = PINA;
      replyBuf[1] = PINB;
      replyBuf[2] = PIND;
      return 3;
   }
   if(rq->bRequest == 3){                          // read port states
      replyBuf[0] = PORTA;
      replyBuf[1] = PORTB;
      replyBuf[2] = PORTD;
      return 3;
   }
   if(rq->bRequest == 4){                          // set ports
      PORTA = data[2];
      PORTB = data[3];
      PORTD = data[4];
      return 0;
   }
   if(rq->bRequest == 5){                          // use usbFunctionWrite to transfer len bytes to DDS
      usb_val = data[2];                           // usb_val!=0 => DDS update pulse after data transfer
      return 0xff;
   }
   if(rq->bRequest == 6){
      PORTB = PORTB | DDS_UPDATE;                        // issue update pulse to DDS
      PORTB = PORTB & ~DDS_UPDATE;
      return 0;
   }
   replyBuf[0] = 0xff;                             // return value 0xff => command not supported
   return 1;
}
```

Program Listing 3

Issue a control transfer command in the out direction (USB_ENDPOINT_OUT) with the data payload stored in the buffer with length len bytes to be sent to the USB device. The variables "request," "value" and "index" have the same meaning as discussed in the Firmware section.

```
usb_control_msg(handle, USB_TYPE_VENDOR or USB_RECIP_DEVICE or USB_ENDPOINT_OUT, request,
                                   value, index, buffer, len, 5000);
```

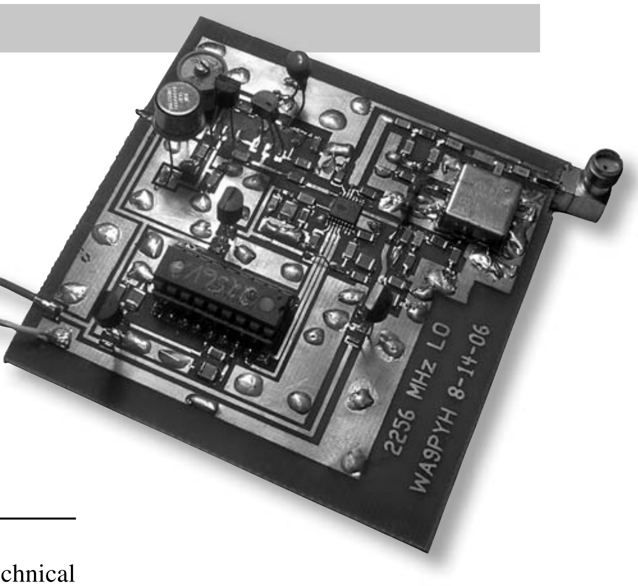
Program Listing 2

The usbFunctionWrite command in the main.c program listing sends the data payload directly to the DDS chip.

```
USB_PUBLIC uchar usbFunctionWrite(uchar *data, uchar len) //sends len bytes to DDS_SDA
{
uchar i;
uchar b;
uchar adr=0;
        while (len!=0){
                b=1;
                for (i=0;i<8;i++){
                        if (b & data[adr]){
                                PORTB = (PORTB | DDS1_SDA) & ~DDS_SCL;
                                PORTB = PORTB | DDS_SCL;
                        }
                        else{
                                PORTB = PORTB & (~DDS1_SDA & ~DDS_SCL);
                                PORTB = PORTB | DDS_SCL;
                        }
                        b=b<<1;
                }
        len--;
        adr++;
        }
if (usb_val){
  PORTB = PORTB | DDS_UPDATE;          // update DDS
  PORTB = PORTB & ~DDS_UPDATE;
  }
return 1;
}
```

Jim Kocsis, WA9PYH
53180 Flicker Ln, South Bend, IN 46637, wa9pyh@arrl.net

A 2256 MHz PLL Local Oscillator



Using a 10 MHz crystal reference oscillator, this PLL tunes a VCO to 2256 MHz.

This article describes the design of a PLL based local oscillator that operates at 2256 MHz. A PIC16F84 PIC (programmable interface controller) is used to load data into a National Semiconductor (NS) LMX2326 PLL (phase locked loop) IC. Using a 10 MHz crystal reference oscillator, the PLL tunes a VCO to 2256 MHz.

I'll describe the operation of the PLL IC, the software that the PIC loads into the PLL, and most importantly, the operation of software on the National Semiconductor Web site that I used to design the very critical loop filter. I will also discuss building the LO, showing the construction sequence and some areas that can cause problems. A spectrum analyzer display of this LO and an HP generator (as a standard) shows the relative purity of this LO.

The cost of all parts if purchased new is approximately $100. Some savings can be realized if you have a well-stocked junkbox, or if you obtain some of the parts at hamfests. Construction time is approximately 10 hours. No special test equipment is needed, but if you have access to a spectrum analyzer, it can be used to check for proper operation, signal purity and the level of the output. A simple frequency counter is used to adjust a crystal oscillator at 10 MHz and a standard DMM is needed to check the voltage regulators and VCO tuning voltage. Unlike other high frequency oscillators that use multipliers, this circuit needs no tuning or peaking of tuned circuits — there are none!

This LO can be combined with a mixer, LNA and filter to make a down-converter for the 2.4 GHz amateur band with 2 meter output. I built a down-converter using this LO and its sensitivity is equal to that of a commercially available unit with a 1 dB noise figure. When developing the down-converter, the mixer and LNA were easy to design compared to the LO. A ham friend warned me that the LO would be, by far, the most difficult part of a down-converter. Boy, was he ever right!

A Bit of History

I enjoy trying new (for me) technical projects. I wrote an article in the Mar/Apr 2000 issue of *QEX* describing my design of a down-converter for 1691 MHz weather satellite reception.[1] I wanted to apply what I had learned from that design to a 2.4 GHz down-converter. I thought of modifying that design for operation at 2256 MHz, but while the LO I described in that article worked, I was not happy with the noisy output. I had not understood loop filter design and some layout issues as well as I do now. It was basically cut-and-try for the loop filter resistor and capacitor values, until it was "good enough" to get pictures from the weather satellites. The PLL IC used in that design, a Motorola MC12179D, came with a Microsoft *Excel* file for loop filter design. It was difficult to understand and use. That was 1997. Moving up to 2005, I discovered that National Semiconductor provides a very complete and easy-to-use Web site — called "WebBench/*EasyPLL*." It is a true masterpiece and it's free! (It is not software that can be downloaded — you must run it on their Web site.) Considering the function it is designed to perform (loop filter design), it is about as simple as you will find. I knew from my reading and experience that the design of a PLL loop filter would either make or break the performance of my new down-converter. If the loop filter design is not proper, you won't hear weak signals and those that you do hear will have noise on them. I found that National Semiconductor makes a PLL IC that fits my needs exactly — the LMX2326. Besides, Motorola discontinued production of the MC12179D PLL a few years ago.

About this same time I saw an article describing a down-converter that used the LMX2326, and operated around 1600 MHz.[2] It uses a PIC to program the PLL. This was proof that I was on the right track when I chose the LMX2326. I didn't have any experience with a PIC, so I learned the basics of programming, changed the code in that article and programmed my own PIC.

Before using *EasyPLL*, I had tried manually computing the loop filter values using methods described in magazine articles and application notes. With each manual method, however, I always hit a "brick wall." Progress stopped when they didn't provide all the information needed. They assumed I knew what default values to use, what units to use, and so on, so I was never able to finish the design using the manual methods. I quickly switched to *EasyPLL* and began seeing real progress in filter design. If you've entered valid numbers, *EasyPLL* shows a "Congratulations on designing your loop filter!" message. That message is *very* encouraging when you're just getting started!

Please note that the LMX2326 PLL IC is *very* small. See Figure 1. The IC body measures 0.2 × 0.17 inch with eight pins on each of two sides. The pins are 0.016 inch wide and spaced 0.026 inch center-to-center. If you're not used to working with parts this small you may find it a challenge to build this project. You'll need steady hands and good vision (probably with the help of a magnifier/glasses) to solder the PLL IC onto the circuit board. You might also want to ask for help from someone with more experience working with surface mount ICs.

Figure 1 — The LMX2326 PLL IC is *very* small, as this photo shows.

[1]Notes appear on page 25.

Theory

A block diagram of the circuit is shown in Figure 2, and Figure 3 shows the complete schematic diagram. The reference oscillator (exactly 10 MHz) is divided by 5, and the VCO output (approximately 2256 MHz) is divided by 1128 inside the PLL. Both divided signals are combined in the phase comparator at 2 MHz. The charge pump in the PLL produces pulses that are filtered by the loop filter to drive the VCO to exactly 2256 MHz. For more information on PLLs please review the PLL section of recent versions of *The ARRL Handbook*, the Mini-Circuits Web site and a downloadable book from National Semiconductor called *PLL Performance, Simulation and Design*, also known as "Dean's Book."[3, 4, 5] They are all excellent, and are required reading if you want to understand PLL operation fully. Other excellent articles are listed in Notes 6 through 11.

When using *EasyPLL*, there are HELP pages available for almost every parameter. I suggest that you print out all the HELP pages in *EasyPLL*, and put them into a 3-hole punched 8½ × 11 inch notebook for easy reference. I found this to be much easier than opening each file as needed and switching back and forth between them. I added my own notes to each page to help me retain what each page discussed. Also, if a paragraph didn't apply to this project, I crossed it out.

When designing a loop filter for fast switching between frequencies (such as in cell phones), there are trade-offs between signal purity and switching time. Since switching time is not a concern here (the LO operates at a fixed frequency) the design can be optimized for purity by setting Optimize to "Max High Order Cap" in *EasyPLL*. In the example in the "Setup and Using *EasyPLL*" sidebar, the value for Max Lock Time is set to 300,000 microseconds.

When designing a receiver or downconverter, you must pay very close attention to noise in the LO, since any noise there is combined with the desired signal. Sources of noise are: poor loop design (wrong values), long Vtune and power supply printed circuit traces that act as antennas, VCO not loaded with 50 Ω, poor power supply bypassing/filtering and poor grounding of the VCO case and ground leads.

I found a wide-range VCO to be trouble. Consider that when using a VCO with a range of 150 MHz / V, 100 µV of noise can modulate the carrier 15 kHz! My first design used such a VCO, and I was unable to attain a clean signal. I changed to a narrow range VCO (15 MHz / V) and was then finally able to attain a clean signal. I read in several of the references that wide-range VCOs have more noise on their output as compared to narrow range units.

In an early version of my LO, which used a wide-range VCO, I also had long circuit traces between the loop filter and the Vtune input to the VCO (a poor design). I'm sure this contributed to my difficulty with attaining a clean output.

Impedance matching of the VCO output is very important. On the Z-Communications Web site I found that for lowest noise the VCO should "look into" as near a pure 50 Ω load as possible.[12] The three 18 Ω resistors at the VCO RF output and the 50 Ω loads they in turn "look into," result in a 50 Ω load being presented to the VCO. See the Z-Communications Application Note, AN-102 "Proper Output Loading of VCOs." (See Note 12.)

Signal levels at each pin are important. If the signal level is too low at a given point, the circuit won't work. If it's too high it may not work or you may damage an IC. The circuit uses an attenuator to adjust the level of the 2256 MHz feedback to well within the limit shown on the data sheet. The attenuator network (R9-R11) has a 50 Ω input and output impedance. Signal levels are shown in the block diagram in Figure 2.

The reference oscillator (10 MHz in this design) must be very clean, with very low phase noise. Any noise in the reference oscillator will show up in the LO output at a level 20 Log N dB higher. (N is the divide-by ratio in the PLL — in this case 1128.) Any drift in the reference is also multiplied by N, but since most crystal oscillators are very stable this isn't a big concern. The reference oscillator in this LO is a crystal oscillator from a *QST* Technical Correspondence Letter by John Clark, K2AOP.[13] I was looking for a good, clean circuit when I saw this article. What caught my attention was that the phase noise was so low that the ARRL laboratory couldn't measure it — that was good enough for me so I incorporated it into my LO!

A VCO running "open loop" has a very broad noise spectrum that is too high to allow it to be used as-is for a local oscillator. VCOs must be locked to a very clean reference to

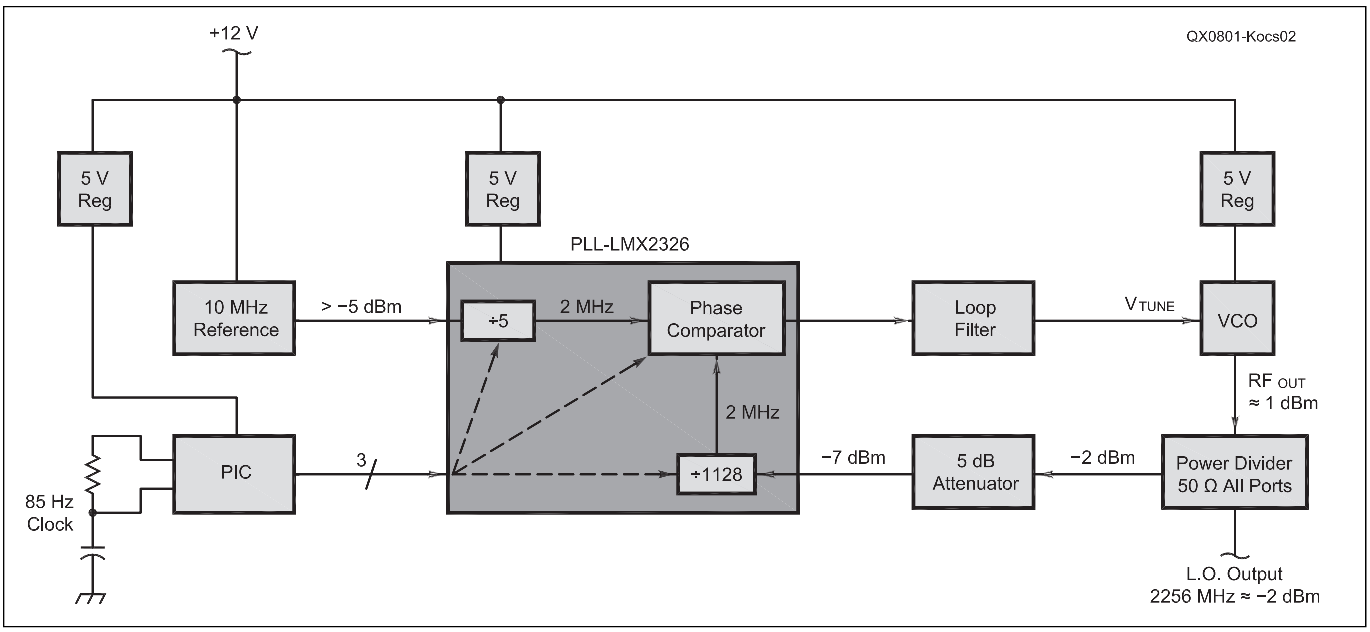


Figure 2 — This block diagram shows the operation of the local oscillator circuit.

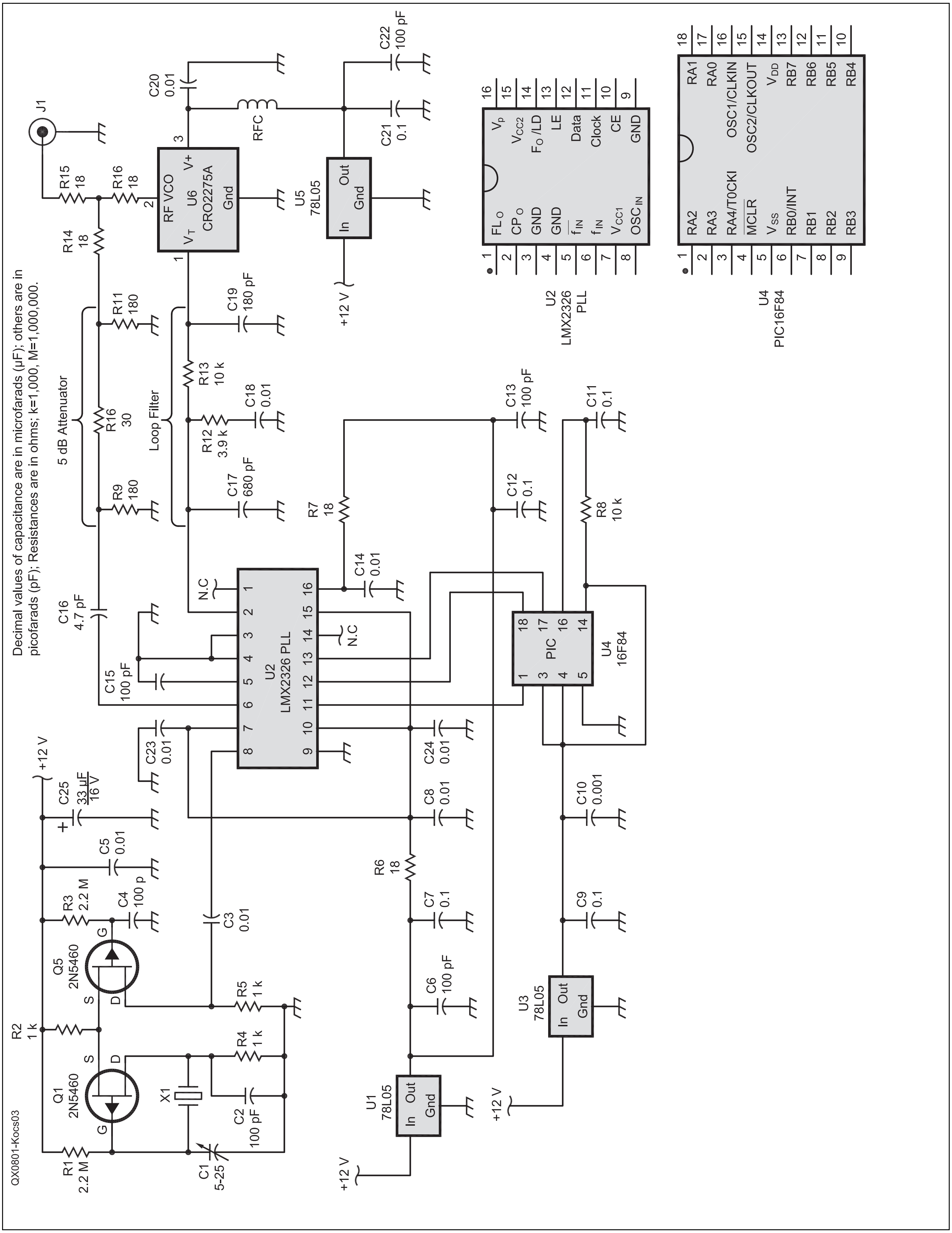


Figure 3 — Here is the schematic diagram of the local oscillator circuit.

produce a clean output. Also, they drift a lot unless "locked" to a stable source. The loop filter and PLL IC actually reduce the noise in the sidebands of the VCO output — up to the bandwidth of the loop filter. Beyond the loop filter bandwidth, the noise from the oscillator is the same as if the VCO was running open loop. This is pointed out very clearly in *The ARRL Handbook* PLL discussion (See Note 3) and several of the other references. It is not intuitively obvious that this can be true — I doubted it until I saw the noise spectrum of my LO reduced just as the *Handbook* said!

A PLL/VCO oscillator will have a noise versus frequency spectrum that has a characteristic shape with a flattening of the noise outside the loop filter bandwidth. The flattening of the noise occurs at a difference from the carrier center frequency equal to the filter bandwidth. In an earlier version with a loop filter bandwidth of 3 kHz I could see a slight increase in noise starting at exactly 3 kHz away from the carrier. When using the LO with my down-converter I could hear an increase in noise exactly 3 kHz away from zero beat when tuning across a carrier — exactly as predicted in *EasyPLL*!

For the lowest noise, you should use the lowest N number (largest step size) possible; it's easier to filter the reference "spurs" and there is much less phase noise due to the lower N number.

When I first began using *EasyPLL*, I specified using a comparison frequency of 100 kHz. The 100 kHz channel spacing came from the original article in the *RIG Journal*. That application needed to set the LO to several frequencies that were spaced multiples of 100 kHz. This low of a frequency wasn't necessary in my application but I didn't think to change it to 2 MHz — I could have specified a higher frequency that resulted in a lower divisor for both the N and R dividers. *EasyPLL* is so smart that it suggested increasing the comparison frequency from 100 kHz to 2 MHz so that there would be less noise in the carrier — that's pretty smart software! (Remember that any noise in the reference is multiplied by 20 log N, where N is the divide ratio.) In switching from 100 kHz to 2 MHz, the noise multiplication dropped by a very significant 46 dB. (For a comparison frequency of 100 kHz, the noise multiplication is 20 Log 225600 = 107 dB but for a comparison frequency of 2 MHz the noise multiplication is only 20 Log 1128 = 61 dB.)

You should download the datasheet for the LMX2326 PLL IC.[14] Review all the information in it along with the other references.

Physical Layout

I took extra care when designing the power distribution portion of the circuit. The goal was to prevent coupling signals from one stage to another over the power lines. To accomplish this, I used one regulator each for the PLL, VCO and PIC. There are lots of bypass capacitors on the power lines to provide each IC with a very good ac ground on the power leads. Figure 4 shows the circuit board pattern that I created. An etching-pattern version of this image is available on the *QEX* Web site.[15]

The power supply and Vtune lines are kept physically separated from the RF output — you don't want any RF getting on these lines. The RF output has a low impedance, and the Vtune input has a very high input impedance (it is typically a varicap) — as a result the Vtune is by far the most susceptible to stray input signals. The pin-outs of the PLL and the VCO are arranged such that the power supply and Vtune lines must cross each other at some point. I used an RF choke to feed dc to the VCO while keeping the Vtune trace short.

Note that the path from the PLL to the VCO consists of very little circuit board trace and mostly loop filter components — the path should be as short as possible so there is no "antenna" to pick up stray signals.

Figure 3 — Parts List

Capacitors	*Value / Description*	*DigiKey p/n, other mfgr*	*Price*
C1	5-25 pf miniature variable	see text	
C2, C4, C6, C13, C15, C22	100 pF chip size 1206	478-1480-1-ND	\$4.18/10
C10	0.001 µF chip size 1206	478-1530-1-ND	\$3.03/10
C3, C5, C8, C14, C18, C20,C23, C24	0.01 µF chip size 1206	478-1542-1-ND	\$3.03/10
C7, C9, C11, C12, C21	0.1 µF chip size 1206	478-1556-1-ND	\$2.20/10
C16	4.7 pF chip size 1206	478-1464-1-ND	\$3.52/10
C17	680 pF chip size 1206	478-1490-1-ND	\$5.13/10
C19	180 pF chip size 1206	478-1483-1-ND	\$5.13/10
C25	33 µF 16 V epoxy dipped tantalum	478-1894-ND	\$1.75
Resistors			
R1, R3	2.2 MΩ chip size 1206	RHM-2.2M-ECT-ND	\$.91/10
R2, R4, R5	1 kΩ chip size 1206	RHM-1K-ECT-ND	\$.91/10
R6, R7, R14, R15, R16	18 Ωm chip size 1206	RHM-18-ECT-ND	\$.91/10
R8	10 kΩ chip size 1206	RHM-10K-ECT-ND	\$.91/10
R9, R11	180 Ω chip size 1206	RHM-180-ECT-ND	\$.91/10
R10	30 Ω chip size 1206	RHM-30-ECT-ND	\$.91/10
R12	3.9 kΩ size 1206	RHM-3.9K-ECT-ND	\$.91/10
R13	10 kΩ size 1206	RHM-10K-ECT-ND	\$.91/10
Transistors			
Q1, Q2	2N5460 JFET, P chan, TO-92	2N5460OS-ND	\$0.46
ICs			
U1, U3, U5	78L05 5 V reg TO-92	LM78L05ACZNS-ND	\$0.73
U2	LMX2326 PLL IC	LMX2326TM-ND	\$3.20
U4	PIC16F84 PIC, 4 MHz, 18 pin DIP	PIC16F84-04/P-ND	\$6.13
U6	2250-2300 MHz VCO	CRO2275A see text	\$29.95 In quantities of 3
Misc			
RFC	15 µH. RF choke	M1376-ND	\$0.30
X1	10.000 MHz fundamental	ICM 35UAAFF21OG	\$38.35
	18 pin DIP IC socket w/flat solder tabs		see text
J1	SMA female connector	ACX-1230-ND	\$4.37

QX0801-Kocs04

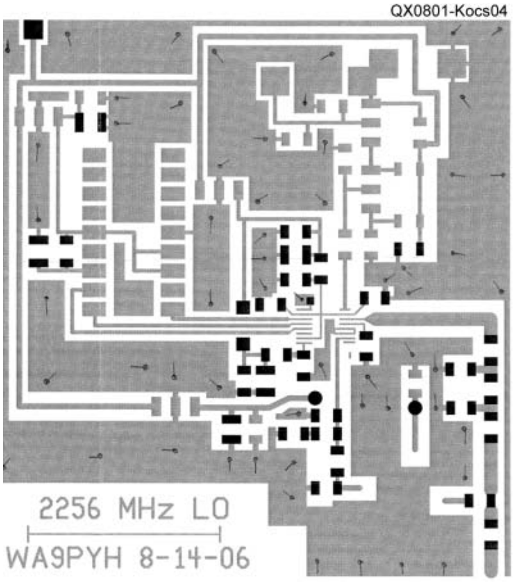


Figure 4 — This circuit board pattern shows the location of the wires used to connect the circuit ground traces with the ground foil on the bottom of the circuit board. A copy of the circuit board etching pattern is available on the *QEX* Web site. See Note 15.

WA9PYH

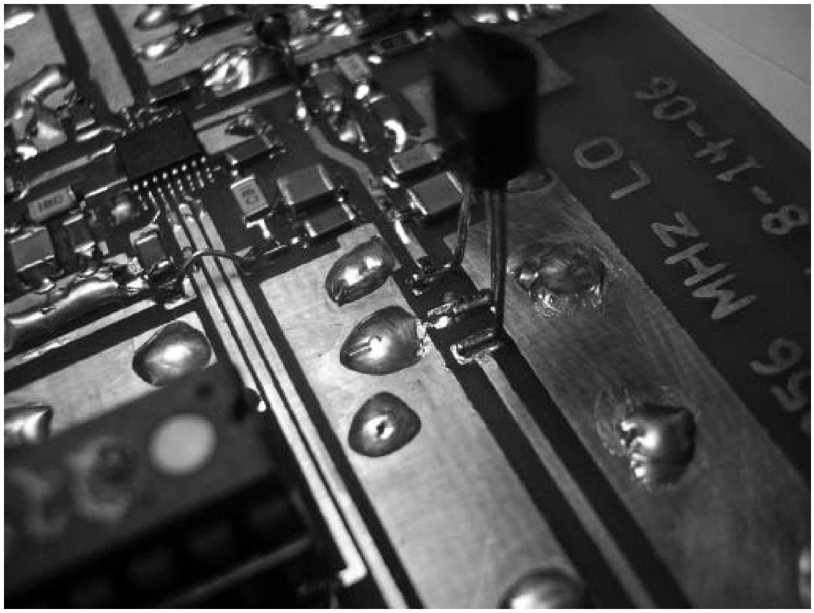


Figure 5 — This photo shows how the leads of a regulator are bent to solder the "feet" to the circuit board traces. The solder "blobs" are covering ground wire connections to the bottom of the board.

Construction

The lead photo shows the completed board with all parts installed. Take your time installing the components. They are very small, and it is very easy to ruin the PC board. Use bright lights and a magnifier so that you can see what you're doing clearly. Move parts into place slowly and skip coffee for a day. You'll need good vision and steady hands when soldering the PLL IC and VCO.

Plated through holes for grounds would have been nice but since there is no simple way to make them at home, I use another technique. At the 59 locations where a ground is needed, I drilled a small hole (0.020 inch diameter), inserted a ¼ inch long piece of no. 26 wire, and soldered both sides to connect that point to ground. See Figure 4 for the locations and the direction each wire must face. The wires must be bent going the directions shown so they don't interfere with component placement at some locations. (Drilling those holes took about an hour using a small drill press.) The line on the board that is located between my call sign and the date is exactly 1 inch long, and is used to check that the artwork scaling is exactly 1:1.

All components are surface mount except for the voltage regulator ICs, RF choke, PIC socket, crystal, trimmer capacitor and the FETs. You will need to bend their leads on those components to fit the pads. Figure 5 is a close-up view of one of the regulators mounted to the board.

A socket is shown for the PIC. There's no problem with lead length here, plus you can reprogram the device if you want to change frequency (within range of the VCO, of course).

The soldering iron should have a grounded tip with a 3 prong plug so there is no chance of a static discharge into any of the parts. The board should be connected to the same ground during assembly. You should use a grounded wrist strap so your body doesn't inject static electricity into the delicate parts. I have found a Weller WP-25 to be a suitable soldering iron, and I use a Highland 2272 grounded wrist strap.[16] These are available from Digi-Key. Suitable soldering flux and desoldering braid are also listed in Note 15.

If you've never worked with surface mount parts, practice on a scrap board until you're comfortable positioning and soldering them. I use the following technique: Use a small sponge soaked in water to wipe the soldering iron tip frequently, place a small amount of liquid soldering flux where the part attaches to the board, place the component on the board near where it goes, slide it into place using a toothpick, hold it in place with the toothpick, put a very small amount of solder on the tip, solder one side, let it cool, then solder the other side. Use just enough solder to secure the component. Examine a commercially made board containing surface mount (SMT) parts to see the correct amount of solder.

A 0.060 to 0.070 inch flat blade tip will work fine for all parts except the ground plane connections. A wider tip (0.125 inch wide blade) is a good size to use for soldering the ground feed-through connections on the ground side of the board. The ground side is a big heat sink that can cool a small tip quickly resulting in a poorly soldered connection, so use a large tip.

The board must be clean and free of all solder flux when you're done, especially in the loop filter and RF areas. Any remaining flux may change the properties of the loop filter and short some of the RF signal to ground. Spray flux remover can be used but its vapors can be rather toxic. I use rubbing alcohol and a small brush to remove most of the flux, then do a final cleaning with the spray flux remover.

Sequence of Parts Installation

1) Put a small amount of soldering flux where each component or grounding wire is soldered to the board.

2) Install the 59 grounding wires through all holes. Rotate each jumper into approximately the correct position as shown in Figure 4. Place a small amount of flux on the wire and ground plane side of the board, solder the ground plane side first, then rotate the circuit side of the wire into the required position, place a small amount of flux on the circuit side and solder that side. Cut both ends of the wire with an X-Acto knife so the wire doesn't extend to where it can short to another trace or catch on any surface.

3) Install the jumper wire that crosses over the three traces that go between the PLL and PIC. See Figure 6.

4) Install the LMX2326 PLL IC. Pin 1 is marked by a faint small dot on top of the IC as shown in the Data Sheet — make sure that you have it positioned properly! There is a small unconnected "dash" on the circuit

board located to the right and slightly above where the PLL IC is located. That "dash" should line up with the pins on that side of the IC. This will ensure that you have the PLL IC located properly. Position the IC pins *exactly* over the 16 pads. Hold it in place with a toothpick and solder one of the corner pins. Do this by touching the iron tip to the trace, and *not* the IC pin. Allow the solder to wick up to the IC pin from the trace. Solder the pin at the opposite corner in the same manner. If the IC has moved and is not positioned properly, unsolder it and try again. Minimize the time heat is applied to the IC. If there's any chance you have overheated the IC, replace it, because they are very inexpensive. If all has gone well so far, solder the remaining pins by allowing the solder to wick up to the remaining IC leads. If solder bridges occur between the pins, apply some liquid flux to the spot and use desoldering braid to remove the extra solder. I did this successfully several times.

5) Install all resistors and capacitors except those near the VCO (R9-16, C17-22 and the RF choke). (Note that there are no pads for C25 but its location is shown in Figure 6. It was added after I noticed that I had failed to include a large valued filter capacitor on the +12 V input to the crystal reference oscillator. Adding it reduced the noise spectrum outside the loop bandwidth by 6 dB.)

6) Install the VCO. The VCO is installed *after* the PLL so that if you mess up soldering the PLL and cannot recover your mistakes, you only lose the time you spent, the cost of the board and a few dollars in parts. The VCO is the second most expensive ($30) and most difficult-to-remove part. Make sure the solder wicks up into the cavities for the power, V_{Tune}, RF output and ground. See Figure 7. Also see Note 12, AN-107 "How to solder Z-Comm VCOs" and AN-100/1 "Mounting and Grounding of VCOs."

7) Bend the PIC IC socket pins outward so they lie flat on the circuit board. Trim them if they extend beyond the solder pads where the socket is mounted. Solder all pins to the board.

8) Install each regulator by first bending ⅛ inch of all three leads forward, then solder only the input and ground leads (the right and center leads when looking at the flat side). Their orientation is shown in Figure 6. Power up the board and verify that 5 V appears at each unconnected output lead. (I installed the regulators backwards and fried the PLL and a wide-range VCO — that was a very expensive and time-consuming mistake.) Once you've verified that 5 V dc appears at the three output leads, solder those leads in place.

9) Install the two FETs. Their orientation is also shown in Figure 6. Note that they face opposite directions.

10) Install the remaining components near the VCO (see Step 4, above).

11) Install the crystal and trimmer capacitor.

12) Install the output connector. See Figure 8.

13) Install the PIC into its socket.

Power-Up and Check-Out

1) Apply 12 V dc power. The circuit as shown should draw approximately 43 mA.

2) At power-up, the VCO will drift for 5 or 6 seconds, then snap to 2256 MHz. The tuning voltage at the VCO V_{Tune} pin should read very close to 1.46 V dc when measured with a high impedance DMM.

3) Attach a frequency counter to the Q2 drain. Adjust C1 until it reads exactly 10.000000 MHz.

4) If at all possible, view the output on a spectrum analyzer to verify output purity. If you have access to an RF power meter or a spectrum analyzer that can indicate absolute power in dBm, verify that the output level is approximately –2 dBm.

5) The individual data bits can be seen on pin 18 of the PIC, the clock for loading each data bit on pin 1 and the Load Enable (LE) on pin 17. Note that there will be 21 or 20 pulses on pins 17 and 18, but only 1 pulse on the LE pin for each of the N, R and F registers. The reason for this is that the LE pin loads 20 or 21 bits into the selected register and does this all at once.

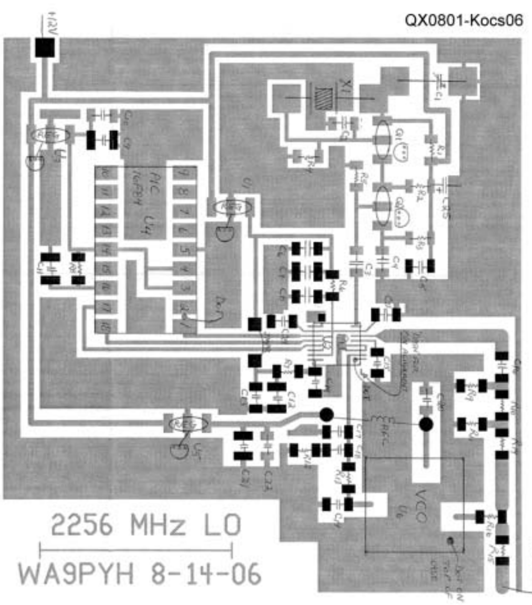


Figure 6 — This parts-placement diagram shows the location and orientation of the various components. All of the components use surface mount.

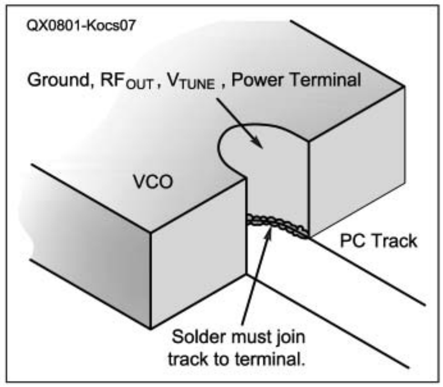


Figure 7 — This drawing illustrates the mounting technique used for the three connections to the CRO2275 Z-Communications VCO.

About the Parts

The crystal must be a fundamental type. If you use a third overtone type of crystal, the circuit will oscillate at ⅓ the frequency marked on the crystal case. This will make finding a crystal a bit difficult. For my frequency, a third overtone 30 MHz crystal is needed, and one was not available at hamfests. The part specified from ICM is a special low phase noise fundamental type crystal. It is a bit expensive ($47) but will produce a very clean 10 MHz signal when used in the FET circuit. I tried some really cheap crystals (from a PC plug-in board) in the oscillator circuit and at their natural frequency they were very noisy when viewed on a spectrum analyzer.

I bought three VCOs from Z-Comm even though their "policy" is to sell a minimum of five. I talked to an applications engineer (Ajay) to plead my case to buy only three (one for the board as shown, one for the down-converter in final configuration and one spare). He was very helpful and arranged for me to buy the three I needed. If you want only one for yourself, go in with some fel-

WA9PYH


Figure 8 — The SMA output connector solders to the edge of the circuit board.

Computing the R and N Register Values for a 10 MHz Reference And 2256 MHz Output

Sections 1.7.5 and 1.7.6 of the LMX2326 datasheet have more detailed information about these calculations.

I selected a PLL comparison frequency of 2 MHz for my LO, so the R value is 5 and the N value is 1128. (Note 10 MHz / 5 = 2 MHz and 2256 MHz / 1128 = 2 MHz.)

Computing the bit pattern for R

The R register is loaded into the LMX2326 by sending a 21 bit serial data stream. The last two bits, C1 and C2, are called Control Bits. They specify the register being sent the data bits. To load data into the R register they are both set to 0. The PIC software picks off individual bits from three 8-bit bytes (the bytes are the values 0, 12 and 128 in the program listing). The first and second 8-bit bytes are sent in their entirety. Only the first 5 bits of the third byte are sent. Figure S1 shows the significance of each bit and the order in which they are sent.

The actual R value is located in bits R1 through R14 in binary form. The weight of each bit is shown for the entire 14-bit R number. The weight of each bit in each 8-bit byte is also shown. The value of 5 (10 MHz reference frequency divided by 5 = 2 MHz comparison frequency) is set by turning on bits R1 and R3.

To set R to a different value, compute its value in binary and turn on (set to 1) those same bits in the three bytes. You will likely use a very low number so that only the first or second byte will be different from mine. For example, if your reference is 20 MHz and you want a 2 MHz comparison frequency, set R to 10, by turning on the following bits: R2 and R4. All other bits will be off (set to 0).

Computing the bit pattern for N

The N register is loaded into the LMX2326 in the same way as the R register, but requires that you perform a series of calculations first. The N register consists of 21 bits, the last two bits are the Control Bits (as in R) but for the N register they are set as follows: C2 = 0 and C1 = 1.

In my local oscillator, N is set to 1128. This value was determined by the LO output frequency and the comparison frequency: 2256 MHz / 2 MHz = 1128. There are two counters/dividers in the "N" section of the LMX2326. They are called the A "swallow" counter and the B counter. Together they divide by 1128. See the calculation examples below and the example provided in section 1.7.6 of the LMX2326 datasheet. In all three examples P = 32 — it is a fixed value in the LMX2326.

Example 1 — My Circuit

Step 1: Assume N = 1128, P = 32, Frequency out is 2256 MHz, the comparison frequency is 2 MHz.

Step 2: Set B = N / P = N / 32 = 1128 / 32 = 35.25. Drop the decimal portion and set B = 35.

Step 3: Set A = N – (B × P) = N – (B × 32) = 1128 – (35 × 32) = 1128 – 1120 = 8. So, A = 8.

In summary, the A "swallow" counter is set to 8 and the B counter is set to 35.

Note that the A counter is set with bits N1 through N5 and the B counter is set with bits N6 through N18, and they are spread across all three 8-bit bytes.

The bit pattern for N = 1128 is shown in Figure S2.

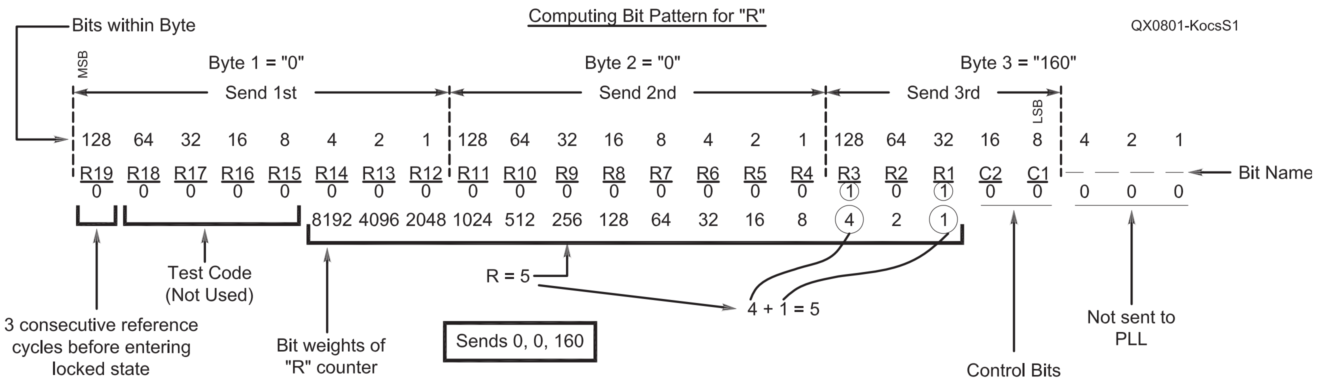


Figure S1 — Here is a chart showing the method used to determine the bit pattern to be loaded into the LMX2326 R register.

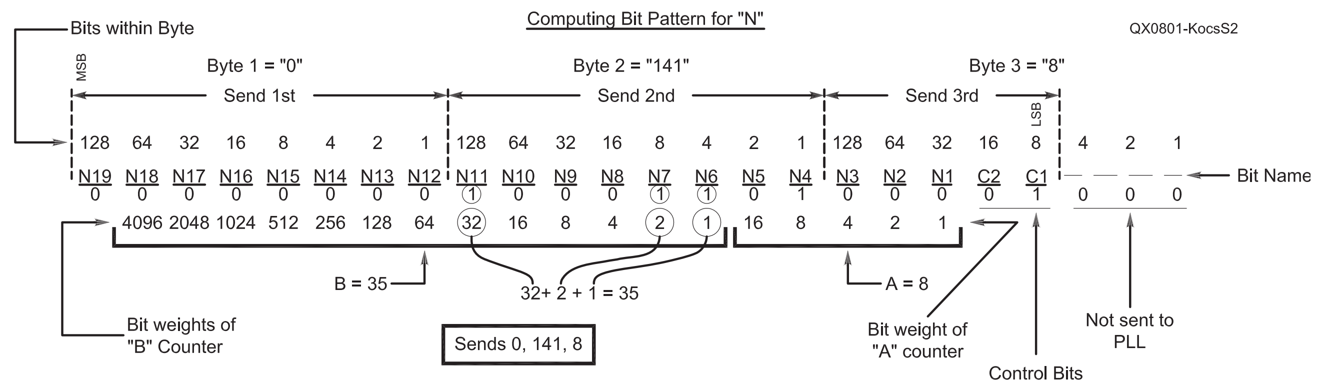


Figure S2 — This chart shows how to compute the bit pattern to be loaded into the LMX2326 N register.

WA9PYH

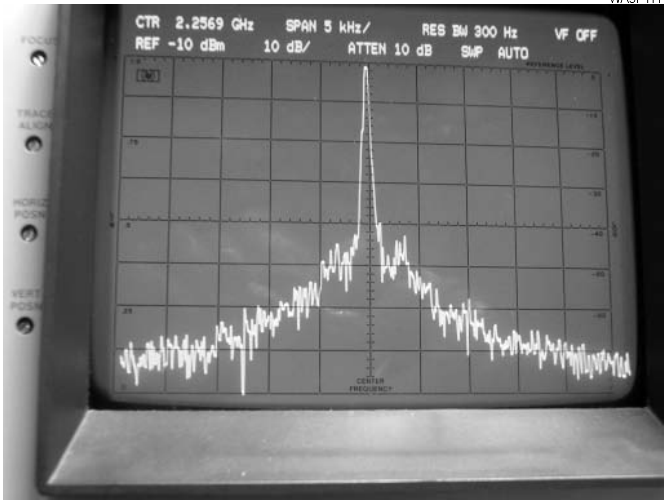


Figure 9 — This photo of the spectrum analyzer display shows the local oscillator output.

WA9PYH

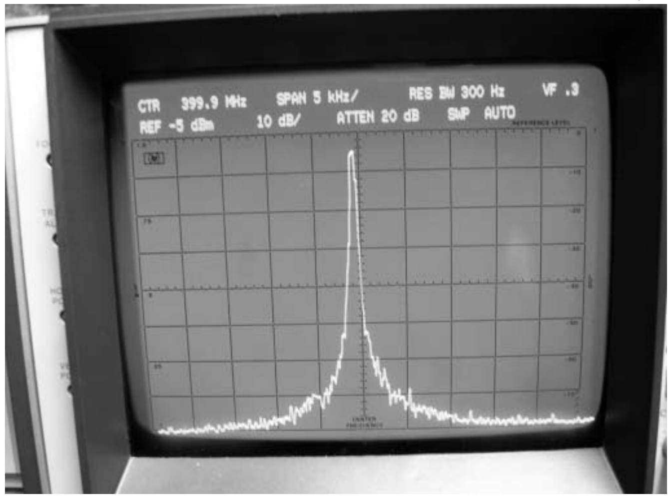


Figure 10 — For comparison with the LO spectrum display shown in Figure 9, this display shows the output from an HP 8640 signal generator set to 400 MHz.

low hams and buy a small quantity. The datasheet for this VCO is not available on their Web site, but it is available on the *QEX* Web site. (See Note 15.) The VCO I used has been replaced by a new lead free part. It is a CRO2275LF. Z-Comm is following the new industry Reduction of Hazardous Substance (RoHS) standard of having no lead in components. Electrically, the new part is the same as the "leaded" version and will solder into place using 60/40 or 63/37 solder.

I'm not in the business of selling programmed PICs but if readers have no other way to obtain a programmed PIC, I can provide them already programmed for this frequency only. Please send $8 for each PIC — US addresses only.

The PIC IC socket is one that I found at a hamfest. I couldn't determine if those available from Digi-Key had pins that are flat and can be bent out toward the sides of the socket body, so look for these in your junk box and at hamfests. Only one 18 pin socket is needed, and should cost less than $0.50.

The pads for the 0.1 µF chip capacitors are wider than the pads for the other capacitors. This is because the 0.1 µF capacitors I had on hand were wider than those shown in the parts list. The type shown in the parts list will fit the board, and actually have room to spare.

C1, the 25 pF variable capacitor used to tune the reference oscillator, is one that I found in my junk box. It measures ⅜ inch in diameter, with tabs that are bent to fit the pads on the circuit board. I was not able to find any suitable units at Digi-Key. Hopefully you have one in your junk box or can find one at a hamfest.

J1, the output connector, also came from a hamfest and costs around $0.50.

Software

The PIC program file (in assembly language — human readable) and the HEX file (loaded into the PIC) are on the *QEX* Web site. (See Note 16.) Look through the assembly program to find the loops that strip off and send one data bit at a time, the clock and load enable (LE) signals to the PLL IC. It is very well commented, and after spending a little time examining the program flow you will see how the bit patterns are generated. Note that the PIC software uses the "Counter Reset Method" described in paragraph 1.7.4 of the LMX2326 data sheet. The book I used to learn PIC programming and to understand this program is *Easy PIC'n* by David Benson.[17] [Note that this is not Dave Benson, K1SWL, of Small Wonder Labs. — *Ed.*]

The Sidebars show how to calculate the R and N register values and set up an account to use *EasyPLL* on the National Semiconductor Web site.

Results

The bandwidth I entered into *EasyPLL* for the version shown in this article was 10 kHz. Unfortunately, I don't see that same dip in the noise on the spectrum analyzer. See Figure 9. Also the noise isn't as low as predicted by *EasyPLL*. I'm attributing some of the differences to not having plated through holes in my design. Perhaps the filter needs

WA9PYH

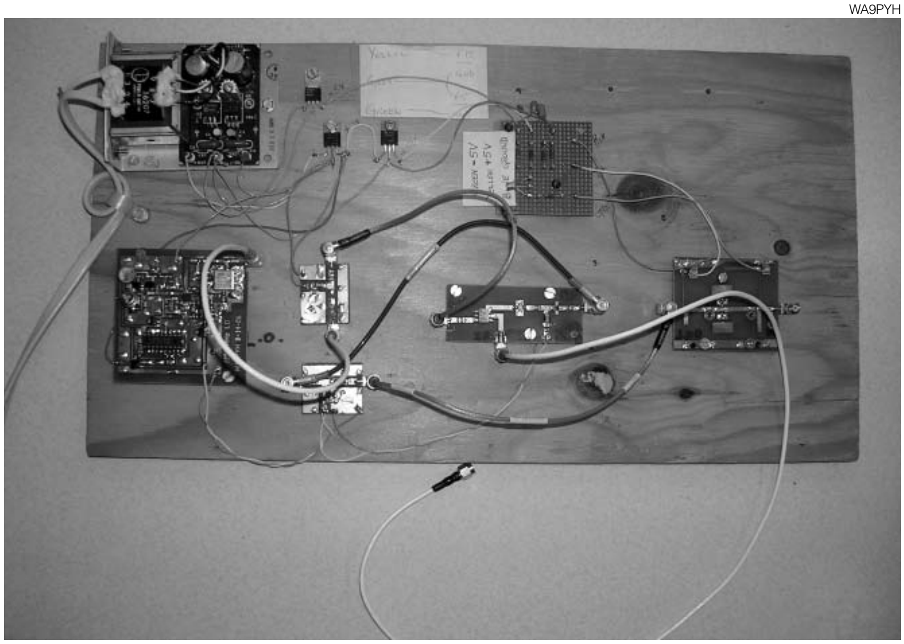


Figure 11 — This 2400 MHz downconverter for satellite operation uses the 2256 MHz LO, shown in the lower left corner of the "breadboard."

Setup and Using *EasyPLL*

First you must open an account on the National Semiconductor Web site:

1) Connect to **www.national.com**.
2) Click "Wireless" under the "Design" heading.
3) On the right side, under "My Designs," click "sign-on here."
4) Click on "here" in "You may create your own personal work space on National's Web site here."
5) Fill out all the information: e-mail address, password, re-enter password, name, address and so on.
6) Click "create."

From now on, the Web site will recognize you (on the same PC) or log on from any other PC using your e-mail address and password.

The Web site provides access to all your previously stored designs, plus you can update a design and save it under a new name.

Using EasyPLL

1) Connect to **www.national.com**.
2) Under "Design" click on "Wireless."
3) Click on "start here."
4) Select the following:
   - Loop filter type: Passive Loop Filter
   - PLL System Specifications:
     - Min output freq 2256 MHz
     - Max output freq 2256 MHz
     - Channel spacing 2000 kHz
     - Crystal Freq 10.0 MHz
     - Max Power Supply 5.5 V
   - PLL Selection Options:

   Check only: single PLLs, Integer PLLs, Automatically Narrow PLL Choices, Voltage Check.
5) Click "Now View Recommended Parts."
6) Select LMX2326 for the PLL.
7) Select "any" for the VCO, and then edit the data:
   - Freq min 2250 MHz
   - Freq max 2310 MHz
   - Gain 15 MHz/V
   - Tune volts min 1.5 V
   - Tune volts max 4.5 V
   - Pout 0 dBc
   - Power Supply 5.0 V
   - VCO cap 50 pF
   - 10 kHz noise floor –114 dBc

   (All this data is available on the Z-Comm Web site for the CRO2275A VCO. Note: For the frequency tuning voltage values, use the "Typical tuning curve" graph on page 2 of the data sheet, not the numbers listed in the table on page 1.)
8) Click "Create a design." (A block diagram and many fields appear.)
9) At this point each parameter has a HELP icon that you can click. Each help file provides a lot of important information on that parameter. I suggest printing out each of the HELP files (use Shift-Print Screen, then paste the clipboard into "Paint" or other graphics software, then print it out). As a "beginner," I was constantly opening the HELP files — it is more efficient to have all the HELP files available at once.
10) Modify/verify the values in "Advanced Settings."
    - Uncheck "allow 2 parallel capacitors"
    - Filter order 3rd order
    - Resistor Tolerance 10 %
    - Capacitor Tolerance 10 %
11) Modify/verify the values in "Loop Filter Specification" as follows:
    - Comparison freq 2000 kHz
    - Charge pump gain 1.25 mA — see Note 18.
    - VCO gain 15 MHz/V
    - VCO input cap 50 pF
    - Output freq 2256 MHz
    - Phase margin 50° 0%
    - Loop Bandwidth 10 kHz 20%
    - T3/T1 Ratio 40% 20%
12) Modify/verify the values in "Loop Filter Optimization."
    - Spur Offset freq 2000 kHz
    - Initial freq 2256 MHz
    - Ending freq 2256 MHz
    - Tolerance 1000 Hz
    - Optimize for Min High Order Cap
    - Max lock time 30,000 µs
    - Max spur gain 100,000 dB
    - Max High Order Cap 50 pF
13) Scroll to the top of the page and verify all values to make sure none have changed. Print the entire page.
14) Click "View Recommended Component Values." This is when the software does its magic and computes the values of the loop filter resistors and capacitors. In about 20 seconds the screen updates, showing the values. Print the entire page. At the upper right, save your design by clicking "Rename Design." Enter a name that will identify it, then enter any additional information in the "comments" box.
15) Click "Analyze a Design."
16) Under "Phase Noise" click "Go."
17) The curves show the phase noise contribution of various components.
18) Select various traces to see the noise contribution of each component. Close this window when you're done examining the predicted noise spectrum.
19) Click "Build It."
20) Click on "Design Check."
21) Note the Design Check parameters, the result and comments for each. All show "OK" except for three parameters.

    A) "Capacitor Dielectric Check" — This is telling/reminding you to use loop filter capacitors that have a dielectric that has a very nearly or zero temperature coefficient. Types NP0/C0G are preferred. Unfortunately, Digi-Key does not offer a 0.01 µF for C18 with an NP0/C0G dielectric. I used what I had in my parts box.

    B) "Discrete Sample Effects Check" – I don't understand the comments for this parameter. However, I know that lock time is not a concern in this design so this can probably be ignored.

    C) "Low Filter Order Check" — the spur levels can be reduced but since I could not even see them on the spectrum analyzer I ignored this comment. Also, if using a higher order filter, more parts will be required.

further design work. The capacitor that I used for C18 is a type that the Design Check part of *EasyPLL* says *not* to use. See step 21.A of the Set-up and Using *EasyPLL* Sidebar for a discussion on this issue. Trying new values in the actual circuit requires removing existing resistors and capacitors and installing new values. A circuit board can only take so much of this before the traces start to lift off the board. I used *EasyPLL* and Dean's Book until everything looked as good as possible, and then I built the circuit. As we say at work, "It's time to shoot the engineer and build the product."

Figure 10 shows the spectrum of my HP8640 signal generator at 400 MHz for comparison with the spectrum of my LO, shown in Figure 9. Note that the LO spectrum isn't as "clean" as the HP signal generator. The LO spectrum was not as clean as predicted in *EasyPLL*, but I believe is good enough to use for satellite work.

I constructed a down-converter for 2400 MHz satellite operation that is not in final form as this is written — it is a true breadboard, and is shown in Figure 11. I compared the sensitivity of my down-converter to a commercial down-converter (a model AIDC 3731 from K5GNA) with a 1 dB noise figure. I found that they had the same sensitivity when receiving a very weak signal (–127 dBm). My down-converter has no input filter to remove the noise from the image frequency (LO – IF = 2256 MHz – 144 MHz = 2112 MHz in this case). When adding the filter, the noise figure may be even lower!

The LO in the K5GNA down-converter is much cleaner than my LO, and slightly cleaner that the HP8640. Apparently, mine is adequate for weak signal reception.

Summary

I learned a *lot* developing this project. Getting to see the output on a spectrum analyzer is a real treat. Better yet, when using a slow clock frequency for the PIC, you get to see the VCO free run and drift, then snap (lock on) to 2256 MHz when the PLL is loaded with the data to make it run.

I talked with the author of the *RIG Journal* down-converter article (see Note 2) via e-mail and he provided lots of help. We used digital pictures of my layout and the spectrum analyzer, voltage measurements, and so on to discuss problems I encountered. As a result, we became good friends. He's not a ham but is a very dedicated and capable homebrewer. He lives in England.

I strongly urge you to read all the literature you can find on PLLs — in particular *The ARRL Handbook*, Mini-Circuits' *VCO Designers Help Notes* on their Web site, and most importantly "Dean's Book." PLLs are a different beast but you *can* learn their operation. If you don't understand something, contact the manufacturer and the authors of the articles. Do a Web search on "RF PLL Design." I'll help as much as possible via e-mail and *snail-mail*.

This design can be used for other frequencies as well. This will require changing the PIC variables, loop filter values and the VCO. Z-Comm and Champion are two manufacturers of narrow range VCOs in a package style that will fit this board layout. While Mini-Circuits makes VCOs that will fit this board, they don't make a narrow range VCO in the range I needed for this project. The PLL IC operates from 100 MHz to 2800 MHz, so you can build an oscillator for any frequency in this range using the correct VCO type, reference oscillator frequency, loop filter values and constants in the PIC program.

The level is too low for most mixers (–2 dBm). Most require +7 dBm or more, so I use a MAR-6 to increase the output level for proper injection level.

The PIC clock is set by R8 and C11. The values shown produce a very low clock frequency — about 85 Hz. I used a low frequency clock for the PIC clock so that I can monitor the three lines going to the PLL IC. I can literally watch the code being loaded into the PLL! It takes six seconds to load the data and start up the PLL. I haven't tried a higher frequency but plan to do so later.

Please send any questions or comments to me by e-mail or US Mail. I am open to suggestions to improve the design. If you choose to build this LO, I would be very interested in hearing from you. Good luck — I hope you learn as much as I did!

Notes

[1]Jim Kocsis, WA9PYH, "A Synthesized Down-Converter for 1691-MHz WEFAX," *QEX* Mar/Apr 2000, pp 48-55.

[2]*Remote Imaging Group Journal*, No. 65, June 2001, pp 62-81.

[3]Mark Wilson, K1RO, *Ed.*, *The ARRL Handbook*, PLL section, Chapter 10, "Oscillators and Synthesizers," 2008 Edition, pp 10.33-10.46. *The ARRL Handbook* is available from your local ARRL dealer, or from the ARRL Bookstore, ARRL order no. 1018. 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**.

[4]**www.minicircuits.com/pages/app_notes.html** Scroll down to "VCOs."

[5]**www.national.com/appinfo/wireless/files/deansbook4.pdf**.

[6]"Design a PLL for a Specific Loop Bandwidth," *EDN*, Oct 12, 2000.

[7]"Design Loop Filters for PLL Frequency Synthesizers," *Microwaves & RF*, Sep 1999.

[8]"PLL Synthesizers," *EDN*, Mar 14, 1997.

[9]"Model PLL Dynamics & Phase Noise Performance," *Microwaves & RF*, May 2000.

[10]"Dealing with PLL Generated Phase Noise," *RF Design*, August 1997.

[11]"An Analysis & Performance Evaluation — A Passive Design Technique for Charge Pump PLLs," National Semiconductor Application Note AN1001.

[12]**www.zcomm.com/support/application_notes.shtml**.

[13]John Clark, K2AOP, "A Simple, Well-Behaved Crystal Oscillator," Technical Correspondence, *QST*, Sep 2004, p 67. Also see changes and corrections in *QST*, Oct 2004, p 39 and Dec 2004, p 37.

[14]**www.national.com/ds/lom/lm2306.pdf**. The page will open in Adobe Acrobat. Click "save as" and store it on your PC. Although the file is named "2306," it covers the LMX2326 as well.

[15]The files associated with this article are available for download from the *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **1x08_Kocsis.zip**.

[16]A suitable soldering iron is a Weller WP-25, Digi-Key WP-25-ND $35.99, with tips ST1-ND $4.35 and ST4-ND $4.35, a suitable grounded wrist strap is Highland 2272, Digi-Key SCP-172-ND, $7.46, Chemtronics desoldering braid, Digi-Key 80-2-5-ND, $3.20 and Kester type SP-44 soldering flux, Digi-Key KE-1700-ND, $1.28. See **www.digikey.com**.

[17]David Benson, *Easy PIC'n*, Square 1 Electronics, PO Box 1414 Hayden, ID 83835; **www.sq-1.com.** This book is out of print. A new version, *Easy Microcontrol'n* is available from Jameco — **www.jameco.com**. Order p/n 215061B for $32.75.

[18]The LMX2326 is capable of being set to two charge pump levels: 1.25 mA and 1.00 mA. Setting it to 1.00 mA allows lower values for the loop filter resistors. These lower values result in less noise contributed by the resistors. The LMX2326 is usually run at 3 V but if run at 5 V, the charge pump levels increase by 25% resulting in a charge pump level of 1.25 mA (1.00 × 125% = 1.25).

Jim Kocsis, WA9PYH, has been a ham since 1964. He earned his Amateur Extra class license in 1986. His interests include manually sent CW, PSK, casual DXing, ragchewing on HF, QRP CW Field Day operating and homebrewing just about any ham radio equipment. He has designed and built a downconverter and dish for 1691 MHz weather satellite reception, a 436 MHz circularly polarized Yagi for satellite work, and is currently building a downconverter for 2400 MHz satellite work. He has also built an azimuth and elevation rotator system that interfaces with SatPC32. He makes his own circuit boards, including the board for this article, but they are limited to SMT components. Jim is a Test Engineer for Honeywell Aerospace, with more than 30 years of experience. Some of Jim's other interests include cooking and baking, reading travel books, traveling and noncompetitive bicycling.

QEX

Maynard Wright, W6PAP

6930 Enright Dr, Citrus Heights, CA 95621, w6pap@arrl.net

More *Octave* For Transmission Lines

In this installment, we learn how to model transmission lines using lumped resistive, capacitive and inductive circuit elements.

In "*Octave* for Transmission Lines," we used *Octave* to calculate the input impedance of a transmission line, and we explored some uses for such a calculation.[1, 2] There are, though, some aspects of transmission line theory and practice that we can't model entirely by determining just the input impedance. We might, for example, want to insert a transmission line into a circuit that contains a transmitter and an antenna to see what happens. We often do that by considering the reflection coefficients at particular junctions, and the SWR on a particular line. For our purposes, though, a practical circuit usually consists of a transmitter, one or more lengths of transmission line, some matching devices, and an antenna. The interactions among all those circuit elements can be complex and we need to consider more than just, for example, the mismatch between the line and the antenna to make sense of the various signals involved.

One way to do that is to build a circuit model in which each transmission line segment is represented by a "transmission matrix" or "ABCD matrix.[3, 4] The use of this matrix is convenient for the solution of a variety of network problems, including those involving transmission lines.

For our purposes, though, I think it would be useful to be able to represent the transmission line, which consists of distributed series resistance and inductance and shunt capacitance and conductance, as a combination of lumped elements. That will allow us to study the line, and the other units attached to it, using circuit theory in a more intuitive manner than the ABCD matrix allows.[5]

It turns out that such a transmission line model is possible, and it has been around for several decades.[6, 7] At a particular frequency (and only at that frequency), and for a line of uniform characteristics throughout its length, we can model the transmission line as in Figure 1, the elements having the following values:

$Z_A = Z_0 \times \tanh(n\, l / 2)$
$Z_B = Z_A$
$Z_C = Z_0 / \sinh(n\, l)$
where:

Z_0 = the characteristic impedance of the line.

n = the complex loss coefficient of the line (also known as the "complex propagation constant").

l = the length of the line in units the same as those used to define n.

When modeling something other than a transmission line, the two series arms of the T-equivalent network may differ. When considering a uniform length of line, however, the line "looks" the same from either end if the far end is terminated in the same impedance for each "look."

Figure 1 shows an unbalanced circuit. We'll use unbalanced circuits throughout this discussion, but each of the circuits can be used to represent a balanced circuit by simply splitting the series impedances between the two series arms of the "T" to form an "H."

Does this model actually work? We can try it out by replacing the expression for input impedance in Table 1 of "*Octave* for Transmission Lines" with a calculation based on a T-equivalent network. (See Note 1.) The modified code appears in Table 1 of this article.[8]

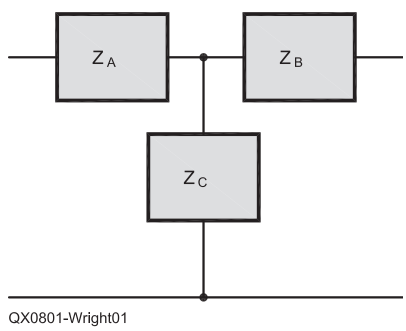


Figure 1 — T-equivalent network for transmission line.

Note that, in the substituted code, we have defined the three complex impedances of the T-equivalent network using the equations above. We have then combined the terminating impedance with the T-equivalent network using circuit theory to combine series and parallel impedances, and we have ended up with a program that yields the same answers as does the code in Table 1 of the earlier article. This gives us some assurance that the model is correct.

Since we have complex impedances for the T-network, we could assemble a physical artificial line using resistors, capacitors, and inductors that would represent the line closely at the particular frequency for which the T-network elements are valid. Such networks were frequently used in modeling transmission lines and other circuits in the days before digital computers became generally available.

In an interesting sidelight on history (from our modern perspective), Bartlett notes that the T-network calculations sometimes yield negative resistances, and that it is impossible to use the results of such a calculation. (See Note 6.) Negative resistances are no problem for our computers, but in 1931 Bartlett had no computer and he had no op amps from which to synthesize negative resistances for an analog simulator, either of which would have solved his problem. Bartlett instead counsels breaking the transmission line up into smaller segments, which would usually eliminate the negative resistances, but which would increase the number of passive components required to build the artificial line. If we encounter negative resistances in our T-equivalent models, we'll use them as is.

Having a transmission line represented as a lumped-element circuit, we might simply use the code we have in *Octave* to produce a group of circuit elements for use in a *SPICE* simulation.[9] Some early versions of *SPICE* make no provision for transmission lines, while other versions feature line simulations that are not very general and that may not

[1]Notes appear on page 31.

Table 1
***Octave* Code for Transmission Line Analysis**

```
#! /usr/bin/octave -qf
# Print header
printf("\n\n     *** TRANSMISSION LINE CALCULATIONS ***\n");
# Enter input data from keyboard
f = input("\n                               FREQUENCY IN MHz: ");
d = input("\n                         LENGTH OF LINE IN FEET: ");
a = input("\n                 ATTENUATION IN dB PER 100 FEET: ");
v = input("\n                VELOCITY FACTOR AS A PERCENTAGE: ");
Zo = input("\n               CHARACTERISTIC IMPEDANCE IN OHMS: ");
Rt = input("\n     REAL PART OF TERMINATING IMPEDANCE IN OHMS: ");
Xt = input("\nIMAGINARY PART OF TERMINATING IMPEDANCE IN OHMS: ");
# Convert inputs as required
a = a ./ 1e2;  # convert dB per 100 feet to dB per foot
a = 0.1151 .* a;  # convert dB to nepers
c = 9.836e8;  # speed of light in feet per second
lambda = c ./ (1e6 .* f);  # wavelength of signal in vacuum
lambda = (v ./ 1e2) .* lambda; # adjust lambda for velocity
B = (2 .* pi) ./ lambda; # calculate Beta
Zt = Rt .+ j .* Xt;  # calculate complex terminating impedance
# Calculate elements of T-equivalent network
# ZA = input series element
# ZB = output series element
# ZC = shunt element
ZA = Zo .* tanh((a .+ j .*B) .* d ./ 2.);
ZB = ZA;
ZC = Zo ./ sinh((a .+ j .*B) .* d);
# Calculate input impedance
#Zd = Zo .* tanh((a .+ j .* B) .* d .+ atanh(Zt ./ Zo));
#Zd = Zo .* ((Zt .* cosh((a .+ j .*B) .* d) .+ Zo .* ...
#sinh((a .+ j .*B) .* d)) ./ (Zt .* sinh((a .+ j .*B) .* d) ...
#.+ Zo .* cosh((a .+ j .*B) .* d)));
Zd = ZA .+ (ZC .* (ZB + Zt)) ./ (ZC .+ ZB .+ Zt);
# Print results
for k = 1:columns(Zd)
   if imag(Zd(k)) < 0
      printf("\n\n  INPUT IMPEDANCE = %8.5g - j%-8.5g\n\n", ...
      real(Zd(k)), abs(imag(Zd(k))));
   else
      printf("\n\n  INPUT IMPEDANCE = %8.5g + j%-8.5g\n\n", ...
      real(Zd(k)), imag(Zd(k)));
   endif
endfor
# End input impedance program
```

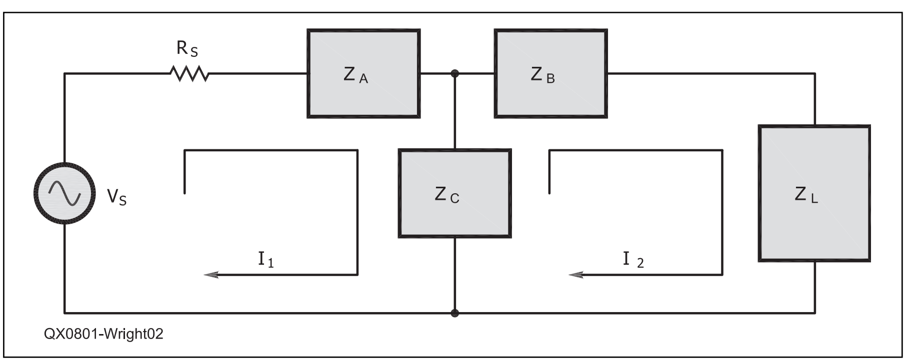


Figure 2 — T-equivalent network with transmitter, one section of line, and load.

be applicable to problems we would like to solve. We can use the *Octave* functions *fopen, fprintf,* and *fclose* to write the results of our T-equivalent network calculations to a file that can be pulled into a *SPICE* source file, ready to have its line numbers and node numbers edited to conform to the circuit into which it's been placed. The code in Table 2 can be added to that in Table 1 to produce such a "*SPICE*-ready" output file.

The "*SPICE*-ready" file for some typical input values is listed in Table 3. Note that you must change the line numbers (prefixed with "xx" in the *Octave* output) and the node numbers (prefixed with "Y") to fit properly into the *SPICE* file for which you need the transmission line. Depending on the version of *SPICE* you use, you may need to do some reformatting and you might want to replace the exponentials with expressions of units such as pF or mH that are acceptable to *SPICE*. Be cautious: if the file specified in the *fopen* statement already exists and contains other data, the *SPICE* statements will be appended to the file contents.

As amateurs continue to experiment at low frequencies, it may become important to be able to consider transmission lines of complex impedance. So far, we've assumed that the characteristic impedance is always real (as is often done in radio work), but at audio frequencies most lines exhibit an impedance having an angle of about 45°, making the imaginary component about the same value as the real component. At 136 kHz and other low frequencies of interest to amateurs, it may not always be possible to ignore the imaginary component of characteristic impedance. The transmission line models we have been building are capable of handling complex characteristic impedances with no modification, although as is appropriate for most amateur calculations, we've been neglecting the imaginary component of the characteristic impedance.

Now let's try to do something more with our T-equivalent model. We'll postulate a transmitter (Figure 2) with a 50 Ω source impedance ($V_S + R_S$) connected by a section of transmission line to an antenna ($\mathbf{Z_L} = R_L + jX_L$).[10] We'll use our T-network to represent the transmission line. Note that we've shown two independent currents flowing around the two loops in the circuit. The voltage drops around each loop must add up to zero according to Kirchhoff's Second Law (Kirchhoff's Voltage Law) as described on page 4.4 of the *ARRL Handbook*. (See Note 5.) We can write equations for the two current loops as follows:

$$V_S = (R_S + \mathbf{Z_A} + \mathbf{Z_C}) \times I_1 - \mathbf{Z_C} \times I_2$$

$$0 = -\mathbf{Z_C} \times I_1 + (\mathbf{Z_C} + \mathbf{Z_B} + \mathbf{Z_L}) \times I_2$$

Rather than including voltage sources as negative numbers on the right side of the equation, we have algebraically moved them to the left side, changing their signs in the process. Since the I_2 loop has no voltage source, its form is unaffected by the move.

Once we have the circuit laid out, we can see that we know everything of interest except I_1 and I_2. We may have to determine the antenna impedance from measurements or from calculations.[11] We might calculate V_S from the known power output of the transmitter but, when interested only in comparative calculations, I often set V_S to 2 V. With a perfectly matched load and no attenuation, the voltage across the load is then 1 V, a convenient value for comparison with other, less than ideal, load voltages.

We can solve for the two unknowns, I_1 and I_2, using a variety of techniques. One method that will work well for us is the method of determinants.[12] We form a matrix using the coefficients of the two currents as the elements:

$$D = \begin{pmatrix} (R_S + Z_A + Z_C) & -Z_C \\ -Z_C & (Z_C + Z_B + Z_L) \end{pmatrix}$$

We then form another matrix by substituting the column vector comprising the voltage sources to the left of the equal sign above for the column whose number corresponds to the number of the current we wish to find. For I_1, for instance:

$$D_1 = \begin{pmatrix} V_S & -Z_C \\ 0 & (Z_C + Z_B + Z_L) \end{pmatrix}$$

We then find I_1 by using determinants as follows:

$$I_1 = \frac{\text{Determinant } (D_1)}{\text{Determinant } (D)}$$

We can make a similar substitution of the column vector into the second column of D to find I_2.

Since $\boldsymbol{Z_A}$, $\boldsymbol{Z_B}$, $\boldsymbol{Z_C}$, and $\boldsymbol{Z_L}$ are generally complex, manual solution using this method is tedious and error prone. We'll use *Octave* to ease the burden and to ensure against "eyes glazed over" errors.

We might want to experiment with different antenna impedances and transmission line types and lengths, and we may need to solve for these currents several times for a particular problem. Having to enter all the data over and over again every time we run *Octave* would be inconvenient. We'll replace the input lines in Table 1 with statements that assign values directly to the various variables. Note that when you are working with an *Octave* script file, you can quickly change between input statements and assignment statements so that you can arrange to input from the keyboard only those values you want to change often during a series of *Octave* "runs."

We'll put the calculations into a loop that computes all the unknown currents. Doing this will allow us to increase the complexity of the circuit, and the orders of the determinants, without having to rewrite any code. The code for this calculation is given in Table 4. Note that we've retained the input impedance calculation.

Most of the code here conforms to the *Octave* code we've used in previous articles. We need, though, to examine the code under the comment "# Prepare matrices for solution of currents."

As described in "*Octave* for Transmission Lines," a comma in a matrix assignment in *Octave* delineates two elements in the same row. A semicolon terminates a row. The matrix *right_side* thus consists of two rows of two elements each. Since that qualifies *right_side* as a square matrix, *Octave* won't complain later when we compute the determinant of that matrix. The matrix *left_side* is a column matrix, so every delimiter is a semicolon. In this case, there are two elements, matching the 2 × 2 *right_side* matrix.

The block of code under the comment "# Solve for currents" calculates the determinants of the various matrices and uses them to calculate the two unknown currents. The line *denom = det(right_side)* calculates the value of the denominator matrix. Since it's the same for each unknown current, it's calculated once before any of the loops begin.

The outer of the two nested for loops, using the loop variable *m,* steps through all the unknown currents. The inner loop, using the variable *n*, replaces all the elements of a column of *right_side* (elements corresponding to a particular current) with elements from "*left_side.*" The "for" loop that prints out the results works as does the similar loop in "*Octave* for Transmission Lines."

The code in Table 4 shows a mismatch

Table 2
Additional Code for Table 1 to Write *SPICE*-Ready Output File

```
# Write output to a SPICE-ready file
ftag = fopen("/home/mwright/Octave/spice_file.txt", "at", "native");
# Calculate and print out values for ZA
fprintf(ftag, "\nxx1 RA  Y1  Y2   %8.5g", real(ZA));
if imag(ZA) > 0
   LA = imag(ZA) / (2 * pi * f * 1e6);
   fprintf(ftag, "\nxx2 LA  Y2  Y3   %8.5g", LA);
elseif imag(ZA) < 0
   CA = 1 / (2 * pi * f * 1e6 * -imag(ZA));
   fprintf(ftag, "\nxx2 CA  Y2  Y3   %8.5g", CA);
else
   fprintf(ftag, "\nxx2 RAA  Y2  Y3   %8.5g", 0);
endif
# Calculate and print out values for ZB
fprintf(ftag, "\nxx3 RB  Y3  Y4   %8.5g", real(ZB));
if imag(ZB) > 0;
   LB = imag(ZB) / (2 * pi * f * 1e6);
   fprintf(ftag, "\nxx4 LB  Y4  Y5   %8.5g", LB);
elseif imag(ZB) < 0;
   CB = 1 / (2 * pi * f * 1e6 * -imag(ZB));
   fprintf(ftag, "\nxx4 CB  Y4  Y5   %8.5g", CB);
else
   fprintf(ftag, "\nxx4 RBB  Y4  Y5   %8.5g", 0);
endif
# Calculate and print out values for ZC
fprintf(ftag, "\nxx5 RC  Y3  Y6   %8.5g", real(ZC));
if imag(ZC) > 0
   LC = imag(ZC) / (2 * pi * f * 1e6);
   fprintf(ftag, "\nxx6 LC  Y6  0   %8.5g", LC);
elseif imag(ZC) < 0
   CC = 1 / (2 * pi * f * 1e6 * -imag(ZC));
   fprintf(ftag, "\nxx6 CC  Y6  0   %8.5g", CC);
else
   fprintf(ftag, "\nxx6 RCC  Y6  0   %8.5g", 0);
endif
fprintf(ftag, "\n");
# Close file
fclose(ftag);
# End program
```

in that the source and load impedances are 50 Ω, but the characteristic impedance of the line is 75 Ω. To see how much trouble that will cause us, let's consider the matched case first. We could do that by changing the values in Table 4 and running the script in *Octave* (a good exercise), but we'll take a more direct approach here because everything is matched. The input current (I_1) will be 2 V / 100 Ω = 0.020 A. We know from the input data and the matched condition that the loss is 6 dB, so the output current, flowing into the same impedance as does the input current, must be approximately one half the input current, or 0.010 A.[13]

The code in Table 4 produces the following output, where blank lines have been eliminated for brevity:

LOOP 1 CURRENT = 0.015794 + j0.00094068

LOOP 1 CURRENT MAGNITUDE = 0.015822

LOOP 2 CURRENT = 0.0059861 + j0.0075107

LOOP 2 CURRENT MAGNITUDE = 0.0096043

INPUT IMPEDANCE = 76.179 – j7.515

The power dissipated in the load (antenna) is:[14]

$P_{ant1} = (LOOP2CURRENTMAGNITUDE)^2 \times real(\mathbf{Z}_L)$

$P_{ant1} = (0.0096043\ \text{A})^2 \times 50\ \Omega$

$P_{ant1} = 0.00461\ \text{W}$

The power that would be expended in the antenna if there were no transmission line between the source and the load is:

$P_{ant0} = abs(V_S / (R_S + \mathbf{Z}_L))^2 \times real(\mathbf{Z}_L)$

$P_{ant0} = 0.02\ \text{W}$

The insertion loss in dB is $20 \times \log(P_{ant0} / P_{ant1})$ = 6.37 dB — not a bad penalty over the 6 dB loss we would incur if we used a "properly" matched line of the same length.[15] We can, if we wish, calculate the voltage reflection coefficient, the SWR, and other parameters of interest at either the transmitter or load using *Octave* or data derived from the *Octave* "run."

What happens if we add a section of 600 Ω transmission line of length, say, 200 feet between the 75 Ω line and the antenna? Since our code for finding the currents is capable of handling more loops, we can do it. We need, though, to create separate T-equivalent circuits for each section of line. We'll name the secondary constants a_1, a_2, v_1, v_2, $\mathbf{Z}_{01}$, $\mathbf{Z}_{02}$, and so

Table 3

***SPICE* Ready Output File**

```
xx1 RA  Y1  Y2     5.801
xx2 LA  Y2  Y3  3.1746e-06
xx3 RB  Y3  Y4     5.801
xx4 LB  Y4  Y5  3.1746e-06
xx5 RC  Y3  Y6    -2.0674
xx6 CC  Y6  0   2.5229e-10
```

Table 4

Modified *Octave* Code for Transmission Line Analysis

```
#! /usr/bin/octave -qf
# Print header
printf("\n\n     *** TRANSMISSION LINE CALCULATIONS ***\n");
# Specify transmission line data
f = 7.01;    #   FREQUENCY IN MHz
d = 100;    #   LENGTH OF LINE IN FEET
a = 6.0;     #   ATTENUATION IN dB PER 100 FEET
v = 83;      #   VELOCITY FACTOR AS A PERCENTAGE
Zo = 75;    #   CHARACTERISTIC IMPEDANCE IN OHMS
# Specify source (transmitter) and load (antenna)
VS = 2.0;
RS = 50;
Rt = 50;
Xt = 0;
# Convert inputs as required
a = a ./ 1e2;  # convert dB per 100 feet to dB per foot
a = 0.1151 .* a;  # convert dB to nepers
c = 9.836e8;  # speed of light in feet per second
lambda = c ./ (1e6 .* f);  # wavelength of signal in vacuum
lambda = (v ./ 1e2) .* lambda; # adjust lambda for velocity
B = (2 .* pi) ./ lambda; # calculate Beta
ZL = Rt .+ j .* Xt;  # calculate complex terminating impedance
# Calculate elements of T-equivalent network
# ZA = input series element
# ZB = output series element
# ZC = shunt element
ZA = Zo .* tanh((a .+ j .*B) .* d ./ 2.);
ZB = ZA;
ZC = Zo ./ sinh((a .+ j .*B) .* d);
# Prepare matrices for solution of currents
right_side = [RS + ZA + ZC, -ZC; -ZC, ZB + ZC + ZL];
left_side = [VS; 0];
# Solve for currents
denom = det(right_side);
for m = 1: rows(right_side)
   numerator = right_side;
   for n = 1: rows(right_side)
      numerator(n, m) = left_side(n);
   endfor
   current(m) = det(numerator) / denom;
endfor
# Print out currents
for m = 1:rows(right_side)
   if imag(current(m)) < 0
      printf("\n\n  LOOP %d CURRENT = %8.5g - j%-8.5g\n", ...
      m, real(current(m)), -imag(current(m)));
   else
      printf("\n\n  LOOP %d CURRENT = %8.5g + j%-8.5g\n", ...
      m, real(current(m)), imag(current(m)));
   endif
printf("\n\n  LOOP %d CURRENT MAGNITUDE = %8.5g\n", ...
m, abs(current(m)));
endfor
# Calculate input impedance
Zd = ZA .+ (ZC .* (ZB + ZL)) ./ (ZC .+ ZB .+ ZL);
# Print input impedance
for k = 1:columns(Zd)
   if imag(Zd(k)) < 0
      printf("\n\n  INPUT IMPEDANCE = %8.5g - j%-8.5g\n\n", ...
      real(Zd(k)), abs(imag(Zd(k))));
   else
      printf("\n\n  INPUT IMPEDANCE = %8.5g + j%-8.5g\n\n", ...
      real(Zd(k)), imag(Zd(k)));
   endif
endfor
# end program
```

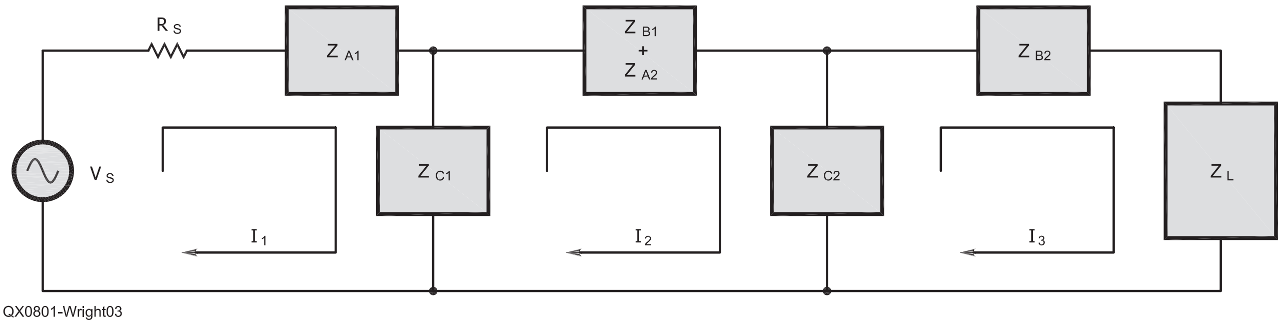


Figure 3 — T-equivalent network with source, two sections of line, and load.

on, and the T-equivalent network elements $\boldsymbol{Z_{A1}}$, $\boldsymbol{Z_{B1}}$, $\boldsymbol{Z_{C1}}$, $\boldsymbol{Z_{A2}}$, $\boldsymbol{Z_{B2}}$, and $\boldsymbol{Z_{C2}}$ where each element containing the numeral 1 is associated with the 75 Ω line and each element containing the numeral 2 is associated with the 600 Ω line. The circuit is shown in Figure 3. Note that, in addition to adding a length of 600 Ω line, we've adjusted the length and attenuation of the 75 Ω line just to keep things interesting.

Since $\boldsymbol{Z_{B1}}$ and $\boldsymbol{Z_{A2}}$ are in series, we have combined them. You can add them algebraically while they are expressed as complex impedances, but only with caution after they are converted to R, L, and C values. The real parts and inductive imaginary parts may be added algebraically, but if there are any capacitances involved, they must be added using the rule for series capacitances given in the *ARRL Handbook*. (See Note 5.)

There are three current loops, I_1, I_2, and I_3, in the model of Figure 3. This results in three simultaneous equations: the column vector *left_side* is three elements in length and the square matrix *right_side* is a 3 × 3 matrix. The elements of *right_side* are determined as before by using the coefficients of the Kirchhoff's voltage law equation for each current loop. The code is listed in Table 5.

Note that we've omitted the input impedance calculation. We could have calculated $\boldsymbol{Z_{in}}$ as we did before, but it's become more complicated, with the addition of the third current loop, and it's probably easier at this point to obtain the input impedance, if we need it, using V_S and I_1:

$\boldsymbol{Z_{in}} = (V_S - I_1 \times R_S) / I_1$

Running the code in Table 5 results in the following outputs, again with blank spaces removed:

LOOP 1 CURRENT = 0.030151 - j0.0089586
LOOP 1 CURRENT MAGNITUDE = 0.031454
LOOP 2 CURRENT = -0.011358 + j0.00047854
LOOP 2 CURRENT MAGNITUDE = 0.011368
LOOP 3 CURRENT = 0.011888 - j0.00086264
LOOP 3 CURRENT MAGNITUDE = 0.011919

The power dissipated in the load (antenna) — see Note 14 — is:

P_{ant1} = *(LOOP 3 CURRENT MAGNITUDE)*$^2 \times$ *real*$(\boldsymbol{Z_L})$

P_{ant1} = 0.00710 W

The power that would be expended in the antenna if there were no transmission line between the source and the load is still, as above, P_{ant0} = *0.02* W and the insertion loss in dB — see Note 15 — is:

$20 \times \log (P_{ant0} / P_{ant1}) = 4.50$ dB. This value includes the various reflection losses at, and interactions among, the various junctions in the system as well as the loss in each section of transmission line. The losses themselves add up to about 0.4 dB, so we can see that we are "losing" quite a bit of power due to reflections. As noted above, an actual transmitter would probably exhibit a lower source impedance than the "ideal" source we are using.

At this point, we have currents I_1 and I_2, or I_1, I_2, and I_3, depending on which model we are using, and we can use them to calculate the voltages at any point in either of our circuits. From there, we can go on to impedances, SWR values and reflection coefficients if we wish to analyze either circuit in more detail using familiar measures of impairment. Over the decades, radio folks have usually worked with SWR and reflection coefficients, while telecommunications folks have most often worked with impedances. Having all those measures in our "toolbox," though, is not a bad thing. We can pick and choose the most useful technique for any particular situation.

It may seem that, by combining $\boldsymbol{Z_{B1}}$ and $\boldsymbol{Z_{A2}}$, we have obscured access to the junction between the two sections of transmission line, a significant reflection point. Now that we have I_2, though, we can use $\boldsymbol{Z_{A2}}$ independently along with the other currents and impedances to determine the impedance, or other parameters, at that junction.

One of the things we ought to learn about *Octave*, as well as about any other software, is when to "punt," so to speak, and move on to another tool. *SPICE* is better suited than *Octave* for most simulations with multiple current loops, since the circuit layout may be entered directly from a schematic diagram of the circuit of interest, and we don't need to manually work the polynomial coefficients into the code. (See Note 9.) The exercise we have just finished ought to verify that *SPICE* is easier to use. If we had used *Octave* and *SPICE* in conjunction here, we would have used *Octave* to generate the T-equivalent transmission line models and would have then plugged them into *SPICE* models of the circuits. Although it was convenient to combine $\boldsymbol{Z_{B1}}$ and $\boldsymbol{Z_{A2}}$ in the model of Figure 3, it would probably be more convenient to keep them independent if we are using *Octave* to build models for *SPICE*. The extra *SPICE* node required is a trivial penalty for the convenience of not having to combine the two impedances.

We could, in addition, add a tuner or other circuits with lumped elements to our *Octave* model. Again, we would probably be better off using *SPICE*.

One situation in which *SPICE* is much easier to use than *Octave* is when transformers are involved. A model of a transformer requires controlled sources.[16] A controlled source is a current or voltage generator that is controlled by a voltage or current elsewhere in the circuit. We can certainly model a circuit that contains a transformer (or a balun) in *Octave*, but it complicates considerably the task of working out the polynomial coefficients. *SPICE* handles transformers directly from the circuit diagram without such intervention from the user.

Although we've found that *Octave* can be used to solve relatively complex transmission line problems, its principal value in that regard may be in generating transmission line models for use in *SPICE*. Our code may also be useful to someone who doesn't have access to *SPICE* or who would like to independently verify results obtained from a *SPICE* circuit model.

Notes

[1]Maynard Wright, W6PAP, "*Octave* for Transmission Lines," *QEX*, Jan/Feb, 2007, pp 3-8.

[2]John W. Eaton, *GNU* Octave *Manual*, Network Theory Limited, 1997 (see **www.octave.org**).

[3]*Reference Data for Radio Engineers*, Fourth Edition, ITT, 1956, p 660.

[4]Robert A. Chipman, *Schaum's Outline Series: Theory and Problems of Transmission Lines*, McGraw-Hill, 1968, pp 142 and 155.

[5]Mark Wilson, K1RO, Ed., *The ARRL Handbook for Radio Communications*, 2008, The American Radio Relay League, Inc, 2007, Chapter 4. *The ARRL Handbook* is available from your local ARRL dealer, or from the ARRL Bookstore, ARRL order no. 1018. 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**.

[6]A. C. Bartlett, BA, *The Theory of Electrical Artificial Lines and Filters*, John Wiley & Sons, Inc, 1931, p 14.

[7]Frederick E. Terman, *Radio Engineers' Handbook*, McGraw-Hill, 1943, p 195. The formulas for calculating Z_A, Z_B, and Z_C of the T-equivalent network here are equivalent to those in Bartlett, Note 6, but among the required arguments are the primary constants of the transmission line. We'll use Bartlett instead of Terman, as Bartlett's equivalent formulas require only the secondary constants as arguments. In my opinion, that's more convenient for most Amateur Radio applications.

[8]The Octave program files associated with this article are available for downloading from the *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **1x08_Wright.zip**.

[9]Paul W. Tuinenga, SPICE — *A Guide to Circuit Simulation and Analysis Using* PSpice, 3rd Edition, Prentice-Hall, 1995.

[10]The 50 Ω transmitter is convenient for our purposes, but doesn't represent very well an actual modern Amateur Radio transmitter. Various models have been developed for both tube and transistor amplifiers and you may want to use one of them if you are interested in the effects of impedance mismatches on the power dissipation within your final amplifier. (See Note 17.)

[11]nec2c is a port of NEC2 from *FORTRAN* to *C* by Ray Anderson, WB6TPU. nec2c may be downloaded from **http://dg3aaf.no-ip.com:8080/sites/Numerical Electromagnet Code (NEC) Archives.htm**.

[12]C. Ray Wylie, *Advanced Engineering Mathematics*, Fourth Edition, McGraw-Hill, 1975, Section 10.5.

[13]The loss corresponding to half voltage is actually closer to 6.02 dB, but we'll neglect the difference here.

[14]We'll assume here that the resistive component of the antenna impedance is the radiation resistance. In reality, any real antenna will radiate only a portion of the energy accepted and will dissipate the rest in its internal resistance.

[15]Technical Personnel American Telephone and Telegraph Company, Bell Telephone Companies, and Bell Telephone Laboratories, *Telecommunications Transmission Engineering, Volume 1 — Principles*, Western Electric, 1974.

[16]A. E. Fitzgerald, ScD; David E. Higginbotham, SM; Arvin Grabel, ScD, *Basic Electrical Engineering*, Fourth Edition, McGraw-Hill, 1975, page 55.

[17]Mark Wilson, K1RO, Ed., *The ARRL Handbook for Radio Communications*, 2008, The American Radio Relay League, Inc, 2007, Chapter 18.

Maynard Wright, W6PAP, was first licensed in 1957 as WN6PAP. He holds an FCC General Radiotelephone Operator's License with Ship Radar Endorsement, is a Registered Professional Electrical Engineer in California, and is a Life Senior Member of IEEE. Maynard has been involved in the telecommunications industry for over 44 years. He has served as technical editor of several telecommunications standards and holds several patents. He is a Past Chairman of the Sacramento Section of IEEE. Maynard is 2008 President of the North Hills Radio Club in Sacramento, California.

Table 5

```
#! /usr/bin/octave -qf
# Print header
printf("\n\n     *** TRANSMISSION LINE CALCULATIONS ***\n");
# Specify transmission line data
f = 7.01;   #   FREQUENCY IN MHz
# 75 ohm line
d1 = 40;
a1 = 0.75;
v1 = 83;
Zo1 = 75;
# 600 ohm line
d2 = 200;
a2 = 0.05;
v2 = 92;
Zo2 = 600;
# Specify source (transmitter) and load (antenna)
VS = 2.0;
RS = 50;
Rt = 50;
Xt = 0;
# Convert inputs as required
#75 ohm line
a1 = a1 ./ 1e2;  # convert dB per 100 feet to dB per foot
a1 = 0.1151 .* a1;  # convert dB to nepers
c = 9.836e8;  # speed of light in feet per second
lambda = c ./ (1e6 .* f);  # wavelength of signal in vacuum
lambda1 = (v1 ./ 1e2) .* lambda; # adjust lambda for velocity
B1 = (2 .* pi) ./ lambda1; # calculate Beta
#600 ohm line
a2 = a2 ./ 1e2;  # convert dB per 100 feet to dB per foot
a2 = 0.1151 .* a2;  # convert dB to nepers
lambda2 = (v2 ./ 1e2) .* lambda; # adjust lambda for velocity
B2 = (2 .* pi) ./ lambda2; # calculate Beta
ZL = Rt .+ j .* Xt;  # calculate complex terminating impedance
# Calculate elements of T-equivalent network
# ZA = input series element
# ZB = output series element
# ZC = shunt element
ZA1 = Zo1 .* tanh((a1 .+ j .*B1) .* d1 ./ 2.);
ZB1 = ZA1;
ZC1 = Zo1 ./ sinh((a1 .+ j .*B1) .* d1);
ZA2 = Zo2 .* tanh((a2 .+ j .*B2) .* d2 ./ 2.);
ZB2 = ZA2;
ZC2 = Zo2 ./ sinh((a2 .+ j .*B2) .* d2);
# Prepare matrices for solution of currents
right_side = [RS .+ ZA1 .+ ZC1, -ZC1, 0; -ZC1, ZC1...
 .+ ZB1 .+ ZA2 .+ ZC2, -ZC2; 0, -ZC2, ZB2 .+ ZC2 .+ ZL];
left_side = [VS; 0; 0];
# Solve for currents
denom = det(right_side);
for m = 1: rows(right_side)
   numerator = right_side;
   for n = 1: rows(right_side)
      numerator(n, m) = left_side(n);
   endfor
   current(m) = det(numerator) / denom;
endfor
# Print out currents
for m = 1:rows(right_side)
   if imag(current(m)) < 0
      printf("\n\n  LOOP %d CURRENT = %8.5g - j%-8.5g\n", ...
      m, real(current(m)), -imag(current(m)));
   else
      printf("\n\n  LOOP %d CURRENT = %8.5g + j%-8.5g\n", ...
      m, real(current(m)), imag(current(m)));
   endif
printf("\n\n  LOOP %d CURRENT MAGNITUDE = %8.5g\n", ...
m, abs(current(m)));
endfor
# end program
```

J. R. Laughlin, KE5KSC

11918 Pompano Ln, Houston, TX 77072, johnel@earthlink.net

A Squelch Amplifier

Designed for use with an HP 8590A spectrum analyzer in "Zero Span" mode, you may find even more applications for this handy circuit.

I originally designed this circuit to provide squelch action and speaker-driving power for my HP 8590A spectrum analyzer. There are a lot of spectrum analyzers on the market and I am sure this project will be applicable for many of them. Later, I added some useful features that I have applied for several other projects. With the extra features incorporated into the circuit, you should find a considerable number of applications for it.

For use with the 8590A, study the explanations given in the *Operating Manual*, "Using the Analyzer as a Receiver in Zero Span." In my manual this section begins on page 2-35. This manual can be obtained from Manuals Plus.[1] It is available on-line also from Agilent technical support.[2]

In the Zero Span mode, the 8590 acts exactly as a radio receiver, with audio output along with a positive dc level, depending upon signal strength. One thing the manual does not mention is that receiving FM is very easy. You simply choose a resolution bandwidth suitable for the signal, about 10 kHZ for communication signals and around 100 — 200 kHz for TV and broadcast FM. The requirement here is that you tune the 8590A slightly off center frequency so as to use slope detection of the FM signal, to obtain good audio output.

Circuit Features

The circuit features are:

1) Adjustable squelch action.
2) U3 amplifier with the following capability:
   A) inverter or non inverter operation (when being used with the 8590)
   B) adjustable gain, with dc gain of 1 or equal to ac gain
   C) rectification, positive or negative, with filtering of output, if desired
   D) signal input, dc/ac , or ac only
   E) high-end rolloff adjustment
   F) input and output can be connected anywhere
3) Squelch dc voltage comparator has an ac voltage removal filter
4) Additional transistor switch (Q2), controlled by the squelch action, or other functions, can be NPN or PNP.
5) Audio attenuation approximately 70 dB, when squelch is closed
6) Audio power amp, U5
7) Low pass and high pass filtering of U5 audio input, if desired
8) Selectable use of + or – squelch voltage (using U3)
9) Internal audio gain setting (U2 and U5)
10) LED to indicate squelch action
11) Bipolar power supply, ac or dc powered, (for dc, using a +dc to –dc converter).

This photo shows the squelch amplifier connected to an HP 8590A spectrum analyzer, ready for operation.

Rear view of the squelch amplifier package.

[1]Notes appear on page 39.

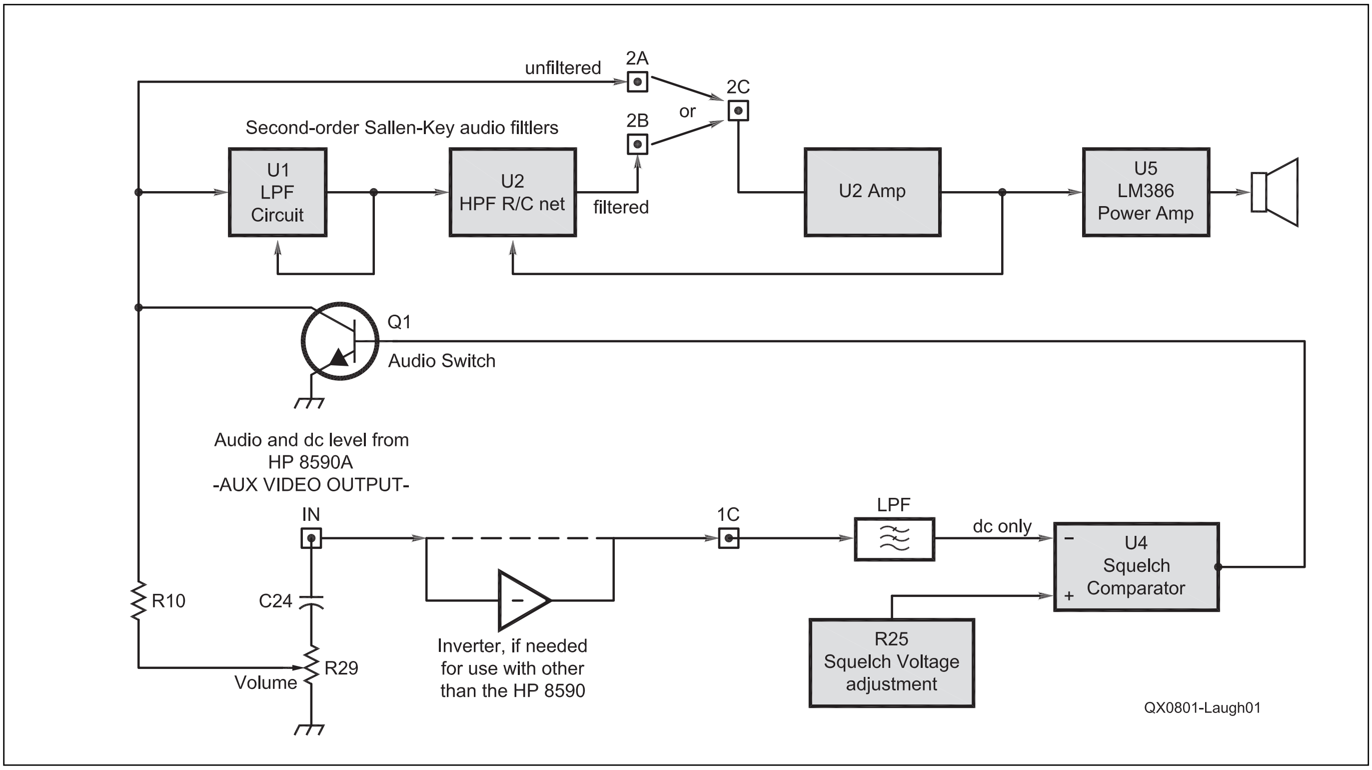


Figure 1 — This block diagram illustrates the basic operation of the squelch amplifier.

Block Diagram

Figure 1 shows the basic circuit configuration block diagram of my design when used with the 8590. The INPUT terminal of the squelch-amplifier circuit board is connected to the AUX. VIDEO OUTPUT connector on the rear panel of the 8590. Looking at R29, the volume control, you see that the ac audio level (dc is blocked by C24) is fed through R10 to the audio switch, Q1. If Q1 is on, it shorts the audio to ground. For Q1 to turn off and permit audio, the output of U4, the squelch comparator, must go negative. U4 is controlled by the changing dc level of the signal occurring at the top of C24.

This is fed to the inverting input of the comparator through the low pass filter, which removes all of the ac component so that only a clean dc level appears there. As mentioned, in zero span mode, the 8590A output dc level goes more positive as signal strength increases. This dictates that the squelch circuit must turn the audio on with this positive going level. As the dc level goes more positive than the squelch voltage setting at the non-inverting input, U4 output goes negative, turning off Q1. If this circuit is used with an output from some circuit (other than an HP 8590) which goes more negative with signal strength increase, you will need to use the inverter between C24 and terminal 1C for proper operation.

Audio from R10 is fed to two filters, U1 and U2 and to terminal 2A. Filtered audio from terminal 2B can be jumpered to 2C, or unfiltered audio from 2A can be used instead. The U2 amplifier drives the LM 386 speaker amplifier. This amplifier provides approximately 1 W of speaker audio. U2 is also part of the high-pass filter, when this filter is used.

The U3 circuit op-amp was originally incorporated to simply provide dc inversion of the input if the circuit is used with a receiver having a negative-going dc output, which is almost universal among most receiving devices. See Figure 2. It is connected as an inverter. As an afterthought, I incorporated some additional features into the U3 circuit, as described in the Circuit Features section of this article.

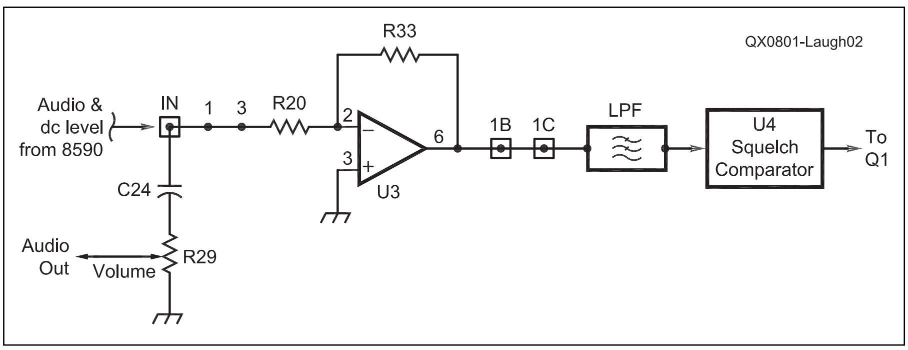


Figure 2 — You can use this circuit addition if your application requires a signal inverter.

Main Schematic

Let's study Figure 3, the main schematic. Note that the input terminal has two solder pads, no. 1 and no. 2 that can be solder-jumpered to each other, or either one can be jumpered to pad no. 3. If the volume control, R29, is not used, you may need to provide a dc return to ground by inserting R35 in its place. In that case, you will have to connect pads Vc and Vh together. For use with the HP 8590 you will need to jumper the Input pad to terminal 1C. This feeds the audio and dc into the low pass filter. The filter consists of three resistors — R32, 21, 22 — and three capacitors — C20, 21and 22. This three-pole filter bypasses the audio and permits the full level of the dc signal to be applied to the inverting

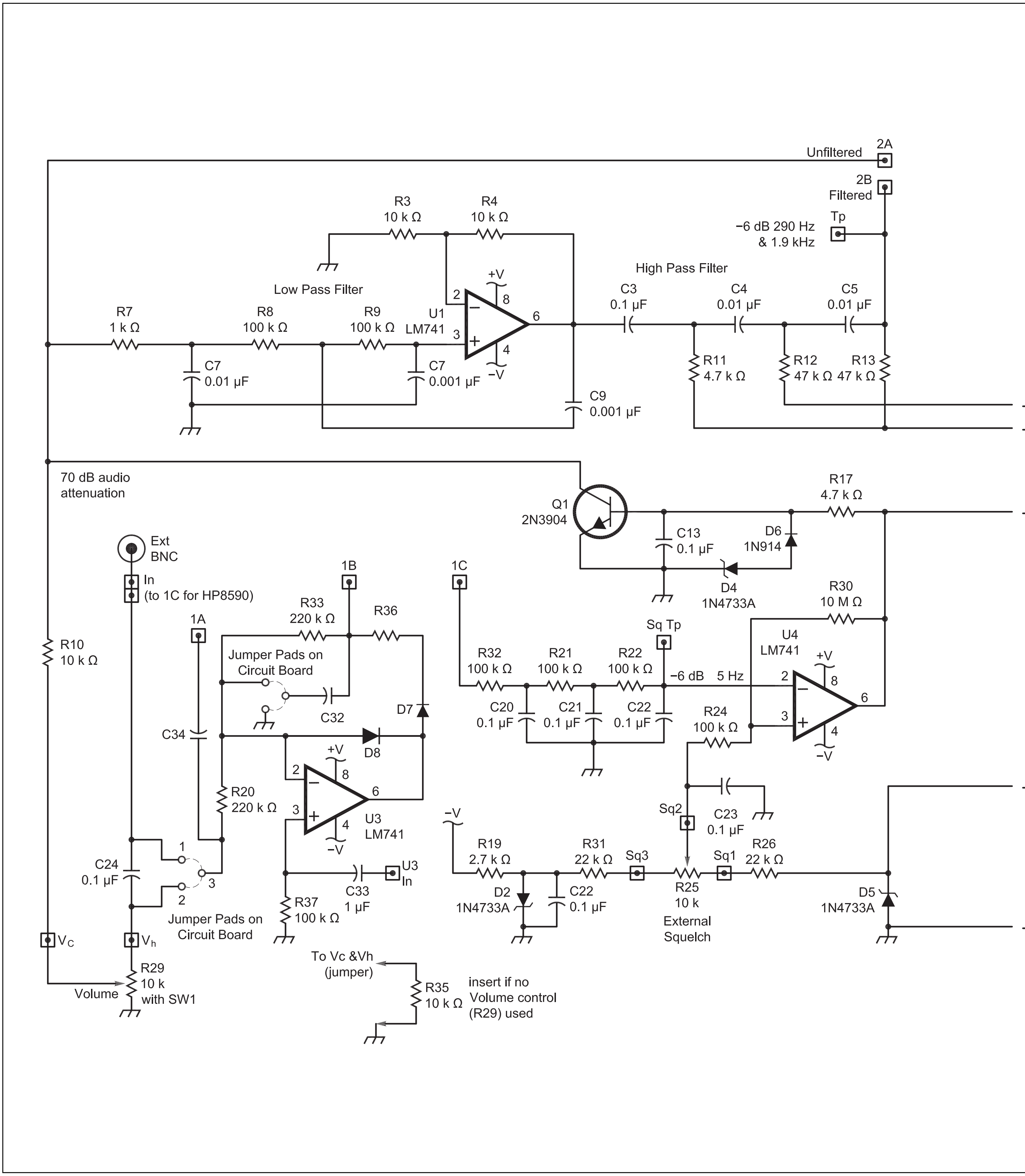


Figure 3—The schematic diagram of the squelch amplifier. The parts list is given in Table 1.

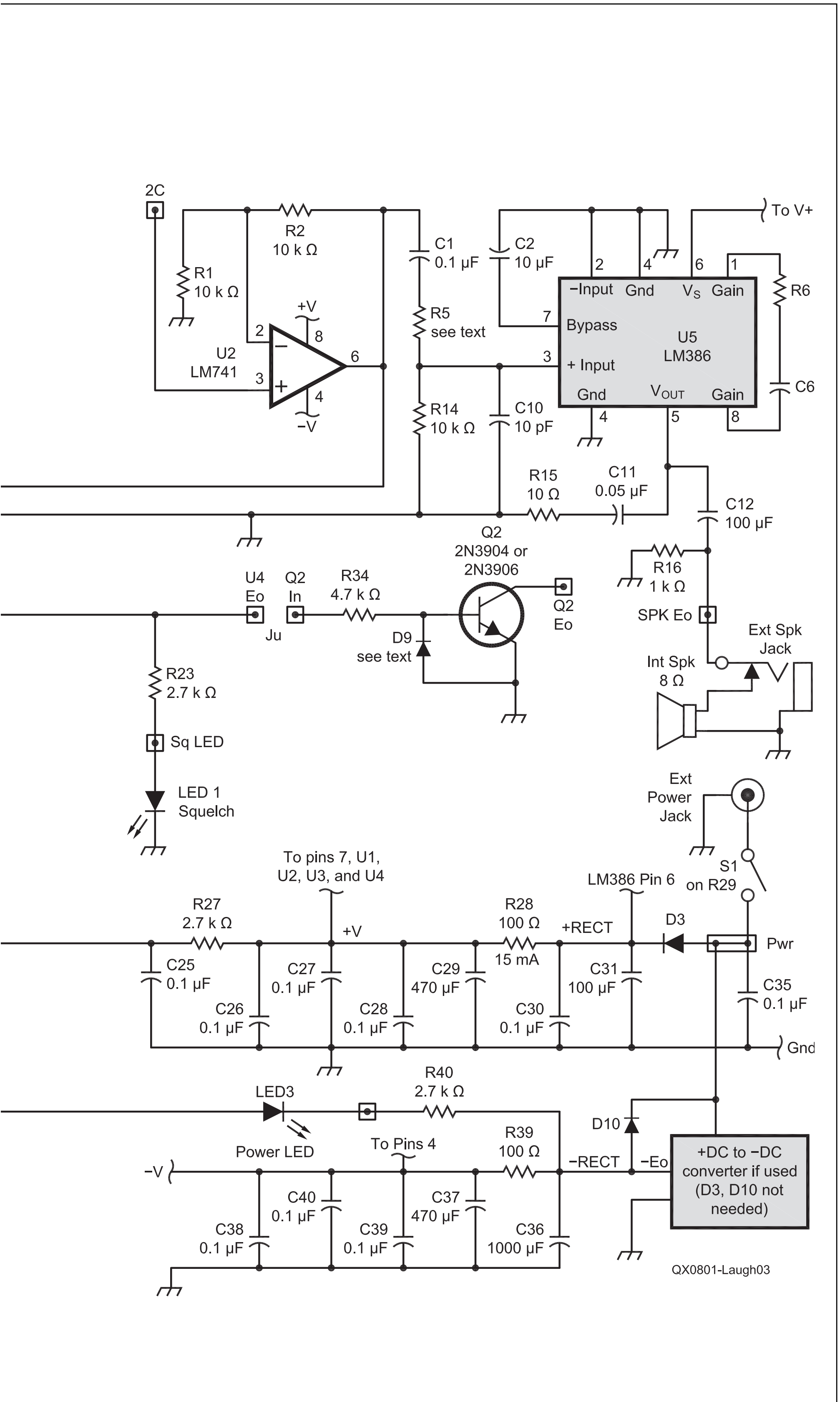


Table 1
Amplifier Parts List

Capacitors:	
No critical values.	
C1, 3, 13, 14, 15, 16, 19, 20, 21, 22, 23, 25, 26, 27, 28, 30, 35	0.1 µF
C2	10 µF
C4, 5, 7	0.01 µF
C6, if used	10 µF
C8, 9	0.001 µF
C10 to set hi roll-off, not used by author	
C11	0.05 µF
C12	100 µF
C17, 29	470 µF
C18, 31	1000 µF
C24, 33	Can be selected to set roll-off, I used 0.1 µF
C32	To set roll-off if used, not used here
Cabinet	Serpac 2019 or equivalent (Jameco)
Speaker	3 inch oval, 8 W
Power Jack	RadioShack 274-1563 or equiv. (M size)
Ext. Speaker Jack	RadioShack 274-248 closed circuit.
Audio Input Connector	RadioShack BNC 278-105
Knobs	To fit controls, your choice
Power Module	AC RadioShack 9 V ac 980-0900, (M size plug) dc RadioShack 10 V dc 273-1614
Power Connectors	RadioShack 274-226 and 274-236 or equiv. (0.062 inch)
Resistors: No values are critical, 10% okay.	
R1, 2, 3, 4, 10, 14, 25, 26, 29, 31,	10 kΩ
R5, 8, 9, 24, 37	100 kΩ
R6 (I did not use R6 in my application.)	0 to open, for gain adjust from 200 Ω to 20 Ω
R7, 16	1 kΩ
R11, 17, 19, 27, 34, 38	4.7 kΩ
R12,13	47 kΩ
R15	10 Ω
R18, 28	100 Ω
R21, 20, 22, 32, 33	220 kΩ
R23, 38 kΩ 2.7 kΩ	
R25	10 kΩ Pot
R29	10 kΩ Pot With Switch
R26, 31	
R30	10 MΩ
R36	see text
LEDs	
Green	RadioShack 276-271
Red	RadioShack 276-270
Machine Tooled Connectors	
From IC sockets, or other sockets using them, Jameco # 78642, 102201, 51626 and so on.	
Semiconductors	*All Available at RadioShack*
U1, 2, 3, 4	LM741
U5	LM386
D1, 3	1N4000 series
D2, 5, 4	1N4733A
D6, 7, 8, 9	1N914
Q1	2N3904
Q2	2N3904 OR 2N3906, see text

input (pin 2) of the comparator, U4. The 6 dB cutoff frequency is approximately 5 Hz.

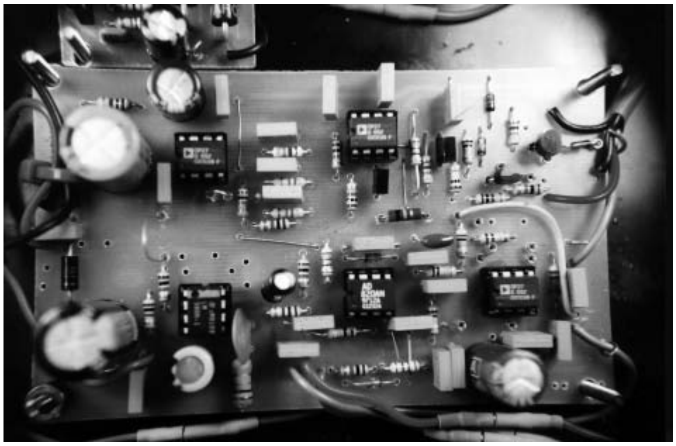
Pin 3 of U4 is adjusted (by R25) so that it is slightly more positive than pin 2 when the signal level is very low or absent. Pin 3 being more positive than pin 2 causes the output, pin 6 to go positive, driving the base of Q1 positive, and turning it on. When Q1 is on, it shorts any audio present at the top of R10 to ground. Thus, audio will not reach U5, the speaker driver. I measured the squelched audio attenuation at 70 dB.

When the dc level rises, pin 2 goes more positive than pin 3, which will drive pin 6 negative, and turn off Q1. This permits the audio to reach U5 and be heard. Negative voltage applied to the Q1 base is limited by diodes D4 and D6. This is below the reverse breakdown of the junction, and at a level high enough to prevent the audio from being distorted by the transistor when it is off. This would happen if negative peaks of audio could forward bias the collector/base junction of Q1. Without the reverse bias, about –0.7 V would do this, but with approximately –5.7 V applied to the base of Q1 by D4 and D6, a negative audio peak of approximately –6.5 V is required. For the audio level from the 8590, this much negative voltage is not needed, but it is available if ever needed.

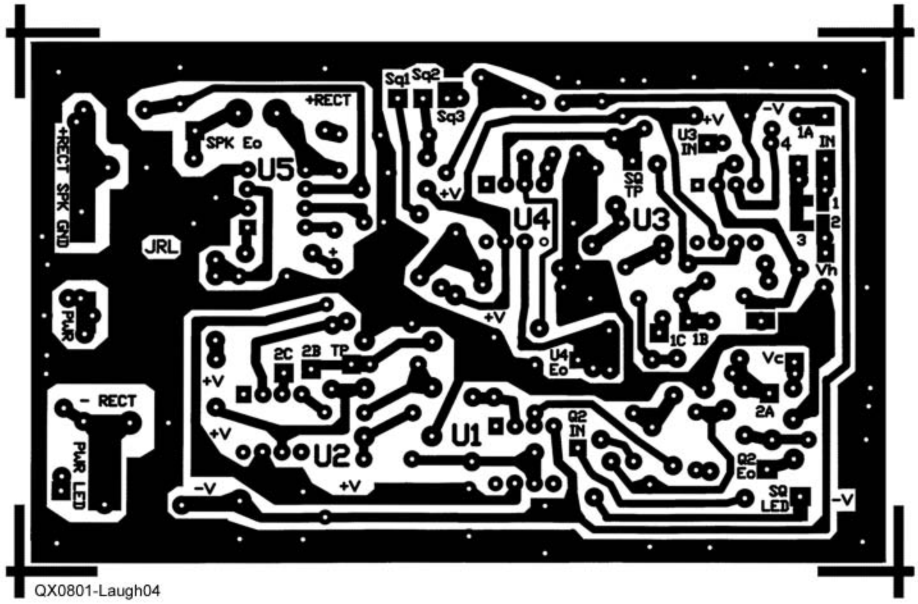
For the 8590, D4 and D6 can both be 1N914 diodes, with D4 then turned around so the diodes are connected cathode to anode. R25 has both positive and negative adjustment, to cover any situation that might be encountered. These voltages are regulated by D2 and D5.

In this case, the residual connection of U2 pin 6 to R12 is of no consequence. The gain of U2 may be increased by reducing the value of R1. Connecting terminal 2B to 2C installs the filters. The U2 output is fed to the U5 input through C1 and R5. Adjustment of the ratio between R5 and R14 will determine the amount of audio delivered to the input. For this application no reduction (R5 = 0 Ω) was about right, with the gain of U5 set at minimum (no connection between pin 1 and 8). When used as filters, it is not advisable to adjust the gain of U1 or U2, as this will alter the roll-off characteristics of the filter action.

Increasing the gain will cause peaks in the response just before roll-off begins. Decreasing the gain will have the opposite effect, resulting in less sharp roll-off characteristics and the over-all response characteristics will not be as "flat." Features such as those incorporated into U3 and the addition of Q2 were included for possible new ideas or uses that might come to mind. The Q2 input and output could be connected anywhere. The U3 input and output are also free to be connected anywhere. This is one advantage of using the terminal points to connect

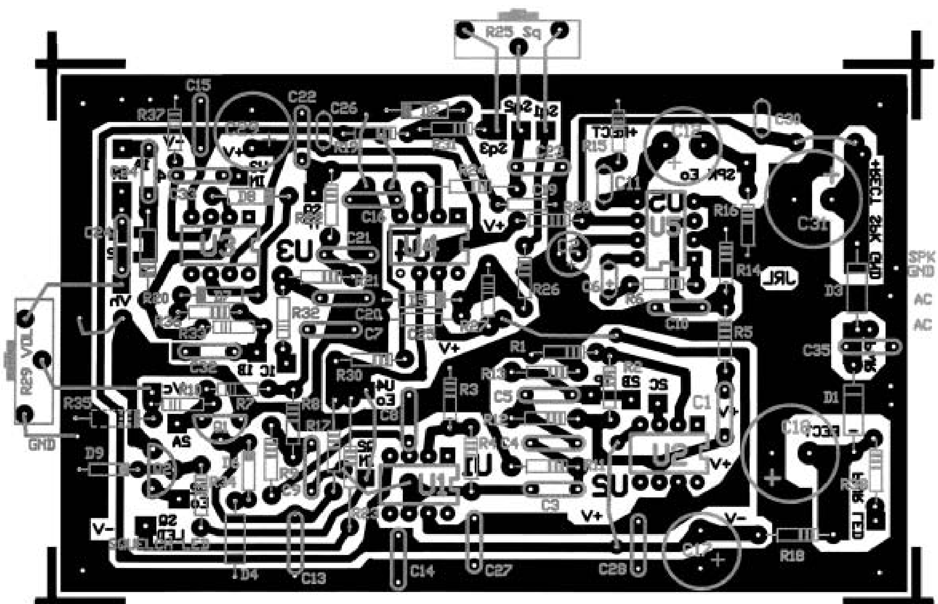


This photo shows the squelch amplifier circuit board. There are several of the machine tooled connectors around the IC in the lower right portion of the circuit board.



QX0801-Laugh04

Figure 4 — Here is a full-size circuit board pattern for the project.



QX0801-Laugh05

Figure 5 — The squelch amplifier parts-placement diagram.

sections of the circuit.

U3 and Q2 can be used together or individually. I might also add that for use with the HP 8590, the squelch option is not really that important but was included for versatility. This entire circuit has numerous possibilities for other applications, open to your imagination. It would be very interesting to hear your ideas on the subject!

Construction

Figure 4 is a circuit board etching pattern.[3] I suggest placing all the jumpers on the circuit board first. There are eight of them. All jumpers are easy to locate on the parts placement diagram of Figure 5 because I "crooked" them to make them show up better. There is a ninth jumper not shown if you use U3, but not the diodes and R36 with U3. This one goes from U3 pin 6 to terminal 1B, a very short jump.

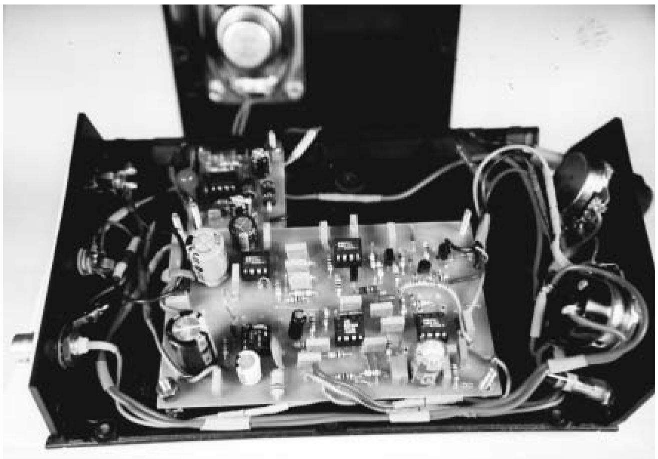
I am almost fanatical about installing any part that I might want to change or experiment with in the future so that it is "pluggable." A really nice way to do this is to install the little "machine tooled" solder tail connectors for these parts. See the parts list in Table 1. The little connectors can be freed from their fixture with a pair of side cutters. A 0.020 inch hole will accommodate the "tail" or if you prefer, a 0.050 inch hole will accommodate the "body" into the circuit board. Any pictures of either circuit board rather clearly show several of these mounted but not being used. They are nice, you will love them!

Another love of mine is to have all connecting wires to the circuit board fitted with pluggable connectors for easy removal and replacement on the board. For this I use the 0.062 inch crimp-type contacts with the wire crimp end removed to shorten them. These can clearly be seen in the photo of the inside of the case. See Figure 6 for more detail about how I prepare these connectors. It is a very good idea to use different, matching color wires and shrink tubing to help prevent plugging in the wrong wires. There are 12 connecting wires on the board. Of course it is quite possible to use a couple of six pin 0.062 inch power connectors to accommodate the wires (see Table 1).

I built my amplifier into a Serpac 2019 plastic case from Jameco. I used some ½ inch stand-offs to mount the circuit boards to the plastic case. I removed some of the molded stand-offs that come as part of the cabinet because they were in the way. Simply, but carefully, drill them down flush with the bottom of the case.

It is always important when mounting the BNC connector to drill and file the hole so the "flats" on the BNC threaded part are tightly fitted to the same shaped hole in the plastic panel. If you do not mount it this way it will soon be loose and twist when you try to place the cable connector onto it. Be careful when drilling holes for the front panel items if you use the Serpac 2019 cabinet, because space is pretty limited here. It is important that you carefully locate where you are going to drill and double check for physical interference of parts between themselves and the cabinet when the panel is in place. Loading and wiring the boards is straightforward.

Be careful as you locate the relative positions for the two boards in the case, if both are



Here is the squelch amplifier built into a Serpac 2019 plastic case.

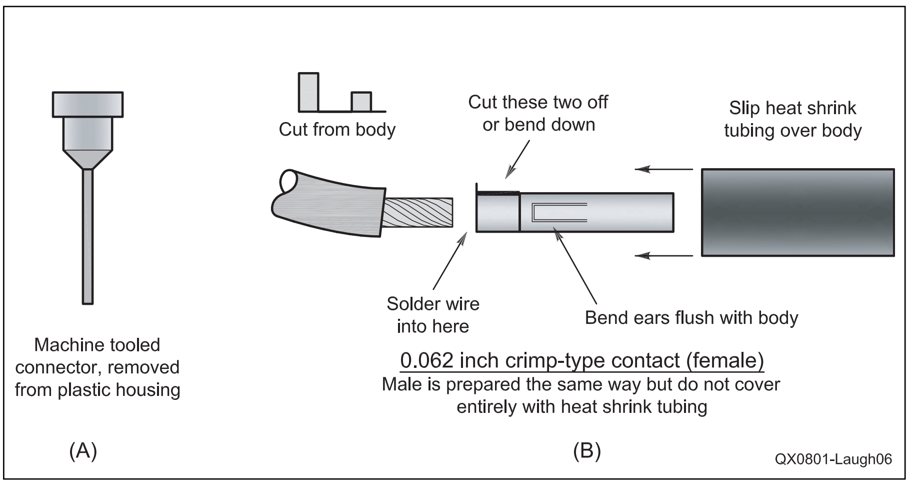


Figure 6 — Part A shows a machine tooled connector removed from its plastic housing, ready to be installed on the circuit board. Part B shows how the author uses 0.062 inch crimp-on contacts to install moveable jumper wires on the machine tooled connectors. See the parts list in Table 1 for Jameco part numbers.

Table 2
+DC to –DC Converter Parts List
All components available from RadioShack except the voltage regulator and Schottky diodes.

Capacitors	*All caps rated at 15 V or higher*
C1, 7, 8	1 µF
C2	10 µF
C3, 6	15 µF
C4	0.01 µF
C5	0.1 µF
Resistors	*All resistors ¼ W*
R1	15 kΩ
R2	22 kΩ
R3, 4	2.7 kΩ
R5	1 Ω
Diodes	
D1, 2 (Jameco MBR1100 part no. 312101.)	1N4000 or Schottky diodes for better output
D3	1N4000
Timer	TLC555/TLC555CP
Transistors	
Q1	2N3906
Q2	2N3904
Voltage Regulator:	
LM 79L05	Easy to find at any semiconductor sales outlet.

used, as the space is a bit tight horizontally. Watch this carefully! After mounting and wiring the components on the front and rear panels I decided to glue them to the bottom chassis. I think you will, also, because of their tendency to be "loose" any time you have the top off and are putting it back on. I simply reamed a 1 inch hole for the speaker outlet on the top and glued a porous cover onto it and mounted the speaker so as to be centered over the hole.

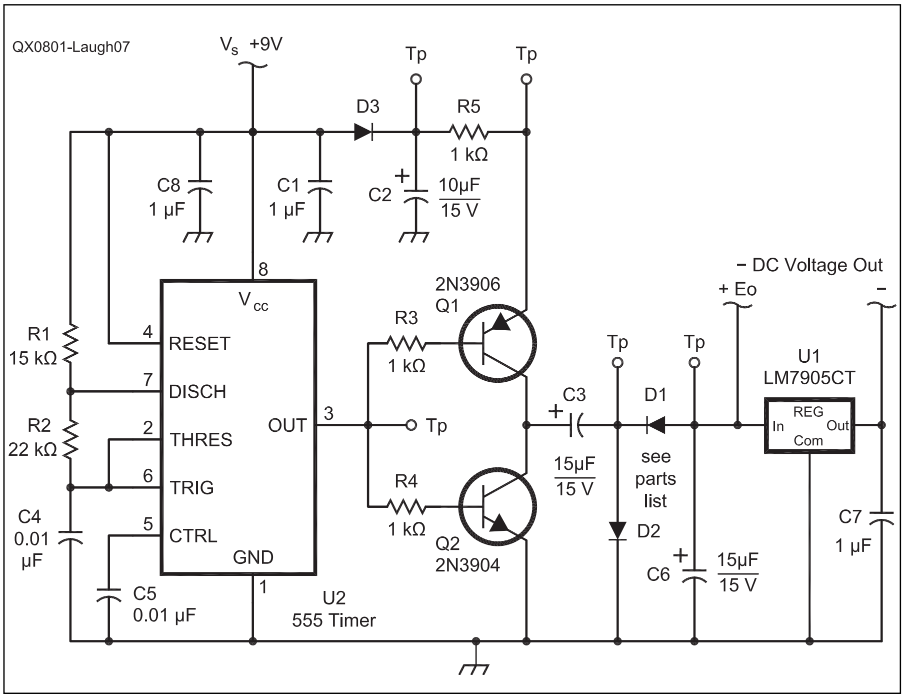


Figure 7 — This schematic diagram shows a +dc to –dc converter that can be used as part of the power supply for the squelch amplifier.

The +DC to –DC converter

This is a separate, small circuit board. This circuit is useful if you might not have, or do not wish to use, an ac power supply module. Figure 7 shows the schematic diagram. Table 2 gives the parts list. The 555 timer generates a constant square wave at approximately 1750 Hz with values shown. This causes Q1 and Q2 to switch on and off, alternately, providing a heavier duty square wave to the negative clamp consisting of D2 and C3. The peak value (minus diode drop) of this negative square-wave voltage charges C6 through D1 resulting in a negative dc voltage with only a small ripple value at C6. Figure 8 is a full-size circuit board patter for the +dc to –dc converter and Figure 9 is a parts placement diagram.

You can determine the current from the output circuit by measuring the voltage drop across the 1 kΩ resistor, R5. D3 was used to slightly reduce the voltage level to the emitter of Q1 compared to the voltage applied to the 555 timer. This ensures that Q1 will be turned OFF with the positive swing of the 555 output.

I have used this little circuit in other applications also, where a –dc voltage was needed but not available. The output current capability can be increased by using heavier duty transistors for Q1 and Q2. Table 3 gives the output voltage versus current using the small transistors. The voltage regulator on the output is optional. I did not use it. You can use

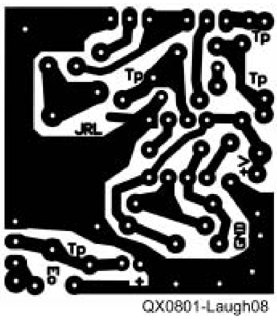
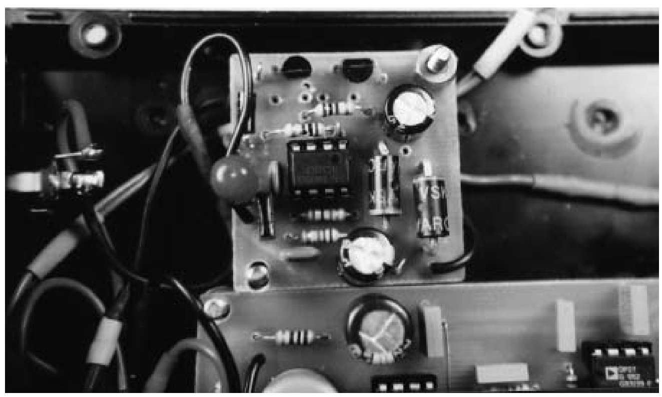


Figure 8 — A full-size circuit board pattern for the +dc to –dc converter.

Figure 9 — The +dc to –dc converter parts placement diagram is shown here.

The speaker is mounted into the top of the project



Here is a close-up view of the +dc to –dc converter.

Table 3
+DC to –DC Converter

E_O versus I_O (at D1 anode) using 2N3904 and 3906 transistors

E_{IN} = 10 V dc

I_O mA	E_O	
0	–9.24 V	D1 and D2 as 1 A Schottky diodes
5	8.8	For less diode loss than regular silicon
15	8.68	
20	8.51	
30	8.35	
40	8.8	
50	7.99	

heavy duty Schottky diodes for D1 and D2 to minimize their voltage drop. (See the parts list in Table 2.) No part values are critical. If this circuit is used with a dc power input, neither D3 or D10 are needed.

Notes

[1]Manuals Plus, 2002 Bethel Road, Suite 105, Finksburg, MD 21048; **www.manualsplus.com**; Phone 801-936-7000.

[2]**www.agilent.com**.

[3]PDF files of the circuit board etching patterns and parts placement diagrams are available for download from the ARRL Web site. Go to **www.arrl.org/qexfiles** and look for the file **1x08_Laughlin.zip**.

John Laughlin, KE5KSC, holds a BS in Electronic Engineering Technology from the University of Houston. He has been employed in the electronic industry for 25 years, and also was a college instructor for 25 years. He holds six CET certifications, along with a Radio-Telephone license, First Class, with radar endorsement.

John has published numerous articles in Popular Electronics, ham radio magazine, 73 Amateur Radio *and other electronics magazines. Other hobbies include walking across the Grand Canyon (three times so far, with more to come!), sailing, billiards and making fine custom billiard cues.* QEX

L.B. Cebik, W4RNL

1434 HIgh Mesa Dr, Knoxville, TN 37938-4443, cebik@cebik.com

Antenna Options

Horizontally Polarized Omni-Directional Antennas: Some Larger Choices

In the last episode, we explored some of the more compact choices for horizontally polarized omni-directional (HPOD) antennas. Each had some advantages and each had some limitations. In this episode, we shall continue the exploration by examining a few larger arrays using more than two independent elements, that is, elements fed in-phase. Stacking HPOD antennas is a familiar technique of increasing the gain in all directions, so we shall also spend a little time on that question.

In the first part of this safari, we employed uniform element sizes to all models. The elements in this episode will be a bit more diverse in diameter, ranging from ⅛ to ½ inch, since the models are based on prototypes. However, we shall retain the 144.5-MHz design frequency, because on 2 meters, the first MHz is the prime territory for horizontally polarized antennas. It is possible to adapt almost any of the designs for field or hilltop service using locally available materials.

The Big Wheel

An interesting and misunderstood semi-constant-current antenna is the *Big Wheel*, first published in *QST* in Sep 1961 (see "The Big Wheel on Two" by R. H. Mellen, W1IJD, and C. T. Milner, W1VFY, pp 42-45). Originally described as three 1-λ loops fed in phase, the antenna is actually a complete circle fed by parallel transmission lines at three equidistant points on the circumference. The outline and patterns appear in Figure 1. Between each transmission line, we find a current peak along the circumference, simulating the constant-current loop action. The model for this antenna uses a ⅜-inch diameter element with 600-Ω NEC-TL lines from a central feed point. The model has a radius of 17.3 inches for a circumference of 108.7 inches.

The maximum gain is about 7.3 dBi at 20 feet above average ground, with less than 0.4-dB variation around the horizon. Note the similarities between the elevation patterns of the constant-current loop (in the preceding episode) and the Big Wheel. However, the Big Wheel requires care in construction, because obtaining a usable feed point impedance for common coaxial cables involves interrelationships among the element diameter, the element radius and the characteristic impedance of the connecting transmission lines. The goal is to obtain a pre-match impedance of about 25 – *j* 25 Ω, so that the addition of a beta or hairpin transmission line stub provides the impedance transformation to 50 Ω. Once obtained, however, the SWR should be less than 2:1 across the entire 2-meter band with very good retention of azimuth pattern circularity.

The original Big Wheel employed an all-tubular construction method that allowed some element warping to arrive at the desired feed point impedance. Nevertheless, because the connections to the rim occur at high-impedance points, the parallel or nearly parallel lines to the hub perform an impedance transformation that demands somewhat finicky adjustment. The antenna remains very popular in Europe, but is perhaps nowadays less well known in the US. There may be arrays of three elements with equivalent performance, but simpler matching schemes.

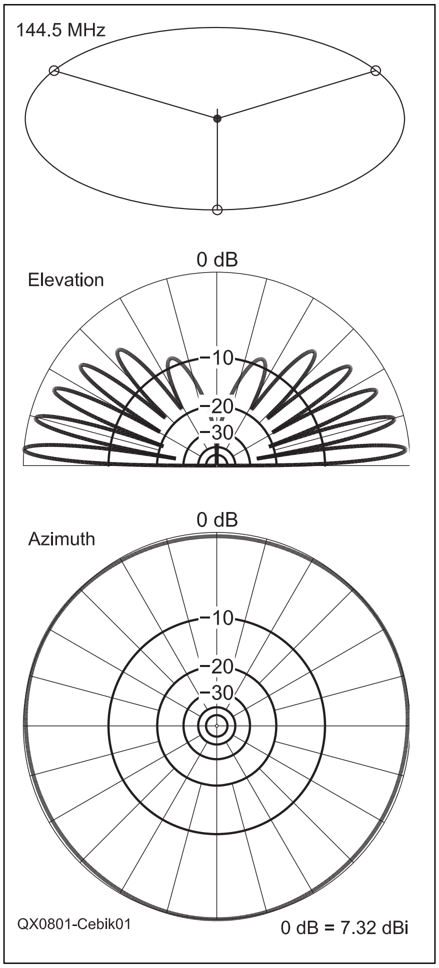


Figure 1 — Big Wheel: general outline and patterns at 20 feet above average ground.

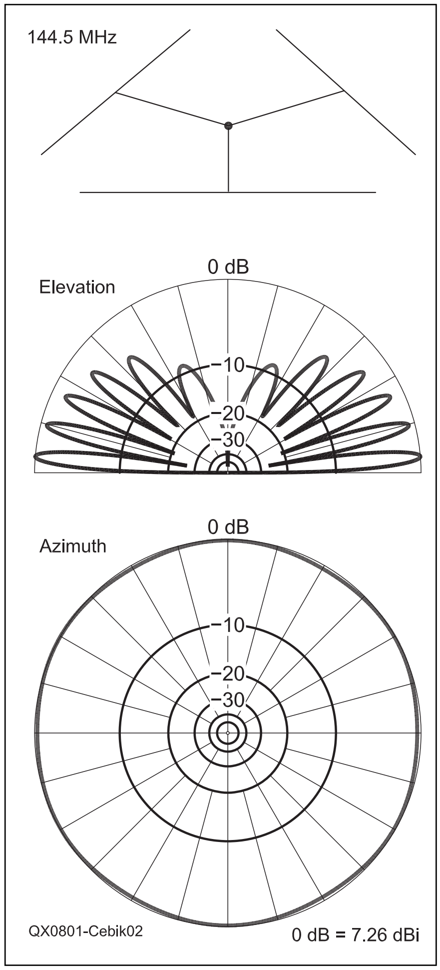


Figure 2 — Triangle of dipoles: general outline and patterns at 20 feet above average ground.

The Dipole Triangle and Wheel

One very straightforward array that yields a very circular horizontal azimuth pattern is a combination of three linear dipoles arranged in a triangle. Figure 2 shows the outline of such an array for 144.5 MHz using 0.5-inch-diameter elements. The success of the array in achieving a true HPOD far-field pattern rests on three factors: the distance of the dipole feed points from the assembly hub, the length of each dipole and the method used to match the triangle to a standard coaxial cable feed line.

The modeled antenna uses a feed-point-to-hub distance of 15.6 inches, with dipoles that are 34.7 inches long. The shortness of the dipoles (relative to the 40.8-inch half-wavelength at the design frequency) does not result simply from the element diameter. Even though the dipole end tips are about 9.6 inches apart, there is considerable interaction between any one dipole and its two mates. Varying the distance of the dipoles and their individual lengths also varies the distance between element tips. However, for this example, judicious juggling of the variables produced individual dipole feed point impedance values very close to 50 Ω. A 50-Ω cable to the hub thus performs essentially no impedance transformation and therefore does not restrict the available operating bandwidth of the array.

As the patterns in Figure 2 demonstrate, the triangle is capable of producing an almost identical set of patterns to those yielded by the Big Wheel. In fact, the modeled deviation from perfect circularity is about 0.1 dB. (The maximum gain for the Big Wheel seems superior by a small amount, but the average gain around the Big Wheel's azimuth pattern is closer to 7.15 dBi due to a slightly greater range between maximum and minimum gain values.) Perhaps the only two disadvantages of the triangle are physical: it requires more area than a circle, and the free ends may be more susceptible to local wind and weather.

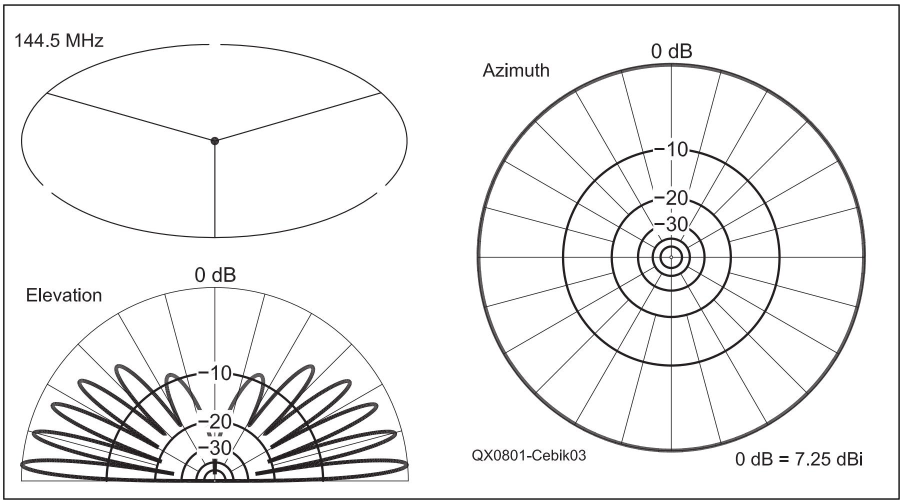


Figure 3 — Three-dipole wheel: general outline and patterns at 20 feet above average ground.

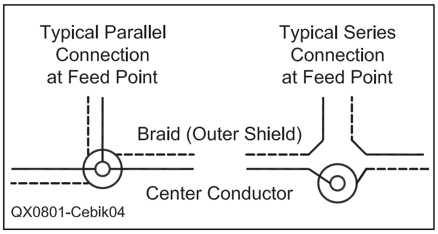


Figure 4 — Alternative parallel and series feed point connections for an assembly of three identical elements, such as used in the triangles and wheels.

We may curve the dipole elements and form a circular version of the same array. Figure 3 shows the outline of a three-dipole wheel. With 0.5 inch-diameter elements, the radius is 15.7 inches for the 144.5-MHz antenna. The resulting circumference is 98.6 inches. With dipole tip spacing of about 1.1 inches, each dipole occupies 31.7 inches of the circumference of the circle. Like the dipoles of the triangle, the dipoles are set for close to a 50-Ω feed point impedance to allow

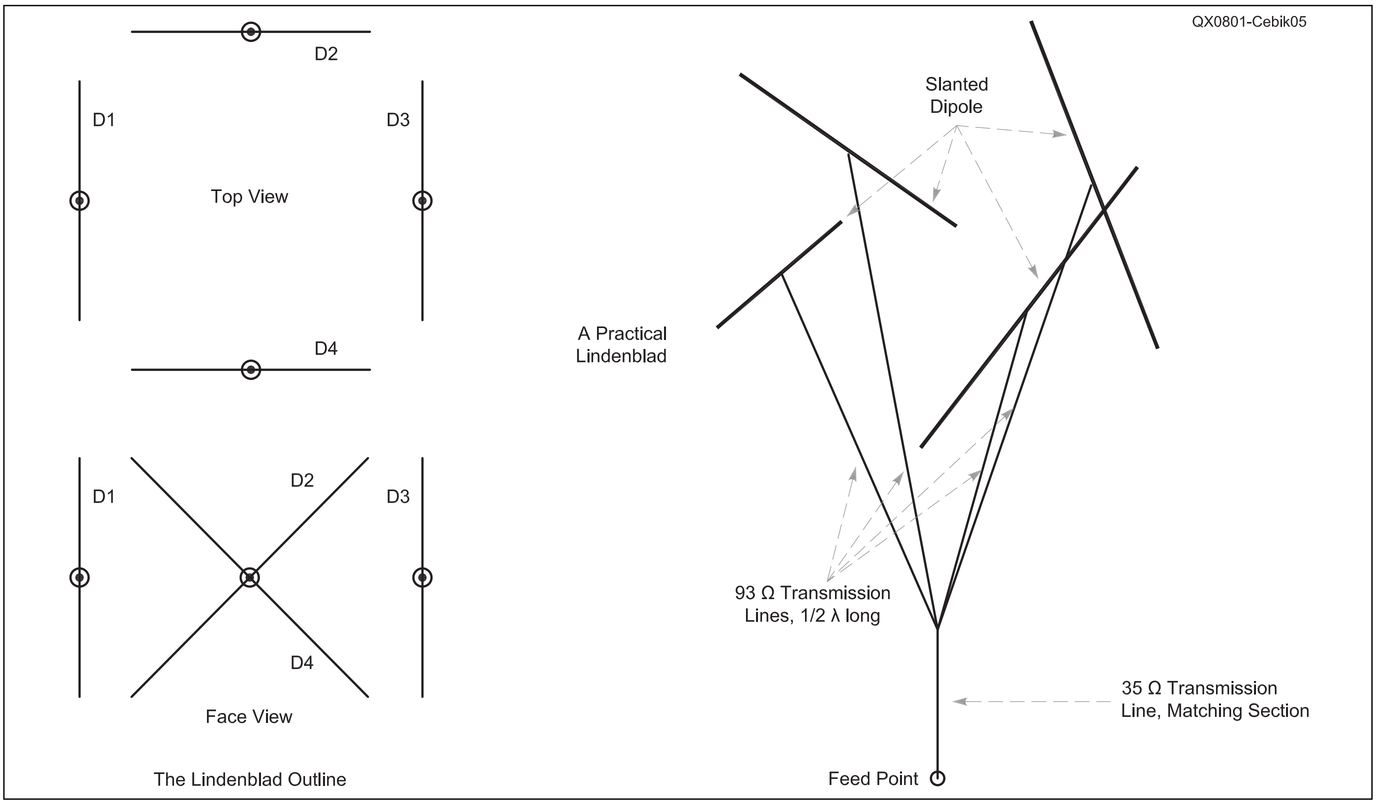


Figure 5 — The Lindenblad: outline and a practical Lindenblad array.

the use of 50-Ω lines to the hub without significant impedance transformation. Note the shorter lengths of the dipole compared to those in the triangle, largely due to both the curvature and the close coupling of dipole ends.

The three-dipole wheel requires somewhat more planning than the simple triangle, since we need three support arms. A non-conductive arm leading to a T at the end would allow the support to fit into the dipole ends to permit a bit of tip-spacing adjustment. At the feed point gaps, the dipoles will also require insulated plugs, which should be as small as feasible. If we add tube bending into the construction equation, the three-dipole wheel may be more complex to construct than the triangle, but the final product will form a closed circle and occupy considerably less area. Despite these differences, as shown in the elevation and azimuth patterns, the performance of the wheel is virtually identical to the performance of the triangle.

The remaining question involves matching the set of three 50-Ω impedance values at the hub to a 50-Ω transmission line. Figure 4 shows us two alternatives. A parallel connection of the lines will yield an impedance in the vicinity of 16 Ω to 17 Ω, a difficult value to match without employing a network. As well, any remnant or stray reactance will further complicate matching. Less often employed but perfectly usable under the circumstances of this array (three identical dipoles and connecting lines) is a series connection. (In fact, the models for these arrays use a series connection system, and the patterns shown are no different from those applying separate sources to each connecting line.) The resulting impedance will be in the vicinity of 150 Ω to 155 Ω, and any stray reactance will be too small to seriously affect the final result. A λ/4 section of 93-Ω RG-62 performs the final transformation of the impedance to about 55 Ω. Even tuned to the low end of 2 meters, the SWR only reaches 1.5:1 at the highest end of the band. The system has enough broadband capability to allow adjustment of the lowest SWR value anywhere in the band simply by lengthening or shortening the matching line slightly.

The Lindenblad

In one sense, our last option is not a true HPOD; that is, a horizontally polarized omni-directional antenna. The *Lindenblad* is actually a circularly polarized array with equal horizontal and vertical components within the point-to-point radiation pattern. Its origins lie in the pioneering work of N. E. Lindenblad, who first proposed the antenna design almost off-hand in a broad article on television transmitting antennas. (See N. E. Lindenblad, "Antennas and Transmission Lines at the Empire State Television Station," *Communications*, vol. 21, Apr 1941, pp 10-14 and 24-26.) After World War II, Brown and Woodward (who made numerous contributions to VHF and UHF antenna design) developed the idea in detail from Lindenblad's patent papers. (See G. H. Brown and O. M. Woodward, "Circularly Polarized Omnidirectional Antenna," *RCA Review*, vol. 8, Jun 1947, pp 259-269.) They envisioned possible aviation uses for the antenna. The overall goal for the antenna was omni-directional coverage in the X-Y plane (parallel to ground) with circular polarization.

Figure 5 shows two ways of looking at the Lindenblad. The left side shows face views and an overhead view of the array. We have four slanted dipoles, each equidistant from a center point. For circular polarization in the X-Y plane, that is, for equal horizontal and vertical components to the radiation pattern, the degree of dipole slant and the distance from the center point are interdependent. At a distance of about λ/4, the required slant angle is 45°. (There are refinements to the calculations. See Appendix 1, "Some Overlooked Antenna Basics for DX and Off-World Communications," *Proceedings of the 2006 Southeastern VHF Society Conference*, pp 250-252, for further information. See earlier portions of the article for information on the modified Lindenblad that may be more useful for lower angle satellite communications.)

The right side of the sketch shows the dipoles and their required interconnections for an effective array. Since the dipoles are fed in-phase and have individual feed point impedances close to 105 Ω in the arrangement shown, four RG-62 λ/2 lines provide a net parallel junction impedance of about 25 Ω. A λ/4 length of 35-Ω cable (usually composed of parallel sections of 70-Ω cable) completes the final impedance transformation to 50 Ω. The 0.125-inch diameter aluminum dipoles are each 40.1 inches long. Other feeding arrangements are certainly possible.

The single elevation pattern in Figure 6 shows a pattern quite unlike the earlier elevation patterns. The combination of vertical and horizontal lobes, even at a height as low as 20 feet, tends to fill the outline of the total radiation pattern in the plot. At the TO angle of 4.6°, the maximum gain is 6.15 dBi, with an overall gain variation of about 0.9 dB. Note that the total pattern is a bit squared off, mostly as a result of the combined horizontal components of the slanted dipoles. However, unlike the previous antennas that we have examined, the Lindenblad's point-to-point performance is not strictly proportional to the far-field radiation pattern. For example, notice the differing strengths of the horizontal and vertical components in the far-field pattern in the middle. The bottom pattern is a ground-wave plot using a distance of 1 mile and a receiving height of 20 feet. In this pattern, the vertical and horizontal components are nearly equal.

Since the Lindenblad maintains its pattern over a considerable bandwidth and since the antenna has a usable SWR bandwidth that is wider than the 2-meter amateur band, the array is suitable for use as an omni-directional antenna for both ends of the band. The models that accompany these notes contain a version of the antenna centered at 146 MHz for full-band coverage.[1] The major disadvantage is that each component of the array's radiation (and reception) pattern is weaker than most of the other antennas in our selection of options. A second

[1]Models for the antennas discussed in these notes are available at the ARRLWeb site. Go to **www.arrl.org/qexfiles** and look for **01x08_AO.zip**. All

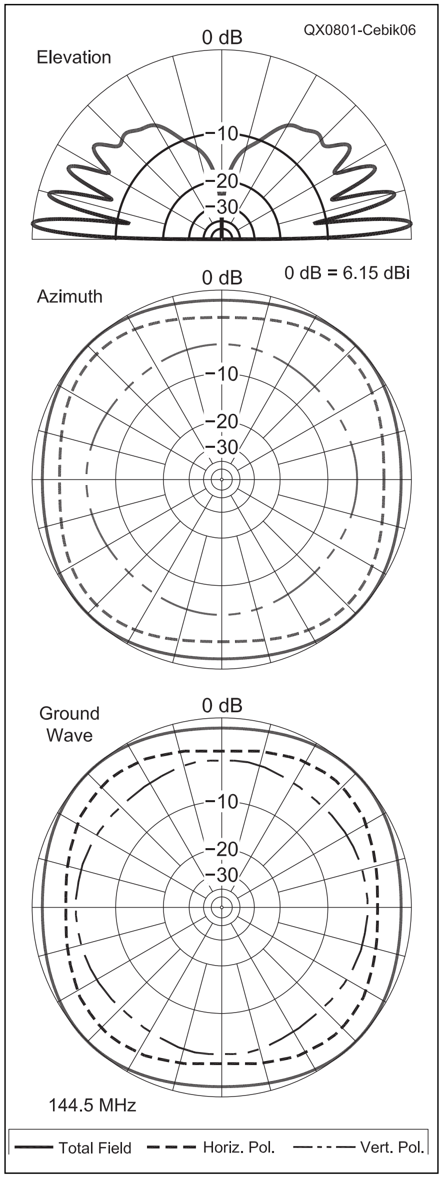


Figure 6 — Lindenblad patterns at 20 feet above average ground.

Table 1
Performance of three-dipole wheels in a stack of two with the lower antenna 20 feet (240 inches) above average ground.

Stack Spacing Inches	*λ*	*Max. Gain dBi*	*TO Angle degrees*	*Feed Z (x2) R ± j X Ω*
40.84	0.5	9.24	4.3	64.3 – *j* 23.1
51.05	0.625	9.98	4.2	54.1 – *j* 20.1
61.26	0.75	10.49	4.1	49.1 – *j* 14.4
71.47	0.875	10.64	4.1	48.6 – *j* 8.6
81.68	1.0	10.46	4.0	51.3 – *j* 4.9

Note: Feed Impedance values based on the use of the single-antenna λ/4 93-Ω match described for a single three-dipole wheel antenna (Z = 55.2 – *j* 8.2 Ω).

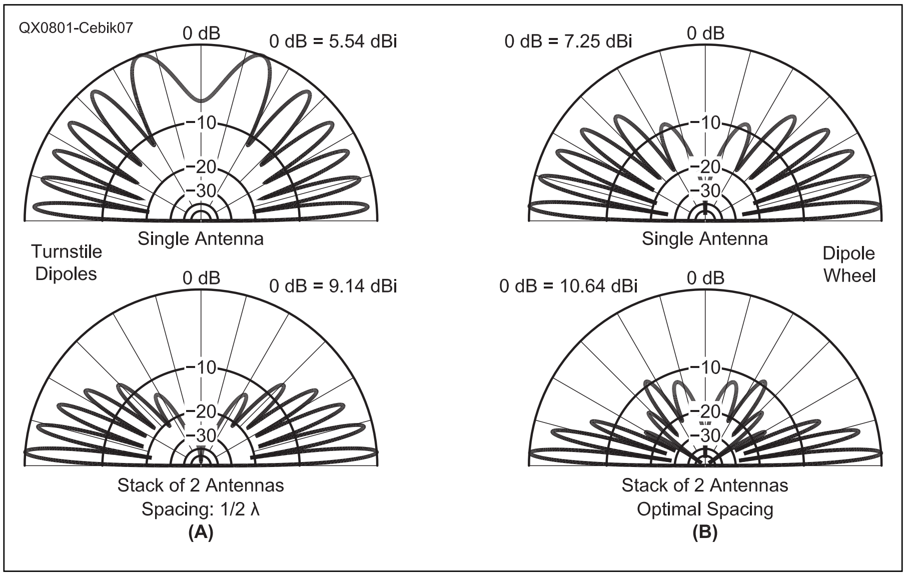


Figure 7 — When λ/2 stack spacing is optimal (with turnstile dipoles, for example) and when it is not (with the dipole wheel, for example, which shows maximum gain with 7λ/8 spacing).

Lindenblad between 0.5 λ and 1 λ above the first and turned 45° will not only improve performance (to a maximum gain of over 8 dBi), but will also circularize the pattern.

Stacking Larger Omni-Directional Arrays

One popular configuration for any of the larger omni-directional antennas is a vertical stack of two. Because we may be tempted to misapply some rules of thumb derived from other antenna types, we should devote a small space to this topic before we close. Like horizontal dipoles, the horizontally polarized arrays with circular patterns increase gain when we stack two such antennas an optimal distance apart and feed the two antennas in-phase. At the design frequency, 144.5 MHz, a wavelength is about 81.7 inches, which makes stacking fairly convenient.

We need to know what separation distance is optimal for these arrays. One popular separation value is a half wavelength. The temptation to use this value arises from and is applicable to special circumstances. On the left, in Figure 7, we find the elevation patterns of a single pair of dipoles arranged as a turnstile. Because radiation is stronger at high elevation angles, the use of λ/2 spacing in a stack of two pairs of turnstile dipoles is very productive. The use of λ/2 spacing with horizontal antennas tends to attenuate very high angle radiation and to make the energy available at lower angles. The maximum gain of a single turnstile pair is about 5.5 dBi (with a 20-foot height above average ground) in the lowest lobe. With λ/2 spacing, the lowest lobe shows better than 9-dBi gain when the lower turnstile is at 20 feet over the same type of ground.

On the right in Figure 7, we have elevation patterns for the three-dipole wheel. The single antenna pattern uses a 20-foot height. However, by nature, the three-dipole configuration does not show very high-angle energy levels. In fact, the lowest lobe has a gain of about 7.25 dBi. Therefore, the automatic use of a stacking space of λ/2 is not necessary and we are free to seek out the separation that yields maximum gain in the stack's lowest lobe, as pictured in the lower elevation plot. Table 1 provides modeled data for various stacking distances when the height of the lower three-dipole wheel is 20 feet above average ground.

Gain honors go to a stack spacing of ⅞ λ, and the elevation plot in Figure 7 uses this value. However, stacking distances between ¾ λ and 1 λ would not show any detectable differences in performance. Noticeable in the table is the fact that the two antennas interact so that the impedance values shift with each change in stacking height. Obtaining a closer impedance value to 50 Ω may require us to change the lengths of the 93-Ω matching sections.

The three-dipole wheel exhibited virtually no fluctuation in the gain around the perimeter. However, construction variations may create very small distortions in the pattern. The variations remain in a stack of two such antennas at any stacking distance. One way to smooth the azimuth pattern is to orient the spokes at a 60° offset between the upper and the lower antennas for a 3-element array and at a 45° offset for a 4-element array. The offset technique will smooth the azimuth patterns of stacks having up to 1 dB or greater fluctuations in gain around the horizon. For example, Figure 8 shows the azimuth pattern differences when we stack Lindenblads without and with a 45° offset. The squarish pattern of a single Lindenblad reappears in the aligned stack. However, with the offset, the pattern is perfectly circular.

The offset technique of creating circularly polarized azimuth patterns applies only to arrays using independent elements fed in-phase. Phase-fed elements, such as those in a turnstile, may suffer from the same treatment in a stack of two.

In-phase feeding of two HPOD antennas in a stack uses the same general rules and procedures employed in any stacking situation. For 50-Ω feed points, the most widely used procedure is to employ a pair of λ/4 75-Ω lines to a parallel junction. The 100-Ω transformed impedances together form a close match for the usual 50-Ω main feed line in most amateur installations.

However, the three-dipole HPODs that we have examined in these notes offer an alternative potential if the main feed line happens to be a length of surplus 75-Ω hard-line. Since the individual dipole impedances match the 50-Ω connecting lines, we may bring these lines all the way to a central position before we wire

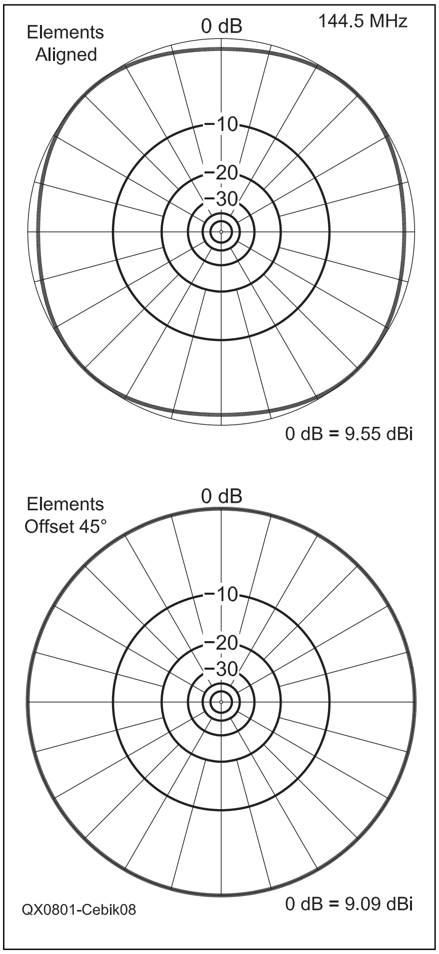


Figure 8 — An example of when rotational offsets between antennas in a stack can improve the circularity of an azimuth pattern: a Lindenblad two-stack with 1 λ separation and a bottom height of 20 feet above average ground, with the elements aligned and with them offset by 45°.

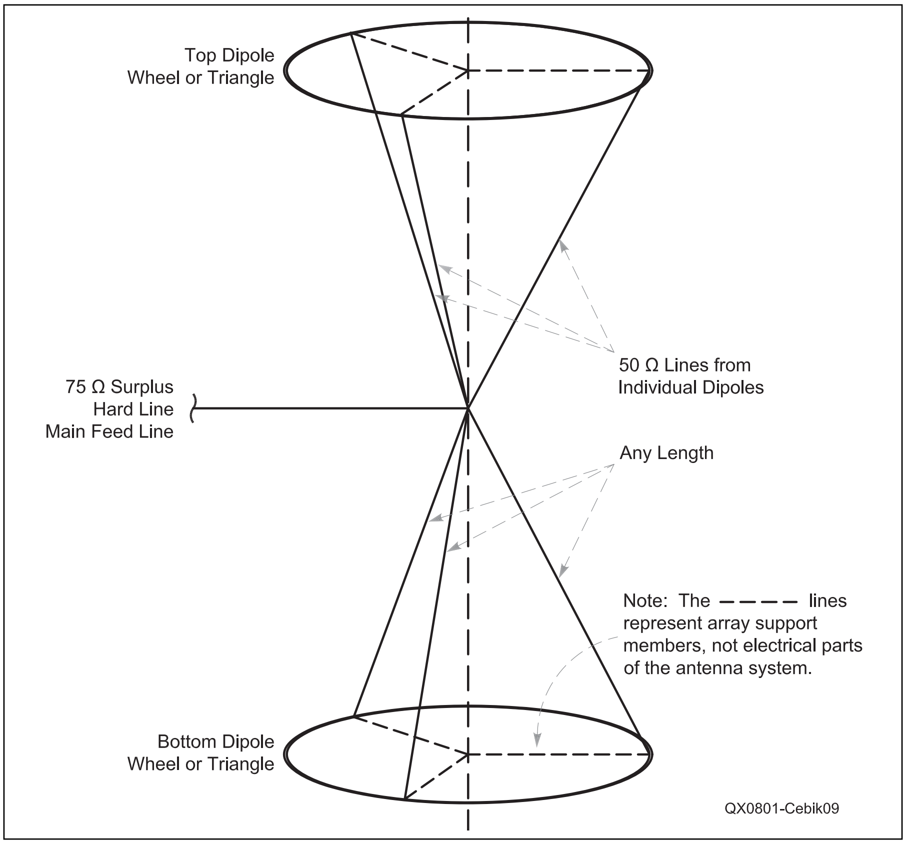


Figure 9 — An alternative system for feeding three-dipole arrays in-phase with a 75-Ω main feed line. It is equally applicable to the three-dipole triangle or the three-dipole wheel.

them in series. Figure 9 shows the general scheme. The two resulting 150-Ω impedance values in parallel provide a close match for the hard-line. A secondary function of the sketch in Figure 9 is to suggest an alternative method of routing the support elements for the three-dipole wheels in the stack. As a support system, the idea is less applicable to the dipole triangle.

Conclusion

In our voyage through the land of horizontally polarized omni-directional antennas, covering this and the preceding episode, we have encountered many schemes. One common feature of most of them is the presence of one or more features that calls for precise, if not downright finicky adjustment, with a resulting narrowing of the region in which we may obtain a nearly perfect circular azimuth pattern. The smaller the array, the more problematical some of the critical features become.

Of the lot, perhaps the larger three-dipole arrays are the least problematical. Once we obtain the proper physical dimensions, the matching becomes routine. As well, the larger arrays best maintain their circular azimuth patterns over a broad bandwidth. That feature may be less important during operation than it is during construction. With a broader design bandwidth, small variations in construction precision create fewer problems in the antenna's performance.

Nonetheless, our survey has unearthed many older and newer designs for the HPOD at VHF. One or more of the options should serve almost any need. QEX

Letters to the Editor

A Direct-Reading Reflection Coefficient and Power Meter (Nov/Dec 2007)

There are several errors in the my article, and several items need clarification. I apologize for the number of errors, and thank Bob Kopski, K3NHI, for pointing out most of them.

In the subtitle, delete the words "piece of," so it reads "This easy to construct test instrument combines power measurement and reflection coefficient measurement capabilities."

Bob Kopski's call is given incorrectly in paragraph 2 of the Introduction. It should be K3NHI.

In the second paragraph under Construction, the references should be to the AD8307AR and AD8307AN.

Figure 1 is from Zack Lau's *QST* article, with the original component numbering, which is not the same as the numbering of the corresponding components in Figure 2. In particular, R4 of Figure 1 corresponds to R1 of Figure 2.

The connector in the lower left corner of Figure 2 should be J2.

There are a number of corrections for Figure 3: The plus and minus signs in the two sections of U4 are reversed. [See Figure 1 in this Letters column for a corrected Figure 3. — *Ed.*] The wire from the wiper of R18 to pin 5 of U4 should not be connected to the 9 V dc wire it crosses. I mentioned in the text that the TLV2462CP op amp specified for U4 is a legacy of an earlier design, and could be replaced by a JFET input op amp such as the TL082 if the input to Pin 5 of U4(B) is shifted to Pin 6 of U3. Another reason to make this change is that the TLV2462 is rated for a maximum supply voltage of only 6 V dc, and it needs to operate from the 9 V dc supply. The recommended DVM module has a very high input impedance and an internal input bias of approximately 5.8 V dc, so it needs the instrumentation amplifier U3 and the buffer U4(A) as a driver. Some DVMs on the market can live with a low common-mode voltage, and could be connected directly between the R18 wiper and S1, eliminating the need for U3 and U4(A).

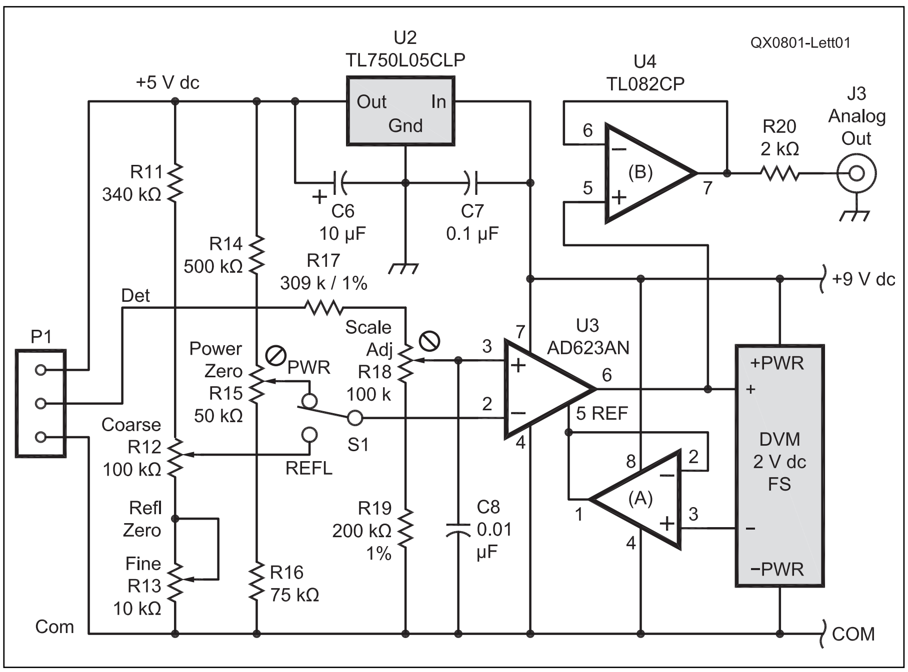
John B. Murphy, K6ILN, wrote with a question about the URL listed for the DVM module. Somehow the initials and URL got transposed. The module is from Marlin P. Jones & Associates, Inc, and the URL should be **www.mpja.com**.

— 73, Ralph Gaze, W1RHG, 35 Linda Terrace, Portsmouth, RI 02871; **rgaze@arrl.net**

The *Star-10* Transceiver (Nov/Dec 2007)

Dear Readers,

In the Nov/Dec 2007 issue we published Part 1 of an article by Cornell Drentea, KW7CD, "The Star-10 Transceiver." This article generated an avalanche of correspondence — far more than I have seen on



any other *QEX* article! I could easily fill the entire Letters section of several issues with correspondence about this one article. There was a significant amount of correspondence expressing praise for the design effort and the introduction to the design of the Star-10, with the outstanding list of specifications presented in Part 1.

While many readers expressed their pleasure at reading about this exhaustive design and construction effort, some readers took issue with statements in the article. Regrettably, this led to copies of correspondence to the *QEX* Editor being posted on various Web sites and e-mail reflectors long before we had an opportunity to review or respond to the comments, let alone publish any discussion in this issue. Tempers flared and any opportunity for a technical discussion quickly disintegrated.

QEX, as our subtitle declares, is "A Forum for Communications Experimenters." As such, we invite reasoned disagreements presented as technical discussions on the merits of a circuit, theory or idea. We will not, however, participate in attacks on character or debates about the extent of an author or reader's technical understanding.

One longtime *QEX* subscriber and professional radio engineer wrote to say:

"There is nothing 'so called' about software defined radios, and nothing old-fashioned about the quadrature modulator and demodulator direct conversion designs used in many SDRs. … (Even the most) anti-SDR person would be hard pressed to argue that the flexibility of SDRs is merely 'perceived.' Nor is there anything 'controversial' about their performance — performance can be measured."

I will acknowledge that as Editor, I probably should have removed the paragraph that made those statements. I honestly don't believe the author intended the comments as an attack on any person or group of radio enthusiasts, however, and no particular radio design or designer was mentioned in the original manuscript or the printed article.

Several points seemed to stand out as I reviewed the negative comments about this article. Some readers believe that:

• The article presented software defined radios in a negative light.

• The performance specifications made claims without any data to back them up.

There are several technical points here that may not be familiar to all *QEX* readers. Many articles could be written about the technical issues on both sides of these coins. We won't take the space here to delve into detail, but would invite reasoned technical presentations on the topics.

There are a number of ways to describe software defined radios, or SDR. Some will argue that SDR really only describes a radio that converts an RF signal to baseband and then uses various digital signal processing techniques to manipulate the signal in software routines to demodulate the signal. Others would include a broader category of radio that uses traditional conversion to intermediate frequency stages, with DSP stages to filter or otherwise process the signals.

The article in the Nov/Dec 2007 issue was only Part 1 of a planned 3 Part article. Parts 2 and 3 go into quite a bit more detail about the circuitry as well as the performance measurements presented in Part 1. I believe many of the questions about performance claims will be answered when the remaining two parts are published in *QEX*. We had planned to publish Part 2 in this issue, but as I mentioned in Empirical Outlook, we were unable to complete the schematic diagrams in time for this issue, so you will have to wait until the Mar/Apr 2008 issue to read Part 2. Our intention is to follow that with Part 3 in the May/June 2008 issue.

— 73, Larry Wolfgang, WR1B, QEX Editor,
lwolfgang@arrl.org

Dear *QEX* Editor:

Congratulations on one of the finest transceiver designs I've ever seen; the *Star-10* transceiver. Cornell Drentea has created a classic, perhaps the most optimized application of existing state-of-the art technology applied to superhet HF broadband design yet published. I only hope that commercial manufacturers of amateur HF transceivers take these specifications as a new performance standard.

I think this is a very well written, however abbreviated, article. I suspect that many (even advanced) readers will have difficulty with the spurious analysis. I understand Cornell's technique of spur analysis, but this might be news to many. Indeed, advanced spurious analysis, so critical in building broad-banded, multi-octave transceivers probably deserves a multi-part article unto itself!

To be fair, this is a very complex design, and the number of technologies converging at a very high level is a challenge for all but the most well-read radio designers. Indeed, I'm certain advanced radio designers worldwide will reference this article for decades to come. For my part, I needed clarification of several points from Cornell.

As far as dynamic range goes, it is clear that the *Star-10* is an outstanding radio. However, Cornell might have been a bit more particular with his specifications. Hopefully, this will become clearer in the subsequent parts. For instance, he specifies 500 Hz sensitivity at –136 dBm and quotes the NF at 15 dB. The noise power in 500 Hz at this NF would be –132 dBm, or 4 dB *higher* than the sensitivity specification. Cornell told me that the initial NF as calculated with his software and reported in Figure 3 was 15 dB for –132 dBm, but the NF with the preamp on is actually 11 dB, which should place the total noise at –136 dBm. That is the same as the sensitivity specification. Normally, sensitivity is taken to be at a 10 dB SINAD, or in this case, –126 dBm. Cornell also said that his IIP3 measurements were taken with the preamp on, so this –126 number can be taken as the basis of a crude DR (CP1/sensitivity), two-tone DR, and the composite DR. I hope Cornell clears this up in the subsequent installment, or at least in corrections and additions in *QEX*. Some simple classic calculations can be derived from Cornell's published specifications:

"Crude DR" — With a CP1 taken at about 10 dB below IIP3, we get +35 –(–126), or a whopping 161 dB. Obviously this must include the action of the AGC, but it must be pointed out that his AGC does not affect the front end, so the DR of the preamp + mixer is indeed 161 dB! The preamp is outstanding to reflect only an 11 dB NF with that kind of IIP3. These numbers are not, in and of themselves unusual. The mixer alone, with a 15 dB NF and IIP3 of +45 dBm has a 500 Hz DR of 157 dB (assuming a CP1 of +35 dBm (IIP3 – 10 dB)).

Two-tone DR is taken to be where the worst case third-order product equals 10 dB below the sensitivity. This is the point where the third-order product just begins to desensitize the receiver. IIP3 is +45 dBm, which implies a pair of input tones of at least –15 dBm. Therefore, the two-tone DR (defined for a 10 dB SNR) is –126 –(–15) or about 111 dB. Again, this is outstanding.

Composite dynamic range as defined in part one of the article is stated as 150 dB, which must be measured directly since it involves the AGC action of the receiver. This number is quite reasonable, given the above specifications and proper AGC design.

Although the preamp specification is outstanding, it is quite believable. A mixer with +45 dBm IIP3 is also outstanding, but quite possible using +27 dBm LOs. Indeed these better IIP3/LO ratios are possible using FET-rings and resonant gate circuits. (I was part of the Siliconix team that designed the Si8901 in 1983. Ed Oxner, KB6QJ, then optimized the performance. (See "The Real Si8901 Story," *QST* Sep 1993, p 79.) Perhaps an interesting aside is the fact that our boss at Siliconix was Rudy Severns, N6LF, former editor of *QEX*! (It's a small world!) Another ham, Van Brolini, NS6N, was also directly involved with the Si8901 design. In retrospect, that was quite a group!

Cornell surprised me when he told me that his mixer used a Calogic High-Speed DMOS Quad FET Analog Switch Array, the SD5000. The Si8901 is simply the SD5000 quad analog DMOS switch reconfigured into a FET-ring mixer by using a new metal mask on the die. Our Si8901 design was as simple as that: a new metal mask on the quad SD5000 die. He also indicated that to insure the high dynamic range in the LNA, he used an adaptation of a Norton amplifier utilizing two high dynamic range CP-650 FETs in push-pull. The preamplifier, as well as other front end and first IF functions use class A devices and 24 V dc to facilitate the high dynamic range. Therefore, Cornell used the best combination of LNA and mixer technology to optimize front end NF and DR. There is nothing miraculous about these specifications. Indeed, high-end HF radios for decades have used the Si8901 to achieve >100 dB DR.

The dynamic range specifications are quite impressive. The LO synthesizer performance, however, is what catches the more discerning eye. Maintaining 25 dB SNR with a –110 dBm signal and a –20 dBm interference signal 5 kHz removed from the desired –110 dBm signal reflects world class synthesizer and filter design. Here, the interference signal is only 5 dB below the upper IIP3 DR. The fact that this is a general coverage radio (implying a far greater challenge in synthesizer design) makes these specifications all that more impressive. According to Cornell, laboratory details of the test configurations and conditions to support these outstanding specifications backed up by pictures of the phase noise as well as blocking dynamic range using world class equipment similar to that used in the ARRL labs will be presented in Parts 2 and 3.

I am looking forward to the second installment. Indeed, this may be the finest amateur HF radio ever built and I'm certain it will generate considerable attention.

— Sincerely, Robert J. Zavrel Jr, W7SX, ARRL TA, Sr. RF Engineer; **w7sx@aol.com**

Dear *QEX* Editor:

Judging by some letters I received, there seems to be some misunderstanding about what composite linear dynamic range means.

As stated on page 9 in Part 1 of my article, composite linear dynamic range is defined as the ability to funnel a given large RF signal range into a final transducer without compressing the receiver and using multiple AGCs. This is a terminology used by some RF designers to characterize one particular kind of dynamic range.

Contrary to some beliefs, this kind of dynamic range is easy to measure. One simply measures the MDS, and with the AGCs on, finds the compression point of the entire system. AGCs keep the system out of compression to the highest point possible. This definition gives the 150 dB results as shown in Figure 3 of the article.

Of course, there are other kinds of dynamic range. They have been addressed separately in the specifications section of the article. Laboratory details of the test configurations, and conditions to support the specifications (which are actually the laboratory results), backed by pictures of the phase noise as well as blocking dynamic range results using equipment similar to that used in the ARRL laboratory, will be presented in Parts 2 and 3 of the article. All tests have been performed in the well equipped KG6NK laboratory.

Please note that phase noise plots (photographs) will show performance from 500 Hz to 20 kHz. This is because in CW work, we are interested in the very close in performance. A picture of all equipment used in the tests will be included in Part 3.

In addition, my MDS specification assumes a 0 dB SNR and not the more common 10 dB SINAD point. This is a matter of preference on my part, since with a good synthesizer, one can observe such signal well into the MDS, on a spectrum analyzer or an RMS voltmeter. As shown in Figure 3, the computer analysis gave an absolute MDS of –132 dBm or a 15 dB noise figure. In the final tests, however, the MDS was measured at –136 dBm after doing some tweaking. This resulted in an 11 dB noise figure (not 15 dB), which I forgot to change in reporting the specifications. I apologize for any inconvenience this might have caused.

The two tone dynamic range test was performed with 20 kHz tone spacing. A 2 kHz tone spacing test was not performed. A 5 kHz offset (or closer) blocking dynamic range (BDR) test was administered, however, and recorded using an HP-3561 dynamic signal analyzer. The results will be shown in Part 3 of the article.

I hope this answers the questions received. More detailed information will be included in the remainder of the article.

— 73, Cornell Drentea, KW7CD, 757 N Carribean Ave, Tucson, AZ 85748; **cdrentea@aol.com**

Dear *QEX* Editor,

Cornell Drentea's article on his *Star 10* transceiver is a beast to assimilate, but after scanning it three times it is starting to make sense. I think his front-panel label, "Made in USA," is a nice touch.

I doubt Cornell "irradiated" his aluminum chassis parts (see the caption under the lead photo, on page 3), but I'll bet he did *irridite* them (a chemical process for aluminum that produces a fingerprint-resistant frosty surface often found in professional-quality electronic equipment).

I'm looking forward to seeing Part 2.

— 73, Keith Kunde, K8KK, 8355 Dalepoint Rd, Independence, OH 44131; **kakunde@att.net**

A Low-Cost Atomic Frequency Standard (Nov/Dec 2007)

Dear Larry,

Here are two items that might be helpful to readers undertaking this project. One, the power supply specification for the rubidium oscillators is very tight on ripple (1 mV). The unit is particularly sensitive at 60 and 120 Hz. The lock circuit for the crystal oscillator frequency modulates the crystal oscillator very slightly, so that the applicable harmonic "straddles" the Rb absorption line. The FM frequency is about 60 Hz, making the locked signal 120 Hz/degree. The signal is weak and is therefore easily masked by power supply ripple. This shows up by the oscillator never finding lock. I discovered this the hard way.

The other item is that if the 10 MHz output is used to control microwave oscillators, the FM will become significant. The spec is "artful" in that it shows the spectral purity starting for frequencies above 120 Hz. One solution is to phase-lock the crystal oscillator with a narrow bandwidth to the Rb oscillator's output.

— 73, Donald Haselwood, W4DH, 18727 Crooked Ln, Lutz, FL 33548; **dhaselwood@verizon.net**

Thanks Donald,

I heard from a number of readers who have successfully completed this project. Your tips should prove helpful to anyone thinking about building this project or looking for ways to improve the performance.

— 73, Larry, WR1B; **lwolfgang@arrl.org**

QEX

In the Next Issue of QEX

Milton Cram, W8NUE, and George Heron, N2APB, present NUE-PSK31. This stand-alone, battery-operated digital modem includes transmit/receive text display and a graphic band-spectrum and tuning-indicator display. PSK31 without a separate computer. You will want to build one of these!

Cornell Drentea, KW7CD, continues his series about the design of his Star-10 transceiver. In this part, Cornell gives us a detailed look at the transmit/receive half octave filters, the 75 MHz first IF and the conversion to the second IF at 9 MHz. He also explains the operation of the master reference unit, which produces the stable signal that serves as a reference to phase lock all of the local oscillators.

Upcoming Conferences

12th Annual Southeastern VHF Society Conference

April 25-26, 2008
Orlando, Florida

Call for Papers

The Southeastern VHF Society is calling for the submission of papers and presentations for the upcoming 12th Annual Southeastern VHF Society Conference to be held April 25-26, 2008, in Orlando, Florida. Papers and presentations are solicited on both the technical and operational aspects of VHF, UHF and Microwave weak signal amateur radio. Some suggested areas of interest are:

- Transmitters
- Receivers
- Transverters
- RF Power Amplifiers
- RF Low Noise Preamplifiers
- Antennas
- Construction Projects
- Test Equipment and Station Accessories
- Station Design and Construction
- Contesting
- Roving
- DXpeditions
- EME
- Propagation (Sporadic E, Meteor Scatter, Troposphere Ducting, for example)
- Digital Modes (WSJT and others)
- Digital Signal Processing (DSP)
- Software Defined Radio (SDR)
- Amateur Satellites
- Amateur Television

In general, papers and presentations on topics not related to weak signal operation, such as FM repeaters and packet radio will not be accepted, but exceptions may be made if the topic is related to weak signal operation. For example, a paper or presentation on the use of APRS to track rovers during contests would be considered.

The deadline for the submission of papers and presentations is February 29, 2008. All submissions should be in Microsoft *Word* (.doc) or, alternatively, Adobe *Acrobat* (.pdf) file formats. Pages are 8½ by 11 inches, with a 1 inch margin on the bottom and ¾ inch margin on the other three sides. All text, drawings and photos, should be black and white only (no color). Please indicate when you submit your paper or presentation if you plan to attend the conference and present there, or if you are submitting just for publication. Papers and presentations will be published in a bound *Proceedings* book by the ARRL. Send all questions, comments and submissions to the program chair, Steve Kostro, N2CEI at **svhf2008@downeastmicrowave.com**. For further information about the conference please go to **www.svhfs.org** or **www.flwss.net**.

42nd Annual Central States VHF Society Conference

July 24-26, 2008
Wichita, KS

The Central States VHF Society is soliciting papers, presentations, and Poster / tabletop displays for the 42nd Annual CSVHFS Conference to be held in Wichita, Kansas on July 24-26, 2008. Papers, presentations, and posters on all aspects of weak-signal VHF and above amateur radio are requested. You do not need to attend the conference, nor present your paper to have it published in the proceedings. Posters will be displayed during the conference.

Submission Deadlines:

Proceedings and for presentations to be delivered at the conference: May 15, 2008.

Poster for display: July 1, 2008, for notifying us you will have a poster to be displayed at the conference. (Bring your poster with you on July 24!)

Please see the Web site at **www.csvhfs.org** for more information.

KENWOOD

Listen to the Future

TS-480

The Perfect Remote Base Transceiver

Straight Out of the Box!

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

KENWOOD U.S.A. CORPORATION
Communications Sector Headquarters
3975 Johns Creek Court, Suite 300, Suwanee, GA 30024-1265
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