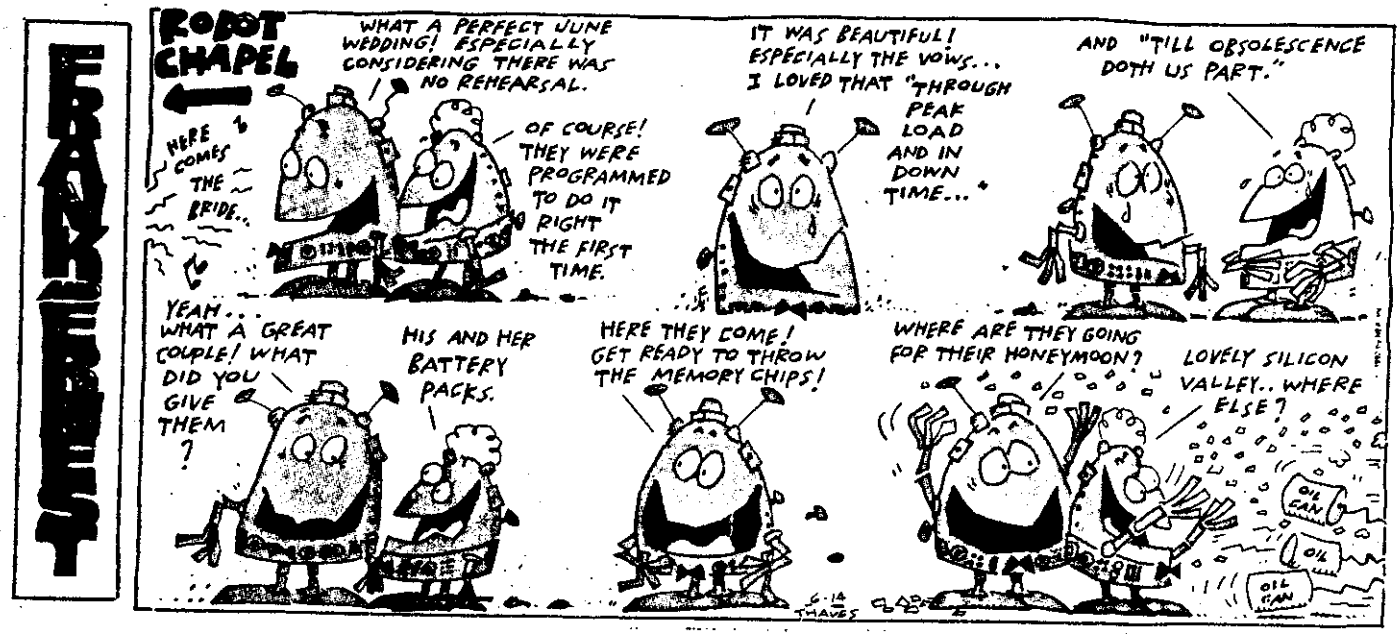
Newsletter Nine-T-Nine

JUNE/JULY 1992

DOUBLE ISSUE

Texas Instruments
USERS GROUP
TORONTO



FROM:
9T9 USERS GROUP
15 KERSDALE AVE.
TORONTO, ONT., M6M-1C9
CANADA

NEWSLETTER 9T9

9T9 USERS GROUP

COMPLAINT FORM

list nature
of complaint
in box below

9T9 USERS GROUP EXECUTIVE COMMITTEE

PRESIDENT Steve Mickelson (657-1494)
VICE-PRESIDENT Neil Allen (236-0842)
SECRETARY/MEMBERSHIPS Randy Rossetto (469-3468)
TREASURER/OFFICER AT LARGE Cecil Chin (671-2052)

LIBRARY DIRECTORS

Gary Bowser (960-0925)
Andy Parkinson (275-4427)
Steve Findlay (416) 727-6807

NEWSLETTER EDITOR

Steve Mickelson (657-1494)

MEMBERSHIP FEE'S

FULL MEMBERSHIP ..... $30.00 / year
NEWSLETTER SUBSCRIPTION ..... $20.00 / year

All memberships are household memberships. A newsletter subscription is only for those who do not wish to attend meeting, but wish to receive our newsletter and have access to our library. You are welcome to visit one of our general meetings before joining the group. If you wish more information contact either our president, in writing, at the club address on the front cover or by phone.

The meetings are usually held on the last Wednesday of each month.(exceptions are December's meeting date, usually mid-month and the months of July and August, when there are no meetings. Consult this issue of Newsletter 9T9 for the date and time of the next meeting. Meetings are usually held at Neil Allen's place, 52 Graystone Gardens,south of Bloor St., just west of Islington Ave., at 7:30 P.M. from 7:30 - 10:30 PM

BBS

The 9T9 Users Group supports the Toronto BBS. The TI Tower BBS #(416) 921-2731, 300/1200/2400 BPS, 24 hrs. Sysop, Gary Bowser

MAILING ADDRESS:

9T9 Users Group, 15 Kersdale Ave., Toronto, Ontario, M6M 1C9, Canada

COMMERCIAL ADVERTISING

Any business wishing to reach our membership may advertise in our newsletter.
The rates are as follows. (width by height):
FULL PAGE (7" x 10") $30.00
HALF PAGE (7" x 5") $15.00
QUARTER PAGE (7" x 2 1/2") $ 7.50
Please have your ad's camera ready and paid for in advance. For more information contact the editor. Don't forget, that any member wishing to place ad's, may do so free of charge as long as they are not involved in a commercial enterprise.

NEWSLETTER ARTICLES

Members are encouraged to contribute to the newsletter in the form of articles, mini programs, helpful tips, hardware modifications, jokes, cartoons and questions. Any article may be submitted in any form by mail or modem. We welcome the reprinting of any article appearing in this newsletter providing credit is given to the author and 9T9. If more information is required, call the editor. The names, 9T9, Nine-T-Nine, Newsletter 9T9, 9T9 Users Group, and Nine-T-Nine Users Group are Copyright.(c).1979-1992, by the 9T9 Users Group of Toronto, Canada, all rights reserved.

DISCLAIMER

Opinions expressed in this newsletter are those of the writers and are not necessarily those of the 9T9 USERS' GROUP. 9T9 cannot assume liability for errors or omissions in articles, programs or advertisements. Any hardware modification or project is presented for informational purposes, and the author, newsletter editor, staff and/or 9T9 Users Group &cannot be held liable for any damage to the user's equipment. All such projects are done at your own risk!

"Sorry, Pop, but when I grow up I've decided to go into the software business".

TIDBITS #60

-By Steve Mickelson, President 9T9 Users Group
Compuserve 76545,1255; Delphi SMICKELSON; GEnie S.MICKELSON

Summer Break:
Although this will be the last issue of Newsletter 9T9, until the August-September issue, I find that I'm already quite busy and involved with summer activities, for my family. My son is in tyke rep. and house league baseball, my daughter has her eighth grade graduation and concert, as well as other family activities, have made free time for computing, difficult indeed!

I missed the May meeting to attend one such activity, and it looks like I'll miss the June meeting as well. At least I've managed to get the newsletter together and drop same by Neil's place before the May meeting. And I hope I can do the same with this edition, as well!

TIC-TALK-TOE:
I apologize to Bob Tisdale, who called me from Peterborough, regarding a problem with the TIC-TALK-TOE program, I converted to Extended Basic. I was not able to help him with the listing and have yet to get a disk out to him, as I had promised. So, I've decided to include a listing of the program, and the load program as well!

```
100 CALL CLEAR
110 CALL SCREEN(11):: CALL SAY("PLEASE WAIT FOR THE PROGRAM TO LOAD")
120 PRINT "Loading..."
130 RUN "DSK1.TICTALKTOE"
```

```
1 ! DO NOT REMOVEITICTALKTOE FOR SPEECH EDITOR CONVERTED TO EXTENDED BASIC & (C)
1989 BY STEVE MICKELSON & PRESENTED TO 9T9 U.G. 3/30/89
2 ! IF YOU ARE USING MECHATRONIC X/BASIC II CHANGE 5130 & 5170 CALL LINK("POKEV"
....) TO CALL VPOKE(...)
10 CALL SCREEN(2):: CALL VDPUTIL2
20 PRINT : : : : : : : : : : : : : : : : : : : : : : : : : :"     COPYRIGHT 198
9": :"-BY STEVE MICKELSON": : :
30 CALL CHAR(152,"80C0E0F0F8FCFEFF")
40 CALL CHAR(153,"0103070F1F3F7FFF")
50 CALL CHAR(154,"FF7F3F1F0F070301")
60 CALL CHAR(155,"FFFEFCF8F0E0C080")
70 CALL CHAR(156,"FFFFFFFFFFFFFFFF")
80 M$="030315622"
90 GOSUB 400
100 M$="150315622"
110 GOSUB 400
120 CALL HCHAR(3,25,152)
130 CALL HCHAR(4,26,152)
140 FOR R=5 TO 8
150 CALL HCHAR(R,CO+4,152)
160 CALL HCHAR(R,CO+27,152)
170 CALL HCHAR(R,CO+3,154)
180 CALL HCHAR(R,CO+26,154)
190 CO=CO+1
200 NEXT R
210 CO=30
220 FOR R=10 TO 13
230 CALL HCHAR(R,CO,155)
240 CALL HCHAR(R,CO-23,155)
250 CALL HCHAR(R,CO-1,153)
260 CALL HCHAR(R,CO-24,153)
270 CO=CO-1
280 NEXT R
290 CALL HCHAR(4,3,156)
300 CALL HCHAR(9,7,156)
310 CALL HCHAR(14,3,156)
320 CALL HCHAR(9,30,156)
330 CALL HCHAR(4,25,154)
340 CALL HCHAR(14,26,155)
350 CALL HCHAR(15,25,155)
360 CALL HCHAR(14,25,153)
370 GOSUB 4270
380 GOSUB 4770
390 GOTO 420
400 CALL HCHAR(VAL(SEG$(M$,1,2)),VAL(SEG$(M$,3,2)),VAL(SEG$(M$,5,3)),VAL(SEG$(M$
,8,2)))
410 RETURN
420 CALL CLEAR
430 CALL SCREEN(2)
440 CALL COLOR(1,16,2)
450 CALL COLOR(2,16,2)
460 CALL COLOR(3,13,2)
470 CALL COLOR(4,13,2)
480 FOR X=5 TO 8
490 CALL COLOR(X,6,2)
500 NEXT X
510 FOR X=9 TO 12
520 CALL COLOR(X,9,2)
530 NEXT X
540 CALL COLOR(13,11,2)
550 CALL COLOR(14,14,2)
560 CALL COLOR(15,8,2)
570 CALL CHAR(136,"C0E070381C0E0703")
580 CALL CHAR(137,"03070E1C3870E0C0")
590 CALL CHAR(144,"0F1F3F7FE0E0E0E0")
600 CALL CHAR(145,"F0F8FCFE07070707")
610 CALL CHAR(146,"E0E0E0E07F3F1F0F")
620 CALL CHAR(147,"07070707FEFCF8F0")
630 CALL CHAR(128,"00183C7E7E3C1800")
640 M$="WELCOME TO TIC-TALK-TOE    Key:"
650 SR=1
660 SC=2
670 GOSUB 3360
680 M$="123"
690 SR=2
700 SC=28
710 GOSUB 3360
720 M$="456"
730 SR=3
740 SC=28
750 GOSUB 3360
760 M$="789"
770 SR=4
780 SC=28
790 GOSUB 3360
800 IF SP=2 THEN 820
810 CALL SAY("HELLO+HELLO+HERE+WE+GO")
820 GOSUB 3950
830 CALL HCHAR(9,10,128,14)
840 CALL SOUND(1000,262,6)
850 FOR X=1 TO 300
860 NEXT X
870 CALL HCHAR(14,10,128,14)
880 CALL SOUND(1000,262,6,330,6)
890 FOR X=1 TO 300
900 NEXT X
910 CALL VCHAR(5,14,128,14)
920 CALL SOUND(1000,262,6,330,6,392,6)
930 FOR X=1 TO 300
940 NEXT X
950 CALL VCHAR(5,19,128,14)
960 CALL SOUND(1500,330,6,392,6,494,6)
970 FOR X=1 TO 300
980 NEXT X
990 IF S1=1 THEN 1280
1000 M$="PRESS"
1010 SR=20
1020 SC=3
1030 GOSUB 3360
1040 M$="1-9"
1050 SR=21
1060 SC=4
1070 GOSUB 3360
1080 M$="TO MOVE"
1090 SR=22
1100 SC=2
1110 GOSUB 3360
1120 IF SP=2 THEN 1140
1130 CALL SAY("GET+#READY TO START#+NOW")
1140 M$="YOU ARE 'X'"
1150 SR=20
1160 SC=21
1170 GOSUB 3360
1180 IF SP=2 THEN 1200
1190 CALL SAY("YOU+ARE X")
1200 M$="I AM 'O'"
1210 SR=21
1220 SC=24
1230 GOSUB 3360
1240 IF SP=2 THEN 1260
1250 CALL SAY("I+AM O")
1260 RANDOMIZE
1270 DIM C(9),W(72,5)
1280 G=G+1
1290 IF G<>72 THEN 1310
1300 G=1
1310 FOR I=1 TO 9
1320 C(I)=0
1330 NEXT I
1340 D=0
1350 C1=0
1360 RESTORE
1370 RAND=RND
1380 IF RAND<.5 THEN 1460
1390 M$="*** I'LL GO FIRST ***"
1400 SR=24
```

```
1410 SC=6
1420 GOSUB 3360
1430 IF SP=2 THEN 1450
1440 CALL SAY("O WILL+MOVE+FIRST")
1450 GOTO 1730
1460 M$="*** YOU GO FIRST ***"
1470 SR=24
1480 SC=6
1490 GOSUB 3360
1500 IF SP=2 THEN 1520
1510 CALL SAY("X WILL+MOVE+FIRST")
1520 IF SP=2 THEN 1540
1530 CALL SAY("YOUR+MOVE+NOW")
1540 CALL KEY(0,KEY,S)
1550 IF S=0 THEN 1540
1560 M=KEY-48
1570 REM +++YOUR MOVE+++
1580 GOSUB 2420
1590 IF E=0 THEN 1640
1600 CALL SOUND(500,110,4,-3,0)
1610 IF SP=2 THEN 1630
1620 CALL SAY("UHOH #TRY AGAIN#")
1630 GOTO 1540
1640 OX$="X"
1650 ON M GOSUB 2760,2800,2840,2880,2920,2960,3000,3040,3080
1660 C(M)=2
1670 GOSUB 2480
1680 D=D+1
1690 C1=C1+1
1700 W(G,C1)=M
1710 IF E=1 THEN 2190
1720 IF D=9 THEN 2190
1730 M=1
1740 IF C(M)<>0 THEN 1790
1750 C(M)=1
1760 GOSUB 2480
1770 IF E=1 THEN 2120
1780 C(M)=0
1790 M=M+1
1800 IF M<>10 THEN 1740
1810 M=1
1820 IF C(M)<>0 THEN 1870
1830 C(M)=2
1840 GOSUB 2480
1850 IF E=1 THEN 2110
1860 C(M)=0
1870 M=M+1
1880 IF M<>10 THEN 1820
1890 IF G=1 THEN 2020
1900 I=1
1910 E=0
1920 FOR M=1 TO C1
1930 IF W(I,M)=W(G,M)THEN 1950
1940 E=1
1950 NEXT M
1960 IF E=1 THEN 2000
1970 M=W(I,M+1)
1980 IF M=0 THEN 2000
1990 IF C(M)=0 THEN 2110
2000 I=I+1
2010 IF I<>G THEN 1910
2020 RAND=RND
2030 IF RAND<.4 THEN 2080
2040 READ M
2050 IF C(M)<>0 THEN 2020
2060 GOTO 2110
2070 DATA 5,7,3,9,1,6,8,4,2
2080 M=INT(9*RND)+1
2090 IF C(M)=0 THEN 2110
2100 GOTO 2080
2110 C(M)=1
2120 OX$="O"
2130 ON M GOSUB 2760,2800,2840,2880,2920,2960,3000,3040,3080
2140 GOSUB 2480
2150 IF E=1 THEN 2190
2160 D=D+1
2170 IF D=9 THEN 2190
2180 GOTO 1520
2190 IF Q=0 THEN 2350
2200 IF Q<>1 THEN 2320
2210 M$="I WIN!!"
2220 SR=12
2230 SC=2
2240 GOSUB 3360
2250 IF SP=2 THEN 2270
2260 CALL SAY("#I WIN#")
2270 GOSUB 3270
2280 Y1=Y1+1
2290 GOTO 3620
2300 Y1=Y1+1
2310 GOTO 2370
2320 REM +++YOU WON+++
2330 X1=X1+1
2340 GOTO 3190
2350 REM +++ITS A TIE+++
2360 X2=X2+1
2370 FOR I=1 TO 5
2380 W(G,I)=0
2390 NEXT I
2400 G=G-1
2410 GOTO 3120
2420 E=0
2430 IF M<1 THEN 2460
2440 IF M>9 THEN 2460
2450 IF C(M)=0 THEN 2470
2460 E=1
2470 RETURN
2480 E=0
2490 Q=C(5)
2500 IF Q=0 THEN 2610
2510 IF C(1)<>Q THEN 2550
2520 IF C(9)<>Q THEN 2550
2530 E=1
2540 RETURN
2550 IF C(3)<>Q THEN 2570
2560 IF C(7)=Q THEN 2530
2570 IF C(2)<>Q THEN 2590
2580 IF C(8)=Q THEN 2530
2590 IF C(4)<>Q THEN 2610
2600 IF C(6)=Q THEN 2530
2610 Q=C(1)
2620 IF Q=0 THEN 2670
2630 IF C(2)<>Q THEN 2650
2640 IF C(3)=Q THEN 2530
2650 IF C(4)<>Q THEN 2670
2660 IF C(7)=Q THEN 2530
2670 Q=C(9)
2680 IF Q=0 THEN 2730
2690 IF C(3)<>Q THEN 2710
2700 IF C(6)=Q THEN 2530
2710 IF C(7)<>Q THEN 2730
2720 IF C(8)=Q THEN 2530
2730 Q=0
2740 RETURN
2750 GOTO 5190
2760 SR=6
2770 SC=11
2780 GOSUB 3440
2790 RETURN
2800 SR=6
2810 SC=16
2820 GOSUB 3440
2830 RETURN
2840 SR=6
2850 SC=21
2860 GOSUB 3440
2870 RETURN
2880 SR=11
2890 SC=11
2900 GOSUB 3440
2910 RETURN
2920 SR=11
2930 SC=16
2940 GOSUB 3440
2950 RETURN
2960 SR=11
2970 SC=21
2980 GOSUB 3440
2990 RETURN
3000 SR=16
3010 SC=11
3020 GOSUB 3440
3030 RETURN
3040 SR=16
3050 SC=16
3060 GOSUB 3440
3070 RETURN
3080 SR=16
3090 SC=21
3100 GOSUB 3440
3110 RETURN
3120 M$="DRAW"
3130 SR=11
3140 SC=3
3150 IF SP=2 THEN 3170
3160 CALL SAY("THE1+GAMES+A1+DRAW")
3170 GOSUB 3360
3180 GOTO 3620
3190 M$="YOU WIN"
3200 SR=12
3210 SC=2
3220 IF SP=2 THEN 3240
3230 CALL SAY("#YOU WIN#+#YOU WIN#")
3240 GOSUB 3360
3250 GOSUB 3950
3260 GOTO 3620
3270 REM  +++NYA NYA SONG+++
3280 FOR REP=1 TO 2
3290 CALL SOUND(900,131,6,262,6)
3300 CALL SOUND(600,110,6,220,6)
3310 CALL SOUND(200,147,6,294,6)
3320 CALL SOUND(900,131,6,262,6)
3330 CALL SOUND(900,110,6,220,6)
3340 NEXT REP
3350 RETURN
3360 REM +++PRINT CHARS+++
3370 FOR X=1 TO LEN(M$)
3380 CALL HCHAR(SR,SC,ASC(SEG$(M$,X,1)))
3390 CALL SOUND(10,880,6)
3400 SC=SC+1
3410 NEXT X
3420 RETURN
3430 REM +++XO PRINT+++
3440 IF OX$="O" THEN 3550
3450 IF SP=2 THEN 3470
3460 CALL SAY("GOOD MOVE")
3470 CALL HCHAR(SR,SC,136)
3480 CALL HCHAR(SR,SC+1,137)
3490 CALL HCHAR(SR+1,SC,137)
3500 CALL HCHAR(SR+1,SC+1,136)
3510 FOR Z=1 TO 20
3520 CALL SOUND(1,440,3)
3530 NEXT Z
3540 RETURN
3550 IF SP=2 THEN 3570
3560 CALL SAY("I+WILL+MOVE+HERE")
3570 CALL HCHAR(SR,SC,144)
3580 CALL HCHAR(SR,SC+1,145)
3590 CALL HCHAR(SR+1,SC,146)
3600 CALL HCHAR(SR+1,SC+1,147)
3610 GOTO 3510
3620 XX1$=STR$(X1)
3630 YY1$=STR$(Y1)
3640 XX2$=STR$(X2)
3650 M$="SCORE:"
3660 SR=9
3670 SC=26
3680 GOSUB 3360
3690 M$="you -"&XX1$
3700 SR=11
3710 SC=26
3720 GOSUB 3360
3730 M$="me  -"&YY1$
3740 SR=12
3750 SC=26
3760 GOSUB 3360
3770 M$="ties-"&XX2$
3780 SR=13
3790 SC=26
3800 GOSUB 3360
3810 M$="PRESS ANY KEY TO PLAY AGAIN "
3820 SR=24
3830 SC=3
3840 IF SP=2 THEN 3860
3850 CALL SAY("ARE+YOU+#READY TO START#+AGAIN")
3860 GOSUB 3360
3870 CALL KEY(0,K,S)
3880 IF S=0 THEN 3870
3890 S1=1
3900 FOR X=5 TO 18
3910 CALL HCHAR(X,1,32,32)
3920 NEXT X
3930 CALL HCHAR(24,1,32,32)
3940 GOTO 830
3950 REM INTRO/VICTORY TUNE
3960 BEAT=100
3970 FOR X=1 TO 2
3980 RESTORE 4240
3990 FOR Y=1 TO 15
4000 READ DUR,HZ
4010 CALL SOUND(DUR*BEAT,HZ,2)
4020 NEXT Y
4030 NEXT X
4040 FOR X=1 TO 1
4050 RESTORE 4250
4060 FOR Y=1 TO 13
4070 READ DUR,HZ
4080 CALL SOUND(DUR*BEAT,HZ,2)
4090 NEXT Y
4100 NEXT X
4110 RESTORE 4260
4120 FOR X=1 TO 7
4130 READ DUR,HZ
4140 CALL SOUND(DUR*BEAT,HZ,6)
4150 NEXT X
4160 RESTORE 4240
4170 FOR X=1 TO 15
4180 READ DUR,HZ
4190 CALL SOUND(DUR*BEAT,HZ,2)
4200 NEXT X
4210 FOR WAIT=1 TO 200
4220 NEXT WAIT
4230 RETURN
4240 DATA 3,523.2,587.1,659.1,698.1,659.1,587.2,523.1,392.3,523.2,587.1,659.1,69
8.1,659.1,698.2,784
4250 DATA 4,523.1,440.1,523.4,587.1,659.1,587.2,523.1,392.2,523.1,392.1,523.1,58
7.3,523
4260 DATA 2,392.1,330.2,392.1,330.1,392.1,440.3,392
4270 RESTORE 4870
4280 FOR CL=9 TO 14
4290 READ N1,N2,N3
4300 CALL SCREEN(CL)
4310 CALL SOUND(-1200,N1,3,N2,3,N3,3)
4320 GOSUB 4480
4330 FOR WAIT=1 TO 10
4340 NEXT WAIT
4350 NEXT CL
4360 CALL COLOR(9,16,1)
4370 CALL COLOR(10,16,1)
4380 CALL COLOR(11,16,1)
4390 CALL COLOR(12,16,1)
4400 FOR X=1 TO 10
```

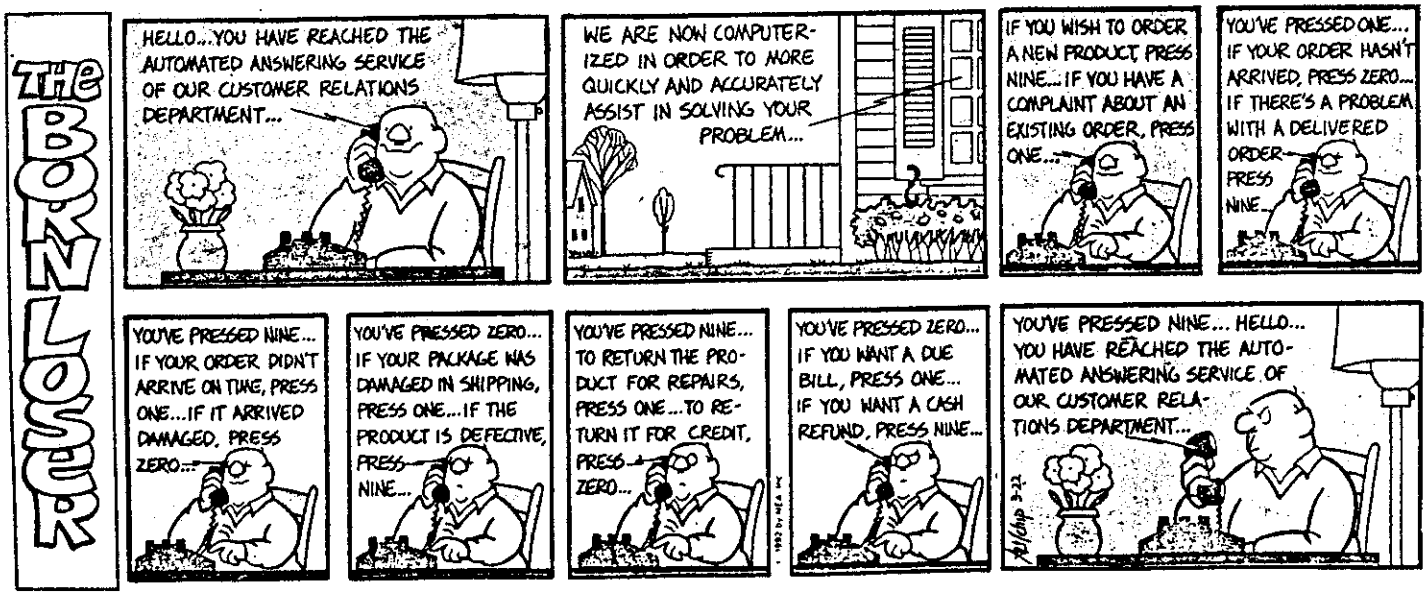
```
4410 FOR Z=2 TO 16
4420 FOR WAIT=1 TO 3
4430 NEXT WAIT
4440 CALL COLOR(16,Z,1)
4450 NEXT Z
4460 NEXT X
4470 RETURN
4480 ON CL-8 GOSUB 4500,4530,4560,4590,4620,4650
4490 RETURN
4500 Q$="0608 9T9 U.G."
4510 GOSUB 4690
4520 RETURN
4530 Q$="0617 PRESENTS"
4540 GOSUB 4690
4550 RETURN
4560 Q$="0911 9T9 tic"
4570 GOSUB 4690
4580 RETURN
4590 Q$="0919talk toe"
4600 GOSUB 4690
4610 RETURN
4620 Q$="1208uses XB"
4630 GOSUB 4690
4640 RETURN
4650 Q$="1217and SPEECH"
4660 GOSUB 4690
4670 RETURN
4680 RETURN
4690 SR=VAL(SEG$(Q$,1,2))
4700 SC=VAL(SEG$(Q$,3,2))
4710 Q$=SEG$(Q$,5,LEN(Q$)-4)
4720 FOR X=1 TO LEN(Q$)
4730 CALL HCHAR(SR,SC,ASC(SEG$(Q$,X,1)))
4740 SC=SC+1
4750 NEXT X
4760 RETURN
4770 REM CALL CLEAR
4780 CALL SCREEN(14)
4790 PRINT "TIC-TALK-TOE" IS INTENDED":"FOR USE WITH OR WITHOUT THE":"  spe
ech synthesizer":"          AND
4800 PRINT :"extended basic":::
4810 PRINT "CHOOSE YOUR OPTION":::
4820 PRINT "1 - TIC-TALK-TOE WITH":"     MUSIC & speech":
```

```
4830 PRINT "2 - TIC-TALK-TOE WITH":"     MUSIC ONLY":::
4840 INPUT "ENTER 1 OR 2":SP
4850 IF (SP<>2)*(SP<>1)THEN 4840
4860 RETURN
4870 DATA 220,262,330,262,330,392,330,392,494,392,494,587,494,587,698,587,698,8
0
4880 SUB VOPUTIL2
4890 CALL CLEAR :: CALL INIT ::CALL LOAD(8196,63,232)
4900 CALL LOAD(16360,80,79,75,69,82,32,38,12,80,79,75,69,86,32,37,164,80,69,69,7
5,86,32,37,36)
4910 CALL LOAD(9491,100)
4920 CALL LOAD(9508,2,224,37,20,3,0,0,0,2,0,0,100,200,0,37,18,4,192,2,1,0,1,4,3,
2,32,12,4,32)
4930 CALL LOAD(9536,32,24,18,184,192,32,131,74,2,1,37,0,208,160,131,18,9,130,2,3
4,255,255,4,32,32,44)
4940 CALL LOAD(9562,4,197,209,34,36,255,9,132,19,21,4,195,60,224,37,18,200,5,131
,76,200,5,131,78,200,5)
4950 CALL LOAD(9588,131,80,2,5,64,0,161,68,2,131,0,1,17,6,2,5,65,0,161,67,6,196,
200,4,131,76)
4960 CALL LOAD(9614,200,5,131,74,4,192,192,66,5,129,4,37,254)
4970 CALL LOAD(9636,2,224,37,20,3,0,0,0,4,192,2,1,0,1,200,1,37,18,4,32,32,12,4,3
2,32,24,18,184)
4980 CALL LOAD(9664,200,32,131,74,37,0,184,32,131,18,37,19,2,3,0,2)
4990 CALL LOAD(9680,4,192,192,67,4,32,32,12,4,32,32,24,18,184,216,224,131,75,37,
0,5,131,136,3)
5000 CALL LOAD(9704,37,18,22,242,192,32,37,0,2,1,37,2,192,131,2,34,255,254,4,32,
32,36)
5010 CALL LOAD(9726,4,192,216,0,131,124,2,224,131,224,4,96,0,112)
5020 CALL LOAD(9740,3,0,0,0,4,192,2,1,0,1,4,32,32,12,200,32,131,74,37,18,2,1,0,2
,4,32,32,12,4,32)
5030 CALL LOAD(9770,32,24,18,184,192,32,131,74,208,32,37,19,4,32,32,48,4,91)
5040 CALL LOAD(8194,39,04)
5050 SUBEND
5060 SUB CHAR(A,A$):: L=LEN(A$)
5070 A$=A$&RPT$("0",16-L)
5080 FOR I=1 TO 16 STEP 2
5090 A1$=SEG$(A$,I,1)
5100 A2$=SEG$(A$,I+1,1)
5110 IF A1$<":" THEN A1=VAL(A1$)*16 ELSE A1=(ASC(A1$)-55)*16
5120 IF A2$<":" THEN A1=A1+VAL(A2$)ELSE A1=A1+ASC(A2$)-55
5130 CALL LINK("POKEV",767+8*A+(I+1)/2,A1)
5140 NEXT I
5150 SUBEND
5160 SUB COLOR(A,B,C)
5170 CALL LINK("POKEV",2063+A,(B-1)*16+C-1)
5180 SUBEND
5190 END
```

9T9 Recall:

One last thing, I would like to ask the executive, as well as any readers, contribute to the newsletter, by writing something about the TI computer, the club, how it has changed or affected you, what hardware/software you like, or would like to see, any, or all, of the above, and/or any other aspect of the TI. If you were to make an informal speech to the TI-ers at our 9T9 Recall Reunion, what would you say that you think would be of interest to us? A paragraph or article would do. I would like to get as many contributions, as possible, compiled for the October issue of the newsletter. You can send your submission to me c/o the club's mailing address, or bring it to the September meeting.

So have a safe and fun summer, see you in September!



CONSOLE DEBUGGING HELP
By John Guion
Dallas TI Home Computer Group

PROBLEM AREAS

1) Console will not power up

2) Keyboard errors

3) Intermittent console lock up

4) Module errors

5) Joystick port errors

6) Video output difficulty

7) Sound problems

The TI-99/4A Console and Peripheral Expansion System Technical Data manual available from Texas Instruments' Dealer Parts Department [(806) 741-2265] will serve as an excellent source for schematics and part location guide.

The information contained herein is only intended for use as a reference for possible debugging procedures. It is not intended as a repair guide for the common user with little or no knowledge of digital electronics or the basic structure of the TI-99/4A system. The author assumes no responsibility for damages resulting from improper use of this information.

This information should only be reproduced in its entirety with credit to the original author.

1) CONSOLE WILL NOT POWER UP

1.1 General information.

Failure of the TI-99/4A console to power up and produce the TI title screen is a common problem that is also the hardest to track down and fix since failure of nearly any component in the console or power supply can cause this.

The following are not intended as solutions to the problem, but merely as points to check that may aid in finding the actual problem and fixing it.

Unless a particular part is suspected, replace any socketed chips possible with known working equivalents before desoldering any components. Since the socketed chips are common causes of lock up, eliminating them as possible problems first may save excess soldering on the board. The console will power up if the sound chip is removed entirely, but not if that chip is shorted internally.

A simple TTL logic probe can be used for tracing signals in the circuit. An oscilloscope may also be used and has the advantage of being able to check clock signals for proper frequency. When a signal should exist as an output from a particular device, be sure to check that device's input for proper signals before attempting to replace the component. When checking for locked up signals, try to trace all signals back through the circuit to the point of origin. A set of schematics (available from several sources, including TI) will help greatly in this part of debugging. Tracing locked signals can determine whether or not the signal is missing due to a faulty component that it must pass through or what power up operation was occurring during lock up.

1.2 Console power up procedure.

A. TMS9900 CPU resets and addresses low ROM locations.

B. TMS9900 initializes.

C. TMS9900 sets up workspace registers in MCM6810 RAM.

D. TMS9900 begins GROM read.

E. TMS9900 enters delay loop for about 1/4 second.

F. TMS9919 sound chip is disabled.

G. TMS9918A VDP chip is initialized.

H. 4116 VDP RAM is initialized (requires about 1 second).

I. Title screen is loaded into VDP.

J. TMS9919 sound chip emits beep.

K. TMS9900 CPU enters keyboard scan.

L. System is ready for use.

1.3 Voltage/signal checklist.

A. Check power supply for +5V, +12V, and -5V. Lack of -5V often results in a gray flickering screen on power up. Check for +5V on chips throughout board. Check TMS9900 for -5V at pin 1; +5V at pins 2, 33, 59, and 64; and +12V at pin 27. If any voltages are missing, check for shorts on main board. Replace power supply if necessary.

B. Check TMS9900 pins 8, 9, 25, and 28 for clock signal. If not found, check TIM9904 clock generator pins 1, 2, 3, and 4 for clock signal. If not found, check TIM9904 supply voltages (+5V at pin 20, +12V at pin 13), crystal, and tank circuit. If no external problem can be found, possible TIM9904 failure.

C. Check TMS9918A pin 39 and pin 40 for the 10.73863 MHz clock. If missing, check crystal and oscillator circuit. Otherwise, check TMS9918A pin 36 and pin 37 for clock outputs. If not found, remove GROMs and sound processor (located next to GROMs) and test again for clock. If missing, possible TMS9918A failure. Reinsert GROMs and sound processor after tests.

D. Check TMS9918A pins 14 (-CSW) and 15 (-CSR) for lock up. If locked up, check memory enable from pin 6 of 74LS32 and pin 13 of 74LS138 located next to MCM6810. Trace signal to find possible failure.

E. Check TMS9918A pin 13 (MODE) for lock up. If locked up, trace signal back to TMS9900. Also check for other components that may be locking up this line (it is used as A14). If no other fault can be found on that line, possible TMS9918A failure.

F. Check TMS9918A pin 1 (-RAS), pin 2 (-CAS), and pin 11 (-R/W) for lock up. If locked up, possible TMS9918A failure.

G. Check TMS9918A pins 17 through 24 (data lines) for signals. If missing, trace to fault. Possible TMS9918A or TMS9900 failure.

H. Check TMS9918A pins 3 through 10 (RAM address/data lines) for signals. If missing, possible TMS9918A failure.

I. Check 4116 RAM pin 14 (DATA OUT) on each chip for signal. Each chip missing signal may be at fault as well as TMS9918A.

J. Check TMS9900 pin 62 (READY) for lock up. If locked up, check TMS9900 pin 6 (-RESET) for signal. If pin 6 is locked up low, possible TIM9904 failure. If high, possible TMS9900 failure. If TMS9900 pin 6 is not locked up, trace circuit back from pin 62 to find fault.

K. Check all three GROMs (CD2155, CD2156, and CD2157) at pin 10 (-CS) and pin 15 (GREADY) for signals. If either is missing, remove all three GROMs and test pin 10 again for signal. If the signal at pin 10 does not exist, trace back through circuit to find failure. If signal exists, replace GROMs one at a time until GROM that causes lock up on pin 15 is found.

L. Check all three GROMs for signal on pin 11 (MO/A14) and pin 12 (M1/DBIN). If missing, trace circuit to find break in signal path.

M. Check each GROM for -5V at pin 14, +5V at pin 9, and -.8V to -.6V at pin 16. If missing, check for broken trace. If -.8V/-.6V is missing or at -5V, check diode connected to that line.

N. Remove sound generator. If console powers up, check pin 16 for +5V, pin 4 for clock from TMS9918A, pin 5 (-WE) for signal, and pin 6 (-CS) for signal from 74LS138 closest to MCM6810. If these signals exist, possible sound chip failure.

O. Check TMS9918A pin 36 for composite video output. If missing, check TMS9918A crystal and clock circuit and pin 16 (-INT) for interrupt signal. If signals exist, possible TMS9918A failure.

P. Check GROMs for clock on pin 13. If missing, check clock output on TMS9918A pin 37. If signal on TMS9918A exists, check for break in signal path. If not, check TMS9918A oscillator circuit. If oscillator operates, possible TMS9918A failure.

Q. Check pin 20 (-CS) of console ROMs for lockup. If locked up, trace circuit back to find fault.

R. Check pins 7 and 9 through 15 of 74LS138 nearest I/O port to determine memory area accessed during lock up. Check pin 4 (-MEMEN) for lock up. If no signal can be found on pin 7 or pins 9 through 15, possible 74LS138 failure.

S. Check pin 11 (-CS) of MCM6810 RAMs for lock up. If locked up, trace circuit back to find fault.

T. Check TMS9901 pin 5 (-CE) for lock up. If locked up, check 74LS138 nearest I/O port for failure. Check TMS9901 pin 11, 17, and 18 for lock up. If locked up, trace circuit back to find fault.

2) KEYBOARD ERRORS

2.1 General information.

After much use, the keyboard will sometimes malfunction and key presses will not appear to have any effect or will only work part of the time (either single keys or groups of keys). Keys may also show multiple entries even though only one key was pressed.

The TI-99/4A's keyboard is set up using an X-Y matrix to allow encoding of output signals from a 74LS156 to drive the interrupt inputs of the TMS9901. This method requires only 15 lines to encode all 48 keys. Keyboard failure is almost always a mechanical problem, but occasionally one of the computer's support components will fail and cause similar problems.

2.2 Possible causes and solutions.

A. If only one key works intermittently or not at all, that single keyswitch is probably dirty or damaged. Some keyboards may allow for disassembly and repair while others make single key repair less practical than replacement of the entire unit.

B. If a group of keys has suddenly failed to work properly, it is likely that the switches in the keyboard are good and that some common component has failed. Typically, this is a broken wire or faulty driver chip. Consult a keyboard schematic to determine control lines common to groups of keys. When a common line is found, check continuity from the keyboard to the main board. If continuity exists, check loading resistors on the control lines from the keyboard connector before replacing any chips.

C. If the ALPHA LOCK key fails to operate properly and the console has been modified with the addition of a diode in the ALPHA LOCK circuit, remove the diode and replace with a piece of wire. The diode is added by some users to allow use of joysticks with the ALPHA LOCK depressed, but it sometimes introduces a timing problem and is not reliable.

D. If some keys do not work at all and others result in improper entries, check the keyboard plug connector for proper alignment.

E. If a group of keys with a control line common to the 74LS156 fail to function and continuity exists to the main board, use a logic probe to check for output pulses on pins 4 through 7 and 9 through 12 of the 74LS156.

F. If the entire keyboard fails to function and continuity exists to the main board, replace the 74LS156 and test again.

G. If 74LS156 replacement has no effect, replace the TMS9901 and test again.

3) INTERMITTENT CONSOLE LOCK UP

3.1 General information.

Occasionally, a console will suffer from lock up during regular use of software. This may be caused by either a software error, a hardware error, or a disturbance of the system. Assuming that software errors and outside disturbance (such as bumping the flex-cable connector accidentally) have been eliminated as possible causes, several conditions may cause random locking up of the console during use.

Causes for random console lock up fall into three main categories: power supply, heat, and poor connections.

3.2 Possible causes and solutions.

A. Check the computer's supply line voltage. The transformer input should be approximately 117VAC. The output of the consoles internal supply should be +5V, +12V, and -5V. Aprroximately 5% variations for each of these are tolerable.

B. Test the computer in different surroundings. If the computer consistently works at another location or with different equipment attached, it may be affected by some components connected to it or by electrical interference from the 117VAC supply.

C. If lock up only occurs after some period of use, the problem may be heat related. Make sure the console's ventilation slots are not blocked. Check the heatsink of the TMS9918A for sufficient heat-conducting grease as well as the TIM9904 if it has a heatsink attached. A cooler switching power supply may be installed to help further lower operating temperature.

D. If the console fails to run certain modules reliably (such as Extended BASIC) and such modules often need reseating several times, the GROM port should be changed. This typically occurs with modules that use contacts on both sides of the edge connector in the module, as is the case with Extended BASIC. Cleaning the GROM port may also help, but the problem will probably appear again shortly unless a new port is installed.

When replacing or cleaning the port, be sure to remove the felt or foam wiper from the clip-on cover to the port. The wiper may be removed with a small screwdriver and solvent. Replace the plastic cover once the wiper material has been removed. Do not use any type of lubricant since this will attract dirt and cause further problems.

E. A poor connection at the consoles I/O port on the side may also lead to occasional problems. If problems persist after checking for a secure connection, remove any device connected to the side port and use alcohol and a stiff piece of paper to clean the inner contacts on the device.

Remove the main board from the console and clean the edge connector with a pencil eraser followed by wiping with an alcohol-treated pad. Only light rubbing is needed with the eraser in order to remove surface oxidation and produce a clean surface. Excessive rubbing will not help and can remove too much plating from the board, especially if this procedure is repeated multiple times.

F. Check the power plug at the back of the console for tightness. A loose connection may cause occasional power failure. If this plug is loose, use pliers or other suitable device to bend the pins in the receptacle SLIGHTLY towards each other. A small piece of electrician's tape around the plug will also help secure the connection.

4) MODULE ERRORS

4.1 General information.

Certain modules may fail to work in the console either intermittently or at any time. Although this is usually the fault of the GROM port, modules do sometimes fail or have other reasons for not working. Before attempting any repair, however, it is important that the module in question be tried several times in the suspect console as well as another console. Be sure that other modules are also tested in the suspect console. This will help in finding the faulty component.

If the console does not *offer* TI BASIC as an option from the module selection screen and displays an "INSERT CARTRIDGE" message when no module is inserted, the console cannot find TI BASIC in GROMs 1 and 2 (chip numbers CD2156 and CD2157, respectively). Replace GROM 1 and test again. If some of the commands in BASIC fail to function, replacement of GROM 2 may also be required.

4.2 Possible causes and solutions.

A. If various modules have intermittent problems and often require reseating the modules in the GROM port, the GROM port should be replaced or cleaned. This condition is usually noticed first with modules that use contacts on both sides of their edge connectors, although it may occur with any module. Such a condition usually worsens with further use and is commonly a problem with consoles that have modules removed and inserted often.

B. If one particular module consistently fails to place a module selection of the TI menu, that module is most likely defective. This is also the case with modules that consistently lock up the computer when a specific function is attempted, indicating some portion of that module's program is damaged. If the module is a Texas Instruments produced device, the GROMs' part numbers usually indicate their GROM addresses relative to each other. A memory editing or debugging program may be useful in determining a faulty GROM by allowing viewing of the GROM contents. Since only five 8K GROM address ranges may exist in a module (at GROM addresses >6000, >8000, >A000, >C000, and >E000), the number of 8K blocks locatable by the memory editor or debugger should be equal to the number of GROM chips in the module. If less blocks are found, GROM chips should be selectively removed until the faulty chip is found. If the module also has a ROM chip in it, the ROM contents may also be examined with a memory editor. The console allows for one 8K block of CPU memory at >6000.

C. With the exception of some specialized GROM-emulating modules, nearly all non-TI produced modules contain only ROM and no GROM. This ROM is usually a single chip that may not be repaired. A few companies produced bank switched ROM modules (TI Extended BASIC also contained bank switched ROM in addition to GROM). If a ROM only module is determined to be at fault, it may contain bank switching components that have failed. Internal inspection of the module must be done to determine if repair is possible with replacement of bank switching components. Few, if any, aftermarket module manufacturers will sell replacement ROM chips. Module replacement is usually necessary.

D. If most non-TI modules will not run in the console, check the TI title screen for a 1983 copyright notice. If found, the console contains the TI operating system which prevented use of non-TI modules. This operating system may be bypassed with some software programs or aftermarket hardware adaptors. Replacement of GROM 0 in the console (chip number CD2155) with a chip from an earlier production console will replace the newer operating system and permanently solve the problem.

5) JOYSTICK PORT ERRORS

5.1 General information.

The TI-99/4A supports two eight-point joystick controllers, each with a single fire button control. Both units share a common port using a 9-pin DB9 male connector on the side of the console.

With the exception of a single driving line to each joystick, all five control lines from each joystick (UP, DOWN, LEFT, RIGHT, and FIRE) are connected together at the plug. Interference between signals is avoided by the use of diodes on each control line in each joystick unit. Adaptors for using other brand joysticks (i.e. Atari) alter the pin assignment configuration as well as supply the diodes in the adaptor unit as they are not normally used in other joysticks.

It is suggested that a joystick with a suspected problem be tested with another console as well as testing the console with a different joystick (if available) since joystick port problems may be either the fault of the console or of the joysticks. Joystick errors that are accompanied by keyboard errors are usually the fault of the console unit itself and not the joysticks.

When checking the joystick signal and driver lines for continuity, check the circuit then switch the test leads used and test again. Since diodes are used, existing continuity may fail to appear unless polarity is reversed on the testing device.

5.2 Possible problems and solutions.

A. If one position on a single joystick fails (as opposed to the same position on two simultaneously connected units), it is most likely a mechanical problem related to that unit. Disassemble the joystick and test for continuity across the contact points on the position in question. The circuit should be completed when the joystick is moved towards that position. Also check for continuity from each of the lines from the joystick to the plug. If continuity exists, replace the diode connected to the control line affected by the problem position and test again.

B. If the joystick will not operate the UP position, check the ALPHA LOCK key to make sure it is not depressed. Since the ALPHA LOCK may be sticking, go to TI BASIC and make sure that lower case can be entered as well as upper case when the ALPHA LOCK is depressed. If not, consult the section on keyboard repair. If the console has been modified with the addition of a diode to allow use of the UP position on the joysticks when the ALPHA LOCK is depressed, remove this diode and replace it with a piece of wire. In many consoles, the diode introduces a timing problem that will prevent proper function of the joystick and/or the ALPHA LOCK key. This commonly appears as an ALPHA LOCK malfunction after the console has been in use for a few minutes.

C. If one position fails on both joysticks and the ALPHA LOCK function has been tested, check for continuity from each joystick control line to the plug. If continuity exists, check each key on the keyboard for proper function since all five control lines from the joysticks are connected directly to the keyboard control lines. If keyboard failure also occurs, replace the 74LS156 in the console and test again. If problems still exist, replace the TMS9901 and test again.

D. If one joystick unit fails entirely, but the other unit (or the same unit plugged into the other port of an adaptor), check for continuity from the driver line to the joystick (pin 2 or pin 7 on the plug, depending on which joystick is in question). If continuity exists, check (or replace) the driver transistor connected to that control line in the console located near the joystick port connector. Also check the appropriate driver line for a short to ground which would indicate a faulty capacitor or transistor between the driver line and ground. If errors still exist, replace the 74LS156 in the console and test again.

E. If both joysticks fail entirely, check each joystick with the above method. If problem is still not found, replace the TMS9901 in the console and test again.

6) VIDEO OUTPUT PROBLEMS

6.1 General information.

The TI-99/4A console uses a five-pin DIN plug at the back of the unit to supply a composite video output for use with either a composite monitor or a video modulator used in conjuction with a television set. The composite video signal is created by the TMS9918A VDP chip and amplified by an external circuit located within the console.

When diagnosing video output problems, first eliminate software as a possible problem. If one particular program causes problems, it is probably not the fault of the system and no hardware repair is possible.

Since either the monitor or television used may cause many problems, connect the console in question to a known good monitor or TV. Also, connect a different console to the existing monitor or TV. This will easily determine whether it is the computer or monitoring device which is at fault. Only computer and video modulator problems are discussed here.

6.2 Possible problems and solutions.

A. If using a video modulator, testing the console with a known good composite monitor will easily determine if the modulator is at fault. If a monitor is not available, simply recheck all connections from the console to the television and try various switch combinations on the modulator unit. Due to the low cost, availability, and difficulty of repairing video modulators, problems with the modulator are best solved by replacement.

The exception is that occasionally the two leads extending from the modulator to the television will have broken connections. Simply shortening or replacing these leads may fix a variety of output problems that occur with this unit.

B. If the console has no video output with either a monitor or video modulator, then either the DIN plug receptacle, the console's amplifier circuit, the TMS9918A VDP chip, or a TMS9918A support component has failed. Each case assumes that the console is functioning and has not locked up. This may be determined by listening for the "beep" that accompanies power up or "blind typing" operations that would activate peripherals such as disk drives. If no evidence can be found that the computer is operating with the exception of video output, consult the section concerning power up problems.

To locate the faulty component, first test the monitor cable or modulator for continuity of signals lines to the DIN connector. If continuity exists, remove the main board from the console and inspect the DIN connector for bad solder joints or misplaced pins. Test again for continuity from the cable or modulator to the main board. If continuity exists, use either test leads or a soldered wire to connect pin 36 of the TMS9918A to pin 4 of the DIN receptacle (CAUTION: Be sure to connect the proper pins since voltages on certain pins of the DIN receptacle can damage the TMS9918A).

Remember to remove this jumper after testing. This procedure bypasses the external amplifier completely. A proper video image should appear if the amplifier circuit is defective, but the image will appear dimmer than normal.

If the video output appears to function, any component in the amplifier circuit may be faulty. Unless an obviously faulty component can be located, replace the two transistors (Q200 and Q201) first as they are most likely to fail.

If no video image has appeared, check pin 40 of the TMS9918A for a pulsed logic signal (this should be 10.73863 MHz) from the video oscillator circuit. If this signal is missing, first check other components in this circuit for damage then replace the crystal located next to the TMS9918A. If this signal exists, check pins 13, 14, and 15 on the TMS9918A. If signals exist here, replace the TMS9918A. If the signals are missing, trace these to their points of origin to find the break in the signal path.

C. If the video image is distorted, first check the computer with another monitor since this is usually the fault of the monitor or modulator/TV combination. If the cause is determined to be the console, bypass the external amplifier in the method described above. If the only effect is dimming of the image, remove the jumper and continue. If the distortion is corrected and a dim image appears, the fault is in the amplifier circuit. If not, remove the jumper and adjust the tuning coil near the TMS9918A VDP chip. If the problem still is not corrected, replace the TMS9918A and test again.

D. If a console only outputs monochrome (black and white) video, adjust the tuning coil located near the TMS9918A. If color does not appear anywhere within its full range of adjustment, replace the 10.73863 MHz crystal located next to it. If this has no effect, replace the TMS9918A VDP chip and test again.

E. If the TI title screen appears "jumbled" on power up and shows unwanted or oddly shaped characters, but not at other times, console GROM 0 (CD2155) may be faulty. This may also be accompanied by errors in using module software.

Replace GROM 0 and test again. If this has no effect or if screen display is disturbed with unwanted or improper characters when running assembly programs, a 4116 RAM chip in the console may be faulty. Faulty RAM is usually accompanied by an inability to run programs in console BASIC. There are eight identical RAM chips in the console, and any one may cause the described symptoms (these chips, located in a row below the TMS9918A, may have a different number than 4116, such as 8216, but are still compatible).

One of the easiest ways to find the faulty RAM is to acquire an extra 4116 dynamic RAM chip. Bend the legs together slightly so that it may be "piggy-backed" on top of an installed 4116 RAM (it may need to be held in place by hand). Put the chip on top of each 4116 chip in turn until some effect on the screen is noticed (be sure that all the pins are touching the lower chip or this method will not work). Test all the 4116 RAMs in the console. If one chip shows to have some effect on the display, check it repeatedly to ensure that the 4116 chip used for testing is making good contact. If the effect is repeatable, replace that RAM. Also test any remaining chips since multiple RAMs may be faulty. If no effect can be found when testing the RAMs in this manner, replace the TMS9918A and test again.

7) SOUND PROBLEMS

7.1 General information.

The TI-99/4A uses the TMS9919 sound processor to produce one noise and up to three tone outputs for use by the monitor or television's amplifier and speaker. This chip also has the ability to accept an external sound signal input that may be joined with its output, as in the case of the TI Speech Synthesizer. Output level from the sound chip is programmable, but this signal must still be amplified in order to drive a speaker.

Three versions of the sound processor chip were used in the TI-99/4A, each being numbered differently. The earlier consoles used chips numbered either SN76489 or SN76494. The later chips were numbered SN94624. The earlier chips required a 3.58 MHz clock signal derived from the TMS9918A VDP chip and supplied to the sound processor through a piece of hand-wired coaxial cable. In later units, the sound chip was revised to use the same 447.443 KHz clock as the GROM chips and no coaxial cable was used. The different clocks are not interchangeable.

Essentially, four problems may exist that concern the sound processor: console lock up due to chip failure (see section on power up problems), improper sound frequency, lack of sound output, and inability to accept sound input from another device.

CONSOLE DEBUGGING HELP Cont'd.

7.2 Possible problems and solutions.

A. If the sound chip produces tone frequencies excessively higher or lower than normal, first check the chip number to see if the proper chip has been installed for the clock wiring on the board. A board wired for the later clock should have continuity between pin 14 of the sound processor and pin 37 of the TMS9918A VDP chip. A board wired for an earlier clock will have continuity between pin 14 of the sound processor and pin 38 of the TMS9918A VDP chip. If this type of sound problem has developed after a period of proper operation, this check is not necessary since the chip could not have properly operated with incorrect wiring. If proper wiring exists, check pin 14 for the appropriate clock signal frequency. If the clock signal is missing, check the clock output pin of the TMS9918A VDP chip corresponding to the proper clock frequency. If missing, replace the TMS9918A chip. If the signal exists at the TMS9918A, inspect wiring between the sound processor and the TMS9918A. If the sound processor does have the proper clock on pin 14, replace the sound processor and test again.

B. If no sound output from the system exists, check the sound processor using the method described above. Check pin 6 (-CS) and pin 5 (-WE) of the sound processor with a logic probe for activity during a CALL SOUND command from BASIC. Also check or replace the 100 microfarad capacitor between pin 7 of the sound processor and pin 3 of the five-pin DIN output port on the back of the console. If no sound exists, replace the sound processor and test again.

C. If the sound processor will output proper sounds to the monitor but will not accept sound input from another device (usually the Speech Synthesizer), check for 330 ohm resistance between pin 9 of the sound processor and pin 44 of the I/O port on the side of the console. If this circuit is open, replace the resistor between these two points. If this has no effect, check for a short to ground from pin 4 of the cassette interface port. If this is shorted, replace the .001 microfarad capacitor between this line and ground. Also inspect transistor Q400 (connected through two resistors to pin 4 of the cassette port) as well as the capacitor across it for shorting. Replace if necessary.

≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈

ARCHIVING—A HEADACHE?

SPIRIT OF 99

By: Andy Frueh, Lima UG

A lot of people are puzzled by archiving and how to use Barry Boone's Archiver. What follows is both a reference guide and explanation of Archiver III. It is not meant to totally replace the documentation for this program. Actually, I haven't seen a distribution copy that comes with a set of instructions. There may be hidden features of ArcIII that aren't obvious to me (for example, Disk Utilities by John Birdwell has a feature to figure decimal-to-hex conversions).

What exactly is archiving? Putting it simply, when you archive you take a file or a set of files, and group them as one file then compress them so they take up less disk space. Some software comes archived. These ALMOST always include the archiving program. Examples are Jack Sughrue's PLUS! and the Complete Adventure disk set.

What is the purpose of archiving? Well it started out as a money saver for modem users. It is faster, and thus cheaper, to send 90 archived sectors as 1 file, than 120 sectors for 3 programs. Now it is also a means of backing up disks. You can save each of your disks as a one file, squashed archive. You can specify whether you want compressed files or not. The reason you have a choice is that some unusual files actually take up more space when they are compressed. Another useful application of archiving is when you have programs you want to keep, but don't need ready to use. You can keep archives of all these files instead of taking up disk space.

OK, now that you have the "what", here's the "how". As far as I know, the only archiver is Barry Boone's program. Its operation is completely different from Archiver II. Rather than add new features to past versions, Archiver was completely re-written. It usually contains an XB LOAD program, but may be loaded from E/A. The program's filename is usually ARC1. It can be found on almost all of the bulletin boards, as a commercial version with Geneve utilities, in user group libraries, with other Fairware programs or from the author. Chances are, you can definitely get a copy.

t things first, so get the program loaded. After that, y
e notice. Press any key to pass this. You then see a menu. Each
ion is described in detail below.

Archive Files - These options are largely self-explanatory. As you may
ssed, this option archives files. Pressing one will deliver a set of
These are "Source Drive (1-Z)". Yes, you can have drive numbered
and A-Z. Then comes, "Output Drive (1-Z)". You may use one drive.
will prompt you to change disks when needed. It is highly recommended
use a blank output disk, since archives may fill or almost fill a
Next comes "Output Filename". This is usually the name of the disk you
hiving, or some related heading. For example, a set of D/V 80 articles
named "ARTICLES". The following prompt is "Pack all Files? (Y/N)". If
wer "Y" then all the files on the source disk are archived. If you
"N", then when Archiver is working, you are asked "Include filename?
If you answer "Y" then that file is archived, otherwise it is ignored.
a handy feature if you have programs and files for example, and need
perated. This process repeats for each of the files on the source disk.
al prompt is "Compress? (Y/N)". Saying "Y" and Archiver attempts to
each file so it takes up less space. Remember that some unusual file
will actually get LARGER if compression is attempted. When all the
s are answered, press REDO to correct an error in your answers, BACK to
to the menu, or any other key to continue. When Archiver is done
ming any operation, pressing a key goes back to the main menu.

2) Extract Files - This is the opposite of archiving. It will let you
(extract) files from an ARC file. You are first asked for the source
. Next you input the source filename. After that, you are asked for the
t drive. It must be stressed that the output drive for ALL operations of
ver should be different than the input drive. You may run out of spave or
rite a file accidently. Output disks should be blank.

The next prompt asks, "Extract all files?" If you answer "Y" then every
stored in the ARC file will be taken out. If you answer "N" then when
cting starts, the program asks, "Include filename?" for every separate
in the archive. Again, press REDO (to restart this option), BACK (returns
in menu), or any other key to continue.

3) Catalog Disk - This is fairly self explanatory. Simply input the
e drive name. The program will ask if you want a printout. If you answer
then you are asked for the printer name. If there are more files than can
isplayed, then [more] is printed on the screen and pressing a key advances
screen.

4) Catalog ARC File - If you aren't sure what files are contained in an
ive file, than this option tells you. You are asked for the source drive,
ce filename, and whether or not you want a printout of the list of files.

5) File Copy - This option will copy a file (obviously). Simply supply
source drive and filename, and the output drive and filename.

6) File Rename - Again, this option should explain itself. Give the
rce drive and filename, then the output filename.

7) File Delete - Supply the source drive and filename.

8) File Un/Protect - You first supply the source drive and filename. You
then asked "Protect?" If you answer "Y" the file is protected. Otherwise,
e protection is lifted.

9) List Text File - This will display or print a D/V 80 file. Give the
urce drive and filename. You are then asked if you want the file printed or
t.

10) Load FW - This returns to Funnelweb. Simply give the drive number on
ich the UTIL1 file is located.

NOTE: When an I/O error occurs, pressing a key returns to the main menu.
you have a Geneve, this is for you. Using a sector editor, find the string
E08C00 and replace it with D8018C00.

I think that this should get people on the road to understanding archiver.
emember that it is fairware, so if you find it very useful, send the author
Barry Boone) a donation.

This article/item comes from the January 1991 issue of BITS, BYTES PIXELS
Charles Good, editor), the newsletter of the Lima OH 99/4A User Group, P.O.
ox 647, Venedocia, OH 45894.]

System Commands

System	PET	Apple II	TRS-80	Atari	TI 99/4	Sorcerer	ZX80
AUTO mm, n		*	*		Number		
BREAK mm					*		BREAK
CLEAR	CLR	*	*	*		*	*
CLEAR n			*				
CLOAD	LOAD	LOAD	*	*	OLD	*	LOAD
CLOAD?	VERIFY		*				
CONTINUE	CONT	CONT	CONT	CONT	*	CONT	CONT
CSAVE	SAVE	SAVE	*	*	SAVE	*	SAVE
DELETE mm		DEL	*		*		
EDIT mm	cursor	cursor	*	cursor	cursor		cursor
HOME		*					
HIMEM		*					
LIST mm-nn	*	*	*	*	*	*	*
LOMEM		*					
MAN		*					
NEW	*	*	*	*	*	*	*
RESEQUENCE mm, nn					*		*
RUN mm	*	*	*	*	*	*	
SYSTEM	SYS	CALL - 151	*	BYE	BYE	BYE	
TROFF		NOTRACE	*		UNTRACE		
TRON		TRACE	*		TRACE		
UNBREAK					*		
(Screen Format)	40 by 24	40 by 24	64 by 16	40 by 24	32 by 24	64 by 30	32 by 24
(Character Resolution, m by n)		7 by 8	2 by 3	8 by 8	8 by 8	8 by 8	
(Total pixels)	128 full-screen	280 by 192	128 by 48	320 by 192	256 by 192	512 by 240	

String Functions

System String Functions	PET	Apple II	TRS-80	Atari	TI 99/4	Sorcerer	ZX80
ASC (string)	*	*	*	*	*	*	
CHR$ (code)	*	*	*	*	*	*	*
CODE (string)							*
FRE (X$)	*		*	*		*	
INKEY$	GET	GET	*		CALL KEY		
LEFT$ (string,n)	*	*	*			*	
LEN (string)	*	*	*	*	*	*	
MID$ (string, p,n)	*	*	*		SEG$	*	
POS (str 1, str2, n)					*		
RIGHT$ (string,n)	*	*	*			*	
STR$ (expr)	*	*	*	*	*	*	*
STRING$ (n,char)			*				
TLS (string)							*
VAL (string)	*	*	*	*	*	*	
VARPTR var			*	ADR			

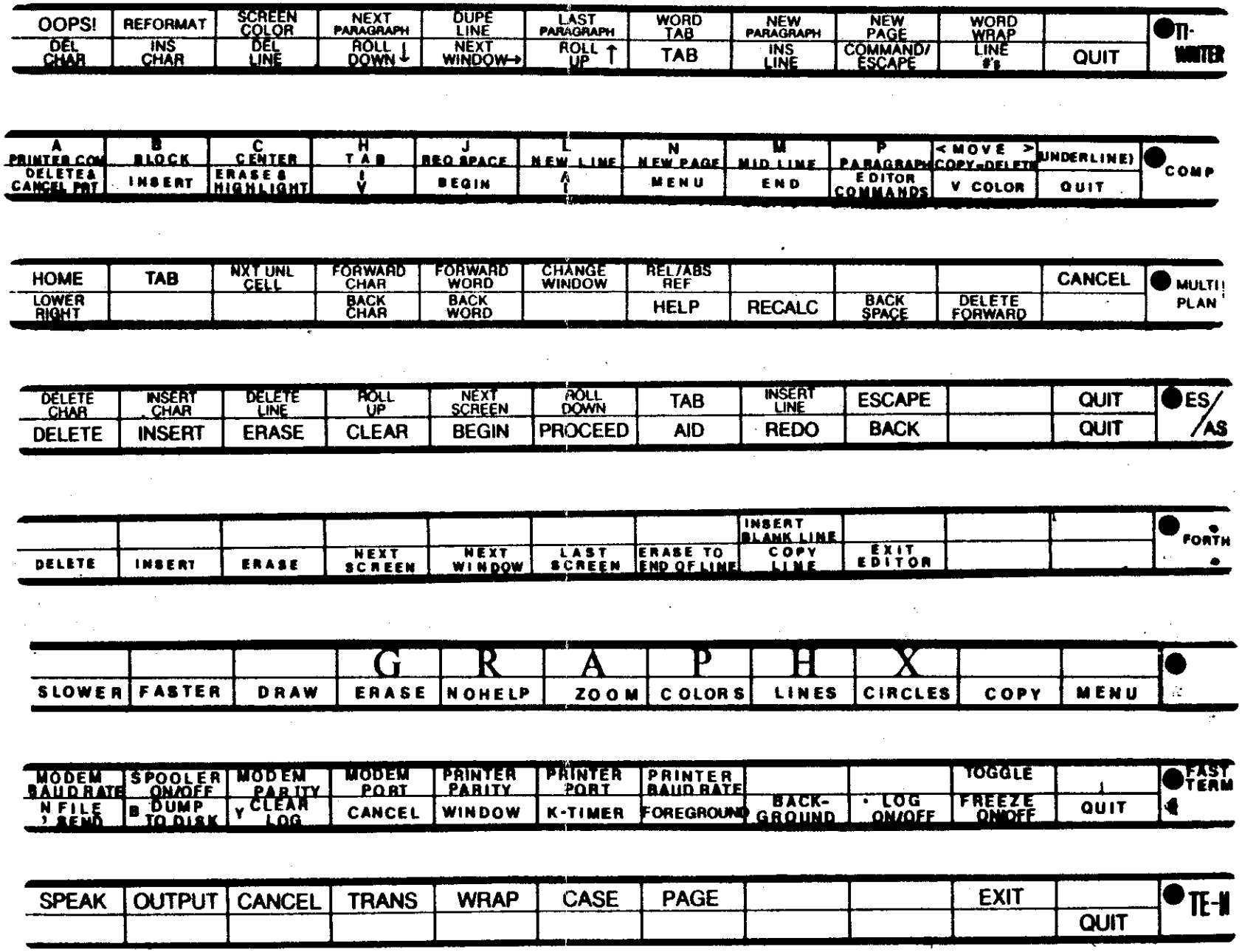
BASIC Math and other Functions

BASIC FUNCTIONS	PET	Apple II	TRS-80	Atari	TI 99/4	Sorcerer	
(Precision)	9	10	6 or 16	10	14	6	
ABS (expr)	*	*	*	*	*	*	*
ATN (expr)	*	*	*	*	*	*	
CINT (expr)			*				
CDBL (expr)			*				
CLOG (expr)		*		*			
CSNG (expr)			*				
COS (expr)	*	*	*	*	*	*	
ERL (expr)			*				
ERR (expr)			*				
EXP (expr)	*	*	*	*	*	*	
FIX (expr)			*				
FRE (expr)	*		* (also MEM)	*		*	
INT (expr)	*	*	*	*	*	*	
LOG (expr)	*	*	*	*	*	*	
MOD (expr)		*					
POS (expr)	*	*	*			*	
RANDOMIZE	RANDOM		RANDOM		*		*
RND (0)	*	RND	*	*	RND(1)	*	*
RND (expr)		*	*				
SCRN (x,y)		*					
SGN (expr)	*	*	*	*	*	*	
SIN (expr)	*	*	*	*	*	*	
SPC (expr)	*						
SPC (num)		*		NULL		*	
SQR (expr)	*	*	*	*	*	*	
TAN (expr)	*	*	*		*	*	
TI (expr)	*						
USR (X)	*	*	*			*	*
AND, OR, NOT	*	*	*			*	*

BASIC Statements

BASIC Statements	PET	Apple II	TRS-80	Atari	TI 99/4	Sorcerer	ZX80
General Statements							
APPEND			•		CALL CLEAR		•
CLS		•				EN	
CALL address		•			•		
CALL CHAR				COLOR	•		
CALL COLOR				STICK	•		
CALL JOYSTK		HCOLOR=		SETCOLOR	•		
CALL SCREEN				SOUND	•		
CALL SOUND							
CLOSE	•					•	
COLOR = n	•	•	•	•	DEF		
DATA	•	•					
DEF FN (name)			•				
DEFINT			•				•
DEFDBL			•				
DEFSNG			•				
DEFSTR	•	•		•	•		
DIM var(k)				•			
DISPLAY		HPLOT				•	
DRAWTO		•		•	•		
DSP var	•		•			•	•
END			•	•	•		
EOF						•	•
ERROR (mm)		•		•	•		
FOR . . . TO . . . STEP, NEXT	•		•			•	•
General Statements		•		•			
GOSUB linenum, RETURN	•	•	•				
GOTO linenum	•	•		•			
GR					CALL HCHAR		•
GRAPHICS				•	•		
HLIN . . . AT		•	•		e	•	
IF expr THEN linenum	•		•		e		
IF expr THEN . . . ELSE		•	•			•	
IF expr GOSUB . . . RETURN	•	•	•			•	•
IF expr GOTO	•		•	•	•	•	•
IN (port)		IN # expr	•			•	
INPUT "msg", var	•	•	•	•	•		
INPUT # n, var	•	RECALL	•				•
LET var = expr							
LPRINT "msg" or LPRINT var							
NEXT var				TRAP	e	•	
ON ERROR GOT linenum		ONERR	•	•	•		
ON expr GOSUB, RETURN	•	•	•		•		
ON expr GOTO linenum	•				•		
OPEN							
OPTION BASE (x)		PR # expr	•	•			
OUT portnum, val		POL		•	CALL GCHAR	•	•
PADDLE		•	•				
PEEK	•			•	(e) Call Load	•	
POINT		•		•			•
POP		•	•	•	(e) Display At	•	
POKE locn, val	•	•	•				
PRINT "msg" or PRINT var	•	•	•	POSITION		•	
PRINT@		•	•		e		
PRINT #	•			•	•	•	
PRINTUSING				•	•		
PTRIG				•	•	•	•
READ var, var . . .	•	•	•	•	•		
RECALL		•	•			•	
REM	•	•	•	•	•		
RESET (x,y)		•					
RESTORE	•	•	•	•			
RESUME linenum		PLOT, HPLOT	•	PLOT		•	•
SET (x,y)		•	•	•	•		
SPEED = expr		•	•			•	
STOP	•	•	•	•	•		
STORE		•	•				
TAB	•	•					
TEXT					CALL VCHAR		
UPDATE		•				•	
VLIN . . . AT		•					
VTAB (x)		•					
WAIT A,B,C	•	•					

This Basic language reference chart may be of some use in transposing programs from one computer's Basic to another's. Stars indicate the existance of the command listed at the left for a particular computer. A small (e) in the 99/4A column indicates the existance of a command in Extended Basic.


OOPS! | REFORMAT | SCREEN COLOR | NEXT PARAGRAPH | DUPE LINE | LAST PARAGRAPH | WORD TAB | NEW PARAGRAPH | NEW PAGE | WORD WRAP | | TI-WRITER
DEL CHAR | INS CHAR | DEL LINE | ROLL DOWN ↓ | NEXT WINDOW→ | ROLL UP ↑ | TAB | INS LINE | COMMAND/ ESCAPE | LINE #'s | QUIT
A PRINTER CON | B BLOCK | C CENTER | H TAB | J REQ SPACE | L NEW LINE | N NEW PAGE | M MID LINE | P PARAGRAPH | < MOVE > COPY DELETE | UNDERLINE | COMP
DELETE& CANCEL PRT | INSERT | ERASE & HIGHLIGHT | | BEGIN | | MENU | END | EDITOR COMMANDS | V COLOR | QUIT
HOME | TAB | NXT UNL CELL | FORWARD CHAR | FORWARD WORD | CHANGE WINDOW | REL/ABS REF | | | | CANCEL | MULTI PLAN
LOWER RIGHT | | | BACK CHAR | BACK WORD | | HELP | RECALC | BACK SPACE | DELETE FORWARD | | |
DELETE CHAR | INSERT CHAR | DELETE LINE | ROLL UP | NEXT SCREEN | ROLL DOWN | TAB | INSERT LINE | ESCAPE | | QUIT | ES/AS
DELETE | INSERT | ERASE | CLEAR | BEGIN | PROCEED | AID | REDO | BACK | | QUIT
DELETE | INSERT | ERASE | NEXT SCREEN | NEXT WINDOW | LAST SCREEN | ERASE TO END OF LINE | INSERT BLANK LINE / COPY LINE | EXIT EDITOR | | | FORTH
SLOWER | FASTER | DRAW | G ERASE | R NOHELP | A ZOOM | P COLORS | H LINES | X CIRCLES | COPY | MENU
MODEM BAUD RATE | SPOOLER ON/OFF | MODEM PARITY | MODEM PORT | PRINTER PARITY | PRINTER PORT | PRINTER BAUD RATE | | | TOGGLE | | FAST TERM
N FILE / SEND | B DUMP TO DISK | Y CLEAR LOG | CANCEL | WINDOW | K-TIMER | FOREGROUND | BACK-GROUND | LOG ON/OFF | FREEZE ON/OFF | QUIT
SPEAK | OUTPUT | CANCEL | TRANS | WRAP | CASE | PAGE | | | EXIT | | TE-II
| | | | | | | | | | QUIT