MARK THIS DATE: MON. OCT. 20, DEPTFORD - MYARC'S LOU PHILLIPS AND GENEVE!

THE DATA BUS

ISSUE :
VOL. 4
NO. 8
- - - -
SEPT
1986

THE DELAWARE VALLEY USERS GROUP

P.O. BOX 6240 STANTON BRANCH, WILMINGTON DE 19804

4TH THURS. 6:30-9:30 P.M. CHRISTIANA MALL COMMUNITY ROOM

KENT COUNTY, DE COURTHOUSE <CHAPTER> DEPTFORD, NJ MUNICIPAL BLDG.
2ND THURS. 7:00-9:00 P.M. <MEETING> 3RD MONDAY 6:45-9:00 P.M

President's Message

Summer is over and it's almost computer season. I always refer to computer season as starting in the middle of October and ending in the middle of April. It just seems to be the natural time of year when most people sit down and work with their computers. In the summertime we usually find other things to do and this has been borne out by historical trends. Almost every year during the summer months most of us are doing other things and our meeting attendance is smaller than normal, then around October our attendance starts picking up and usually our largest attendance is in February. This trend has been noted by other users groups, too (regardless what kind of computer they have). So we look with great anticipation to a new computer season. We have a lot of things to look forward to like the new computer from Myarc and other new hardware from other companies, along with some of the new programs that keep showing up like C99 and Funnel Writer. C99 opens up alot of new possibilities for converting some of our old basic programs to assembly language speed.

We are looking forward to our October meeting in New Jersey when the President of Myarc Lou Phillips will demonstrate the new TI compatible computer. Lou told the chairman of the New Jersey Chapter of the DVUG that he will have some of the new computers for sale at that meeting. This once again shows that our computer will not die, in fact, new products continue to come out to support it. We are still trying to learn the full potential of our computer. New products like the Gram Kracker have shown us how to modify the operating system and save modules to disk so they can be modified to your personal needs. We can even access the GPL language interpreter inside our computer (many people did not know that there was a hidden language inside our computers). Cor Comp's X-10 can convert your computer into a home security system. So, as you can see we are still exploring new ways to use our old $50 computers.

In the three years since Texas Instruments decided to stop producing the 99/4A, our users group not only survived, but actually doubled in size. We have even expanded to include two new chapters to our users group and we now support three bulletin boards.

Now that we have the two new chapters, we need to update our constitution with a few changes. I intend to have several copies of our constitution with the proposed changes at our next meeting for all the members to review and discuss. We will vote on those changes at the October meeting. The intent of these changes is to incorporate some of the things that we have already agreed to in some of our business meetings and to officially recognize the Delmarva and New Jersey chapters.

Since we estimate that it cost about $10 per member to cover the basic expenses of the users group and about $5 to cover each chapter's functions, we would like to recommend the following changes to the constitution:

1. The Student membership be increased from $5 to $10.
2. When a chapter recruits a new member or gets a member to renew his or her membership, $10 will go to the DVUG to cover the newsletter and other basic cost. The other $5 will remain with the chapter to help fund its operations.
3. Create a correspondent's membership of $10 for people who only want our newsletter and live more than 75 miles away.

The intent of these changes is to distribute the users group's money in a more equitable manner.

We will also propose some changes to explain the responsibilities of a chapter chairman and to emphasize our intent to operate our users group as a non for profit organization.

DELAWARE VALLEY USER'S GROUP: SEPT 1986

DVUG EXECUTIVE COMMITTEE MEMBERS IN 1986

PRESIDENT .....TOM AUGUST
VICE PRESIDENT .....JIM DAVIS
SECRETARY .....LYNN ACQUARD
TREASURER .....TOM KLEIN
SGT. AT ARMS .....JIM FOLZ
DELMARVA CHAPTER CHR.....CHARLES BOWER
SO.JERSEY CHAPTER CHR.....ERROL LANSBERRY

DVUG BBS: (302)322-3999 Anytime
(609)429-7792 Anytime
(302)674-1449 6:00 PM-6:00 AM

For general information, you may contact

TOM KLEIN	Pa.	(215)494-1372
GIL or KAY QUILLEN	Del.	(302)492-8189
JACK SHATTUCK	Del.	(302)764-8619
BUTCH FISHER	N.J.	(609)783-8276

A Delaware Valley Users Group membership includes monthly newsletter (DATABUS), library and software privileges, plus other special benefits. Annual membership rates are: Family or Individual $15; Students $10; Newsletter only (beyond 75 mi) $10.

PLEASE TRANSMIT YOUR NEWSLETTER COPY TO: The Data Bus Editor ---- Jim Folz, Telephone (302)995-6848, or use the DVUG mailing address shown on Page One. NEWSLETTER COPY WILL NOT BE ACCEPTED FOR AN ISSUE AFTER THE 2ND THURSDAY OF EACH MONTH.

DVUG ADVERTISING RATES FOR THE DATA BUS:
1/4 page = $ 5/issue, or $ 45/12 issues
1/2 page = $ 8/issue, or $ 75/12 issues
Full page = $15/issue, or $125/12 issues

DELAWARE VALLEY USERS GROUP MEETINGS

Plenary meetings: Delaware's Christiana Mall on Rte. 7, at I-95 Exit 4-S, in the Community Room. Enter between J.C.Penney and Liberty Travel inside the Mall.

DELMARVA CHAPTER: Kent County Courthouse, Basement Conference Rm #25, Green & State Streets, Dover, Delaware. Use the Green St. side entrance.

SO.JERSEY CHAPTER: Deptford Municipal Bldg, Cooper Ave. and Delsea Drive, (Rtes. 534 & 47), in Gloucester County. Enter and park in rear of the building.

CONTENTS OF THE SEPTEMBER ISSUE OF THE DATA BUS:

NOISE on The Data Bus	Page 2
TI 99/4A Hex Dump Article	Pages 3-4
Paid Advertisements	Pages 4,6,8
Keyboards	Page 5
DVUG Charter Change Proposal	Page X1
TI 99/4A Hex Dump Program	Pages 7,9
Corrections	Page 10
TOUCHTONE FREQUENCIES	PAGE 10

NOISE on The Data Bus

by Jim Folz

Why is it that every newsletter editor (even new ones) has to see his name on some article? I didn't know until now. It seems that editors pick-up all kinds of information bits that are too small to sustain an article. In the future, you can expect to find those goodies here.

On Monday, October 20, Lou Phillips (Myarc) will visit the South Jersey Chapter. I understand that he will be bringing Myarc's new computer offering GENEVE. That chapter meets at the Deptford Municipal Building. Don't miss this one.

On Sunday, September 21, the TRI-STATE Computer Fair will be held at the Ramada Inn in New Castle. Schedule-10 A.M. to 4 P.M. Admission-$3.50 For more information call (201)533-1991. DVUG is looking at setting a table up at this fair. If you would like to work at our table please contact Tom August or Jack Shattuck.

On Saturday, October 11, the North Eastern 99'ers Computer Club will be holding its 2nd Annual Computer Workshop in Pawtucket, R. I. For information call Robert Lavatin (617)695-7461

Attention Plato Enthusiasts! Control Data Corporation has offered discount rates for its educational software. For more information contact L. B. Lewytzkyj, Control Data Corporation, 8100 34th Avenue South, HQB02G, Minneapolis, MN 55440, (612)853-3162

The Executive Board needs someone to fill the Refreshment, Equipment and Recruitment Chairs. If you are interested please contact Tom August or Jim Davis. Please volunteer.

In the future, the Main Chapter meeting format will be as follows: 6:30-7:00 Demonstrations and Socializing, 7:15-7:45 Education Committee Program, 8:00-9:15 Raffles, Business Meeting and Discussion. It is a good meeting. Y'all come.

Please note the change of the phone number for the BBS in New Jersey. Also note the new BBS phone number in the DelMarVa Chapter area. While it was not in operation at press time, it should be operating by the time you receive this issue.

For the last two newsletters, the Student membership rate was incorrectly shown as $5. Student rates are $10.

Barry Boland reminds us that a forum for TI computer users is available. GO TEX-200 gets you to the right place.

TI-99/4A HEX DUMP PROGRAM

by Norm Sellers

It is often necessary to execute an assembly language program from a BASIC program. For example, if you are writing an assembly language subroutine that must have arguments, it is necessary to execute this program from a BASIC program. Another example is when running an assembly language program that hangs the system. It is often difficult to get back in the debugger without destroying some of the memory that you need to look at in order to debug your program.

To solve these problems, I have written an "all purpose" DUMP program in BASIC which will run under Extended BASIC or Editor/Assembler BASIC (although some features in DUMP only work under E/A). I chose BASIC since this will never allow the program to destroy a normal assembly program. The Extended BASIC cartridge puts BASIC in high memory while assembly is in low memory, and the E/A cartridge does the opposite.

Description of the DUMP Program:

The DUMP Program first asks if you wish to print the results. If you have no printer, it is not necessary to change the program--just answer "N" to this question. Note: If you have a printer, it may be necessary to modify the printer name in line number 250 to match your configuration.

Next DUMP asks whether to CALL INIT. If you are running your assembly program, answer "Y". Statement 490 LINKs to your program. If you have arguments, they should be put in this statement. If you are dumping a previously run program, answer "N". This negative answer sends the DUMP Program immediately to the Dump portion of the program at statement 560; however, if you are running your assembly language program, more questions are asked.

When you are running your assembly language program, the names of all object files that are needed to be loaded are requested. When you enter the name of each one, be sure that the disk containing the corresponding object file is on the correct disk drive before pressing <ENTER>.

After all object files have been loaded, just press <ENTER> when the next object file name is requested. Then you are asked if you need the TI DEBUGGER and the BSCSUP, BASIC Support Utilities (these are needed if your assembly subroutine has arguments or if you wish to pass an error code from your assembly subroutine to the calling BASIC program). DEBUGGER from TI only works with E/A unless you convert it. If you answer "Y" to either question, you are asked to mount the Assembler disk containing DEBUGGER and BSCSUP. When this is mounted, press <ENTER>. You are then asked to mount your 'RUN' disk. This is needed only if your program looks for a disk file while running.

If your assembly routine is able to return control to the DUMP Program, you will then hear a 3 note musical chord which lasts one second. At this point, you are now in the TI DEBUGGER if you requested it in the beginning. The first command you need to give the DEBUGGER is 'U' to toggle the screen for BASIC since its screen characters are not using the offset of >80 and are not visible until this command is entered.

Any other DEBUGGER commands may be given now for a quick scan of memory etc. When you are finished with the DEBUGGER, enter 'Q' to Quit. The control is then returned to the DUMP Program. At this point, whether you have run an assembly program or are dumping a previous run, DUMP asks you to enter 'B' for Backwards. This is primarily used to Dump the REF/DEF table backwards from >4000, although you can dump any PEEKable area in reverse order. If you request 'B' for Backwards, the next question DUMP asks is 'DUMP RETURNS?'. This feature should be used with E/A only. If you answer 'Y', the routines found in the REF/DEF table are printed and displayed on the screen if the routine was executed by your program while in execution. The way DUMP determines this is to look at the registers in the BLWP Routines. Subroutine Register 14 is printed as 'RETURN'. This shows you where in your program that this routine was last called from (the statement preceeding this address is the calling BLWP). Also Subroutine Register 15 is printed which is the saved status byte recorded when the BLWP routine was last exited. R13 is printed which gives the calling routine's register workspace at that time. If the return address found in Subroutine Register 14 is outside of your program, this entry in the REF/DEF table is not printed or displayed in the Dump. Of course, you guessed it. In order to do this, the DUMP Program must have asked for the upper and lower address bounds of your program. It should be noted here that several BLWP routines in the REF/DEF table use the same registers; therefore, extraneous routines will be printed. Just check your program return addresses to see which are impossible.

After the BLWP routines are printed, the DUMP Program proceeds to ask for start and stop data addresses to dump. DUMP then displays and prints (if requested), in HEX and Character formats, the memory between the lower and upper address bounds of the data in your program. If your data appears at multiple places in memory, while DUMP is running, hold the space bar down until the following menu appears at the bottom of the screen:

"P=SWCH,C=CHG ADR"
"E=END,Q=QUIT,ELSE CON"

If 'P' is entered, the printer is turned on or off (like the FORTH word SWCH). If 'C' is entered, the dumping address is requested, and is accepted either in HEX or in DECIMAL (as is used by BASIC in the PEEK function). If 'E' is entered, the dump is exited and DEBUGGER is started if, it was requested. Also if 'Q' is entered, the program QUITs. 'P' and 'C' can be used together.

Detailed Description of the Program DUMP:

The DEF Function in statement 230 is used to convert addresses in the DUMP from 16 bit numbers to 15 bit signed numbers and visa versa. Statements 260 through 460 ask all questions needed to set up the desired environment with the printer, and the assembly language program to be debugged, as described in a previous paragraph. If you are running your assembly routine or program, this section also calls INIT and LOAD in order to run your assembly program.

Statements 470 through 490 call LINK to your program or routine. If you need arguments to call your subroutine, put them in statement 490. You need not save this version of the DUMP Program if you do not need this version again--just RUN the memory version. Statements 500 through 530 announce the completion of your program while statements 540 and 550 call DEBUG if you selected this option.

The variable BK is used to signal 'forwards' or 'backwards' dumping--the value 0 meaning 'forwards', the value 1 meaning 'backwards'. Statements 560 through 590 ask you whether to dump 'forewards' or 'backwards' and set BK accordingly.

If you selected 'backwards', you are then asked whether to 'DUMP RETURNS'. If your answer is 'Y', the program goes to line 1830. The variable ODADR is set to 16376 (the decimal equivalent to >3FF8 which is the address of the last REF/DEF entry. You are then asked for the starting HEX address and last HEX address in your program. Each of these are converted to decimal by a GOSUB routine at 2810 through 2960. These values are saved in variables STRT and LST.

Statement 1920 PEEKs the 8 byte REF/DEF entry where A1 through A6 are the routine name and A7 with A8 give the routine's entry address.

Statement 1930 is used to terminate the REF/DEF table scan by looking for a routine with a name that does not start with a letter. When this condition occurs, the line 2460 loads the beginning Dump address, ADRS with STRT. At line 2510, BK=1 to Dump memory forwards. After requesting the starting and ending addresses, line 730 begins the memory dump. The GOSUB 750 routine converts the 15 bit signed address into a 16 bit unsigned address (lines 750 through 810). Lines 880 through 910 with the GOSUB 2580 determine this address in HEX as a character string (lines 2560 through 2740 adds on to the right of the character string HX$ two bytes of hex digits equal to the numeric value contained in the variable VALA). Line 990 PEEK's at 8 bytes starting at DADR for the DUMP. Lines 1000 through 1240 convert the 8 bytes PEEK'ed into character HEX for printing and displaying in lines 1250 through 1600. Lines 1280 through 1440 remove any non-printable characters from the printer since these often mean something else to the printer (i.e. changing to condensed print, changing lines per inch etc.). The actual printing only occurs every other time these statements are executed, as controlled by the variable PY$. This was done to allow printing twice as much per line as is displayed on the screen.

Line 1620 advances the Dump address by 8 bytes. Lines 1630 and 1640 calculate the new 16 bit address from the signed 15 bit address in ADRS.

Lines 1650 through 1740 determine whether to continue by going to line 730 for the next portion of the dump, or whether the keys 'P', 'C', 'Q' or 'E' have been pressed. If 'Q' was pressed, the program QUITS. If 'E' was pressed, the Dump is Exited and DEBUGGER is again called, if this option was chosen when the program was first started. If 'P' is entered, the printer is turned on or off (like the FORTH word SWCH). If 'C' is entered, the memory location to dump can be changed in Hex or in Decimal.

Again, when the DEBUGGER starts execution, the first command needed is 'U' to toggle the screen. This time when the DEBUGGER is exited, the program STOPs.

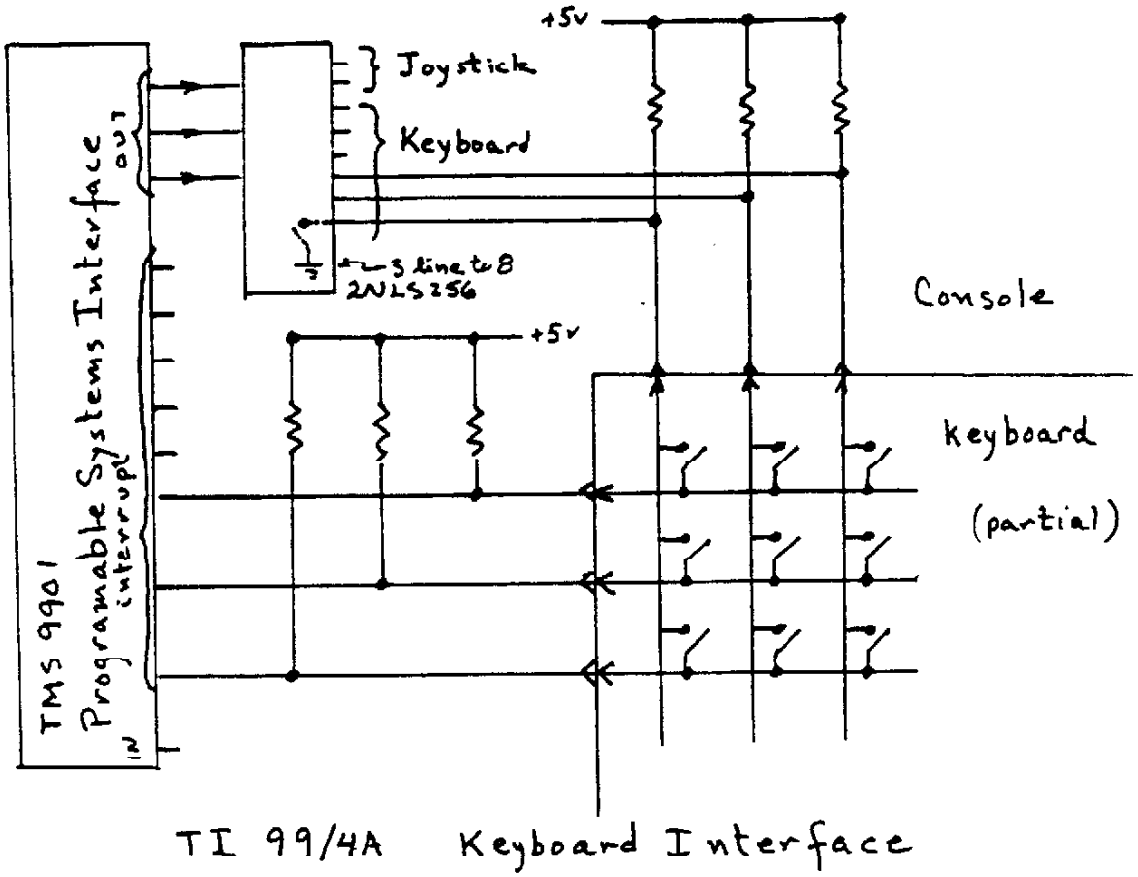
FOR PROGRAM LISTING

SEE PAGES 7 AND 9

Keyboards
by Jim Davis and Jim Folz

At the August meeting, Jim Davis led a discussion on keyboards and Don Newsom showed a keyboard project that he has been working on. Jim has provided us with a schematic for reference. Remember that the 2NLS256 chip essentially grounds "columns" alternately during a key scan. When a key is pressed, the grounding during a key scan also grounds an input to the TMS 9901 chip generating an interupt. The computer uses the "column" info from the key scan and the "row" info from the interupt to enter a table and look up the key that was pressed.

You missed a good presentation if you missed Keyboards. Next month Jim will discuss "ON ....." commands. Don't miss it.



PROGRAM

```
100 REM SAVE DSK2.DUMP
110 REM
120 REM GENERAL BASIC DUMP
130 REM
140 REM BY NORM SELLERS
150 REM
160 L32=32768
170 GOTO 230
180 DEF ST$(N)=SEG$("      ",1,6-LEN(STR$(N)))&
STR$(N)
190 B,J,LST$,TYP$,STRT$,R15B,R15A,R14,R13B,R13A,
HXR14$,HXR13$,BK$, B2,B1,STRT,RET$,R14B,R14A,OBJ$,
LST.LIN$.ENT$.DDADR.BSC$
200 ANS$,SD$,PY$,LIND$,BK,ADRS$,R,PRT$,DEB$,
LINP3$,LINP2$,LINP1$, LINP0$,S,CALL INIT,CALL
LOAD,CALL LINK,CALL SOUND,VAL,CALL KEY,INT
210 DI,DI$,HW4$,HW3$,HW2$,HW1$,HXADR$,D$,L,DADR,I,
D,ADRS,DIGT,SAV,CALL PEEK,A8,A7,A6,A5,A4,A3,A2,A1,
H$,HXN$,HX$,VALA,VALU,PRT
220 PRNTR$,TYE$,ADRE$
230 DEF CHG(B)=B-65536*SGN(B)
240 REM @P-
250 PRNTR$="RS232.BA=4800.DA=8"
260 INPUT "PRINT(Y,N)?":PRT$
270 IF PRT$<>"Y" THEN 300
280 OPEN #1:PRNTR$
290 PRT=1
300 INPUT "CALL INIT,Y OR N? ":ANS$
310 IF ANS$<>"Y" THEN 360
320 CALL INIT
330 INPUT "ENTER OBJ NAME=DSK":OBJ$
340 IF OBJ$="" THEN 370
350 CALL LOAD("DSK"&OBJ$)
360 GOTO 330
370 INPUT "ENTER ENTRY NAME=":ENT$
380 INPUT "NEED DEBUG,Y OR N ":DEB$
390 INPUT "NEED BSCSUP,Y OR N ":BSC$
400 IF (DEB$<>"Y")*(BSC$<>"Y")THEN 470
410 INPUT "ASSM DISK":ANS$
420 IF BSC$<>"Y" THEN 440
430 CALL LOAD("DSK1.BSCSUP")
440 IF DEB$<>"Y" THEN 470
450 CALL LOAD("DSK1.DEBUG")
460 INPUT "RUN DISK":ANS$
470 REM PUT ARGS IN STMT 450 IF NEEDED
480 PRINT "LINK("&ENT$&")"
490 CALL LINK(ENT$)
500 CALL SOUND(1000,110,2,165,2,262,2)
510 FOR I=1 TO 1000
520 J=I*I
530 NEXT I
540 IF DEB$<>"Y" THEN 560
550 CALL LINK("DEBUG")
560 BK=1
570 INPUT "B FOR BACKWARDS THRU MEM=":BK$
580 IF BK$<>"B" THEN 630
590 BK=-1
600 INPUT "DUMP RETURNS?":RET$
610 IF RET$="Y" THEN 1830
620 PY$="N"
630 INPUT "H OR D,START ADDR=":TYP$,ADRS$
640 INPUT "H OR D,END ADDR=":TYE$,ADRE$
650 IF TYE$="H" THEN 680
660 LST=VAL(ADRE$)
670 GOTO 710
680 H$=ADRE$
690 GOSUB 2810
700 LST=D
710 IF TYP$="H" THEN 820
720 ADRS=VAL(ADRS$)
730 GOSUB 750
740 GOTO 860
750 DADR=ADRS
760 IF ADRS<0 THEN 800
770 IF ADRS<=32767 THEN 810
780 DADR=CHG(ADRS)
790 GOTO 810
800 ADRS=CHG(ADRS)
810 RETURN
820 REM CALL DEC2(ADRS$,ADRS)
830 H$=ADRS$
840 GOSUB 2810
850 ADRS=D
860 HXADR$=ADRS$
870 GOTO 730
880 HX$=""
890 REM CALL HEX2(ADRS,HXADR$)
900 VALA=ADRS
910 GOSUB 2580
920 J=LEN(HX$)
930 IF J=4 THEN 980
940 IF J<4 THEN 970
950 HX$=SEG$(HX$,J-3,4)
960 GOTO 980
970 HX$=SEG$("0000",1,4-J)&HX$
980 HXADR$=HX$
990 CALL PEEK(DADR,A1,A2,A3,A4,A5,A6,A7,A8)
1000 REM CALL HEX2(A1,HX$)
1010 HX$=""
1020 VALA=A1
1030 GOSUB 2580
1040 VALA=A2
1050 GOSUB 2580
1060 HW1$=HX$
1070 VALA=A3
1080 HX$=""
1090 GOSUB 2580
1100 VALA=A4
1110 GOSUB 2580
1120 HW2$=HX$
1130 VALA=A5
1140 HX$=""
1150 GOSUB 2580
1160 VALA=A6
1170 GOSUB 2580
1180 HW3$=HX$
1190 VALA=A7
1200 HX$=""
1210 GOSUB 2580
1220 VALA=A8
1230 GOSUB 2580
1240 HW4$=HX$
1250 LIN0$="  "&HXADR$&"  "&HW1$&" "&HW2$&" "&
HW3$&" "&HW4$&" "
1260 PRINT LIN0$;TAB(0);
1270 PRINT ST$(DADR);TAB(9);CHR$(A1);TAB(11);
CHR$(A2);TAB(14); CHR$(A3);TAB(16);CHR$(A4);
TAB(19);CHR$(A5);TAB(21); CHR$(A6);TAB(24);
CHR$(A7);TAB(26);CHR$(A8);
1280 IF PRT$<>"Y" THEN 1610
1290 IF (A1<32)+(A1>126)THEN 1300 ELSE 1310
1300 A1=32
1310 IF (A2<32)+(A2>126)THEN 1320 ELSE 1330
1320 A2=32
1330 IF (A3<32)+(A3>126)THEN 1340 ELSE 1350
1340 A3=32
1350 IF (A4<32)+(A4>126)THEN 1360 ELSE 1370
1360 A4=32
1370 IF (A5<32)+(A5>126)THEN 1380 ELSE 1390
1380 A5=32
1390 IF (A6<32)+(A6>126)THEN 1400 ELSE 1410
1400 A6=32
1410 IF (A7<32)+(A7>126)THEN 1420 ELSE 1430
1420 A7=32
1430 IF (A8<32)+(A8>126)THEN 1440 ELSE 1450
1440 A8=32
1450 REM
1460 SD$=ST$(DADR)
1500 LIN$=SD$&"   "&CHR$(A1)&" "&CHR$(A2)&"  "&
CHR$(A3)&" " &CHR$(A4)&"  "&CHR$(A5)&" "&CHR$(A6)&
"  "&CHR$(A7)&" "&CHR$(A8)&" "
```

```
1510 IF PY$="Y" THEN 1560
1520 LINPO$=LINO$
1530 LINP1$=LIN$
1540 PY$="Y"
1550 GOTO 1620
1560 LINPO$=LINPO$&LINO$
1570 LINP1$=LINP1$&LIN$
1580 PRINT #1:LINPO$
1590 PRINT #1:LINP1$
1600 PY$="N"
1610 IF (ADRS>LST)THEN 2550
1620 ADRS=ADRS+8*BK
1630 IF ADRS>=0 THEN 1650
1640 ADRS=CHG(ADRS)
1650 CALL KEY(0,R,S)
1660 IF S=0 THEN 730
1670 PRINT "P=SWCH,C=CHG ADR"
1680 INPUT "E=END,Q=QUIT,ELSE CON:":SEL$
1690 IF POS(SEL$,"E",1)>0 THEN 2550
1700 REM 69 FOR 'E' 81 FOR 'Q'
1710 IF POS(SEL$,"Q",1)>0 THEN 2570
1720 IF POS(SEL$,"P",1)>0 THEN 1750
1730 IF POS(SEL$,"C",1)>0 THEN 630
1740 GOTO 730
1750 IF PRT$="Y" THEN 1760 ELSE 1780
1760 PRT$="N"
1770 GOTO 1730
1780 PRT$="Y"
1790 IF PRT=1 THEN 1730
1800 OPEN #1:PRNTR$
1810 PRT=1
1820 GOTO 1730
1830 DDADR=16376
1840 INPUT "START PGM CODE HEX ADDR-":STRT$
1850 H$=STRT$
1860 GOSUB 2810
1870 STRT=D
1880 INPUT "LAST PGM CODE HEX ADDR-":LST$
1890 H$=LST$
1900 GOSUB 2810
1910 LST=D
1920 CALL PEEK(DDADR,A1,A2,A3,A4,A5,A6,A7,A8)
1930 IF (A1<85)+(90<A1)THEN 2420
1940 ADRS=A7*A8
1950 GOSUB 750
1960 IF (DADR<-25600)*(DADR>-26626)THEN 2400
1970 CALL PEEK(DADR,B1,B2)
1980 ADRS=B1*B2+26
1990 GOSUB 750
2000 IF (DADR<-25600)*(DADR>-26626)THEN 2400
2010 CALL PEEK(DADR,R13A,R13B,R14A,R14B,R15A,R15B)
2020 R14=R14A*R14B
2030 IF (R14<STRT)+(LST<R14)THEN 2400
2040 HX$=""
2050 VALA=A7
2060 GOSUB 2580
2070 VALA=A8
2080 GOSUB 2580
2090 HXADR$=HX$
2100 HX$=""
2110 VALA=R13A
2120 GOSUB 2580
2130 VALA=R13B
2140 GOSUB 2580
2150 HXR13$=HX$
2160 HX$=""
2170 VALA=R14A
2180 GOSUB 2580
2190 VALA=R14B
2200 GOSUB 2580
2210 HXR14$=HX$
2220 HX$=""
2230 VALA=R15A
2240 GOSUB 2580
2250 VALA=R15B
2260 GOSUB 2580
2270 LINPO$=CHR$(A1)&CHR$(A2)&CHR$(A3)&CHR$(A4)&
CHR$(A5)&CHR$(A6)&" ADDR="&HXADR$
2280 LINP1$=" WSPTR="&HXR13$
2290 LINP2$="RETURN="&HXR14$
2300 LINP3$="STATUS="&HX$
2310 PRINT LINPO$
2320 PRINT TAB(8);LINP1$
2330 PRINT TAB(8);LINP2$
2340 PRINT TAB(8);LINP3$
2350 IF PRT$<>"Y" THEN 2400
2360 PRINT #1:LINPO$
2370 PRINT #1:TAB(8);LINP1$
2380 PRINT #1:TAB(8);LINP2$
2390 PRINT #1:TAB(8);LINP3$
2400 DDADR=DDADR-8
2410 GOTO 1920
2420 INPUT "STRT DATA ADDR-":STRT$
2430 H$=STRT$
2440 GOSUB 2810
2450 STRT=D
2460 ADRS=STRT
2470 INPUT "END DATA ADDR-":LST$
2480 H$=LST$
2490 GOSUB 2810
2500 LST=D
2510 BK=1
2520 IF PRT$<>"Y" THEN 730
2530 PRINT #1:" "
2540 GOTO 730
2550 IF DEB$<>"Y" THEN 2570
2560 CALL LINK("DEBUG")
2570 STOP
2580 REM SUB HEX2(VALA,HX$)
2590 VALU=VALA
2600 HXN$=""
2610 SAV=INT(VALU/16)
2620 DIGT=VALU-SAV
2630 D=DIGT
2640 GOSUB 2750
2650 HXN$=H$&HXN$
2660 VALU=SAV
2670 IF VALU>0 THEN 2610
2680 L=LEN(HXN$)
2690 IF L=2 THEN 2730
2700 FOR I=L TO 1
2710 HXN$="0"&HXN$
2720 NEXT I
2730 HX$=HX$&HXN$
2740 RETURN
2750 REM SUB HDIG(D,H$)
2760 IF D>=10 THEN 2790
2770 H$=STR$(D)
2780 RETURN
2790 H$=CHR$(55+D)
2800 RETURN
2810 REM SUB DEC2(H$,D)
2820 D$=SEG$(H$,1,1)
2830 IF D$>="A" THEN 2840 ELSE 2860
2840 D=ASC(D$)-55
2850 GOTO 2870
2860 D=VAL(D$)
2870 L=LEN(H$)
2880 FOR I=2 TO L
2890 DI$=SEG$(H$,I,1)
2900 IF DI$>="A" THEN 2910 ELSE 2930
2910 DI=ASC(DI$)-55
2920 GOTO 2940
2930 DI=VAL(DI$)
2940 D=D+DI
2950 NEXT I
2960 RETURN
2970 END
```

TOUCHTONE FREQUENCIES

From the Chattanooga Users Group

Your computer can generate sounds that will dial your phone. It has to be a dual tone since a single tone is not always accepted properly. Use these in a program to get the sounds and then hold your telephone near the speaker to see if it works. This information has been around a long time, so I don't know who originally came up with it.

```
1-CALL SOUND(100,1209,0,697,0)
2-CALL SOUND(100,1336,0,697,0)
3-CALL SOUND(100,1447,0,697,0)
4-CALL SOUND(100,1209,0,770,0)
5-CALL SOUND(100,1336,0,770,0)
6-CALL SOUND(100,1447,0,770,0)
7-CALL SOUND(100,1209,0,852,0)
8-CALL SOUND(100,1336,0,852,0)
9-CALL SOUND(100,1447,0,852,0)
0-CALL SOUND(100,1336,0,941,0)
```

Corrections

by the Editor

For some reason, I could not get TI-WRITER to print lines 1940, 1980, 2020, 2620, and 2940 from the Hex Dump Program properly. They should read as follows:

```
1940 ADRS Equals ( A7 x 256 ) Plus A8
1980 ADRS Equals ( B1 x 256 ) Plus B2 Plus 26
2020 R14  Equals ( R14A x 256 ) Plus R14B
2620 DIGT Equals VALU Minus ( SAV * 16 )
2940 D    Equals ( D x 16 ) Plus DI
```

DELAWARE VALLEY USERS' GROUP
P. O. BOX 6240
STANTON BRANCH
WILMINGTON, DE 19804

Subscription expires with date on label

FIRST CLASS MAIL

USA 22

TO:

0000 FOREST LANE USERS GROUP
JOE GILLO - 4413 CORNELL DR
GARLAND TX 75042