QEX 13 February 1983

# The ARRL Experimenters' Exchange

## Second ARRL Packet Conference

The ARRL's Second Amateur Radio Computer Networking Conference will be an all-day meeting held on Saturday, March 19, 1983 in a room set aside within the West Coast Computer Faire in San Francisco, CA. The Faire itself runs from Friday, March 18 through Sunday, March 20. So, if you are interested in participating, the best scheme is for you to plan to see the Faire on Friday so that you can be free to attend the packet gathering on Saturday. The exhibits are not quite so crowded on Friday; however, it's elbow-to-elbow exhibits on Saturday and Sunday. Alternate Plan B would be for you to attend the packet conference on Saturday and the Faire exhibits on Sunday afternoon.

Technical papers will be presented by a wide variety of speakers. Here is a partial list of conference speakers:

Den Connors, KD2S
Terry Fox, WB4JFI
Dave Henderson, KD4NL
Hank Magnuski, KA6M
Dan Morrison, KV7B
Margaret Morrison, KV7D
Harold Price, NK6K
Paul Rinaldo, W4RI

The program will include:

- the evolution of packet radio,
- AX.25 packet protocol,
- several aspects of the development of the Tucson Amateur Packet Radio (TAPR) Terminal Node Controller (TNC) hardware and software,
- Packet Assembler/Disassembler (PAD) and Packet Adaptive Modem (PAM) designs by the Amateur Radio Research and Development Corp. (AMRAD).
- The Pacific Packet Radio Society (PPRS) will conduct live packet radio demonstrations throughout the Faire.

Participants are urged to preregister by sending a check payable to "Henry S. Magnuski" in the amount of $19 (to cover both Faire and the packet conference tickets) to:

311 Stanford Ave,
Menlo Park, CA 94025.

The price includes a personal copy of the proceedings which will contain all of the packet conference papers. Tickets will be mailed to you. Late registration at the door will be possible at slightly increased price and inconvenience. Also, there will be an opportunity to purchase copies of the conference proceedings by mail after the conference for those who are unable to attend. The price and where to order the copies will be announced later.

Plans are for a luncheon for those interested at the nearby Salmagundi Resaurant on Saturday at noon. Saturday evening will be open for random access.

Anyone wishing to appear on the program or have a paper included as part of the proceedings should contact Paul Rinaldo, W4RI, on 703-734-0878 (days/eves) immediately. There is a little flexibility in the previously announced February 5 deadline for camera-ready papers, but not very much and still get the proceedings printed in time for the conference.

## Doug DeMaw, W1FB Announces Retirement

Doug DeMaw, W1FB has announced his intention to retire as ARRL Technical Department Manager in May 1983. He and his wife, Jean, will move to their 40-acre farm in central lower Michigan. Doug has made many contributions to the technical part of QST and to the numerous ARRL books produced over his time at Hq. That will not end, however. Doug will continue work for the League as a part-time contributing editor and columnist for QST, along with tree farming and some consulting.

## QEX and QST Articles

QEX is in need of manuscripts on rf/analog subjects such as receiver and transmitter design, ATV, modulation techniques, vhf/microwave experimentation, design techniques to minimize RFI. Also, it would be useful to have some articles on modifying existing types of hf antennas to provide coverage on the new WARC bands. Also welcome are articles on radio interfaces for personal computers, packet radio and other digital communications topics.

In the QEX queue for upcoming issues are articles on minimum-shift keying (bibliography), Sinclair ZX81 RTTY receive program, diode voltmeters, Ten-Tec transceiver modifications, 88-mH inductor stacks, an hf frequency standard, the Novation D-Cat modem, and more. Most of the articles in the queue are feature-length, and it is not possible to fit more than one of the longer ones in each issue and stay within budget. The shorter ones of about one printed page in length are easy to fit in and get published without much delay.

Also, keep sending letters to the editor with comments on articles, tips for experimenters and wish-list items.

QST is looking for construction articles of widespread appeal. There is a particular need for material about microwaves, notably hardware articles that show how to build one's own microwave gear. Other topics of high interest are hf linear amplifiers, receivers, band-switching solid-state transmitter and test equipment. There is no need at this time for tutorial text or articles about keyers, antenna tuners or antennas.

# Correspondence

### Automatic QSP Design Needed

A need has arisen in our emergency services work for a cross-band patch variously referred to as a portapeater, bonzai box and, undoubltedly, many other names! The objective is to provide extended-range communications through the use of readily available transceivers which, when coupled together through "the black box" would allow automatic relaying of local transmissions (on voice or high speed data) to more-distant points not readily accessed from the originating point.

Three example situations are:

A. Using two in-hand vhf transceivers (one on 2 meters and one on 1 1/4 m) to be employed without accessing the innerds of either radio, but merely using the audio out and microphone in and push-to-talk of each to comple to "the black box" which would handle level, impedance and VOX operations as a portable repeater to temporarily substitute for a repeater out of service in an emergency, or one not available in a desired location. As envisioned the device would not require the features of a permanent repeater including i-d, time-out timer and other more-exotic features. Operation would entail the use of adapting plug/cables to "any" readily available pair of transceivers which would be placed on a hill and "baby-sat" by a mobile unit or even left unattended, if necessary. Use of two radios in the same band would, of course, require sufficient antenna separation which could also be done if essential.

A1. Using the same example as above but with one of the transceivers being on hf-ssb/fsk and experiencing some noise. Here, in order to achiieve the same results a "smart" squelch might be used on the hf receiver.

B. Emergency operations here are now starting to employ a number of vhf/uhf units located at some emergency operating center with a need to be located in some rather poor rf locations (basement of police department, or in a shelter) and having a need to communicate for considerable distances. A low-powered vhf/uhf simplex radio (located at the same site as a number of other base/mobile/handheld units) carrying, for example, digital data destined to feed terminals many miles away over the hills can communicate with a nearby amateur station having good antennas and height advantage and able to operate from a stable location while automatically relaying the two-way transmissiion via vhf or hf without intermod problems via "the black box." (A cw i-d insertion could be used here as long as the tone does not conflict with either 2125/2295-Hz (RTTY) or the Bell 103 tones of 1070/1270 and 2025/2225 Hz.) State Disaster Support Area bases expect to utilize Amateur Radio during initial hours of a disaster and will likely have any number of radios and a need for more than one hf path to handle the volume of traffic among affected counties and to the state capitol. Field Day experiences have shown us for years the impracticability of trying to locate two or more 40-meter stations at the same site. Using "the black box" a number of vhf simplex paths can be generated from the disaster comm center going out to a variety of scattered hf installations - paths increasing in number based on the traffic volume - each on a different 40/80-meter frequency to the distant point(s) and reverse black box links down into the distant comm center(s).

The "black box" needs to be simple, use a given standard plug arrangement which can be adapted to by each radio owner, operated from the same 12-volt supply, contain a common speaker and microphone for the operator who baby-sits the box. Level controls, impedance matching, some LED or level-setting indicators and VOX/anti-VOX adjustments would seem necessary.

I've been trying for some time to locate someone in my Section who would take on this project without success. Apparently it takes someone who is "emergency" oriented as well as one who has a feel for what we're trying to accomplish.

One other important element I should mention is that the system should be capable of passing high-speed data - maybe 1200 bauds. Tests with government are now starting to link Z80-CP/M-compatible micros via Amateur Radio and the state microwave system.

I'd appreciate any ideas and help you might be able to provide. - Robert N. Dyruff, W6POU, SCM - Santa Barbara Section, 1188 Summit Road, Santa Barbara, CA 93108, 805-969-3073.

### Finding Hidden Signals

Please find enclosed a clipping from the Wall Street Journal of December 17. I have bracketed an item that I think is of interest to the readers of QEX. - Michael S. Bilow, N1BEE, Forty Plantations, Cranston, RI 02920.

Ed Note: The clipping says that a west-coast firm has developed a receiver that can find hidden signals (even when they are brief, or when they use frequency hopping) using "instantaneous direction finding" technology.

### Teleconference Radio Net Changes

Please make the following changes to the information which appeared in QEX 12:

The Phoenix repeater's call sign is now KA7DSY/R. Cherryville, NJ was incorrectly listed as Cherry Hill.

Add: Milwaukee WI WB9ZCT/R 145.13
Baltimore MD W3VPR/R 147.105
San Francisco CA (to be determined)

On Thursday, June 2 at 8:30 EST, Joe Riesert, W1JR, will speak on "Antennas and Antenna Systems, Where is the State of the Art Going?". - Rick Whiting, WØTN, 4749 Diane Dr, Minneapolis MN 55343, 612-870-2071 (days).

### Want to Try CCW Experiments with Hungary?

We have a few students dealing with the theoretical and practical problems of coherent cw. We should be very happy to find radio amateurs in the USA to arrange common experiments on 14 MHz this summer. Our freq. reference is an HP 10544A standard. A Gschwindt, HA5WH, Tusnadi u 25, H-1125 Budapest, Hungary.

### Feedback on VADCG TNC 2716 EPROM Modification

Robert Anderson has used a diode AND gate to provide a chip-select signal for the 2716 PROMs used on a modified TNC board. The logic is correct, but the use of the diode circuit may cause problems.

The logic zero level at a standard TTL output is about 0.1 volt, and most TTL-compatible inputs require that a logic zero input level be between 0.0 and 0.8 volts. Unfortunately, if the diodes and pull-up resistor are used, the chip-select voltage for the 2716 PROMs may be about 1.0 volt which is outside of the limit specified by the manufacturer. I'd recommend using a standard 7400-series AND gate, even though it may have to be "kluged" onto the TNC board.

(continued on page 11)

# PROM Programmer/Reader and Utility Software for 2708 and 2716 PROMs

By G. M. Palmer,* K8LG

While there have been many articles on PROM programmers in the past, this article offers a programmer and comprehensive software that can be used with any microcomputer system having two programmable 8-bit parallel I/O ports and running CP/M (tm Digital Research) with Microsoft BASIC version 4.51 or higher. The programmer and software described here will allow the user to:

read PROMs,
program PROMs,
compare PROMs,
read data files,
read hex files,
print the contents of PROM,
etc.

This combination of hardware and software has made the use of PROMs in developing microprocessor controllers and hobby projects effortless. The programmer may be constructed at very little cost and will do the same job as a much-more-expensive one.

**Programmer**

The PROM programmer is constructed on a Vector model 3662 board that measures 4.5 by 6.5 inches. The wiring was done using standard wire-wrap methods. Construction is not critical if one follows good bypassing and power-distribution practices. The input and output connections are all made on the 44 pins along the edge of the board. The only critical wiring on the entire project is the cable that connects the programmer board to the I/O ports of the computer. This cable should be of the flat ribbon type, no longer than three to four feet, and have ground wires every other wire (as the 34-conductor disk cables). It was found that if this practice was not followed it was possible to get crosstalk between the lines.

The logic diagram of the programmer is shown in Fig. 1. To make the project less complicated, two PROM sockets were used (one for 2708, U4, and one for 2716, U5). The first I/O port (Port A) is used to send data to and from the PROMs and is connected directly to the I/O lines of both PROM sockets. The address lines of the two PROM sockets are connected to output lines of three 4-bit binary counters (U6, U7 and U8). These counters are used to set the address lines on the PROMs. The address counters are set to zero by toggling bit 0 of Port B high and then low. For normal counting the bit 0 is held low. To advance the PROM address, bit 2 of Port B is toggled high then low. This action also triggers a single shot, U1A, that is used to generate a 10-us flag to allow software polling when operating at high speed. This flag, which is low during the count advance, is available on Port B bit 7.

*Mechanical & Aerospace Engineering, West Virginia University, Morgantown, WV 26506.

The hardware shown here is also used to read PROMs as well as program them. The read/write status is controlled by Port B bit 3. When bit 3 is low the PROMs may be read, and if it is high they may be programmed. To program a PROM, a pulse or series of pulses must be provided by the programmer. The length, number, and sequence of pulses will depend on the PROM type. The 2708 requires a 1-ms 25-27-volt pulse on pin 18 with valid data on the data lines. Each address is programmed in sequence this way to complete one cycle. One hundred cycles are required to program the 2708. The 2716, on the other hand, requires 25-27 volts on pin 21 during programming with a 50-ms pulse (TTL) on pin 18 with valid data on the data lines. Only one cycle through the addresses is required with the 2716.

The required pulses for programming the 2708 and 2716 are generated by single shots. The 1-ms 2708 pulse (U1B) is triggered by toggling Port B bit 1 high and then low. The TTL output of U1B is then used to generate the 25-volt pulse by transistors Q1 through Q3. The status of the pulse is available on Port B bit 6 (low while pulse is in progress). The 50-ms 2716 pulse is generated by U2 when bit 4 of Port B is toggled high and then low. The status of this pulse is available on Port B bit 5 (low during pulse).

**Reading PROM**

To read a PROM, either 2708 or 2716, the following sequence is used:

1) set Port a as input to CPU,
2) set read/program line to read (Port B bit 3 low),
3) reset address counter (toggle bit 0 low, high, low),
4) read data from PROM through Port A,
5) advance the address (toggle bit 2 low, high, low), and
6) repeat steps 4 and 5 for all addresses.

**Programming PROM**

*2708:* The 2708 is the more complicated of the two PROMs to program in that it requires 100 passes through the addresses for complete programming. The steps shown below are for one cycle which will need to be repeated 100 times.

1) set Porta A as output from CPU,
2) set read/program line to Program (Port B bit 3 high),
3) turn on program power switch (SW1),
4) reset address counter (toggle bit 0 of Port B),
5) output data byte to Port A,
6) pulse 2708 program line (toggle bit 1 of Port B),

7) wait until pulse is over (look for bit 6 to go high),
8) advance address (toggle bit 2 of Port B), and
9) repeat steps 5 through 8 for all 1024 addresses.

**2716:** The process for the 2716 is the same as that for the 2708 except that the program pulse is on bit 4 and the status is on bit 5. The 2716 needs only to be cycled through its addresses once.

**Software**

A PROM programer/reader to be useful must have software to drive it. After using several development systems that included PROM utility programs, it was decided that the software should be able to:

1) read PROM to memory,
2) program PROM from memory,
3) compare PROM with memory,
4) write from memory to disk file,
5) read to memory from disk file,
6) read to memory from Intel hex file,
7) display memory to screen or printer,
8) change memory data, and
9) fill memory with fixed data.

All of these features have been included in the software shown in Fig. 2. The software is written in Microsoft BASIC 4.51. The trade off between using BASIC or assembler was between ease of programming and speed of operation. With the system used (Z80 CPU with 3-MHz clock) it takes about 15 minutes to program a 2708 and 2.3 minutes to program a 2716. For the hobby user or a person who needs to program only a few PROMs a week, the longer time required by BASIC is no problem.

The two ports used for the programmer are in a single MC6821 which is memory mapped at FD00 (Port A) and FD02 (Port B). All the PEEKs and POKEs to these two addresses (variables PA and PB) in the listing are for I/O. The program easily could be altered to use I/O devices other than the MC6821 and port-addressed I/O. With port-addressed I/O, INP and OUT commands would replace the PEEKs and POKEs.

The program is set up so that all data and addresses are in hex. Data is moved between PROM, program array (variable D(I)), disk files, and memory buffer. When the program is started, the user must specify PROM type (2708 or 2716) which sets the size of the data array used (either 1024 or 2048). Once the size is set, all data transfers will be for that type of PROM.

**Program Function**

The program has eleven functions that are specified by two-letter abbreviations. These are summarized below.

**RH** - Read Intel Hex data file. This function will read an Intel hex object file from disk to the program array. The CP/M assembler generates these files during assembly.

**RD** - Read Data file. This function reads a sequential file to the program array and is used to recover data stored by the WD function.

**WD** - Write Data file. This function writes the data from the program array to a sequential data file.

**RP** - Read PROM. Read the data from a PROM in the PROM socket into the program array.

**WP** - Write PROM. Program the data in the program array into the PROM in the PROM socket.

**DD** - Display Data. This function displays the data in the program array on the system console or line printer in hex.

**CP** - Compare PROM. The data in the program array is compared with the data in the PROM in the PROM socket.

**RB** - Read Buffer. This section reads data from a memory buffer to the program array. This is useful when one wants to program a PROM with a program that is being developed in memory on the system.

**CD** - Change Data. The data in the program array may be changed one byte at a time with this function. The address in PROM is specified, the current data, and a prompt for the new data is displayed. The new data is then typed in.

**FM** - Fill Memory. Many times the entire PROM is not to be programmed, in which case the bytes not used should be filled with hex FF since the 27XX-series PROMs program from one to zeros (i.e., a fresh PROM has all FFs). The FM function allows the user to fill any block of addresses in the program array with any data in the range 00 to FF.

**EX** - Exit Program. This brings the program to a stop.

Using the PROM Programmer/Reader is quite simple. Several examples that illustrate the use of the syststem follow.

**Example A:** To program a PROM from an Intel hex object file that has been created by the CP/M assembler, one would: (see example program dialog A)

1) specify if the PROM is 2708 or 2716,
2) place a fresh PROM in the appropriate socket,
3) read the hex file (RH),
4) program the PROM (WP), and
5) compare the PROM with program array (CP) to check for correct programming.

Quite often in development work one has a system using PROMs up and running which need some fine tuning that requires only a few locations in the PROMs to be changed. This can be accomplished as follows: **(Example B)**

1) specify PROM type,
2) place old PROM in socket,
3) read the PROM to memory (RP),
4) change the data (CD),
5) replace the old PROM by fresh one,
6) program the PROM (WP), and
7) compare the data (CP).

Checking a fresh PROM (one that has been erased or is new) may be done as follows: **(Example C)**

1) specify PROM type,
2) place PROM in socket,
3) fill memory array with FFs (FM), and
4) compare the PROM with memory (CP).

If any address contains other than FF they will be displayed on the console. This will verify that the PROM is, in fact, erased.

Reading a PROM, displaying the data, and writing a disk file may be done as follows: **(Example D)**

1) specify PROM type,
2) place PROM in socket,
3) read the PROM,
4) display program array (DD), and
5) write the data file (WD).

Reading a data file and programming a PROM may be accomplished as follows: **(Example E)**

1) specify PROM type,
2) place PROM in socket,
3) read data file (RD),
4) program the PROM (WP), and
5) compare the PROM data with memory array (CP).

**Conclusions**

The hardware and software shown here have been used for a year or so without problems. The use of the programmer and the associated program have been very helpful in the development of several microprocessor-controller applications. The thought of using BASIC for the program was at first dismissed because of speed, but as time passed it was found to be very satisfactory. The penalty paid in speed with BASIC so far has been outweighed by the ease of program development.

While the program and hardware are centered around the I/O ports that are on the author's system, the changes to make them useful on another system should not be too difficult. Most of the popular microcomputers use versions of BASIC developed by Microsoft and similar file I/O statements, making conversion easy.

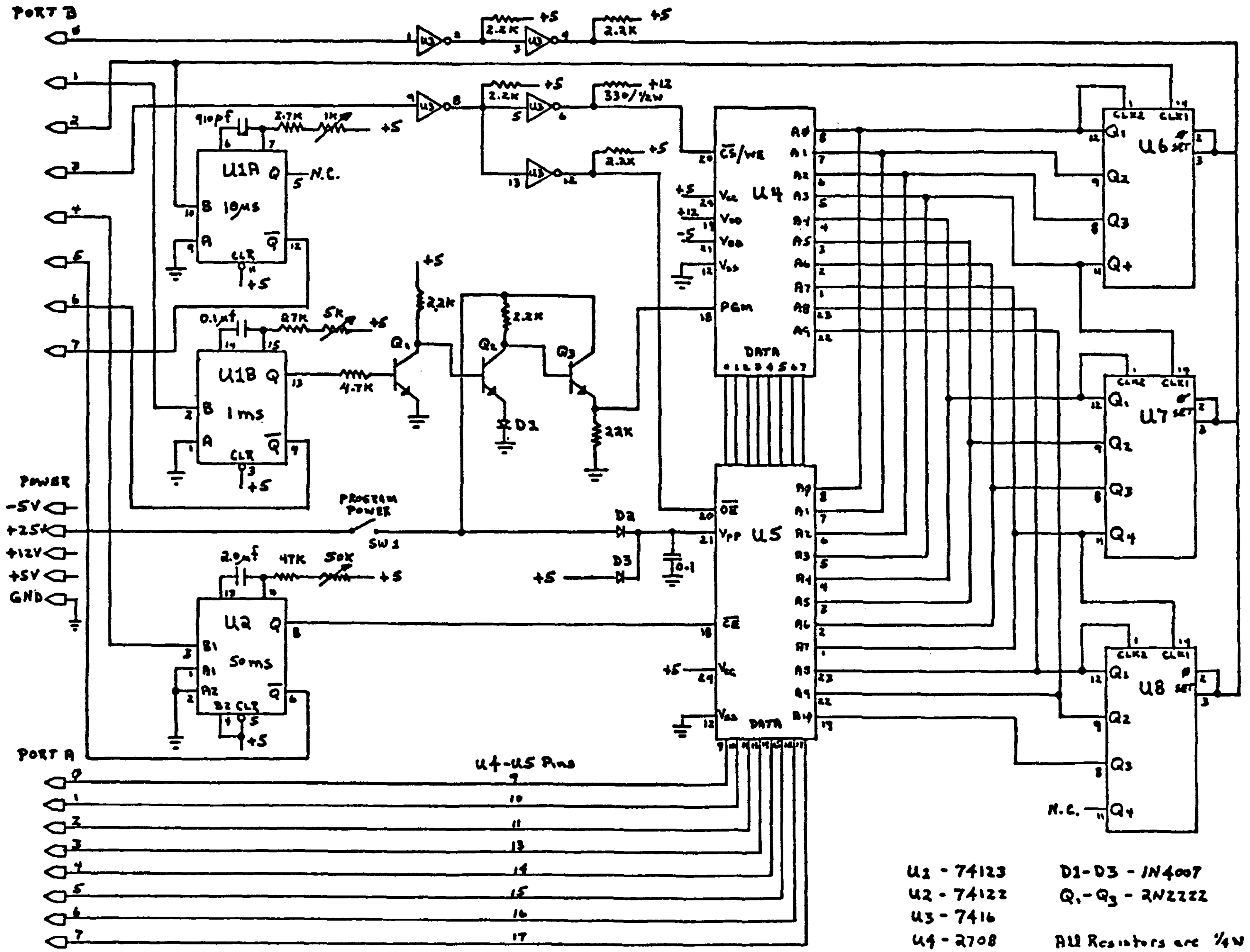



Figure 1. PROM Programmer.

U1 - 74123
U2 - 74122
U3 - 7416
U4 - 2708
U5 - 2716
U6-U8 - 7493

D1-D3 - IN4007
$Q_1$-$Q_3$ - 2N2222

All Resistors are ¼W 5% unless noted.
Chip by pass caps. are not shown.

```
PROM.BAS   PAGE  1


10 '
20 '
30 '   PROM   A 2708 AND 2716 PROM UTILITY PROGRAM
40 '   by: G. M. Palmer
50 '
60 '
70 DEFINT A-Z
80 DIM D(2048),CMD$(11)
90 PA=&HFD00
100 PB=PA+2
110 '
120 '   SET PORT A ALL BITS TO BE INPUT TO CPU
130 '   SET PORT B BITS 0 THRU 4 TO BE OUTPUT FROM CPU
140 '   SET PORT B BITS 5 THRU 7 TO BE INPUT TO CPU
150 '
160 POKE PA+1,0
170 POKE PB+1,0
180 POKE PA,0
190 POKE PB,31
200 PUKE PA+1,28
210 POKE PB+1,28
220 POKE PB,0
230 PRINT "COMMAND SUMMARY (Y OR N): ";
240 LINE INPUT F$
250 IF F$="Y" THEN GOSUB 2530 : GOTO 310
260 IF F$="N" GOTO 310
270 GOTO 230
280 '
290 '   READ COMMANDS
300 '
310 FOR K=1 TO 11
320 READ CMD$(K)
330 NEXT K
340 DATA RD,RP,WD,WP,DD,CP,RB,CB,RH,EX,FM
350 '
360 '   INPUT COMMAND, LOOK UP, AND SENT TO  CORRECT PROGRAM SECTION
370 '
380 PRINT : PRINT
390 PRINT "TUKN OFF PROGRAM POWER"
400 PRINT : PRINT "PROM TYPE (2708 OR 2716) : ";
```

```
PROM.BAS   PAGE  2


410 LINE INPUT TYPE$
420 IF TYPE$="2708" THEN TYPE=1 : GOTO 450
430 IF TYPE$="2716" THEN TYPE=2 : GOTO 450
440 GOTO 400
450 N=TYPE*1024
460 PRINT : PRINT
470 PRINT "FUNCTION : ";
480 LINE INPUT F$
490 J=0
500 FOR K=1 TO 11
510 IF CMD$(K)=F$ THEN J=K
520 NEXT K
530 IF J<>0 THEN GOTO 560
540 PRINT "NOT A VALID FUNCTION"
550 GOTO 460
560 ON J  GOTO 670,1470,1150,1590,830,1320,2160,2310,2710,600,2930
570 '
580 '   EXIT PROGRAM
590 '
600 PRINT
610 PRINT "EXIT PROM"
620 PRINT
630 STOP
640 '
650 '   READ DATA FILE FROM DISK TO PROGRAM ARRAY
660 '
670 PRINT
680 PRINT "READ DATA FILE"
690 PRINT
700 PRINT "DATA FILE NAME: ";
710 LINE INPUT F$
720 OPEN "I",#1,F$
730 FOR K=1 TO N
740 INPUT #1,D(K)
750 NEXT K
760 CLOSE
770 PRINT
780 PRINT "DATA FILE ";F$;" READ WITH ";N;" DATA VALUES"
790 GOTO 460
800 '
```

```
PROM.BAS   PAGE  3


810 '   DISPLAY DATA FROM PROGRAM ARRAY TO CONSOLE OR PRINTER
820 '
830 PRINT
840 PRINT "DISPLAY DATA"
850 PRINT
860 PRINT "OUT PUT TO CONSOLE OR LINE PRINTER (C OR L): ";
870 LINE INPUT F$
880 IF F$="C" THEN DEV=0 : GOTO 910
890 IF F$="L" THEN DEV=1 : GOTO 910
900 GOTO 860
910 J=6
920 FOR K=0 TO 15
930 IF DEV=0 THEN PRINT TAB(J);HEX$(K); : GOTO 950
940 LPRINT TAB(J);HEX$(K);
950 J=J+4
960 NEXT K
970 IF DEV=0 THEN PRINT : PRINT : GOTO 990
980 LPRINT : LPRINT
990 FOR K=1 TO N STEP 16
1000 J=6
1010 IF DEV=0 THEN PRINT HEX$(K-1); : GOTO 1030
1020 LPRINT HEX$(K-1);
1030 FOR L=0 TO 15
1040 IF DEV=0 THEN PRINT TAB(J);HEX$(D(K+L)); : GOTO 1060
1050 LPRINT TAB(J);HEX$(D(K+L));
1060 J=J+4
1070 NEXT L
1080 IF DEV=0 THEN PRINT : GOTO 1100
1090 LPRINT
1100 NEXT K
1110 GOTO 460
1120 '
1130 '   WRITE DATA FORM PROGRAM ARRAY TO DISK FILE
1140 '
1150 PRINT
1160 PRINT "WRITE DATA FILE"
1170 PRINT
1180 PRINT "DATA FILE NAME : ";
1190 LINE INPUT F$
1200 OPEN "O",#1,F$
```

```
PROM.BAS   PAGE  4


1210 FOR K=1 TO N
1220 PRINT #1,D(K)
1230 NEXT K
1240 CLOSE
1250 PRINT
1260 PRINT "DATA FILE ";F$;" WRITTEN WITH ";N;" DATA VALUES"
1270 GOTO 460
1280 '
1290 '   COMPARE PROM DATA WITH DATA IN THE PROGRAM ARRAY. IF NOT THE
1300 '   SAME DISPLAY ADDRESS, PROM DATA, AND PROGRAM DATA
1310 '
1320 PRINT
1330 PRINT "COMPARE PROM"
1340 PRINT
1350 PRINT "ADR";TAB(8);"PROM";TAB(16);"MEMORY"
1360 PRINT
1370 POKE PB,1 : POKE PB,0
1380 FOR K=1 TO N
1390 DP=PEEK(PA)
1400 IF DP<>D(K) THEN PRINT HEX$(K-1);TAB(9);HEX$(DP);TAB(18);HEX$(D(K))
1410 POKE PB,4 : POKE PB,0
1420 NEXT K
1430 GOTO 460
1440 '
1450 '   READ PROM DATA INTO PROGRAM ARRAY
1460 '
1470 PRINT
1480 PRINT "READ PROM"
1490 PRINT
1500 POKE PB,1 : POKE PB,0
1510 FOR K=1 TO N
1520 D(K)=PEEK(PA)
1530 POKE PB,4 : POKE PB,0
1540 NEXT K
1550 GOTO 460
1560 '
1570 '   WRITE(PROGRAM) PROM WITH THE DATA IN THE PROGRAM ARRAY
1580 '
1590 PRINT
1600 PRINT "WRITE PROM"
```

Figure 2A. A 2708 And 2716 PROM Utility Program.

PROM.BAS PAGE 5

```
1610 PRINT
1620 '
1630 '   SET PORT A TO OUTPUT FROM CPU
1640 '
1650 POKE PA+1,0
1660 POKE PA,255
1670 POKE PA+1,28
1680 POKE PB,9 : POKE PB,8
1690 IF TYPE=2 THEN GOTO 1970
1700 '
1710 '   2708 PROM PROGRAM SECTION
1720 '
1730 PRINT
1740 PRINT "2708 PROM PROGRAM"
1750 PRINT "TURN ON PROGRAM POWER AND TYPE ANY KEY TO START: ";
1760 LINE INPUT F$
1770 FOR J=1 TO 100
1780 POKE PB,9 : POKE PB,8
1790 FOR K=1 TO N
1800 POKE PA,D(K)
1810 POKE PB,10 : POKE PB,8
1820 POKE PB,12 : POKE PB,8
1830 NEXT K
1840 NEXT J
1850 '
1860 '   SET PORT A TO INPUT TO CPU
1870 '
1880 POKE PA+1,0
1890 POKE PA,0
1900 POKE PA+1,28
1910 POKE PB,1 : POKE PB,0
1920 PRINT "TURN OFF PROGRAM POWER"
1930 GOTO 460
1940 '
1950 '   2716 PROM PROGRAM SECTION
1960 '
1970 PRINT
1980 PRINT "2716 PROM PROGRAM"
1990 PRINT "TURN ON THE PROGRAM POWER AND TYPE ANY KEY TO START ";
2000 INPUT F$
```

PROM.BAS PAGE 6

```
2010 FOR K=1 TO N
2020 POKE PA,D(K)
2030 POKE PB,24 : POKE PB,8
2040 STAT= PEEK(PB) AND 32
2050 IF STAT =0 THEN GOTO 2040
2060 POKE PB,12 : POKE PB,8
2070 NEXT K
2080 GOTO 1880
2090 POKE PB,10 : POKE PB,8
2100 STAT=PEEK(PB) AND 32
2110 IF STAT=0 GOTO 2040
2120 '
2130 '   READ DATA FORM MEMORY TO THE PROGRAM ARRAY
2140 '   THE BUFFER IS IN MEMORY
2150 '
2160 PRINT
2170 PRINT "READ BUFFER"
2180 PRINT
2190 PRINT "BUFFER STARTING ADDRESS (HEX) : ";
2200 LINE INPUT F$
2210 F$="&H"+F$
2220 AD=VAL(F$)
2230 FOR K=1 TO N
2240 D(K)=PEEK(AD)
2250 AD=AD+1
2260 NEXT K
2270 GOTO 460
2280 '
2290 '   CHANGE DATA IN THE PROGRAM ARRAY
2300 '
2310 PRINT
2320 PRINT "CHANGE DATA"
2330 PRINT
2340 PRINT "ADR : ";
2350 LINE INPUT F$
2360 F$="&H"+F$
2370 AD=VAL(F$)
2380 IF AD>(TYPE*1024-1) THEN PRINT "ADR OUT OF RANGE" : GOTO 2340
2390 PRINT HEX$(AD);TAB(6);HEX$(D(AD+1));TAB(10);":";
2400 LINE INPUT F$
```

PROM.BAS PAGE 7

```
2410 F$="&H"+F$
2420 DP=VAL(F$)
2430 IF DP>255 THEN PRINT "DATA OUT OF RANGE" : GOTO 2390
2440 D(AD+1)=DP
2450 PRINT "MORE (Y OR N) ";
2460 INPUT F$
2470 IF F$="Y" THEN GOTO 2340
2480 IF F$="N" THEN GOTO 460
2490 GOTO 2450
2500 '
2510 '   SUBROUTINE TO DISPLAY PROGRAM COMMANDS
2520 '
2530 PRINT
2540 PRINT "2708/2716 PROM UTILITY PROGRAM."
2550 PRINT
2560 PRINT "COMMANDS :  ";
2570 PRINT TAB(13);"RH....Read Hex file"
2580 PRINT TAB(13);"RD....Read Data file"
2590 PRINT TAB(13);"WD....Write Data file"
2600 PRINT TAB(13);"RP....Read PROM"
2610 PRINT TAB(13);"WP....Write PROM (program)"
2620 PRINT TAB(13);"DD....Display Data"
2630 PRINT TAB(13);"CP....Compare PROM"
2640 PRINT TAB(13);"RB....Read Buffer (memory)"
2650 PRINT TAB(13);"FM....Fill Memory"
2660 PRINT TAB(13);"EX....Exit program"
2670 RETURN
2680 '
2690 '   READ INTEL HEX DATA FILE
2700 '
2710 PRINT
2720 PRINT "READ HEX DATA FILE"
2730 PRINT
2740 PRINT "DATA FILE NAME: ";
2750 LINE INPUT F$
2760 OPEN "I",#1,F$
2770 INPUT #1,A$
2780 BADR=VAL("&H"+MID$(A$,4,4))
2790 GOTO 2810
2800 INPUT #1,A$
```

PROM.BAS PAGE 8

```
2810 CNT=VAL("&H"+MID$(A$,2,2))
2820 IF CNT=0 THEN GOTO 2900
2830 ADR=VAL("&H"+MID$(A$,4,4))
2840 FOR K=1 TO CNT
2850 J=8+2*K
2860 I=ADR-BADR+K
2870 D(I)=VAL("&H"+MID$(A$,J,2))
2880 NEXT K
2890 GOTO 2800
2900 CLOSE
2910 PRINT "HEX FILE ";F$;" READ"
2920 GOTO 460
2930 PRINT
2940 '
2950 '   FILL MEMORY ARRAY
2960 '
2970 PRINT "FILL MEMORY ARRAY"
2980 PRINT
2990 PRINT "STARTING ADDRESS: ";
3000 LINE INPUT A$
3010 SA=VAL("&H"+A$)+1
3020 PRINT "LAST ADDRESS: ";
3030 LINE INPUT A$
3040 EA=VAL("&H"+A$)+1
3050 IF(SA>N OR EA>N) THEN PRINT "ADDRESS OUT OF RANGE":GOTO 2990
3060 PRINT "DATA: ";
3070 LINE INPUT A$
3080 DA=VAL("&H"+A$)
3090 IF DA>255 THEN PRINT "DATA OUT OF RANGE":GOTO 3060
3100 FOR K=SA TO EA
3110 D(K)=DA
3120 NEXT K
3130 GOTO 460
```

Figure 2B. A 2708 And 2716 PROM Utility Program.

### Example A

```
RUN
COMMAND SUMMARY (Y OR N): Y

2708/2716 PROM UTILITY PROGRAM.

COMMANDS :   RH....Read Hex file
             RD....Read Data file
             WD....Write Data file
             RP....Read PROM
             WP....Write PROM (program)
             DD....Display Data
             CP....Compare PROM
             RB....Read Buffer (memory)
             FM....Fill Memory
             EX....EXit program


TURN OFF PROGRAM POWER

PROM TYPE (2708 OR 2716) : 2716


FUNCTION : RH

READ HEX DATA FILE

DATA FILE NAME: CBIOS.HEX
HEX FILE CBIOS.HEX READ


FUNCTION : WP

WRITE PROM


2716 PROM PROGRAM
TURN ON THE PROGRAM POWER AND TYPE ANY KEY TO START ? G
TURN OFF PROGRAM POWER


FUNCTION : CP

COMPARE PROM

ADR      PROM     MEMORY


FUNCTION : EX

EXIT PROM

Break in 630
Ok
```

### Example B

```
RUN
COMMAND SUMMARY (Y OR N): N


TURN OFF PROGRAM POWER

PROM TYPE (2708 OR 2716) : 2716


FUNCTION : RP

READ PROM


FUNCTION : CD

CHANGE DATA

ADR : 200
200   9E  :9C
MORE (Y OR N) ? Y
ADR : 2A4
2A4   54  :10
MORE (Y OR N) ? N


FUNCTION : WP

WRITE PROM


2716 PROM PROGRAM
TURN ON THE PROGRAM POWER AND TYPE ANY KEY TO START ? G
TURN OFF PROGRAM POWER


FUNCTION : CP

COMPARE PROM

ADR      PROM     MEMORY


FUNCTION : EX

EXIT PROM

Break in 630
```

### Example C

```
COMMAND SUMMARY (Y OR N): N


TURN OFF PROGRAM POWER

PROM TYPE (2708 OR 2716) : 2716


FUNCTION : FM

FILL MEMORY ARRAY

STARTING ADDRESS: 0
LAST ADDRESS: 7FF
DATA: FF


FUNCTION : CP

COMPARE PROM

ADR      PROM     MEMORY


FUNCTION : EX

EXIT PROM

Break in 630
Ok
```

## Example D

```
RUN
COMMAND SUMMARY (Y OR N): N

TURN OFF PROGRAM POWER

PROM TYPE (2708 OR 2716) : 2708

FUNCTION : RP

READ PROM

FUNCTION : DD

DISPLAY DATA

OUT PUT TO CONSOLE OR LINE PRINTER (C OR L): C
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D' | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | C3 | 0 | FE | C3 | A2 | 9D | C3 | 8C | 9C | C3 | 95 | 9C | C3 | A3 | 9C | C3 |
| 10 | 85 | 9D | C3 | A0 | 9D | C3 | 92 | 9D | C3 | B0 | 9C | C3 | 3F | 9C | C3 | 33 |
| 20 | 9C | C3 | 39 | 9C | C3 | 2D | 9C | C3 | D | 9D | C3 | 49 | 9D | 69 | 60 | 22 |
| 30 | 42 | 0 | C9 | 79 | 2F | 32 | 40 | 0 | C9 | 79 | 2F | 32 | 41 | 0 | C9 | 79 |
| 40 | 32 | 44 | 0 | 32 | 4 | FA | 6 | A | 21 | 45 | 0 | 3E | 3B | 32 | 0 | FA |
| 50 | 3A | 4 | FA | E6 | C0 | FE | 40 | CA | 67 | 9C | FE | 80 | C2 | 50 | 9C | 3A |
| 60 | 3 | FA | 77 | 23 | C3 | 50 | 9C | 3A | 0 | FA | 2F | E6 | 1C | C2 | 77 | 9C |
| 70 | 3A | 45 | 0 | 32 | 1 | FA | C9 | 5 | C2 | 48 | 9C | 21 | 9B | 9E | CD | 37 |
| 80 | 9E | CD | 95 | 9C | FE | D | CA | 3F | 9C | C2 | 81 | 9C | 3A | 0 | FC | E6 |
| 90 | 1 | CB | 3E | FF | C9 | 3A | 0 | FC | E6 | 1 | CA | 95 | 9C | 3A | 1 | FC |
| A0 | E6 | 7F | C9 | 3A | 0 | FC | E6 | 2 | CA | A3 | 9C | 79 | 32 | 1 | FC | C9 |
| B0 | 6 | A | 3E | F0 | 32 | 0 | FA | 3A | 4 | FA | E6 | 40 | CA | B7 | 9C | 3A |
| C0 | 0 | FA | 2F | E6 | 3C | FE | 24 | C8 | 5 | C2 | B2 | 9C | 21 | 70 | 9E | CB |
| D0 | 37 | 9E | CD | 95 | 9C | FE | D | CA | B0 | 9C | C2 | D2 | 9C | 6 | A | 3A |
| E0 | 1 | FA | 4F | 3A | 40 | 0 | B9 | CA | A | 9D | 32 | 3 | FA | 3E | E0 | 32 |
| F0 | 0 | FA | 3A | 4 | FA | E6 | 40 | CA | F2 | 9C | 3A | 0 | FA | 2F | E6 | 1B |
| 100 | CA | A | 9D | 5 | C2 | DF | 9C | 3E | 1 | C9 | 3E | 0 | C9 | CD | DD | 9C |
| 110 | FE | 1 | C8 | 6 | A | 3A | 41 | 0 | 32 | 2 | FA | 2A | 42 | 0 | 3E | 73 |
| 120 | 32 | 0 | FA | 3A | 4 | FA | E6 | C0 | FE | 40 | CA | 3B | 9D | FE | 80 | C2 |
| 130 | 23 | 9D | 3A | 3 | FA | 2F | 77 | 23 | C3 | 23 | 9D | 3A | 0 | FA | 2F | E6 |
| 140 | 1C | C8 | 5 | C2 | 15 | 9D | 3E | 1 | C9 | CD | DD | 9C | FE | 1 | C8 | 6 |
| 150 | A | 3A | 41 | 0 | 32 | 2 | FA | 2A | 42 | 0 | 3E | 53 | 32 | 0 | FA | 3A |
| 160 | 4 | FA | E6 | C0 | FE | 40 | CA | 77 | 9D | FE | 80 | C2 | 5F | 9D | 7E | 2F |
| 170 | 32 | 3 | FA | 23 | C3 | 5F | 9D | 3A | 0 | FA | 2F | E6 | 7C | C8 | 5 | C2 |
| 180 | 51 | 9D | 3E | 1 | C9 | 3A | 5 | FB | E6 | 80 | CA | 85 | 9D | 79 | 32 | 4 |
| 190 | FB | C9 | 3A | 1 | FD | E6 | 80 | CA | 92 | 9D | 3A | 0 | FD | E6 | 7F | C9 |
| 1A0 | 79 | C9 | F3 | 31 | 80 | 0 | E | 0 | CD | 3F | 9C | CD | B0 | 9C | E | 1 |
| 1B0 | CD | 33 | 9C | E | 1 | CD | 39 | 9C | 1 | 80 | 0 | CD | 2D | 9C | CD | B |
| 1C0 | 9B | FE | 0 | C2 | A2 | 9D | E | 0 | CB | 33 | 9C | 16 | 2 | 21 | 0 | 87 |
| 1D0 | 6 | 2A | C5 | D5 | E5 | 4A | C8 | 39 | 9C | C1 | C5 | CD | 2D | 9C | CD | D |
| 1E0 | 9D | FE | 0 | C2 | C6 | 9D | E1 | 11 | 80 | 0 | 19 | D1 | C1 | 5 | CA | 8 |
| 1F0 | 9E | 14 | 7A | FE | 13 | DA | D2 | 9D | 16 | 1 | C | 79 | FE | 1 | C2 | 2 |
| 200 | 9C | 14 | CD | 33 | 9C | C3 | D2 | 9D | 3E | C3 | 32 | 0 | 0 | 32 | 5 | 0 |
| 210 | 21 | 3 | 9C | 22 | 1 | 0 | 21 | 6 | 8F | 22 | 6 | 0 | 1 | 80 | 0 | CD |
| 220 | 2D | 9C | E | 0 | C3 | 0 | 87 | F3 | 31 | 80 | 0 | CD | DD | 9E | 21 | 48 |
| 230 | 9E | CD | 37 | 9E | C3 | 8 | 9E | C5 | 4E | 3E | 0 | B9 | CA | 46 | 7E | C0 |
| 240 | A3 | 9C | 23 | C3 | 38 | 9E | C1 | C9 | D | A | A | 43 | 50 | 2F | 4D | 20 |
| 250 | 20 | 56 | 45 | 52 | 2E | 20 | 31 | 2E | 34 | D | A | 43 | 55 | 53 | 54 | 4F |
| 260 | 4D | 49 | 5A | 45 | 44 | 20 | 53 | 59 | 53 | 54 | 45 | 4D | D | A | A | 0 |
| 270 | D | A | A | 43 | 41 | 4E | 20 | 4E | 4F | 54 | 20 | 48 | 4F | 4D | 45 | 20 |
| 280 | 44 | 49 | 53 | 4B | 2C | 20 | 54 | 59 | 50 | 45 | 20 | 43 | 52 | 20 | 54 | 4F |
| 290 | 20 | 54 | 52 | 59 | 20 | 41 | 47 | 49 | 41 | 4E | 0 | D | A | A | 43 | 41 |
| 2A0 | 4E | 20 | 4E | 4F | 10 | 20 | 52 | 45 | 41 | 44 | 20 | 44 | 49 | 53 | 4B | 20 |
| 2B0 | 41 | 44 | 44 | 52 | 45 | 53 | 53 | 20 | 4F | 4E | 20 | 44 | 49 | 53 | 4B | 20 |
| 2C0 | 53 | 45 | 4C | 45 | 43 | 54 | 2C | 20 | 54 | 59 | 50 | 45 | 20 | 43 | 52 | 20 |
| 2D0 | 54 | 4F | 20 | 54 | 52 | 59 | 20 | 41 | 47 | 49 | 41 | 4E | 0 | 3E | 3 | 32 |
| 2E0 | 0 | FC | 3E | 11 | 32 | 0 | FC | 3E | 0 | 32 | 1 | FD | 32 | 3 | FD | 32 |
| 2F0 | 0 | FD | 2F | 32 | 2 | FD | 3E | 2E | 32 | 1 | FD | 32 | 3 | FD | 3A | 0 |
| 300 | FB | 3A | 6 | FB | 3E | FF | 32 | 5 | FB | C9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 310 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 320 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 330 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 340 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 350 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 360 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 370 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 380 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 390 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3A0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3B0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3C0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3D0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3E0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3F0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Example D (continued)

```
FUNCTION : WD

WRITE DATA FILE

DATA FILE NAME : SDROMR.1

DATA FILE SDROMR.1 WRITTEN WITH  1024  DATA VALUES

FUNCTION : EX

EXIT PROM
```

## Example E

```
RUN
COMMAND SUMMARY (Y OR N): N

TURN OFF PROGRAM POWER

PROM TYPE (2708 OR 2716) : 2708

FUNCTION : RD

READ DATA FILE

DATA FILE NAME: SDROMR.1

DATA FILE SDROMR.1 READ WITH  1024  DATA VALUES

FUNCTION : WP

WRITE PROM

2708 PROM PROGRAM
TURN ON PROGRAM POWER AND TYPE ANY KEY TO START: G
TURN OFF PROGRAM POWER

FUNCTION : CP

COMPARE PROM

ADR      PROM     MEMORY

FUNCTION : EX

EXIT PROM

Break in 630
Ok
```

# VHF+ Technology

Conducted by Geoff Krauss,* WA2GFP

Recently a Federal Court in Minneapolis declared private use of MDS/HBO (Multipoint Distribution Service/Home Box Office)receivers illegal. This is of interest to vhf+ers because the "pirate-detection" method has grave undertones. The MDS owner apparently sent out crews to note any antenna which **may** be capable of receiving an MDS signal, then cross checked these locations against their subscriber list and sent warnings to all non-subscribers. Could the "surveillance" crew know the frequency of use by merely looking at an antenna? Will an amateur microwave operator be wrongfully notified by a common-carrier operator? If this happens, will a simple note of explanation to the carrier operator be sufficient? Will an MDS operator believe the amateur? The frequency and "scrambling" relationships used by many MDS operators are such that the MDS signal can be an image signal to an amateur 2304-MHz converter utilizing a low-vhf i-f. Many cheap "amateur microwave TV converters" appear to fall into this class; it is a moot point as to whether there is any 13-cm ATV work actually being done.

These octagonally shaped microstrip converter boards seem to be readily available, and there is a way to use them as a true 2304-MHz amateur receiving converter. Each has an rf and an i-f amplifier, a harmonic mixer (needing an injection signal at one half the local-oscillator frequency) and a free-running local osciallator at that injection frequency (e.g, (2304-28)/2=1138 MHz, for a 28-MHz i-f). The free-running oscillator can be injection-locked by one milliwatt or more of a crystal-control signal. I have used an injection source (refer to N6TX's article in <u>Ham Radio</u>, December, 1980) with a 94.8-MHz oscillator followed by an x12 multiplier chain, for a 28.8-MHz i-f.

The modifications (see pictorial) stabilize the oscillator B+ and bring in the locking signal. B+, from a feed-through capacitor in the pc box wall, goes through an rf choke to point A. The B+ land (B) is cut and bridged with resistor R1; capacitors C1, C2 and zener D, are added to operate the oscillator at +10 volts. Coax (from the injection source) is brought through hole (C); the shield is grounded, and the inner conductor is supported by resistor R2. A variable capacitor, from cable inner conductor to device emitter, is used to vary level and frequency. The oscillator collector line (D) was cut while checking with a frequency counter at the converter input. The variable capacitor is trimmed to lock, as verified with a spectrum analyzer.

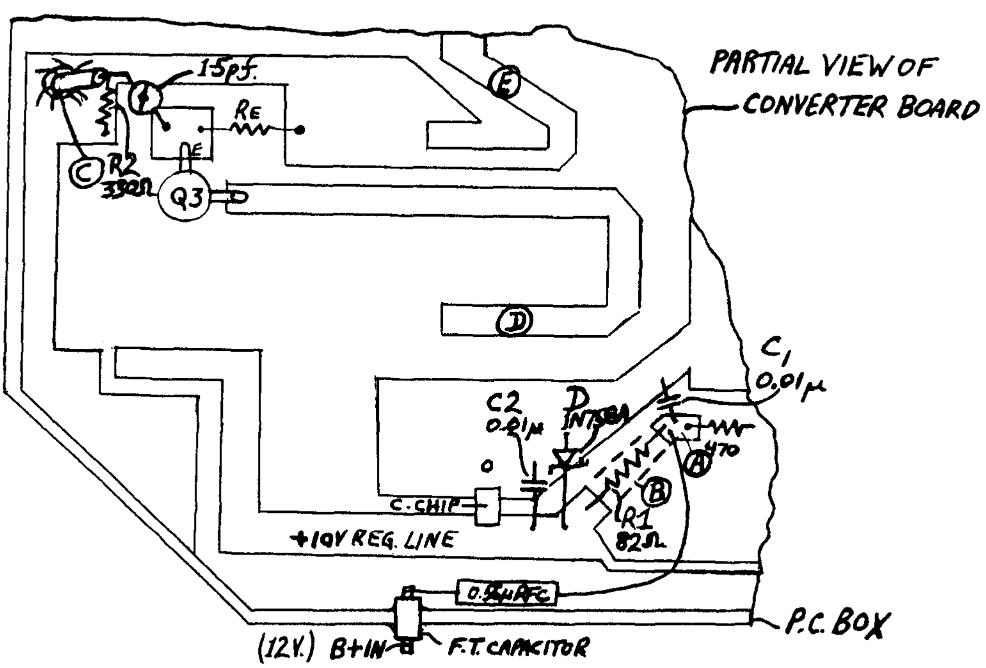
Ideas to try: to provide the needed LO signal (1) find a circuit modification or another device which will change the oscillator to an amplifier; or (2) use at least 5 milliwatts of injection signal at mixer input land E.

One converter supplier (Universal Communications, P.O. Box 339, Arlington, TX 76004) also has a converter with a crystal oscillator and phase-locked loop LO, but this more-expensive unit has not been checked by this writer.

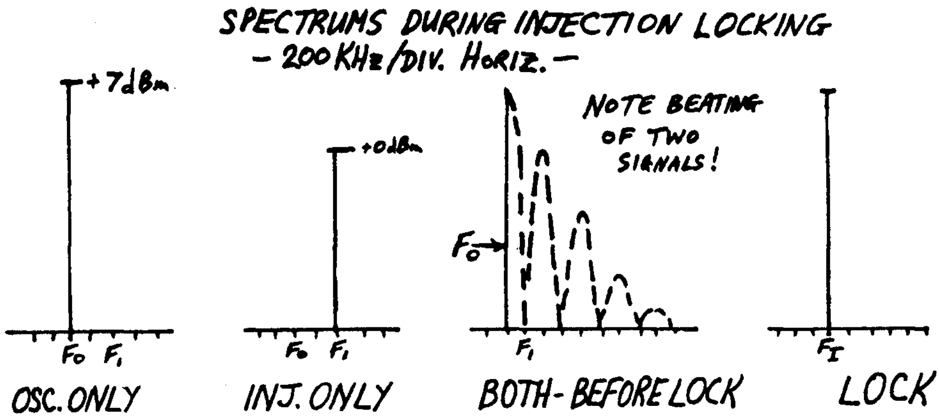
After the above text was submitted for this issue of <u>QEX</u>, I learned of another court decision. A Sacramento, CA court apparently has decided that <u>any</u> amateur 2304-MHz receiving equipment used in <u>that</u> area must first be approved by the local MDS operator to assure that it can't be used to receive local MDS broadcasts. The conversion described earlier in this column might not be a legal ham receiver in Sacramento, nor might the use of a wideband mixer (2-4 GHz) front end for allowing the 2.3- and 3.3-GHz ham bands to be received by switching LO frequencies. Any receiver not having a narrow-bandpass-filter front end might be illegal.

Will wideband 10-GHz rigs be next because they might be used to receive the proposed DBS (Direct Broadcast Satellite) sometime in the future? Will all high-gain dishes be banned, as usable to receive all sorts of non-amateur signals, even if the wrong feed/LNA is in use?

*16 Riviera Drive, Latham, NY 12110.



VHF+ Technology
(continued from previous page)




## CORRESPONDENCE

### Feedback on VADCG TNC 2716 EPROM Modification

(continued from page 2)

The diodes may work just fine with some PROMs, but in some cases they may not, causing marginal performances, glitches in programs and non-reproducible errors.

The AND gate will add a few nanoseconds of delay to the chip-select signal, but in an 8085-based system, this shouldn't cause problems. Bob's idea is a good, but an AND gate improves it and removes any lingering doubts about marginal performance. - Jonathan A. Titus, KA4QVK, P.O. Box 242, Blacksburg, VA 24060, 703-951-9030.

### Bauds

As writers and editors, we accept a challenge and bear a responsibility to our readers: the words we use must be accurate, appropriately expressed, and suitable for transmitting information to readers with minimum confusion, doubt, or mental gymnastics on the part of those readers.

As an active amateur operator and experimenter, active in radioteleprinter and data transmission, as an RTTY repeater operator and licensee since 1969, I enjoy the general trend of the information available in QEX. There's no other source like it at this time.

As a concerned engineer, writer, editor and translator in the telecommunications field, I call your attention to incorrect or inappropriate use of the word "baud". In articles published in QEX (and QST, as well), the question of proper use of the term "baud" remains to be defined. The singular form "baud" is frequently used in cases where the plural form "bauds" is clearly required.

The internationally accepted standards of documentation show that in any case where a modulation rate is greater than one baud, the plural form "bauds" must be used. When using "baud" as an adjective, the singular form is needed. When using "baud" as a noun, the plural form is needed.

Example: "This is a 300-baud modem."
"This modem operates at 300 bauds."

The terms "baud," "baud" and "bits per second" are absolutely not interchangeable, except under very specific circumstances.

The case is identical to use of the term "volt," in defining signal and power applications. You would normally state that, "This is a 230-volt power line." Would you condone a writer's statement that, "This power line operates at 230 volt?"

One can't help but doubt the credibility of manufacturers whose advertising copy contains such significant technical errors, or question the expertise of authors of learned papers whose writing reflects such simple yet glaring errors.

Must we swim though a sea of jargon, buzz words, vogue words and acronyms and, in addition, struggle to penetrate the thickening cloud of semantic errors that befogs much of our computer-related writing today?

Urge your writers, editors and advertisers to pay attention to this very common, very frequent miscarriage of proper technical language usage. - Norman J. Sternberg, W2JUP/WR2AFC/G5ECI, 279 Adirondack Drive, Farmingville, NY 11738.

QEX: The ARRL Experimenter's Exchange is published by the

American Radio Relay League
225 Main Street
Newington, CT 06111 USA
telephone 203-666-1541



The purposes of QEX are to:

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, and

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

Subscriptions are available to ARRL members and non-members at the rates shown on the QEX Subscription Order Card inside this issue.

Subscription orders, changes of address, and reports of missing or damaged copies should be sent to the American Radio Relay League, Newington, CT, USA 06111. Members are asked to include their membership control number or a label from their QST wrapper when applying.

QEX is edited in, and mailed from, McLean, Virginia. Authors are invited to submit manuscripts and letters for publication to Editor, QEX, 1524 Springvale Avenue, McLean, VA 22101. Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and double spaced. Please use the standard ARRL abbreviations found in the latest December issue of QST. Authors should supply their own artwork using black ink on white paper. When essential to the article, photographs may be included. Photos should be glossy, black-and-white positive prints of good definition and contrast.



*QEX: The ARRL*
*Experimenters' Exchange*
1524 Springvale Avenue
McLean, VA USA 22101

Nonprofit Organization
U.S. Postage
PAID
McLean, Virginia 22101
Permit No. 235