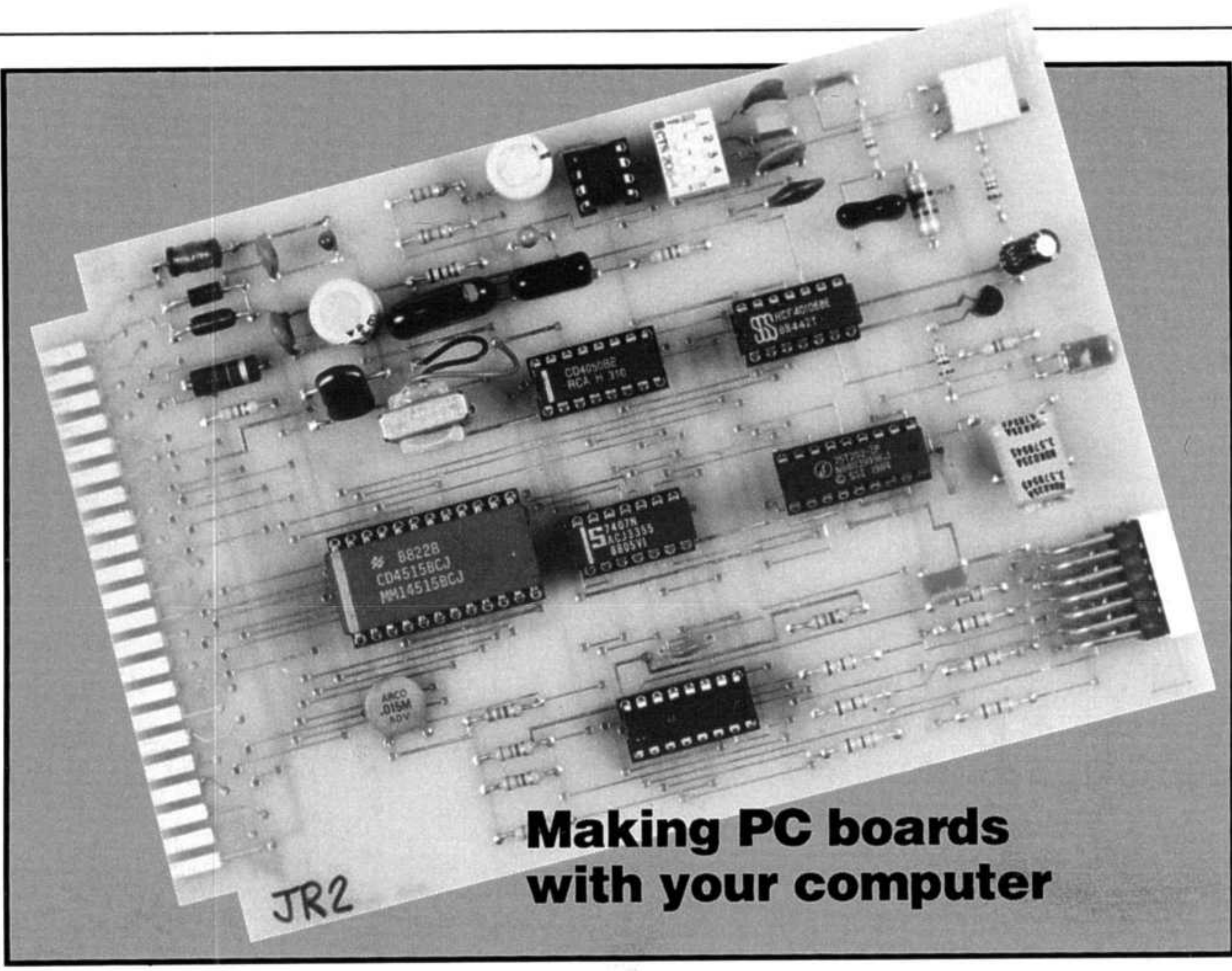
QEX 126 with Gateway

$1.75

ARRL Experimenters' Exchange

AUGUST 1992



Making PC boards with your computer

QEX: The ARRL
Experimenters' Exchange
American Radio Relay League
225 Main Street
Newington, CT USA 06111

Non-Profit Org.
US Postage
PAID
Hartford, CT
Permit No. 2929

QEX (ISSN: 0886-8093) is published monthly by the American Radio Relay League, Newington, CT USA.

David Sumner, K1ZZ
Publisher

Jon Bloom, KE3Z
Editor

Lori Weinberg
Assistant Editor

Mark Forbes, KC9C
Geoffrey H. Krauss, WA2GFP
Stan Horzepa, WA1LOU
Contributing Editors

Production Department

Mark J. Wilson, AA2Z
Publications Manager

Michelle Bloom, WB1ENT
Production Supervisor

Sue Fagan
Graphic Design Supervisor

Dianna Roy
Senior Production Assistant

Circulation Department

Debra Jahnke, *Manager*
Kathy Fay, N1GZO, *Deputy Manager*
Cathy Stepina, *QEX Circulation*

Offices

225 Main St, Newington, CT 06111 USA
Telephone: 203-666-1541
Telex: 650215-5052 MCI
FAX: 203-665-7531 (24 hour direct line)
Electronic Mail: MCI MAIL ID: 215-5052
(user name ARRL)
Telemail address: ARRL
Internet:arrl@mcimail.com

Subscription rate for 12 issues:
In the US by Third Class Mail: ARRL Member $12, nonmember $24;

US, Canada and Mexico by First Class Mail:
ARRL Member $25, nonmember $37;

Elsewhere by Airmail:
ARRL Member $48 nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation.

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

TABLE OF CONTENTS

COLUMNS

AUGUST 1992 QEX ADVERTISING INDEX

THE AMERICAN RADIO RELAY LEAGUE

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interests in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of 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 two 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 correspondence should be addressed to the administrative headquarters at 225 Main Street, Newington, CT 06111 USA.

Telephone: 203-666-1541 Telex: 650215-5052 MCI.
MCI MAIL (electronic mail system) ID: 215-5052
FAX: 203-665-7531 (24-hour direct line)

Canadian membership inquiries and correspondence should be directed to CRRL Headquarters, Box 7009, Station E, London, ON N5Y 4J9, tel 519-660-1200.

Officers

President: GEORGE S. WILSON III, W4OYI
1649 Griffith Ave, Owensboro, KY 42301

Executive Vice President: DAVID SUMNER, K1ZZ

Purpose of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
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 correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and doubled spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook*. Photos should be glossy, black and white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX*.

Any opinions expressed in QEX are those of the authors, not necessarily those of the editor or the League. While we attempt to ensure that all articles are technically valid, authors are expected to defend their own material. Products mentioned in the text are included for your information; no endorsement is implied. The information is believed to be correct, but readers are cautioned to verify availability of the product before sending money to the vendor.

Empirically Speaking...

Change and Continuity at *QEX*

In the life of any magazine there is change, and change has come to *QEX*. Founding *QEX* Editor Paul Rinaldo, W4RI, is leaving Newington to return to Washington, DC, where he will continue to work for ARRL, concentrating on those all-important regulatory and technical issues. He leaves behind him a thriving *QEX* magazine, a publication that Paul started more than ten years ago. In taking his place as *QEX* Editor, I hope to build on what Paul has accomplished.

Whenever such a change takes place it provides an opportunity to reassess directions. We have done exactly that with regard to *QEX*. While the basic direction of *QEX* remains sound, some adjustments are in order. Principally, these adjustments are intended to sharpen the focus of *QEX* in order to better serve our readers, the Amateur Radio experimenter community.

One change we are making is to concentrate more on practical information. While we intend to continue publishing overviews of state-of-the-art technology, we feel that practical development of such technology by amateurs depends on the availability of component and circuit information as well as system-level treatments. Over the coming months you will see these changes become apparent. You'll see one change starting with this issue of *QEX*: a new bimonthly column, "RF." This column is conducted by Zack Lau, KH6CP, of the ARRL laboratory staff. Zack is one of the premier RF experimenters in Amateur Radio, having designed and published circuits and projects for MF, HF, VHF, UHF and microwave bands. "RF" will provide a place in *QEX* for RF experimenters to hold a dialogue. To be of most benefit, "RF" will need to hear from those of you who are developing RF circuits and systems. Sharing the knowledge you gain from your efforts is crucial to advancing the technical capabilities of Amateur Radio.

We plan other changes for the months ahead, some subtle, some not. Some of these changes are in response to reader feedback gathered by our reader survey of 1990; some are in response to individual comments we've received. We hope the evolving *QEX* will respond to your needs better with each issue. This is most likely to happen if you let us know what works and what doesn't in *QEX*. What could we be doing better, and what should be left alone? What areas aren't we covering that desperately need coverage, and where should we be soliciting articles that cover those areas? Answers to these questions are best formulated by the people working in those fields: you.

We want...no, we need to hear from you. If you have electronic mail (email) capabilities, you may be able to communicate with us that way. You'll note an Internet email address at the end of this editorial. If you can send mail to Internet, please feel free to contact me via that address. Some of our columnists and authors also can be contacted via email, and we encourage you to do just that. If you don't have email capability, you can always use the "regular" mail. We're happy to hear from you via whatever medium is most convenient for you. Please help us make *QEX* a magazine that helps you experiment with and learn about Amateur Radio technology with every issue. — *KE3Z, email: jbloom@arrl.org (Internet)*

By Lawrence S. Higgins, W5QMU
2522 Old Hickory Trail
San Antonio, TX 78230

So you want to make PC boards with your computer!?

About four years ago, *Ham Radio* published a neat article on CAD programs for home computers.[1] At the time it had become evident that wire wrapping gets old quickly, except for one-off prototypes. Sure, the circuit is reliable, once wrapped, but the pain of wrapping and trouble shooting the rat's nest resulting is real.

Enter the IBM PC and clones, and several programs available for nontrivial prices, but perhaps within reach. If you are as scared of taping boards over a light box as I am, read on.

The Programs

In addition to the article (note 1), I looked over demos from *Tango*[2], *Orcad*[3] and *SmartWork*[4] (used at the ARRL Lab[5]). But the Advanced Micro-Systems *PC-SCHEME* and *PC-PRO*[6] combo was the only one I felt a ham budget could afford. I soon had the programs, supplied on 5¼ inch floppies, in hand. (The software runs on any IBM PC or compatible computer.) I have a PC clone with 640 k memory, but only 286 k is required. A hard disk is helpful for library recall, but is not required. EGA color would have been nice, but again, I didn't feel I could afford the extra $500; AMS said I could run it on Hercules graphics—something that is not always supported by the more expensive programs. (This article will use file references from the AMS program. However, there should be no problem in translating my suggestions to similar programs.)

You will need a mouse and its associated software installed on your computer. The programs are accompanied by several library disks containing schematic and printed-circuit-board symbols or "footprints." I read the excellent manual and ran the demos (cost $10) several times to familiarize myself with the mechanics of PC layout. I soon learned that you can lay out a PC board directly (using *PC-PRO*), but for anything beyond a very small board, schematic capture is the way to go.

What is Schematic Capture?

Well, first you draw your schematic on an old envelope or whatever. Then you transfer it to the computer screen, viz:

1. Decide, if you can, on drawing size.
2. Start with a blank screen. You will see a bunch of dots which represent the drawing grid.
3. Call up the components to the screen and place as you would on the back of the envelope sketch.
4. Connect them up using the line drawing utility.
5. Label with text.
6. Plot a check drawing and carefully check the

[1]see notes on page 8

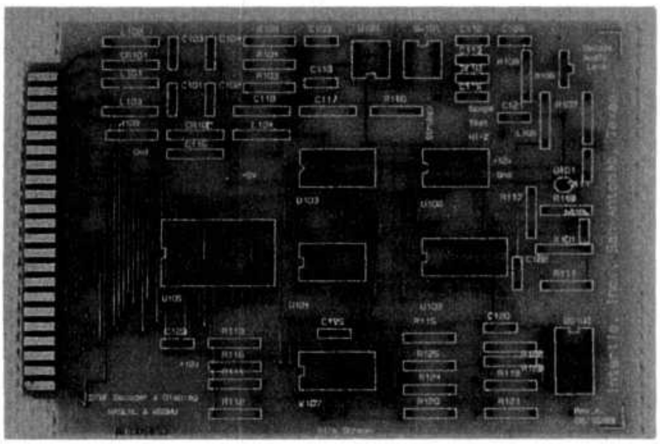


Fig 1—PC board, component side (layer 1) shown, with a silk screen (layer 3); not shown: flip side of board (solder layer). Additional layers used include schedule and drill hole diagram. A solder mask layer is often used, as well. This allows for a protective coating to be applied to all traces on the board, excluding solder points.

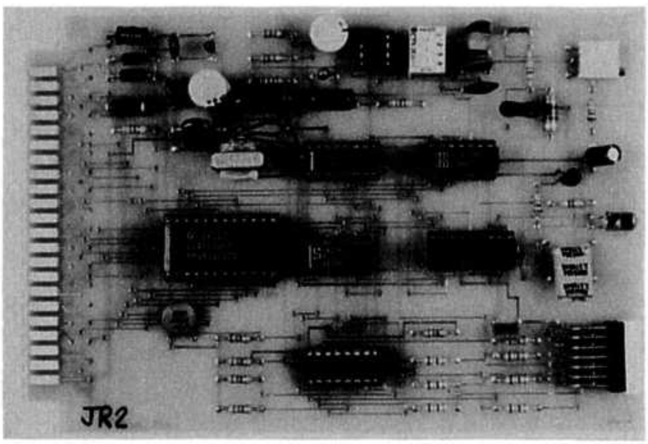


Fig 2—My first board, complete, with parts installed. This is a DTMF decoder with output displayed on a 7-segment LED (lower right corner). The basic board is 4.5 × 6.5 × 0.062 inches and has an edge connector set up for 22 pins on each side with 0.156-inch spacing.

schematic. This is the most important first milestone. Do it very carefully! For a large schematic, I suggest you place all the components before any wiring is done, to be certain you have enough space. (Components are in the library supplied by the software vendor.)

7. When you are really certain that you've gotten it perfect, make an extra backup and proceed with the next step: EDIF (Electronic Design Interchange Format[7]).
8. Load the editor program and the schematic. The menu has several items, but you only need a NET.LST right now, so ignore the rest.

Run EDIF

This program will humble you, especially if you thought you knew how to draw a schematic and, like me, have been doing it for 30+ years. Why? Well, the schematic you drew with the computer has some very special *properties*. These are attributes of each point on the schematic, which must be totally consistent or the EDIF will return error and warning messages. Until you are really familiar with the antidotes for the warnings and errors, you are in for some frustrations. Perhaps I can help prevent some errors now. Here are some dos and don'ts:

1. Each schematic symbol is enclosed in a 4-cornered box called a *cell*. The area within is *protected*, meaning you cannot write anything inside without *removing the protection*. Ordinarily you've got to have a *good* reason.
2. When you select *line* from the menu to begin drawing lines, you are forced to select a *property* of that line. Select *net*, unless you are constructing a symbol. Be very careful not to select some other property.
3. When you select *text*, again, the text must have a property. Use *unassigned* for all incidental labels. But notice: Components are special in their size and value, ie, ohms or microfarads are specified in the part number; their occurrence like R322 or C409 are instances. You *must* select the property properly. (Use F9 to call the property menu.)
4. NETS—At first, I recommend that you manually connect each component terminal to each place it is to be connected. That is, show each NET. (Later, much later, you can experiment with broken nets, buses, etc.) Run EDIF, get the NET.LST printed and use it to check the schematic for proper CAPTURE. Chances are you'll be missing some grounds. Sometimes, grounds are flaky because they need a little space around the triangular symbol.

   Try this: Pull up a GND symbol on the screen. Call for PROPERTY edit. At this prompt, select GROUND. You will see just ONE point highlighted—this is what makes the line you drew from the component terminal to ground a GROUND. If you are not careful in routing your net line from the chip symbol to ground, you may wipe out this point by overlaying it with a point that has NET property and, oops!, you've lost your ground. To get it back on a real schematic, call PROPERTY EDIT, select GROUND, position the cursor over the proper point in the ground symbol and hit ENTER. Voilá—you have your ground back, and the NET.LST will show a ground run from your chip.

EDIF Errors

On my early schematic drawings I was very frustrated at first in my attempt to get an error-free EDIF. Before running EDIF, check your schematic for plenty of space around parts. Don't let them kiss, but provide a run of dots around each. Leave dots all around a ground and 2 dots below the point of the ground symbol.

Run the PROPERTY menu for each item in the menu: UNASSIGNED, NET, INSTANCE, and PART # will give the most trouble. Fix any wrong properties by placing the cursor with correct property called and hit ENTER.

Look at every SYMBOL and make sure all four corner markers are present. You tend to lose these when you move symbols around. Be careful. If you mess up a symbol, ERASE it and call up a FRESH ONE from your symbol library. Relabel the component size (part number) and instance. Remember, the *part number* is ohms or µF, not "R156". "R156" is an *instance.*

When you get to labeling NETS with NET numbers or text, you may find you need an extra line of dots above the net name in order for the schematic capture routine to work properly.

Labels

Another source of error message is duplicate INSTANCES (ie, R126 used twice on the same schematic). Use your check plot to assign INSTANCES, or make a checkoff list so there are no duplicates.

No Net Assigned

This is the most frustrating source of EDIF error message. I have found that if I run the PROPERTY EDIT with NET called that I have sometimes gotten a line with "unassigned" or some other property stuck within a run between two terminals. Simply positioning the cursor over the offending point with NET called and hitting ENTER cures the error.

Another source is in moving parts with the "wires" attached. Although the program offers "rubber banding," this can sometimes lead to inappropriate properties outside a symbol. Fix with PROPERTY EDIT.

Finally, if the error is "CORRUPT SYMBOL," just erase the symbol and put a new one in its place. This usually cures the problem.

NET.LST Errors

Once EDIF runs with zero errors and zero warnings it will be enabled to produce a net list-file NET.LST. I usually print this up, and then using a highlighter pen, mark each net on a copy of the computer-generated schematic. I highlight the net number last to indicate a perfect connection. NET.LST lists *every* connection, including nonconnected terminals. (A terminal must never be left blank. Connect a short run of net wire to it.)

This check is the *second* critical milestone in the PC board capture process. I take a lot of time on this. If a net is incorrect, load the schematic and look at it. Chances are the computer schematic is correct, but some wires have been run too close, and the wires need to be rearranged to produce a little space between them.

EDIF will have failed to see the connections properly. The fix is really easy, but in a too-tightly drawn schematic, it might take a little imagination to arrange parts so EDIF captures correctly.

Once you have verified every net, I suggest a second check against your original hand-drawn schematic. Do it now—once you are in the printed circuit board *(PC-PRO, Tango, SmartWork)* program it can be a lot harder to fix. Keep a record of each check you've done. If you do all these on spare time, you may easily forget. I mark each check schematic and each NET.LST with the date and what I've done with it, ie, net/schematic check, etc.

Your PC Board Layout Begins: PCPRO.PCB

After all corrections are made, go back and rerun EDIF—this time call up PARTS.LST, NET.LST, and PCPRO.LST. The programs will then produce the PCPRO file, which is in effect a bag of your parts to be thrown on a PC board, arranged properly and then routed. To do this, EDIF requires that a file be generated *in advance* of each part number and the type of drawing this represents, expressed as a PC board footprint (ie, if a 4069 requires a 14-pin DIP socket, it must be listed as "DIP-14" in the file PARTS.PKG).

You generate the file yourself using your word processor. The instructions will tell you that this is a simple two-column list as an ASCII file, with the part numbers on the left and the package on the right. All entries *must* be in capitals. In the AMS program, these footprints are listed in little files on the library disks. Simply type the file designations required on the right. Save the file, and copy it onto your EDIF disk. I have a very long file now, but no matter, it can be added to whenever a new part number is encountered.

Potentiometer part numbers should be listed as "K200" to separate them from fixed resistors which should be "200k." The factory instructions should be pretty clear. Look out for spaces before part number, ie, " 5k", which won't match "5k" without a leading space.

EDIF will generate, providing all part numbers have a proper footprint, another list (not printable) which, when PCPRO is loaded, can be called "PCPRO.PCB" and then later renamed as you see fit, as it is stored on a file disk. If a part number has no footprint, EDIF will yield an error message and force you to redo your ASCII file, PARTS.PKG, before an error-free file can be created. Sometimes you will get such an error message and there is a footprint listed. Chances are, the part number is corrupt, ie, leading or trailing spaces, or incorrect properties on the schematic.

The AMS files should contain almost all footprints that you'll ever need. The manual will tell you how to make custom footprints. Some library footprints you will find are of the wrong size/pin spacing for your in-stock parts. You will need to modify those. I had trouble with potentiometers, crystals and transistors. Their transistor footprint (DISC-Q) is set up for a plastic TO-92; if you need to use the round transistors (TO-5, etc) you may wish to redo this one. Remember, the program has trouble drawing real circles, so good luck! Call all the footprints to a PCPRO page and then plot all the different footprints on a piece of paper and keep it with your manual for reference. To get the pin spacing right, especially for capacitors, you may need more than one footprint. Buy a ruler that measures in 1/100 of inches [Sears 9 HT 40303] and measure the pin width of your capacitors and other parts. Make certain to draw an outline large enough. Then modify the library symbol to fit. EDIF the symbol before using it (see manual regarding EDIF for symbols).

Parts such as 4069s, which have 6 inverters, will be represented as individual inverters on your schematic. All 6 should be labeled with *the same* instance number. Thus, this number will call one footprint for one real 4069, with all 6 inverters inside it. (If your program does not provide each element pin number, look them up and assign them using the correct properties.) Your PARTS.LST from EDIF must exactly match a list of parts generated from your hand-drawn schematic.

EDIF also provides a PIN.LST which can be used for checking, but I find it too long and cumbersome. The EDIF.LST will not be used unless you are going to carry your schematic to some other PC board program. Run it to see it. It can also be typed up and used for cross checking.

More on Nets

Study very carefully the pages in your manual regarding nets, buses, etc. The labeling methodology is critical. The illustrations are very helpful.

When making the actual PC board, it is possible to start with a PC program such as *Tango* and go directly to a board layout. But there are *serious* limitations to this approach. First, if there is any complexity at all, checking the board is a lot more difficult because it must be done without the help of a correct net list. Most important is the use of an autorouter. Autorouters require that the board be scheduled, meaning a list of all runs between pins of each part-package footprint be listed. This requires a net list, and it will have to be manually generated if the schematic capture program is absent. Hence, I was pleased to see that the AMS program does it all, without the purchase of additional expensive packages as discussed in the original *Ham Radio* (see note 1). (AMS, and others, offer translator programs which, for instance, will allow you to use their schematic capture program and then import the data to *Tango, SmartWork*, etc.)

We have a schematic, computer generated, which exactly matches the hand drawing as to wire connections (nets) between parts. The properties of each run have been checked and are correct. The footprints of each real component have been matched to each part called up in the computer drawn schematic. We are now ready to do the PC board.

Steps in Creating Your Own PC Boards

1. Load the PC board CAD program and call the file containing your EDIF schematic file. (In the AMS program, the file has the extension .PCB and will appear in the directory after EDIF has been successfully run without error flags.)
2. On the blank board, choose the number of layers.
3. Arrange the parts in a logical manner, within the confines of the size board you desire.
4. Run the scheduling program.
5. Autoroute.
6. Make check plots and edit to reduce number of vias and complexity of runs.
7. Plot final quality.
8. Photograph, submit to board maker and receive finished board(s).

How to Do It

Actually, this part of the whole PC board business is the easiest, but is still time-consuming. A good layout requires very careful parts placement. (More expensive PC programs have added enhancements to optimize parts locations for shortest scheduling runs; AMS lets you do it.)

I found that the instructions are helpful. Use the keys that produce the point-to-point schedules and move parts close by each other.

Use a 50-mil parts location grid. Line up parts in straight rows. Do not stagger packages unless you are sure it will shorten runs.

Once I have the parts lined up on the board, I make a backup and run the autorouter scheduler (PC-SCHED). I then reload the board backup with PCPRO and look at the relative density of the scheduling lines (these are called a "rat's nest"). I make notes of parts to move to shorten lines and to have a uniform density of lines. Give edge connectors plenty of real estate in front.

I then redo the layout repeatedly, perhaps making an autorouter run to determine the percent completion.

Details

A. Layers. The AMS instruction manual has a very complete tutorial on boards, layers, etc. For ham use, I found 7 layers are the usual. (The program can handle 256!)

1. ALL (where the pins poke through)
2. COMPONENT (top) (can have solder traces)
3. SOLDER (bottom) (can have solder traces)
4. SILK SCREEN (parts, labels and your name)
5. DRILL (where to drill holes)
6. SOLDER MASK (necessary only for flow soldering)
7. SCHEDULE (required for autorouter—shows the rat's nest when the board is scheduled [PC-SCHED])

Relax, if a layer gets forgotten, it can be added later.

B. Size. Settle on a size of board. You can make it a little too big at first in order not to crowd parts and traces, and trim it down later on. Be *very* careful to avoid placing parts or traces outside the board edges.

C. Crop markers. I suggest 2-mil crop lines placed 250 mils in from the board edges. Stay inside these. Some board makers like a 250-mil diameter hole 250 mils in from each corner for line-up of tools.

D. Edge connectors. The AMS manual shows how to do edge connectors. These are constructed from *pads*. There is a menu of pads (see below). Double-sided connectors are spooky for autorouting. The pads will be laid on top (component) and bottom (solder) sides of the board. They will be spaced at some standard interval, usually 100 or 156 mils. You can get them laid down right the first time by setting the Y-grid to 100 or 156 mils. They will number themselves automatically.

To get the autorouter to work with a double-sided connector, I found it necessary to draw a connector with hand run traces to a round pad laid on the appropriate side of the board. I used the layout found on VECTOR 3577 PC board edge connectors. I then numbered those accessory pads as I would have numbered the edge pins. The edge pads ("fingers") were then renumbered with higher numbers (ie, 22/44x156 connectors): accessory pads 1-22 component side; 23-44 solder side; fingers 45-67 component side; 68-90 solder side. Before I came on this trick, I found the autorouter was unable to construct vias automatically to get traces to the other side.

E. Pads. On my first board, the 62-mil default pad diameter for DIPS was used. This pad is really much too small for hand soldering (although flow soldering might be okay). A fellow ham suggested that I use elliptical pad's. Voilá—the program already has these in the pad's menu (55 × 100 mils), both vertically and horizontally oriented.

The elliptical pad probably provides a little better strain relief for board flex than a round pad does. If real estate on the PC is too limited for elliptical pads, I suggest the 75-mil rounded pad, if at all possible.

Note: Pads must be "reloaded" when using the PCPRO.PCB schematic capture method. (RELOAD PADS is a menu choice.)

F. Vias. These holes are used to bring the circuit trace through the board on double-sided or multilayer boards. We want to minimize the number of these (zero would be good) in order to keep cost of fabrication down. Vias are plated through by the board maker in order to obviate the need for a thru-board jumper. The vias are defaulted to 50 mils and this is okay. Using the larger pads for DIPS allows easier eyeball checks on the PC board later. You may wish to use a larger via for power lines.

G. Board layout, continued. After getting the best autoroute completion by juggling parts locations, plot the board (PCPLOT) on your line printer. For 8½ × 11 inch paper, a 4½ × 6½ inch board can be plotted with 1.25 magnification. Look very carefully for redundant loops of trace runs. Mark with a highlighter and remove or reposition to minimize run lengths. Often, removing these will open up the board for more comfortable, less crowded PC runs.

If you have a larger plotter, use it now to get check plots and look for shorts, inadequate clearance of traces and excess vias. Keep going back and simplifying the board. Keep sequentially numbered backups in case you get lost. Make extra backups during revision sessions often. (My program has crashed on a power off after 1-2 hours of PCB revisions—sob!)

On my boards, the commonest wasted runs are:

1. Long U-shaped things that decompose into a shorter direct connection.
2. Long wrap-arounds often demanded by lack of real estate on the original autoroute.
3. Nets with 3 or 4 chips to be interconnected with a single run may be overly decomposed into longer unnecessary runs. These can often be shortened.

About the Autorouter

A. You will have an opportunity to make some very critical choices here.
   1. Set the B+ and other power lines to at least 25 mils. Fifty mils is better, if space allows. Grounds should be the same width. Route these separately first.
   2. A special net that needs to be routed early can be given an oddball width.
   3. 15 mils should be the default width. Use 12 mils only if real estate is very dear.
   4. Keep a record of your choices.

B. Routes. The router allows a choice of clearances. The manual suggests a 50-mil grid. I agree, but allow 25 mil minimum distance between component pads and 15 mil between traces and vias. The menu allows choices between right angle and 45-degree routing. Note the factory defaults. Audio boards may benefit from all 45s. The manual offers some sage advice here.

About checking

Bring up the image of the printed circuit that you have laid out and routed on the screen and have an extra copy of the original schematic in hand. Using a highlighter, mark each point-to-point connection on the check plot, then ERASE it from the screen. Keep doing this until all traces are gone from the screen. Errors will be very obvious: A trace won't be found or you'll have something on the screen that is not represented on the original drawing. Use a storage disk to hold partially erased drawings for rechecks, etc.

This procedure shows that I rarely had errors, unless I caused them by moving (previously autorouted) traces around to eliminate vias and excess run lengths. At this point, with all checking done, plot with multicolors or on multisheets and have a friend check your work before final quality plots are made. He will inevitably find 6-8 more runs that you can revise and simplify. Then RECHECK by the screen erasure method and REPLOT, final quality.

You can make quick check plots with your line printer. Most common printers and plotters are supported by these programs. The newer Hewlett Packard DeskJet and similar printers work fine. Of course, a pen plotter is better, but slower. The *TANGO* manual offers suggestions regarding plotters, papers, and pens. Suffice to say, the best plots are made on good paper with a steel pen and ink. Felt pens are okay for check plots. You can save money on the expensive vellum paper by using clay-based finger-painting paper, available in school supply stores. My final plots are drawn twice their actual size, except for the drill drawing, 1:1, on high quality vellum. I use HP VELLUM #17805P and the Staedtler #75PL07H1 holder with their 750PL3C3 pens with a Houston Instruments PC-695 plotter.[8]

Photography

Bring your 2:1 plots to your local graphics photographer for positives on acetate. Have the photographer shoot 50 percent to get actual size positives. The PC board maker requires a plot of the drill locations 1:1, also. The PCPRO program produces a list of numbered drill ("tool") hole sizes matched with X-Y coordinates of the hole (DRILL.LST). These will be needed to drill the board.

Some programs, but not PCPRO, will give you the EXCELLON drill tape for automatic drilling machines. This is an 8-level ASCII Teletype tape. Ought to be easily programmable by someone with a TTY printer. You can skip the final plot process by using the GERBER output file as the plot and send it via modem to some circuit board houses.

Conclusion

I have shown my ups and downs of PC board construction with an inexpensive but excellent PC program. My first board worked without modification or jumpers added later. You can do it too!

Acknowledgments

Thanks to Ernest Reich, III, W5FQA, for doing the photo-ready plots; to Don Kessler, KI6SZ, for checking my layouts; to Dink Stockert, WA5LNL, for constructing the prototype from the first board, as well as pointing out some deficiencies in the layout of parts and the default pad sizes.

Notes

[1]Freeman, E., "Low-Cost PC Board Layout Software," *Ham Radio,* October 1987.

[2]*Tango-PCB* is manufactured by ACCEL Technologies, Inc, 6825 Flanders Drive, San Diego, CA 92121, tel 800 488-0680.

[3]*Orcad* is manufactured by Orcad, 3175 NW Aloclek Drive, Willsboro, OR 97124, tel 503 690-9881.

[4]*SmartWork* is manufactured by Wintek Corp, 1801 South Street, Lafayette, IN 47904, tel 317 742-8428.

[5]Miller, T. and Pagel, P.K., "PC Board Production in the ARRL Lab," *QST,* April 1988.

[6]*PC-SCHEME* and *PC-PRO* are manufactured by Advanced Micro Systems, Inc, 2780 SW 14th Street, Fort Lauderdale, FL 33069, tel 305 975-9515.

[7]Eurich, J.P and Roth, G, "EDIF Grows Up," *IEEE Spectrum,* November 1990.

[8]Houston Instrument Division of Ametek, 8500 Cameron Road, Austin, TX 78753, tel 512 835-0900.

RF

By: Zack Lau, KH6CP
225 Main St.
Newington, CT 06111
email: zlau@arrl.org (Internet)

Welcome to "RF." This column is for true experimenters — those who build projects whose outcome is uncertain. After all, someone has to build that first prototype and get it to work. While all kinds of ideas are certainly welcome in this column, I'm primarily looking for project ideas from people actually trying to implement them, or, better yet, have actually managed to get one to work. Little details about a project, like where to get the parts or figuring out how to tune the marvel up without $100,000 worth of test equipment are not important here, if I feel enough people can benefit from publication of the project. Those interested in purely theoretical exercises are advised to send their letters to our "Correspondence" column, though you will get a (form?) letter back if you mail it to me, and I'll probably save the stamp as I do collect used stamps as a hobby.

Of course, if you want to set up a little mail order business to get rid of a can filled with $300 power GaAs FETs for $10 each to grateful readers, let me know. For this column, an example of a product announcement is: 30-kHz bandwidth 10.7-MHz crystal filters are available from:

A & A Engineering
2521 W LaPalma, Unit K
Anaheim, CA 92801
Phone 714 952-2114

These HC-18 cased filters with 5000-ohm termination resistances are the right bandwidth for satellite WEFAX receivers. Current price is $7.50 each, plus shipping. For those who don't know, buying one or two 30-kHz-wide filters is a real challenge, unless you just happen to know that this supplier has them.

Wideband Microwave Antennas

I've been playing around with a 903-to-5760-MHz log feed for a 2-foot dish. While it doesn't offer a whole lot of gain on the lower bands, it certainly seems to be an acceptable antenna for a portable or rover station. Artwork is not available, since I'm still trying to find time to use a suitable computer drawing program. However, I will share the design procedure so that someone with the appropriate test equipment can develop a really good design.

First of all, what are the characteristics of a log antenna and how do I optimize these for a dish feed? There appear to be two different types of log antenna operation. Not only can you get the elements to resonant as half wave elements, but you can get them to operate on odd multiples as well. It appears to me that there is a bit of a central null with these high order multiples — that the pattern peaks either side of boresite. Actually, this is what you want for a dish feed — the center of the dish usually looks at the feed, and is therefore not useful anyway. So, it looks like a really clever designer could use the high order resonances to maintain the phase center of the dish, and even come up with a pretty optimum illumination pattern. Even the higher gain you get on the higher order resonances can be an advantage, as you don't necessarily want to fully illuminate a dish on the higher bands, due to the difficulties in pointing a dish accurately.

However, not being that clever and not having enough time I opted for the easier log feed approach. This is to make the array as short as possible, so the phase center wouldn't shift too much. This trades-off SWR and gain. The latter really isn't a problem, since we often don't want that much gain out of our dish feed, particularly with the low f/D dishes sometimes obtained. The SWR on the other hand, can be a problem, as the worst case SWR gets worse as the array gets shorter. I've been told that the SWR oscillates with frequency, but I haven't verified it myself. It may be possible to get the SWR minimums to correspond to the amateur bands, but you are on your own at this point. I didn't worry about this too much, since the 80 feet of feed line needed to reach the top of the tower pretty much flattens the SWR at the transmitter end! Not very pretty, but at the K1TR/3 contest site we usually run enough transmit power to compensate for the losses. After all, not everyone enjoys climbing towers to fix equipment. Surprisingly, my prototype seemed to have a pretty good pattern on all bands from 903 to 5760 with the 2-foot, 0.45-f/D dish.

Transverter Topologies

The simplest topology is a bilateral mixer — use the same mixer on receive or transmit. The primary advantage is that it is simple and easy to align, at least if you want something that basically works. Since the same filtering is used for transmit and receive, you can use either a signal generator and tune for best received signal or a spectrum analyzer and tune for best power output. Of course, if you are real picky and want to optimize it for both you will find that you have to compromise in your tuning. Once you have a mixer, the next step is to add an amplifier. A transfer relay is quite useful, since it lets you use the same amplifier on receive and transmit. For those not familiar with transfer relays, these are 4 port devices. Again, optimum tuning is often a compromise — it's difficult to tune for best noise figure without sacrificing output power. However, if you use a latching transfer relay, the transverter will often draw the same current whether on transmit or receive. This is definitely an advantage when running off battery power, as some local oscillators, particularly the phase-locked bricks, can be very voltage sensitive.

The next step is to eliminate the compromise and go to separate transmit and receive amplifiers. There are several ways to achieve this. Perhaps the most elegant is to use a circulator to separate the transmit and receive paths. Typically, a good circulator has only a few tenths of loss and 20-dB isolation. Problem is, I've had little success finding suitable ones. The crude but effective method is to use an ordinary splitter, but this degrades the transmit and receive paths by 3 dB. Not a problem if gain is easy to get, but hardly optimum. A third method is to use separate transmit and receive mixers. This is perhaps the best method for the ultimate transverter, as the transmit mixer can be optimized for best output power and the receive mixer optimized for best noise figure.

A variation to be considered is to use multiple conversions. This is quite useful when dealing with surplus local oscillators. Instead of buying an expensive precision crystal to replace the one that's in there, simply perform another conversion, choosing the frequency so your transverter "tunes the right way." The disadvantage is that you now have two independent variables that determine the frequency, but this often is acceptable. Sometimes it's possible to do both conversions with multiples of a single crystal. For instance, I used a 106.4896-MHz crystal in a 11.5-GHz local oscillator brick (×108 multiplication). By building a ×12 multiplier and performing another conversion, I effectively have a mixer with a 10.223-GHz local oscillator. And, if I somehow figured out how to retune the brick as an ×96 device or got an ×96 brick, I could still use my crystal. A caveat with this approach is that multiples of a conversion oscillator frequency should not be close to the receive frequency. For instance, my first 3456 transverter was a dual-conversion system using the 2160 LO to use a 1296 transverter having an 1152 LO. Not surprisingly, I needed something like a 10-dB pad to prevent the 3rd harmonic of the 1152 LO from blocking an ICOM IC-202 IF receiver. The transverter itself had enough dynamic range to accommodate this unwanted spurious signal. Fortunately, Jim Davey came up with his 3456 transverter (see June 1989 *QST*, p 21) so I just scrapped this dual-conversion system.

If you do buy the expensive crystal, make sure you specify the correct operating temperature. Poor results are inevitable if you try to run a crystal optimized for operation at 75 degrees C at room temperature.

While the crystal manufacturers may not appreciate this, specifying series resonant or parallel load capacitance really isn't the best way to characterize crystals. After all, isn't the idealized model for a crystal a series resonant circuit in parallel with a capacitor? A much better characterization is the motional inductance and capacitance, along with that parallel capacitance. Of course, it's much cheaper to take a crystal and grind it to the right frequency, than to find the right type of quartz and get both the motional inductance and capacitance correct. But, if you spend enough money...

Transverter Dynamic Range — How Good is Good Enough?

While it may sound great to have a transverter with a two-tone dynamic range of 110 dB, what good is it if your IF radio crunches way before your transverter? For instance, the lowly MiniCircuits SBL-1/1X mixer in a good circuit can have a +15-dBm input intercept and 10 dB of conversion loss. With 10 dB of post mixer amplification, this means you need a receiver with a similar input intercept: +15 dBm. If the receiver has a 3-dB NF and 500-Hz bandwidth, a two-tone IMD DR of 108 dB is needed! Even with a 15-dB NF, a 100-dB IMD DR is needed. No wonder the big VHF contesters haul those top-of-the-line radios out to mountaintops.

The situation is different on the microwave bands, as there is usually a lot of directivity associated with antennas. Also, in order to maintain low noise figures, a large amount of gain is often required ahead of the mixer. Finally, a high dynamic range is useful only if the noise of the oscillators is sufficiently low. Otherwise, any weak signals will be masked by noise sidebands.

Switching Regulators

Perhaps the biggest chore in building microwave work is building the necessary power supplies. Given the expense of some of the microwave active devices, this is not the place to cut corners. For fussy applications, switching regulators using inductors to step up or step down the voltage really aren't suitable. Not only do these toroids generate magnetic fields that are difficult to shield, but these designs usually don't like the varying loads often found in amateur designs. They typically respond with high voltage spikes that pose a serious hazard to solid state devices. Fortunately, it is usually possible to get by with another form of switching regulator: charge pump devices. The idea is to generate a square wave and then use diodes to steer charge into capacitors. Not only can a polarity inversion be obtained, but voltages can be easily doubled or tripled. Unlike inductor-based power conversion, the energy is stored as electric fields, so there is little magnetic energy to cause interference. As a result, a simple box soldered together out of unetched circuit board often makes a totally adequate shielded enclosure. Most important, if you vary the load there are no high voltages generated. A disadvantage is that a linear regulator is usually needed to get the final voltage, so the efficiency is probably worse than that of a good inductor-based switching regulator.

Protecting GaAs FETs

I've found that the best protection circuit seems to be a simple current limiter. Not only do they effectively prevent the devices from drawing more than the maximum allowed current, but they are simple enough to have a high reliability. I've had pretty good luck using LM317s as current regulators, the only disadvantage being the several volts of overhead required. For portable use with cheap devices, it may be entirely adequate to just size your voltage regulator appropriately. Many of the regulators are self protecting, and will not pass more than two or three times the usual maximum power. I don't know about you, but when I have to carry the gear more than a mile up a hiking trail I want it to be as light as possible.

The Status and Future of High Frequency Digital Communication

Part III: Simulating the Performance of HF Digital Networks

By Ken Wickwire, KB1JY
232 North Road #17
Bedford, MA 01730

1. Why Simulate Performance?

Most of the resurgent interest in applying digital signaling techniques to HF communications has been concentrated on the operation of a small number of links —frequently single ones. Part of the reason for this is the fact that multipath fading and contention for channel use often cause very poor communications in the HF band: The unpredictable changes in signal-to-noise ratio (SNR) at the receiver caused by these conditions can force amateur packet and commercial modems to drop and then have to reestablish links many times before all the packets of a message are delivered correctly. Sometimes the modems carry out so many retries that their protocols "time out" before all the packets of a message are correctly received. The amount of time a particular HF frequency will support reliable reception is sometimes only a few minutes.

One of the most powerful techniques used in computer networking is **relaying.** The relay nodes are usually chosen because they have access to channels with higher capacity (greater bandwidth) than the *direct* links to destination nodes. Relaying often involves **storage and forwarding** of messages. Storage and forwarding allow the relaying nodes to wait until congestion has cleared before sending on traffic.

Relaying and storage and forwarding of messages are effective in computer networks because computer channels are generally reliable and have high capacity. In VHF and HF radio networks the channels have low capacity (they are only a few kilohertz wide instead of several megahertz wide), and they frequently carry multipath. As we found in Part I, HF channels have the additional disadvantages that they are often full of interfering signals and disturbed by noise. This means that the methods used to route and forward data in computer networks will not always work in HF networks.

Computer networking techniques were developed mainly to react to changes in *traffic* and not to changes in *channel quality*. The rapidly changing quality of the HF channel demands that HF networking techniques adapt themselves to changing traffic *and* channel quality. Figuring out how to do this effectively requires new methods, and a means for predicting the behavior of a randomly and rapidly changing system. This changing channel quality also means that, even though choosing relays properly can be difficult, using them for regular long-distance traffic is absolutely necessary at HF.

Designing an automatic network that is effective over the HF channel involves the proper choice of

- the **equipment** at each node (radio, amplifier, antennas, ALE device, data modem and a computer to control the node), and
- the **network control software** that runs on each node's computer.

Choosing the proper equipment at each node (for example, high-gain antennas or a modem with particular error-correction coding) depends on predictions of link performance and overall network performance (see the discussion on this below). Knowing how well a set of equipment works on particular links does not necessarily tell us how well it will work in a network, and vice versa: effective use of relays can bring high performance for the network as a whole even if some of the links are not very reliable.

In a fully automatic network, control software operates the ALE (linking) and data modems, chooses relays, makes routing decisions, stores and forwards relayed traffic, manages message buffers, performs input and output functions, points antennas if necessary, and so on. All the operators have to do is enter and read messages. Deciding how such a network should be controlled (especially choosing relays and making other routing decisions) also depends on predictions of link *and* network performance.

Both parts of the design of an automatic network therefore require **predictions** of its performance. I mentioned in Part I that there are three basic ways to predict network performance:

- by nonrandom ("deterministic") mathematical analysis,
- by operating an actual network, or
- by "Monte Carlo" computer simulation.

As will become clear in the rest of this part, there are many interrelated steps in the operation of an automatic HF network. Changes in the channel and the message traffic make the outcomes of many of these steps random. For example, the number of attempts before a successful link-up and the amount of time until all the packets of a message are correctly received are random variables. Since the most effective techniques for routing messages through relays use measurements of channel quality (which is random), the routes taken by messages through an HF network are generally also random (see Sec. 7). The complexity of the probability distributions of many of these random variables (they are correlated and their statistics change with time) makes a useful deterministic analysis very difficult.

On the other hand, predicting an HF network's performance by actually operating it is usually too time consuming and too expensive to be a practical approach (except for amateurs): every change in the control software has to be coded and loaded into the computers at each node; proposed changes in equipment require expenditure for actual hardware and for trips to the nodes to carry out installation. Finally, the actual network has to be run long enough to collect reliable statistics on its performance, which may also be expensive.

It turns out that the only reasonable way to predict a commercial HF network's performance for design purposes is to use a simulation.

In HF simulations a **discrete event scheduler** creates messages at random times and chooses random sending and destination nodes according to assumed traffic distributions. An **event manager** keeps track of the messages as they move through the network. Such a simulation is said to be **event driven** or event stepped. A random number generator (used also by the event scheduler) produces changes in SNRs about their mean values (due to fading, interference, etc.) in accordance with the selected time, season and sunspot number. Predictions of the mean SNRs themselves at a particular time come from a **propagation prediction program** (like IONCAP), which may be viewed as input to the network simulation. The traffic distributions and input to the propagation program are dictated by the scenario that is being investigated by the simulation run.

The propagation prediction program usually gives a description of the so-called "quiet ionosphere," that is, an HF channel without fading or bursts of noise and radio interference. Simulations of these short-term changes are added to the long-term (quiet) predictions to create a realistic channel simulation. A network simulation that generates messages and channel changes randomly is called a **Monte Carlo** simulation.

After a simulation has been used in trial-and-error fashion to design a network with seemingly acceptable performance, the network's designers may consider building an actual network according to their design. Data from the actual network's performance are then put into the simulation for **calibration** of the design. Calibration refers to the process of matching the design's predicted performance to its actual performance by adjusting input or protocol parameters. (Amateur networks, which use inexpensive equipment and software, are set up in the opposite order: first they're built, and then they're improved. This is possible because amateur networks don't have to meet stringent performance requirements. The "reversed order" of amateur development offers many opportunities for data collection and efficient network design that are not available to commercial or governmental designers.)

As soon as a design has been calibrated, it should then go through a period of acceptance (or "beta") testing and further adjustment before being turned over to the customer for regular operation.

Fig 3-1 is a flow chart of the design cycle for an automatic HF network whose design is based on Monte Carlo simulation. Note the presence in the cycle of the **network designer**, who has the non-trivial job of finding ways to improve performance that is not acceptable. The following sections discuss some of the details of the simulation shown in the figure.

2. Input to the Simulation

- *Hardware:*

The simulation requires specification of the output

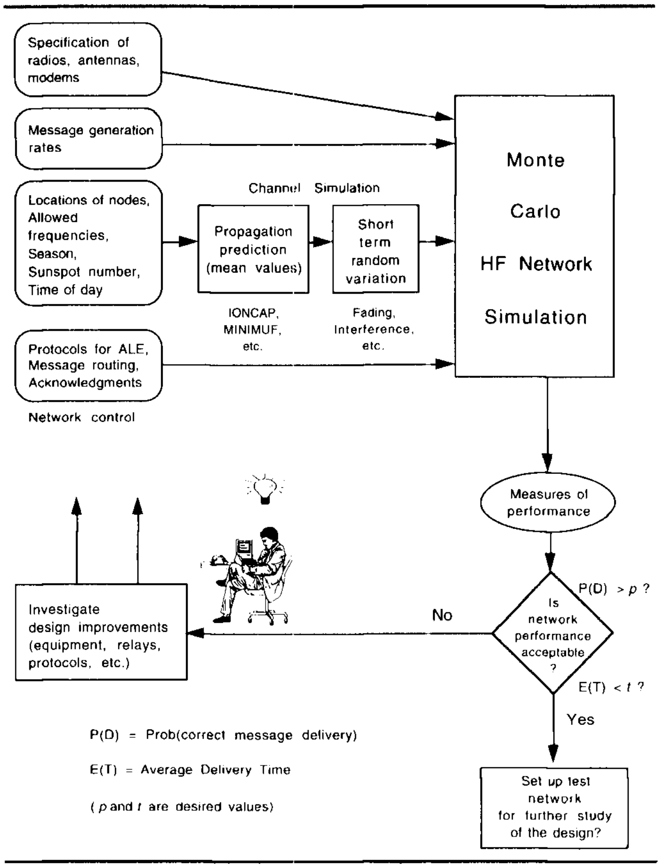


Fig 3-1 Flow Chart of an HF Network Simulation

power of radios and amplifiers, radiation (gain) patterns of antennas, performance and linking protocol of ALE devices and performance and synchronization protocol of data modems, including possible data rates. If special relays with high power amplifiers or high gain antennas are part of the network design, the parameters of this relay equipment are also entered here.

The performance of modems (see Part II), which includes the effects of modulation, error correction interleaving and equalization, depends on the properties of the particular HF channels the modems are designed to operate in: Rayleigh fading, discrete multipath, etc. ALE performance is measured in terms of **linking probability** as a function of SNR in the particular channel; data modem performance is measured in terms of **bit error rate** as a function of SNR in the channel. For some simple HF channel types (for example, Rayleigh fading), modem performance can be expressed analytically; in complicated channels, modem performance is derived from tests with an analog **RF channel simulator**.

• *The offered traffic load:*

The rates at which messages are generated by each pair of nodes, or the fraction of the total traffic generated by each pair of nodes along with the message generation rate for the whole network. (Message generation rates are generally functions of the time of day.)

• *Input to the **channel** simulation:*

The "channel simulation" in Fig 3-1 is software that produces mean values and variances of the SNR on a link at a particular time, along with *short term* variations in SNR due to sudden fading, noise bursts and interference. The means and variances are *long term* statistics, averaged over many days. (More systematic, *medium term*, changes in the channel caused, for example, by Rayleigh (multipath) fading, are accounted for by the modem performance input discussed above under *Hardware*.)

The channel simulation requires as inputs

- the number and locations of the nodes (including special relays),
- allowable frequencies,
- the sunspot number,
- the season,
- the time(s) of day for network operation, and
- statistics on short term fading, radio interference and noise.

Since the best frequency for a link depends on how long the link is, networks of wide extent will require a wide spread of frequencies assigned to some nodes.

Since atmospheric disturbances are not modeled by the usual channel simulations, statistics on their occurrence should also be entered here for subsequent simulation of such disturbances as random events. Many observations have shown that the *long-term* effects of atmospheric disturbances on closely spaced frequencies and closely spaced links are correlated (when one frequency or link improves, so do nearby ones, and vice versa), so a mechanism for introducing **correlated SNRs** on close frequencies and close links is required in the channel simulation.

• *Initial values of channel quality for each link:*

When the network simulation begins, the nodes will not generally have much information on the quality of the links (at various frequencies) they will try to set up. They will have to be provided with initial values of channel quality. Initial link quality values can come, for example, from IONCAP predictions.

• *Network control protocols:*

These protocols are embodied in software programs for message routing, choice of relays, automatic repeat requests (ARQ), acknowledgment (eg, end-to-end or point-to-point), storage and forwarding, message queue management (for example, a first-in, first-out protocol), network flow control, channel access protocol, and so on. This is the part of the simulation that reflects *the main work of the design*. Generally speaking, each network will require its own set of control protocols.

3. Operation of the Network Simulation

The flow of decisions and events within the Monte Carlo HF simulation proceeds roughly in the following sequence (see Fig 3-3 below):

1. Using the statistics of the offered message traffic, choose random sending and receiving nodes.
2. Create a message for sending and receiving stations to exchange at a particular random time.
3. Using the ALE protocol and the output of the channel simulation, determine the outcome of an ALE attempt at the appropriate time. (The frequency used in the ALE attempt is chosen from the set of allowable frequencies using the initial values of channel quality; subsequent ALE attempts generally use frequencies chosen according to active or passive channel quality measurements made during network operation.)
4. If the ALE attempt is successful, go to the next step; otherwise try ALE again according to the protocol for ALE retries. (If there have been too many ALE tries on one frequency, the ALE protocol generally attempts ALE on a new frequency. If ALE still fails, another attempt is usually tried after a random *backoff* time.)
5. Simulate data modem synchronization (if required), including retries. Synchronization is required by modems that use an adaptive equalizer with training sequences.
6. Send one or more data packets and determine if they have been received correctly. If packets have been received without error, send more packets until the whole message has been received correctly. When

the message has been received without error, note its reception and the time it took, for the purpose of performance assessment. If the receiving node is a relay, choose the next node according to the routing protocol and go to step three. If a packet arrives with errors, send a NACK (negative acknowledgment) and follow the ARQ (automatic repeat request) protocol if required. (Some packet radio networks allow acceptance of messages that contain certain kinds of errors.) If the message has not been received properly after the allowed maximum time for reception, "back off" for further retries or go to the next step.

7. Perform step one for another pair of stations or halt the simulation if the desired running time has passed. If the simulation is finished, collect the performance statistics and display them in a form suitable for analysis of the link and network performance.

4. Measures of Network Performance (Output)

The primary measures of performance for both links and networks are the **probability of correct** (or very accurate) **message reception**, *P(D)*, and the **average time for reception**, *E(T)*. *P(D)* is calculated simply as the number of correctly delivered messages divided by the number generated. The next section explains how *E(T)* is calculated. In a complete simulation, these measures are produced as functions of sender-destination pair, time of day, season and sunspot number. The time of day is frequently divided into hourly or two-hourly segments, which are typical time divisions for IONCAP output. Specialized simulations that are used to analyze the effects of failed equipment, etc., may use finer time divisions. Fig 3-2 shows how *P(D)* and *E(T)* might change in a network as the message generation rate increases. The purpose and methods of "queue limiting," which is mentioned in the figure, are discussed below.

Many other assessments of network performance are also useful—especially in analysis of routing and error control protocols. Examples are tabulations of the reasons why ALE or packet transfer attempts fail (fading, contention, etc.) or the details of ARQ operation (for example, how many repeat requests are needed to deliver a particular message). Certain applications require the probability that a message is **delivered correctly by a particular time.** These other assessments arise from the requirements of particular networks and are not calculated by every simulation. Almost every simulation calculates *P(D)* and *E(T)*, however. (*Throughput*—usually for individual links—is of interest to designers and network operators, but of smaller importance to users. The user is normally interested in *P(D)* and delivery times since HF messages are generally short.)

5. Calculation of the Message Delivery Time

Fig 3-3 shows a typical sequence of operations that might be carried out over a real HF link to deliver a digital message correctly from a sender to a relay or final destination. Note the number of operations that have random outcomes. A useful simulation must keep track of how long each operation takes, and then add up all the operation times to get the total message delivery time for the link. If the network uses relays (perhaps with storage and forwarding of messages), then the delivery times for all the links on a message's path from source to destination must be calculated. At the end of the simulation, its output processing section calculates the percentage of generated messages that have been delivered and the average delivery times. (Deciding what should be done with messages that have not been delivered is part of the network design; they might be re-submitted later or sent over some other communications medium.)

6. Desirable Attributes of an HF Network Simulation

To make a simulation most useful in network design, the input and output interfaces should be written so as

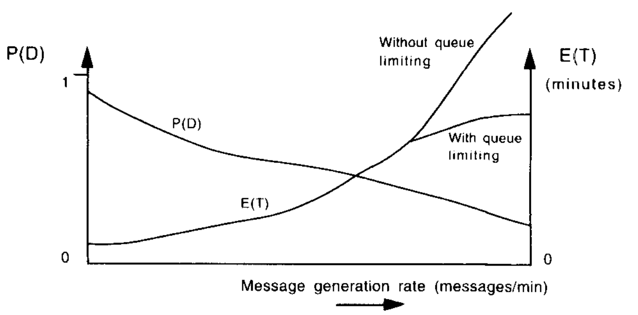


Fig 3-2 P(D) and E(T) for a Link or a Network as Functions of Generation Rate

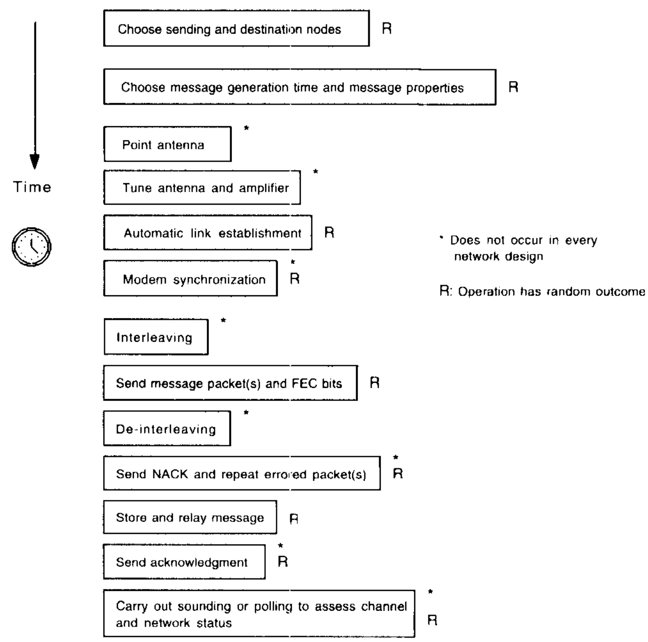


Fig 3-3 Sequence of Operations in an Automatic Digital HF Network

to make changes in input, and interpretation of network operation and performance easy. Future simulations will have menu-driven input interfaces and a wide range of graphical and statistical presentations of performance. A routine for tracing the passage of messages through a network that uses relays is also very useful in the analysis and improvement of message routing techniques. All of these desirable functions can easily be carried out on a modern personal computer.

7. The Importance of Routing and Flow Control

I've pointed out that good performance of an automatic HF network depends not only on effective signal processing (to deal with the rapidly changing channel) but also on relaying. The relays move messages "around" channels that are bad (low SNR or high noise and interference) or nodes that have failed because of congestion or equipment breakdown. In a network with relays there are often many ways to route a message from a sender to its final destination.

The first thing to decide in the design of a routing technique is whether the routing will be **directed** (stations are told by a network control center how to route messages) or **distributed** (stations choose routes on their own using channel quality measurements and other network status assessments). Channel quality measurements in a directed network are often scheduled by the network control station; making scheduled measurements (by means of sounding or polling—see Part I) takes time away from message transfer and must therefore be considered carefully. (As a rough guide, polling or sounding should use no more than about 20 % of network capacity in a commercial or amateur network.) Because of the high overhead required for directed routing over rapidly changing channels, HF networks operating in such conditions are probably most efficient when their routing decisions are distributed.

The choice of good routes can use various criteria: shortest predicted distance (number of links needed), least predicted time, record of past successes, etc. Deciding what criteria to use for route selection and what measurements to make is one of the most challenging parts of HF network design.

There are many **routing strategies**. They can be characterized as **connected routing, flooding** or **connectionless routing**. In so-called connected networks, where connected routing takes place, the route a message will take to its destination is fixed in advance. Fixed routes can be set up by a network control center (directed), or stations can decide for themselves what path they wish messages to take (as in some of the early amateur packet radio routing protocols). Routing through a connected network is called **fixed routing.** (The converse of fixed routing is **broadcasting,** in which one station sends packets to many stations, usually *without* using ALE. Broadcasting is used in AMTOR Mode B, the asynchronous "FEC" mode, used for CQ calls.)

In **flooding,** *any* station that receives a message broadcasts it to all the stations that can hear it until the message reaches its final destination, or the allowed number of rebroadcasts is exceeded. Flooding (which wastes bandwidth) is often used in emergency communications systems for which delivery of a small number of important messages is more important than overall network throughput. **Random routing** (an example of connectionless routing that doesn't usually work well for HF) involves random choices of relay nodes.

The future of HF networking probably lies in connectionless **adaptive routing,** in which routes are selected according to frequent measurements of changing channel and network status. Because of the difficulties of collecting and distributing such measurements in a directed network, adaptive routing is probably most effective in a distributed network. Fig 3-4 illustrates several routing techniques that are used in HF networks.

When a network uses relays, its operation may require **flow control** to keep relay nodes from becoming so congested during periods of heavy traffic that the network fails. Flow control often involves **feedback** from congested nodes to tell sending nodes to slow down their traffic or direct it elsewhere until the congestion ceases. Another aspect of flow control is **queue management,** which involves rules for deciding when, and in what order,

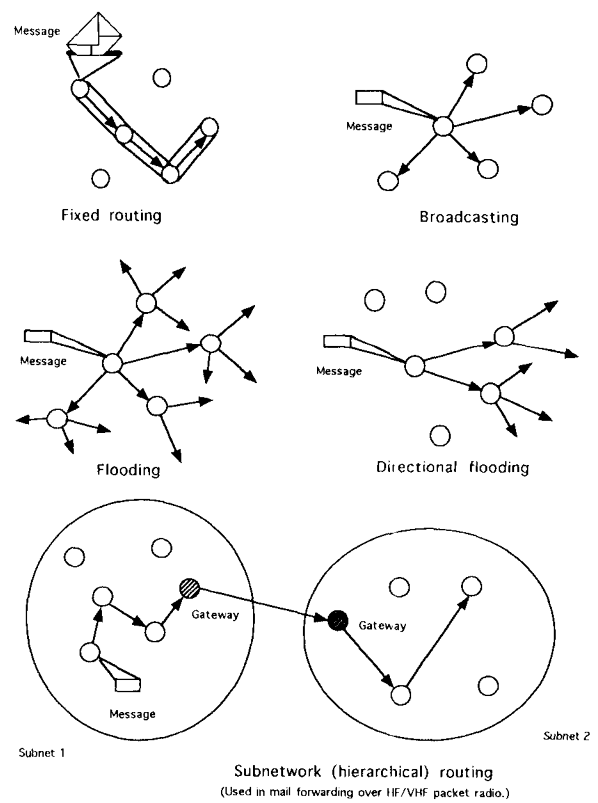


Fig 3-4 Various Routing Techniques

messages in a storage buffer should be relayed. Sometimes this is decided on the basis of *message priority*. Otherwise, considerations of fairness must be weighed in deciding which *messages* should go out ahead of others. In some systems, messages are actually removed from congested relay queues for transmission later, or by another medium (eg, telephone).

This method of **queue limiting** is an extreme example of a set of techniques designed to prevent network failure caused by congestion. Fig 3-2 above illustrates the effect of queue limiting on network performance in a congested network: Even though few messages may get delivered, those that are delivered arrive in a reasonable time.

I have just touched the surface of routing and flow control here. Although choosing these techniques properly is a challenge, they make the difference between good and poor network performance.

8. How can Radio Amateurs Contribute to HF Network Simulation Studies?

Two of the most difficult and expensive aspects of commercial HF network design are calibration of simulation models and acceptance testing of candidate designs. Calibration means checking that modem performance models (bit error rate curves, for example), antenna gain figures and channel predictions (by *IONCAP, MINIMUF,* etc) are an accurate representation of what will happen during actual operation. Acceptance testing involves checking that link and overall network performance meet user requirements.

Amateurs in many parts of the world have been providing calibration and acceptance data for VHF packet radio for several years. Now that HF modems and networking techniques have begun to be used in our hobby, the possibility that thousands of amateurs all over the world will also provide designers with calibration and acceptance data for HF networking is about to be realized.

In such a realization, designers will send computer disks or EPROMs with new networking techniques to amateurs with radios, PCs and packet modems. The amateurs will try out the new techniques at all times of the day and in many kinds of channel conditions (strong multipath fading, high noise and interference, auroral effects, etc). After the trial period they will send data back to the designers for detailed analysis of link and network performance. From time to time the designers will also suggest equipment changes (establishment of new or moved relays, more advanced modems, etc). A rudimentary version of this approach (with extensive but uncoordinated data collection) is already operating in VHF packet radio, and the practice will no doubt expand.

The last part of the series will contain further discussion of amateur participation in digital HF networking.

(Note: Part 4 of this series will appear in the October 1992 issue.)

Bits

AMSAT Annual Meeting and Space Symposium

The AMSAT-NA 1992 Annual meeting and Space Symposium will be held October 9-11, 1992, at INTELSAT's international headquarters in Washington, DC.

The Space Symposium will feature the presentation and publishing of a wide variety of papers on topics of current interest to the Amateur Radio satellite community. Beginners, experienced users, and the curious are welcome to attend and participate.

The current development effort for the Phase 3-D satellite will be discussed as will SAREX missions and the state of packet radio via the birds. Digital signal processing and topics for getting started with satellite communication will also be presented.

Hotel accommodations are available near INTELSAT headquarters. A dinner banquet will be held Saturday evening.

For registration information, or information on presenting a paper, please contact: AMSAT, 850 Sligo Avenue, Silver Spring, MD 20910-4703; tel 301 589-6062.

Components

By Mark Forbes, KC9C
PO Box 445
Rocklin, CA 95677

The first item this month is not so much a specific component, rather a precursor of a trend that will be showing up in many IC components. While the first applications will be microprocessor applications, I expect to see analog, radio and other communications applications in the near future.

As the speed of microprocessors has increased rapidly over the last few years, a problem that hams are keenly aware of has surfaced. Specifically, the lead length on printed circuit boards has become an increasing problem, as the frequencies result in significant inductance in these leads.

To overcome this difficulty, manufacturers began integrating more and more functionality onto a single chip. The resulting lead-length is extremely small—microscopic in length in fact. But, the cost of a chip is proportional to the *area* of the chip, so extremely large-area ICs cost a great deal and have lower yield than small-area ICs.

nChip Inc of San Jose, California, has introduced a set of what it calls Multichip Modules (MCMs). One MCM from nChip is their nC2000, which is a 200-MHz (!) processor that also includes memory, timing and logic chips on a single pin-grid-array (PGA) chip carrier similar to the 80386 and 80486 packages. The chips are connected with copper traces and have individual groundplanes separated by dielectric layers. These reduce resistance and control transmission-line effects, thus allowing the incredible speeds.

The engineers design the interconnects and lay out the MCM for a given order. Then they take the customer's dies (single chips without any carrier) and mount them. These are, as you would expect, pretty expensive at the present time. Typical costs for large quantities are $50 each plus the cost of the customer-supplied chips.

However, you can expect to see these costs come down, as all costs do in IC technology. I would bet that this technology will prevail in many application areas in the next few years.

If you'd like to see what nChip is doing, drop them a line at: nChip, Inc, 1971 N Capitol Avenue, San Jose, CA 95132.

MICROLINEAR POWER CONTROL IC

Another product intended for the laptop computer market has some interesting possibilities for the ham and electronics experimenter. Micro Linear's ML 4860 is a power-control IC designed to give maximum life from a laptop computer's battery. Of course, these same benefits can be gleaned for any battery-powered project, such as an HT, a battery-powered QRP rig, or any of a multitude of applications.

The ML 4860 incorporates two 5-V regulated outputs (with 3 amps available from one), a 12-V output, a 2.5-V high-precision reference voltage and a low-battery indication output. Fig 1 shows the block diagram of the part.

In addition to the regulators, three logic inputs can be translated to another voltage, which could be handy in some applications. The chip also includes a battery switch control to disconnect the battery when an ac adapter is plugged in, and a low-battery output that can be used to prompt the user, turn off unnecessary devices, or any other function that would be appropriate. Also, the 12-V supply can be externally turned off with a logic signal.

More information on the ML 4860 is available from: Micro Linear Corp, 2092 Concourse Drive, San Jose, CA 95131.

ELECTROLYTIC CAPACITORS

Aluminum electrolytic capacitor development is not dead, despite the widespread use of tantalum capacitors. Illinois Capacitor has introduced a new line of low-leakage aluminum electrolytic capacitors. The capacitors are available from 0.1 to 1000 µF and in working voltages from 10 to 50 V dc. The primary advantage of these capacitors to tantalum is the price. The capacitors go for less than 10 cents in low quantities and less than a nickel for high quantities. Contact the manufacturer at Illinois Capacitor Inc, 3757 W Touhy Avenue, Lincolnwood, IL 60645.

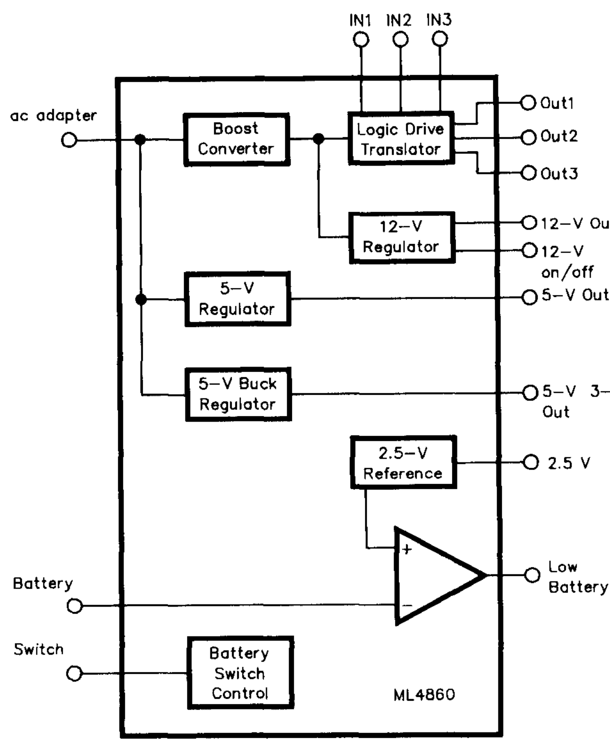


Fig 1

Conducted by: Stan Horzepa, WA1LOU
75 Kreger Drive
Wolcott, CT 06716-2755

DIGITAL COMMITTEE PRESENTS RECOMMENDATIONS FOR HF DIGITAL COMMUNICATIONS

The ARRL Committee on Amateur Radio Digital Communication has reviewed the results of the January 1992 *QST* survey on automatic unattended HF operation of digital stations and has submitted recommendations for ARRL Board consideration at its July 17 meeting.

A clear majority of survey respondents opposed fully-automatic, unattended operation on HF. However, by a ratio of two to one, respondents endorsed semi-automatic operation, where there is a control operator present at one end of the circuit. The Digital Committee recommendations are consistent with the survey results. The committee is recommending that FCC Rules be proposed to permit semi-automatic digital operation below 30 MHz, but not to permit fully-automatic operation. Neither type of operation is presently permitted except under a Special Temporary Authorization (STA) granted to ARRL that will expire next January. The recommendation includes language to protect other amateur operations from interference in the event of a malfunction of the unattended station.

The committee also recommends that the use of unspecified digital codes on HF be allowed with bandwidth limited to 500 Hz below 28 MHz and to 2 kHz between 28.0 and 28.3 MHz to encourage experimentation with more spectrum-efficient systems.

Finally, the committee recommends greater efforts by the League to educate amateurs interested in HF digital operations and to develop technical standards or guidelines for spectrum-efficient digital communications equipment.

ARRL Directors are now studying the recommendations of the Digital Committee in preparation for their formal consideration July 17. At that time, the Board will have the opportunity either to adopt the recommendations, decline to adopt them, adopt them in modified form or postpone consideration.

—from W1AW

NEW PACKET-RADIO SOFTWARE FOR AMIGA COMPUTERS

Black Belt Systems has released Packeterm, a Commodore Amiga specialized communications program designed to operate with TNCs such as the AEA PK-232, Kantronics KAM and similar units.

Packeterm has the ability to provide the packet-radio operator with a unique window for each connection in multiple-connect situations. For dual-port (or more) TNCs, it supports up to eight independent stream ID types. *Packeterm* is totally configurable for the operator's preferences, particularly with respect to providing custom control panels with buttons that can perform virtually any task required.

The Amiga's multitasking operating system provides the ideal environment for packet-radio operations; *Packeterm* operates either fully active or "iconified," allowing 100% monitoring of the packet-radio station by the user, even when another program is being used. This helps satisfy FCC's requirement for continuous monitoring of packet-radio stations used as gateways, etc.

Packeterm is designed to replace the hard-wired host-mode software which previously was considered the ideal way to operate these complex radio modems. In addition to a friendly mouse environment (with built-in on-line help), *Packeterm* can actively parse and operate upon incoming packets in any way the operator likes. For instance, at station AA7AS, *Packeterm* is connected to a Heath ID-5001 weather station and provides connectees with the weather report at their request.

Packeterm requires an Amiga computer with 512 kbytes of memory and Amiga DOS 1.3 or 2.0. The program includes a 244-page manual and sample configurations for AEA and Kantronics packet-radio controllers. For more information, phone 406 367-5509. To order *Packeterm* ($99.95), phone 800 852-6442.

—from Ben Williams, AA7AS, via CompuServe's HamNet

MACS KISS-OFF TNCS WITH *SOFTKISS*

Softkiss version 1.1.t permits a Macintosh computer to operate TCP/IP packet radio using a modem rather than a TNC. The program supports 300-baud HF, 1200-baud VHF, as well as higher data rates. The software is functional at this time, but the user interface is not pleasant (yet).

Softkiss is available from CompuServe's HamNet (filename KISS1T.HQX in library 9) or by Internet FTP (/aw0g/softkiss.1.1t.sit.hqx from akutaktak.andrew.cmu.edu [128.2.35.1]). Note that *Softkiss* requires Net/Mac which is also available from HamNet (filename MTCP22.SIT) or Internet FTP (/aw0g/netmac22.sit.hqx from the same source as *Softkiss*).

—from Aaron Wohl, N3LIW, via Internet

LOGGING SOFTWARE SUPPORTS PACKET-RADIO CLUSTER SPOTTING

LOG-EQF is a logging program for IBM PCs that provides complete TNC control including a monitor window and file transfer. It also tracks DX spots that appear on DX packet-radio clusters.

In addition to packet-radio support, *LOG-EQF* provides the following features:

- Menus with mouse support,
- Kenwood radio interface for control and display of radio parameters,
- Beam headings automatically calculated from your location,
- DXCC country, CQ and ITU zones, and continent determination from a call sign,
- Search or sort log by any field,
- Print logs and QSL labels,
- Note pads for each log entry,
- Previous QSO information presented automatically,
- Run other programs while *LOG-EQF* stays memory resident,
- Selectable date format with automatic UTC adjustments,
- Edit or delete any entry easily, and
- Contest support with QSO rate and multiplier tracking.

LOG-EQF is shareware. You can test drive it for the price of a disk, then register it for only $25. Registered users receive the latest version, full technical support and automatic notification of updates. File import/export and DXCC tracking modules are also provided to registered users.

LOG-EQF is available from your local shareware distributor, BBS, CompuServe's HamNet (filename LOGEQF.ZIP in library 9) or send $5 (US), $8 (outside US) to: EQF Software, Tom Dandrea, N3EQF, 396 Sautter Drive, Coraopolis, PA 15108, tel 412 457-2584.

NEW PACKET-RADIO SOFTWARE AVAILABLE

The following new packet-radio software became available this spring. You may download all of these programs from CompuServe's HamNet library 9 (unless indicated otherwise). They also may be available by other means such as TCP/IP FTP, landline BBSs and users' groups.

Apple Macintosh Software

- *IM/Mac* 1.0b24 Filename: IM_MAC, text, 123211 bytes, TCP/IP SMTP mailer.
- *MacDOVE Packet Decoder* Filename: MACDOV.SIT, binary, 35072 bytes, HamNet library 5, DOVE satellite telemetry decoder.
- *SatTrak* 1.01 Filename: SATRAK.SIT, binary, 258432 bytes, HamNet library 5, program for tracking satellites.

Commodore Amiga Software

- *Satellite Tracking Program for Amiga* Filename: SATTRA.LZH, binary, 190720 bytes, HamNet library 5, program for tracking satellites.

IBM PC Software

- *7PLUS200.EXE* Filename: 7PLUS2.EXE, binary, 161450 bytes, Binary file encoder and decoder.
- *Anti-Burnout Screen Saver for Packet* Filename: BURNOU.ZIP, binary, 15484 bytes, screen-saver for those who monitor packet-radio nodes or DX packet-radio clusters.
- *APLINK Version 6.04* Filename: APL604.EXE, binary, 320942 bytes, mailbox and PBBS software that supports packet-radio and AMTOR ports.
- *BPQ405B.EXE* Filename: BPQ45B.EXE, binary, 166585 bytes, packet-radio switch software.
- *Enhanced PACKHACK* Filename: PHACK3.ZIP, binary, 8960 bytes, program that analyzes packet-radio channel activity. Lists call signs of packet senders with transmitted packets sorted according to frame type. Requires a TNC 2 or TNC-2 clone.
- *HyperLog 2.11* Filename: HYLOG2.EXE, binary, 306825 bytes, HamNet library 6, logging program that includes a packet-radio interface and packet "Speed Keys."
- *Improved Kansas City Tracker/Tuner Software* Filename: KCT500.ZIP, binary, 45952 bytes, HamNet library 5, satellite antenna tracking and Doppler shift tuning software. Includes automatic downlink polarity switching.
- *KaGOLD ver 7.0 for KPC and KAM v 5.0* Filename: KAGOLD.EXE, binary, 346008 bytes, demonstration packet-radio terminal program for host mode Kantronics KPC TNCs and KAM multimode controllers. Reports frames outstanding, tries, packet-radio and AMTOR mode states. Supports file transfers, conference mode (automatic and manual) and includes on-line help concerning TNC setup, parameters and operation.
- *LAN-LINK 2.00* Filename: LL200E.ZIP, binary, 363520 bytes, packet-radio terminal program (shareware).
- *OS/2 Multi-mode pgm for the PK-232 using hostmode* Filename: PHSOS2.ZIP, binary, 170612 bytes, OS/2 multimode terminal program for the AEA PK-232 host mode. Features user-configurable COM ports, split-screen operation, command and parameter entry in mode-sensitive dialog windows. Supports binary file transfer using YAPP, reviewable received text, "snapshooting" the review-buffer to a file, logging to a file, text sending from a file. Also features Heard lists that show paths and a built-in message editor.
- *OS/2 PR pgm for TNC with WA8DED firmware* Filename: THSOS2.ZIP, binary, 128127 bytes, OS/2 packet-radio terminal program for TNCs using WA8DED's replacement firmware. Has same features as OS/2 Multi-mode pgm for the PK-232 using hostmode described above.
- *paKet v 5.1* Filename: PAKET.ZIP, binary, 544611 bytes, packet-radio terminal program that supports four binary file transfer protocols and up to 10 simultaneous windows. Includes on-line TNC help, on-line manual, remote access personal message

system, contest mode and scripting. (Shareware)

- *PCTOR V1.13* Filename: TOR113.EXE, binary, 58952 bytes, HamNet library 6, AMTOR emulator that requires an external HF modem on COM 1 or 2. Includes a multiwindowed user interface that can send files, canned buffers, log to disk. It supports the extended ASCII character set for APLINK compatibility.
- *PkGOLD Enhanced TestDrive* Filename: PKGOLD.EXE, binary, 340049 bytes, demonstration terminal program for all AEA TNCs. It supports AMTOR, Baudot, Morse, packet radio, NAVTEX and "MAILDROP" even while connected. Includes full on-line help system with data about TNC parameters, operation and technical information. Runs in DOS, Windows, DESQview, OS/2 and 8088 to 80486 systems.
- *pktbat13.zip* Filename: PKTBAT.ZIP, binary, 42917 bytes, IBM XT (and greater) program that allows you to play the game of battleship via packet radio. Includes a window to converse with the other player while game is running.
- *TXM* Filename: TXM153.ZIP, binary, 254037 bytes, program that manages interfaces to landline BBSs from packet-radio stations to permit using landline BBS software for packet-radio connections. Improves backbone mail store-and-forward with LZH data compression and other enhancements, while maintaining connectivity with existing MSYS, WRLI and AA4RE PBBSs.
- *WA6AEO Directory Check Server* Filename: DIR20.ZIP, binary, 21504 bytes, server program for PBBS SYSOPs. It checks for new files in a directory and sends a message for PBBS import. Includes a database. Can handle multiple directories.
- *WA6AEO Message Editor* Filename: MSGS14.ZIP, binary, 25600 bytes, program that allows you to compose messages off-line for uploading to your PBBS later. Includes a database. Can run in a DESQview window.

TNC Firmware

- *TAPR TNC-2 Firmware Version 1.1.8* Filename: TNC118.ZIP, binary, 36645 bytes, version 1.1.8 of the TAPR TNC-2 firmware featuring a new host mode, a QRA command and minor other changes.

TRS-80 Model 4 Software

- *Packet Terminal 80* Filename: PT80.CMD, binary, 21760 bytes, packet-radio terminal program for AEA PK-88 TNCs and PK-232 multimode controllers. Features multiconnect screens, monitor, locally editable input line, disk capture, ASCII uploading and downloading.

THE WB3FFV AMATEUR RADIO TELEPHONE BBS

WB3FFV has placed a 24-hour-a-day BBS system on-line that is mainly oriented towards the Amateur Radio community. Access to the BBS is provided via the following telephone numbers: Number 410 661-2475 supports 1200, 2400, 4800, 9600, 19200, 38400, V.32/ V.42/ V.42bis/ MNP1-5.

Number 410 661-2598 supports 1200, 2400, 4800, 7200, 9600, 12000, 14400, 19200, 38400, V.32/ V.32bis/ V.42/ V.42bis/ MNP1-5/ HST.

Number 410 661-2648 supports 1200 and 2400 MNP1-5/ V.42bis, 9600 and 19200 PEP.

Data settings are 8 bits, no parity, 1 stop bit. User login is "bbs." Some of the items that are available for downloading include:

- KA9Q TCP/IP software for the PC (latest official release and test versions),
- KA9Q TCP/IP for the Atari-ST, Apple Macintosh and Commodore Amiga,
- KA9Q TCP/IP for UNIX-based systems,
- KA9Q TCP/IP (The NOS release) (UNIX, MS-DOS and Commodore Amiga),
- KA9Q TCP/IP (version by G1EMM, PE1CHL, PAGRI, etc),
- N2GTE Packet Message Switch (GTEPMS, version 1.2 and 1.3),
- WA7MBL BBS for the PC (versions 3.31, 4.31 and 5.1[2,3,4]),
- WRLI BBS for the PC (versions 10.xx, 11.xx, 12.xx, 13.xx),
- MSYS BBS for the PC running KISS TNCs (version 1.07-1.11),
- AA4RE BBS for the PC (version 2.11),
- F6FBB BBS for the PC (version 5.14 and updates), G8BPQ PC-Node (version 4.05),
- Various BBS utilities and enhancements,
- Several Morse Code tutors,
- Modifications for many ham radios and scanners,
- Digital Signal Processing (DSP) software,
- DX and contesting programs,
- ARRL newsletters and Gateway, and
- W5YI Electronic Edition.

Files may also be FTP'd by packet-radio TCP/IP from 44.60.128.1 (wb3ffv.ampr.org). If you are interested in using UUCP to connect to the BBS, this can be done as it supports Anon-uucp. The login to the system is "uucpanon" and there is no password. The listing of available archives are stored in a file called "FILES" that is located in /usr/spool/uucppublic. To retrieve the files listing, use the following command:

uucp wb3ffv!~/FILES /usr/spool/uucppublic

This will move a copy of the files listing into your uucppublic directory.

—from Howard Leadmon, WB3FFV, via Internet

DISCLAIMER

The descriptions of hardware and software contained in this column are merely reports of what is being offered in the Amateur Radio marketplace. Your Gateway contributing editor, *QEX* and the ARRL in no way warrant the operability, functionality, suitability, or even the existence of what is being offered.

GATEWAY CONTRIBUTIONS

Submissions for publication in Gateway are welcome. You may submit material via the US mail to 75 Kreger Dr, Wolcott, CT 06716-2755, or electronically, via CompuServe to user ID 70645,247, or via Internet to horzepa@evax.gdc.com. Via telephone, your editor can be reached on evenings and weekends at 203 879-1348 and he can switch a modem on line to receive text at 300, 1200 or 2400 bit/s. (Personal messages may be sent to your Gateway editor via packet radio to WA1LOU@N4GAA or IP address 44.88.0.14.)

The deadline for each installment of Gateway is the first day of the month preceding the issue date of *QEX.*

InstantTrack 1.00 Patch Available

Patch #2 to *InstantTrack* 1.00 is now available. This patch cures a bug in Patch #1 to *InstantTrack* that could cause system crashes or other unpredictable behavior when reading an ill-formatted NASA format element file. Users who have installed Patch #1 are advised to install Patch #2 as soon as possible to avoid any problems.

Like Patch #1, Patch #2 is designed to solve the problem created by the recent change in NORAD's computation of the checksum in its 2-line element sets. NORAD changed the value of the plus sign from 2 to 0. Users can work around the problem by replacing each plus sign in the element set file with a space. With Patch #2, this manual operation is not required. The patched *InstantTrack* 1.00b will accept NASA format element sets that use either checksum formula.

Patch #2 is available as ITPATCG2.ZIP or ITPAT2.ZIP on CompuServe's HamNet and several land-line BBS systems, and is also available for broadcast download on AO-16 and LO-19. Patch #2 should soon be available from AMSAT-NA headquarters; call AMSAT- NA for further details (301 589-6062). The patches were created by Paul Williamson, KB5MU.

EEpal

Eagleware has announced a new program, *EEpal,* which intergrates a variety of electrical engineering and design activities in a program for IBM personal computers. *EEpals* features are based in "screens" which are selected by a menu system. EEpal includes 240 screens of ready-to-use formulas, data tables and other functions.

EEpal is programmable. The user may add up to 500 new formulas and data by simply typing them onto a screen. User-added screens become part of the menu-accessed system.

EEpal users may also: tune formulas and watch results change, auto-dial numbers from hundreds of entries in the phone lists, pick up and drop data to and from other programs, curve fit, find Fourier coefficients, solve equations, and use an alarm with appointment list, scientific calculator, matrix calculator, and editor.

EEpal runs in DOS, Windows 3.0, or TSR modes on IBM PCs and compatibles. Only 10 k RAM core is required in TSR mode. A hard drive is required. A modem is required for auto-dial features.

For pricing and availability, contact Randy Rhea, Eagleware (formerly Circuit Busters), 1750 Mountain Glen, Stone Mountain, GA 30087, tel 404 939-0156.

Circuit Search Data Base

Circuit Search announces the release of Version 1.09 of the Circuit Search data base of references to articles containing practical electronic circuit designs.

This dBASE III/dBASE III+ compatible data base contains references to nearly 13,000 articles and papers from over 300 technical and scientific journals and magazines. As an interdisciplinary reference source, circuits can be located by keywords from journals in fields as varied as electronics, astronomy, agriculture, physics, chemistry, nuclear science, education, biomedicine, and many more disciplines where electronics can be applied.

The Circuit Search data base is installed on hard disk on an IBM PC or compatible, and used in conjuction with either its own user-friendly menu-driven front end or a dBASE-compatible data base management program. The data base occupies about 7 Mbytes and comes on a series of diskettes.

Pricing (US dollars): Circuit Search data base, $375 (includes 1 free semi-annual update/revision); 1 semi-annual update, $40; 2 semi-annual updates, $65. A sample diskette including 400 sample references is available for $7.00.

For additional information contact Peter Sawatzky, Sales Manager, Circuit Search, PO Box 268, Breslau, Ontario, Canada N0B 1M0, tel 519 241-1252, fax 519 742-4594.

Personal Radio Instrumentation Catalog

Optoelectronics Inc offers a new 16-page brochure describing the firm's newest hand-held and bench-top instruments.

Contents include descriptions, technical data and tips on how to use frequency finding hand-counters, universal counter-timers for lab and field, PC-based counters with Windows 3.0 for control and display, active preselector bandpass filters and antennas and accessories.

For a copy of this free brochure, contact Bill Owen, Optoelectronics Inc, 5821 NE 14th Avenue, Fort Lauderdale, FL 33334, tel 800 327-5912 or 305 771-2050.