

NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 12, Number 8

September, 1993

Registered by Australia Post - Publication No. NBH5933



NEW

SECTION for IBM COMPATIBLES

Sydney, New South Wales, Australia

\$3

TisHUG (Australia) Ltd.
A.C.N. 003 374 383

TisHUG News Digest

All correspondence to:
C/o 3 Storey St.
Ryde 2112 Australia

TisHUG News Digest

ISSN 0819-1984

I N D E X

The Board

Co-ordinator
Dick Warburton (02) 918 8132
Secretary
Russell Welham (043) 92 4000
Treasurer
Cyril Bohlsen (02) 639 5847
Directors
Percy Harrison (02) 808 3181
Peter Schubert (02) 318 1191

Sub-committees

News Digest Editor
Bob Relyea (046) 57 1253
BBS Sysop
Ross Mudie (02) 456 2122
BBS telephone number (02) 456 4606
Merchandising
Percy Harrison (02) 808 3181
Publications Library
Russell Welham (043) 92 4000
Software Library
Larry Saunders (02) 644 7377
Technical Co-ordinator
Geoff Trott (042) 29 6629

Regional Group Contacts

Central Coast
Russell Welham (043) 92 4000
Coffs Harbour
Kevin Cox (066) 53 2649
Glebe
Mike Slattery (02) 892 8182
Hunter Valley
Geoff Phillips (049) 42 8176
Illawarra
Geoff Trott (042) 29 6629
Liverpool
Larry Saunders (02) 644 7377
Northern Suburbs
Dennis Norman (02) 452 3920
Sutherland
Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00
Associate membership \$10.00
Overseas Airmail Dues A\$65.00
Overseas Surface Dues A\$50.00

TisHUG Sydney Meeting

The September Meeting will start at
2.00 pm on the 4TH September 1993
at Ryde Infants School,
Tucker Street, Ryde.

Printed by
Kwik Kopy Parramatta

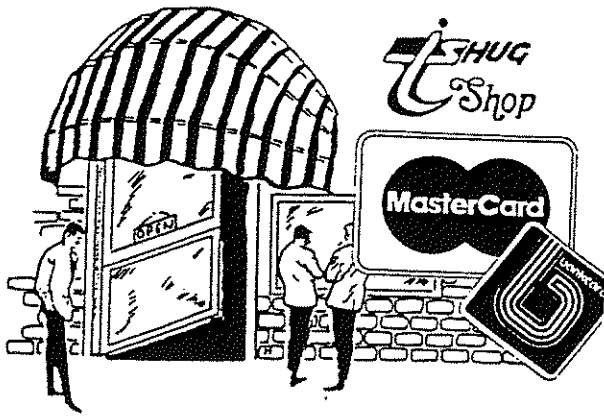
Title	Description	Author	Page No.
AIDKEY	HINTS	ATTO HEINO	20
BULL RUN	PROGRAMME	STEPHEN SHAW	19
CHAOS & GRAPHICS	PROGRAMME	JOE JACOBSON	16
CO-ORDINATORS REPORT	CLUB NEWS	DICK WARBURTON	11
CONVERTING FROM IBM TO TI-99/4A	GEN. INT.	LARRY SAUNDERS	5
DESIGN FOR FUN	PROGRAMME	STEPHEN SHAW	3
EDITORS COMMENTS	CLUB NEWS	BOB RELYES	10
EXTENDED BASIC III	HINTS		15
FROM THE BULLITIN BOARD	REQUEST	CLAUDE	10
FORTH & FILES	GEN. INT.	GEOFF TROTT	16
IBM SECTION			22
LEARNING TO KNOW YOUR TI No.8	PROGRAMMING	PERCY HARRISON	7
LETTER HEAD DESIGN	PROGRAMME	KEN BURGESS	10
LETTER FROM AUSTRALLIA	GENERAL INT.	TONY MCGOVERN	14
MINI-TOWER PANDA BOX EXPANSION POSSIBILITIES	GEN. INT.	C. PEDIGO	14
NOSTALGIA TIME	GEN. INT.	GEOFF TROTT	13
REGIONAL GROUP REPORTS	GEN. INT.		23
TisHUG SHOP	CLUB NEWS	PERCY HARRISON	2
TisHUG SOFTWARE	CLUB NEWS	LARRY SAUNDERS	5
TREASURER'S REPORT	CLUB NEWS	CYRIL BOHLSSEN	9
VINCENT'S CORNER	HINTS	VINCENT MAKER	8

IN MEMORIAM

It is with considerable sadness that we report the death of one of our dedicated 99/4A users, WILLIAM (BILL) WALKER, who passed away last month.

Bill was a keen TI enthusiast who was always willing and eager to help our other TI users in his area.

Our sincere sympathy and condolences are extended to his wife, family and friends, on this very sad occasion.



TISHUG SHOP

with Percy Harrison.

Just as I expected, there were no surprises for me when I browsed through the August TND. You guessed it! There were no new subscribers to the articles submitted for publication. It was however pleasing to see that Robert Brown is back on board again with his articles on TI Games and we must apologise to him for omitting to list his article in the index on the inside front cover. For those members who are interested in games it is on pages 16 to 18 of the August TND and is well worth reading.

At the August meeting we spoke to the members present on a proposal to move our meeting venue from the Top Ryde Infant's School to Meadowbank Primary school situated on the corner of Thistle St and Belmore St Meadowbank which is about 1.5 kilometres south of our present location and readily accessible by bus from Strathfield and Meadowbank stations or about 10 to 15 minutes walk from Meadowbank station depending on how fast one walks.

The advantages this new site offers are:

1. The tables are higher and provide more comfort to those members who bring their systems along to do demonstrations etc.
2. With the introduction of IBM compatible users to the group it will enable us to separate the TI enthusiasts from the IBM compatible users as we will have access to two separate buildings each of which have ample room for the requirements of each group. The real advantage of this is that the TI er's will not have to shout over the IBM er's to be heard and vice versa as they will each have their own domain.
3. The shop will be set-up in the Staffroom away from the other two locations so that when you take time out for a cuppa tea and bickies you will be able to do your shopping and replenish the clubs coffers (we really need your monetary support to keep our head above water).

4. Both rooms that we will be using for setting up computers have an ample supply of power outlets and will handle the current load of the four or so TI systems and the nine IBM compatibles that we anticipate would be set up at any one time.
5. There is plenty of parking in the school grounds close to each room that we will be using. Entry to the parking area is from Gale St located on the western side of the school.

Please take note, we are not trying to set-up two separate factions within the club, both types of users will be free to go to either group or spend part of their time at one group then the balance at the other. Our aim is to satisfy the requirements of both system user's and to encourage an interchange of ideas among all of our members so as to build up our expertise in both areas and establish our club as being the best in this country for both the TI er's and the IBM compatible users and any suggestions from our members that will help us achieve and maintain these goals would be most welcome.

It was initially envisaged that we would make this change from the first meeting in 1994 as that is when our subscription to the present location expires but following the club meeting I was approached by a couple of long standing members and asked if we could make the change earlier so we now plan to move to Thistle St in November. Please make a note of this in your diary so that you remember not to go to the Top Ryde venue on the 6th November as we will meet at Meadowbank school on that day.

This month I am not including a list of what is available from the shop as these have been covered in the May, July and August issues. I would like to draw to the attention of IBM compatible user's that we are now stocking good quality 1.44 Meg 3.5 preformatted HD disks (Apple) made in USA and they are available at \$14.00 for ten. These are not the cheapies that are being sold elsewhere for \$15 per box and which do not all format to the full 1.44 Meg, I know because I purchased two boxes of the cheapies about three months ago and had seven out of the 20 that would not format to the full 1.44Meg so I took them back and finished up paying extra for preformatted disks. I can highly recommend the above Apple disks to our IBM compatible user's. Unfortunately they are not suitable for use on the TI.

For those TI er's who are interested in foreign languages I have just received the following disks which are available at \$2.00 per disk:

Greek (1 disk)
 Vietnamese (1 disk)
 Czechoslovakian (2 disks)
 German (1 disk)
 French (1 disk)
 Russian (1 disk)
 Japanese (2 disks)
 Spanish (1 disk)

Hopefully, Larry will assign club numbers to these and list them either in this issue or the next but in the meantime if you want any of these language disks please let me know. If you are ordering them by mail please remember to include sufficient for return postage.

END OF ARTICLE

The characters are numbered from left to right first then from top to bottom, the number pattern below indicates how the 5x5 grid is numbered!

DESIGNS for FUN

Originally written for TI Basic by Peter Brookes and published in TIDINGS in June 1982. Rewritten for Extended Basic plus The Missing Link by Stephen Shaw 1990.

Main Menu:

0. Set Defaults... resets all parameters to the original default values.
1. Check Settings.. displays current parameter values
2. Select Pattern Area.. each block of pattern may be set to an area of 1, 4, 9, 16 or 25 characters from 8x8 pixels to 40x40 pixels.
3. Select Colours.. Only operational if The Missing link is loaded and set up for 16 colour use Sets screen, foreground and background colours.
4. Select Bias.. Allows selection from 16 predefined biases—for more use option 5. See description below.. (Default 01234789CEF)
5. Create Bias.. Allows input of a bias string from 2 to 112 characters, composed of the numbers 0 to 9 and the letters A to F.

The program is designed so that the selection of combinations at random is made from the bias string. If the string contains the entire range of combinations (0 to F) the program will randomly select any of the 16 possible combinations of 4 pixels. If, however, the bias string contained only 1248 the program would only be able to select from those patterns, thus biasing the result.

6. Reset Randomiser.. It is the nature of the beast that it may become cyclic—probably with a large cycle! Using this option will break the cycle and shift the pattern generation to another range.
7. Select Mode.. either continuous or pattern will stay on screen until you press a key.
8. Continue.. get the patterns going.

During pattern generation:

Press and hold key S to return to main menu.
User prompted pattern change: any key except S & A for next pattern.

To print: Hold CTRL and FCTN together.

To save in TI Artist format:

Once the pattern is on screen and before the definitions at screen bottom start to change, press and hold key A. Indicate drive number, then file name (up to 8 characters, omit _P). Easier to use if user prompt mode is specified.

Patterns saved in TI Artist format can then be cut out into Instances and used for texture or repeated over the full screen as required before printing. Can be used for book jackets or fly leaves or even wallpaper! Enlarge and use as a crosswork design. etc etc.

As listed the equivalent character definitions are given at screen bottom— if you would like to save your TI Artist files without these then amend the following line:

```
2730 FOR @=1 TO 24 :: etc etc
```

Or to view a full screen pattern after the new "characters" have been displayed, make the following amendment:

```
2630 FOR L=1 TO 24
```

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Here is the program listing:

```
100 REM DESIGNS FOR FUN (C) PETE BROOKS 1982 V1.0
110 REM TO TML S SHAW 1990
120 CALL LINK("CLEAR")
130 CALL LINK("CHSIZE",8,8)
140 OPTION BASE 1
150 DIM C$(24),D$(40),M$(2)
160 H$="0123456789ABCDEF"
170 I$="084C2A6E195D3B7F"
180 L$="54433322"
190 M$(1)="CONTINUOUS"
200 M$(2)="USER-CUED"
210 DEF R(X)=INT(RND*X)+1
220 FOR L=96 TO 127
230 CALL LINK("CHAR",L,"00013B3323033F7F")
240 NEXT L
250 GOTO 2270
260 FOR L=1 TO Z*8
270 D$(L)=" "
280 NEXT L
290 S=LEN(S$)
300 FOR L=1 TO Z*4
310 FOR M=1 TO Z
320 A$=SEG$(S$,R(S),1)
330 D$(L)=A$&D$(L)&SEG$(I$,POS(H$,A$,1),1)
340 CALL KEY(1,K,T)
350 IF K=2 THEN 550
360 IF K=1 THEN GOSUB 2670
370 NEXT M
380 D$(Z*8-L+1)=D$(L)
390 NEXT L
400 Q=96
410 FOR L=1 TO Z*8 STEP 8
420 FOR M=1 TO 2*Z-1 STEP 2
430 A$=" "
440 FOR N=L TO L+7
450 A$=A$&SEG$(D$(N),M,2)
460 CALL KEY(1,K,T)
470 IF K=2 THEN 550
480 NEXT N
490 CALL LINK("CHAR",Q,A$)
500 GOSUB 2540
510 Q=Q+1
520 NEXT M
530 NEXT L
540 GOSUB 2610
550 RETURN
560 FOR L=1 TO 24
570 C$(L)=" "
580 NEXT L
590 FOR L=1 TO Z
600 FOR M=1 TO Z
610 C$(L)=C$(L)&CHR$(95-Z+1*Z+M)
620 NEXT M
630 NEXT L
640 FOR L=1 TO Z
650 FOR M=1 TO VAL(SEG$(L$,Z,1))
660 C$(L)=C$(L)&C$(L)
670 NEXT M
680 C$(L)=SEG$(C$(L),1,28)
690 NEXT L
700 FOR L=Z+1 TO 24
710 C$(L)=C$(L-Z)
720 NEXT L
730 RETURN
740 FOR L=1 TO 10
750 CALL LINK("PRINT",L*8-7,1,C$(L))
760 NEXT L
```

```

770 RETURN
780 Z=3
790 GOSUB 560
800 A=1
810 S$="01234789CEF"
820 C=15
830 B=15
840 F=2
850 CALL SCREEN(C)
860 REM
870 CALL LINK("COLOR",F,B)
880 REM
890 CALL SOUND(100,1220,0)
900 RETURN
910 CALL SOUND(100,1220,0)
920 CALL KEY(0,K,T)
930 IF T<1 THEN 920
940 P=POS(H$,CHR$(K),1)
950 IF P=0 THEN 920
960 CALL SOUND(100,1220,0)
970 RETURN
980 CALL LINK("CLEAR")
990 CALL LINK("PRINT",2,20,"CURRENT SETTINGS")
1000 RESTORE 2480
1010 FOR I=1 TO C
1020 READ B$
1030 NEXT I
1040 CALL LINK("PRINT",12,20,"SCREEN IS "&B$)
1050 RESTORE 2480
1060 FOR I=1 TO B
1070 READ B$
1080 NEXT I
1090 CALL LINK("PRINT",22,20,"BACKGROUND IS "&B$)
1100 RESTORE 2480
1110 FOR I=1 TO F
1120 READ B$
1130 NEXT I
1140 CALL LINK("PRINT",32,20,"FOREGROUND IS "&B$)
1150 CALL LINK("PRINT",42,20,"BIAS IS "&S$):: CALL
LINK("PRINT",52,20,"PATTERN REA IS "&STR$(Z*2))
1160 CALL LINK("PRINT",62,20,"MODE IS "&M$(A))
1170 CALL LINK("PRINT",72,20,"PRESS 'S' TO CONTINUE")
1180 CALL KEY(1,K,T)
1190 IF K<>2 THEN 1180
1200 CALL SOUND(100,1220,0)
1210 RETURN
1220 CALL LINK("CLEAR")
1230 CALL LINK("PRINT",20,20,"PATTERN AREA"):: CALL
LINK("PRINT",30,20,"PRESS: OR:")
1240 FOR I=1 TO 5
1250 CALL LINK("PRINT",30+I*9,20,STR$(I)&" "&STR$(
1+I)&"X"&" "&STR$(I)&" "&STR$(1)&"")
1260 NEXT I
1270 CALL LINK("PRINT",120,20,"YOUR SELECTION ?:")
1280 CALL SOUND(100,1220,0)
1290 CALL KEY(0,K,T)
1300 IF (K<49)+(K>53)THEN 1290
1310 CALL SOUND(100,1220,0)
1320 Z=K-48
1330 CALL LINK("PRINT",140,90,Z)
1340 GOSUB 560
1350 RETURN
1360 CALL LINK("CLEAR")
1370 RESTORE 2480
1380 CALL LINK("PRINT",12,20,"SELECT COLOURS")
1390 CALL LINK("PRINT",22,20,"PRESS: FOR:")
1400 FOR I=1 TO 16
1410 READ B$
1420 CALL LINK("PRINT",1*8+26,23,SEG$(H$,I,1)&" "&B$)
1430 NEXT I
1440 CALL LINK("PRINT",160,40,"SCREEN COLOUR ?:")
1450 GOSUB 910
1460 C=P
1470 CALL LINK("PRINT",160,220,SEG$(H$,C,1)):: CALL
LINK("PRINT",169,20,"BACKGROUND COLOUR ?:")
1480 GOSUB 910
1490 B=P
1500 CALL LINK("PRINT",169,220,SEG$(H$,B,1)):: CALL
LINK("PRINT",178,20,"DIFFERENT FOREGROUND
COLOUR ?:")
1510 GOSUB 910
1520 IF B=P THEN 1510

```

```

1530 F=P
1540 CALL LINK("PRINT",178,220,SEG$(H$,F,1))
1550 FOR I=1 TO 200
1560 NEXT I
1570 RETURN
1580 CALL LINK("CLEAR")
1590 RESTORE 2520
1600 CALL LINK("PRINT",10,20,"BIAS SELECTION"):: CALL
LINK("PRINT",20,20,"PRESS: FOR:")
1610 FOR I=1 TO 16
1620 READ S$
1630 CALL LINK("PRINT",1*8+22,30,SEG$(H$,I,1)&" "&S$)
1640 NEXT I
1650 CALL LINK("PRINT",170,20,"YOUR SELECTION ?:")
1660 GOSUB 910
1670 CALL LINK("PRINT",170,220,SEG$(H$,P,1))
1680 FOR I=1 TO 200
1690 NEXT I
1700 RETURN
1710 CALL LINK("CLEAR")
1720 CALL LINK("PRINT",10,10,"BIAS CREATION")
1730 REM
1740 CALL LINK("PRINT",22,20,"USE 0 TO 9 AND A TO F")
1750 REM
1760 CALL LINK("PRINT",32,20,"INPUT BETWEEN 2 & 112
DIGITS LONG")
1770 CALL LINK("PRINT",62,20,"PLEASE ENTER YOUR
STRING :- ")
1780 CALL LINK("INPUT",70,20,S$,112)
1790 CALL LINK("CLEAR")
1800 CALL LINK("PRINT",100,20,"CHECKING...")
1810 IF LEN(S$)>1 THEN 1850
1820 CALL LINK("PRINT",109,20,"YOU'VE MADE A MISTAKE")
1830 CALL LINK("PRINT",119,1,"YOU'LL HAVE TO INCREASE
THE LENGTH OF YOUR STRING")
1840 GOTO 1760
1850 FOR I=1 TO LEN(S$)
1860 A$=SEG$(S$,I,1)
1870 IF A$<>" " THEN 1900
1880 CALL LINK("PRINT",109,20,"PLEASE DO NOT INCLUDE
SPACES")
1890 GOTO 1750
1900 IF (A$<"0")+(A$>"9")*(A$<"A")+(A$>"F")THEN 1750
1910 NEXT I
1920 CALL LINK("PRINT",150,50,"STRING CHECKED")
1930 FOR I=1 TO 200
1940 NEXT I
1950 RETURN
1960 CALL LINK("CLEAR")
1970 RANDOMIZE
1980 CALL SOUND(100,1220,0)
1990 RETURN
2000 CALL LINK("CLEAR")
2010 CALL LINK("PRINT",10,20,"SELECT MODE"):: CALL LINK
("PRINT",20,20,"PRESS: OR:")
2020 FOR I=1 TO 2
2030 CALL LINK("PRINT",40+I*10,40,STR$(I)&" "&M$(I))
2040 NEXT I
2050 CALL LINK("PRINT",80,30,"YOUR SELECTION ?:")
2060 CALL KEY(0,K,T)
2070 IF (K<49)+(K>50)THEN 2060
2080 A=K-48
2090 CALL LINK("PRINT",90,120,A)
2100 CALL SOUND(100,1220,0)
2110 FOR I=1 TO 200
2120 NEXT I
2130 RETURN
2140 CALL LINK("CLEAR")
2150 GOSUB 850
2160 GOSUB 740
2170 GOSUB 260
2180 CALL KEY(1,K,T)
2190 IF K=2 THEN 2260
2200 IF K=1 THEN GOSUB 2670
2210 ON A GOTO 2170,2220
2220 CALL KEY(1,K,T)
2230 IF T<1 THEN 2220
2240 CALL SOUND(100,1220,0)
2250 GOTO 2170
2260 RETURN
2270 GOSUB 780
2280 GOSUB 560
2290 CALL LINK("CLEAR")

```

```

2300 CALL SCREEN(15)
2310 RESTORE 2500
2320 CALL LINK("PRINT",10,20,"MAIN MENU"):: CALL LINK
("PRINT",20,20,"PRSS: TO: ")
2330 FOR I=0 TO 8
2340 READ B$
2350 CALL LINK("PRINT",1*9+28,30,STR$(I)&" "&B$
2360 NEXT I
2370 CALL LINK("PRINT",168,30,"YOUR SELECTION ?:")
2380 CALL SOUND(100,1220,0)
2390 CALL KEY(O,K,T)
2400 IF (K<48)+(K>56)THEN 2390
2410 K=K-47
2420 CALL LINK("PRINT",176,140,K-1)
2430 CALL SOUND(100,1220,0)
2440 FOR I=1 TO 200
2450 NEXT I
2460 ON K GOSUB 780,980,1220,1360,1580,1710,1960,2000,
2140
2470 GOTO 2290
2480 DATA TRANSPARENT,BLACK,MEDIUM GREEN,LIGHT GREEN,
DARK BLUE,LIGHT BLUE,DARK ED,CYAN,MEDIUM RED
2490 DATA LIGHT RED,DARK YELLOW,LIGHT YELLOW,DARK GREEN,
MAGENTA,GREY,WHITE
2500 DATA SET DEFAULTS,CHECK SETTINGS,SELECT PATTERN
AREA,SELECT COLOURS,SELECT BIAS,CREATE BIAS
2510 DATA RESET RANDOMISER,SELECT MODE,CONTINUE
2520 DATA F2481F4812F8124F1248,02468ACE13579BDF,FFEECCOO
,FECS,01234789CEF,1248,8CEF137F,11333311333311C8
2530 DATA 842184218421,124812481248,9AC953F,11224488,
07E07E07E,DB00DB00,969696F08181,1032547698BADCFE
2540 REM PUT CHARDEF'S ON SCREEN
2550 CALL LINK("CHSIZE",6,7):: CHARNO=Q-95 I 1 TO 25
2560 NEG=8*INT(Z*/2+2)! I TO 13 *8
2570 ROW=193-NEG+INT((CHARNO-1)/2)*8
2580 COL=1-((Q/2)<>INT(Q/2))*114 :: CALL LINK("PRINT",
ROW,COL,STR$(CHARNO)&" ":"&A$)
2590 CALL LINK("CHSIZE",8,8)
2600 RETURN
2610 REM DISPLAY NEW PATTERN
2620 BOTTOM=(192-NEG)/8
2630 FOR I=1 TO BOTTOM
2640 CALL LINK("PRINT",1*8-7,1,C$(I))
2650 NEXT I
2660 RETURN
2670 REM TO ARTIST FILE
2680 CALL LINK("PRINT",1,196,"DSK")
2690 CALL LINK("INPUT",1,220,@,1)
2700 CALL LINK("PRINT",1,194,"FILE ")
2710 CALL LINK("INPUT",9,170,@,$,8)
2720 @$="DSK"&STR$(@)&" "&@$
2730 FOR @=1 TO 3 :: CALL LINK("PRINT",@*8-7,1,C$(@)):: NEXT @
2740 CALL LINK("SAVEP",@$,1)
2750 RETURN

```

END OF ARTICLE

CONVERTING from IBM to TI

by Larry Saunders

To convert colour pictures from IBM to TI, the only way to date is GIF to TI Artist in small pieces at a time. In Page Pro terms that would be 33 hor X 16 Ver. That is the bad news. The good news is that a FULL Page Pro page can be converted from IBM to Page Pro in MONO (Black and White) in about 4min 10sec using PCX format and GOFER program on the TI. To convert the same picture from GIF to Page Pro to make a full page would take over 1 Hour or more. Mono pictures converted from IBM to Page Pro by Gofer come out the same as the GIF way but a lot faster.

END OF ARTICLE

TIshUG Software

by Larry Saunders

G036

Games Disk: This disk is full of games for teaching maths, addition, etc.

Used= 316 Free= 42

ALIENADDN1	13*Prog	ALIENADDN2	22*Prog
ALIENADDN3	11*Prog	ALLIGATOR1	13*Prog
ALLIGATOR2	33*Prog	ALLIGATOR3	3*Prog
DEMOLDIVN1	13*Prog	DEMOLDIVN2	20*Prog
DEMOLDIVN3	11*Prog	DRAGONMIX1	13*Prog
DRAGONMIX2	18*Prog	DRAGONMIX3	25*Prog
DRAGONMIX4	10*Prog	LOAD	10 Prog
METEORMPY1	13*Prog	METEORMPY2	33*Prog
METEORMPY3	6*Prog	MINUSMISS1	13*Prog
MINUSMISS2	33*Prog	MINUSMISS3	3*Prog

P037

Page Pro pictures converted by me from CSGD pictures.

Used= 349 Free= 9

4HLOGO	3 I 13	ABE	3 I 13
ANGEL	3 I 13	BATTER	3 I 13
BEL1	3 I 13	BOWL	3 I 13
BOWLER	8 I 13	BUFFALO	3 I 13
CAMERA1	3 I 13	CAMERA2	2 I 13
CHEMIST	3 I 13	COKE	3 I 13
DUCK2	16 I 13	EPSON	5 I 13
FILM	3 I 13	FIRE	3 I 13
FISH	18 I 13	FLAG	3 I 13
FOX1	26 I 13	FOX2	11 I 13
GOLF	3 I 13	GOOSIE	3 I 13
GRAD	3 I 13	GRSHOPR1	14 I 13
GRSHOPR2	8 I 13	HALLEY	3 I 13
HAND	2 I 13	HANGAR2	3 I 13
HENRY	3 I 13	HUSHPUP	3 I 13
INDIAN	3 I 13	JGHOST	3 I 13
KANGA	3 I 13	KARATE	3 I 13
KERMIT	3 I 13	KITTY1	3 I 13
KITTY2	3 I 13	LAMB1	3 I 13
LAMP	3 I 13	LARROW	3 I 13
LIBERTY	3 I 13	LION	3 I 13
MAGGIE	3 I 13	MOTH1	4 I 13
MOTH2	5 I 13	MUSCBOX	3 I 13
NOSMOKE	3 I 13	ORNAMNT	3 I 13
OWL	3 I 13	PANTHER	3 I 13
PENGUIN	8 I 13	PIE1	3 I 13
PIE2	3 I 13	PUMPKN1	3 I 13
PUMPKN2	3 I 13	PUNT	3 I 13
PUPPY	3 I 13	RABBIT	3 I 13
RADIO	3 I 13	RARROW	3 I 13
ROACH1	4 I 13	ROACH2	3 I 13
ROBOT	3 I 13	SPIDER1	4 I 13
SPIDER2	10 I 13	SPIDER3	8 I 13
SPIDER4	8 I 13	STARFISH	5 I 13
SUNSET	3 I 13	TEAKTLE	3 I 13
TEAPOT	3 I 13	TELE	3 I 13
TELESCO	3 I 13	TITCHIP	3 I 13
TILOGO	3 I 13	TULIP	3 I 13
TURKEY	3 I 13	TYSHAT	3 I 13
VGREET	3 I 13	XMSTRE2	3 I 13

AD038

Adventure Disk: This disk includes the diskbase Adventure Module.

Used= 349 Free= 9

FUNHOUSE	42*Prog	GHOSTTOWN	41*Prog
IRONHEART	50*Prog	ISLAND-1	41*Prog
ISLAND-2	50*Prog	LOAD	5*Prog
PYRAMID	41*Prog	UTIL1	13*Prog
UTIL2	25*Prog	VOYAGE	41*Prog

P039

Page Pro pictures: More pictures converted by me.

Used= 339 Free= 19

99/4A	6 I 13	ANCHOR	3 I 13
BALLOON	2 I 13	BICYCLE	3 I 13
BUGGY	3 I 13	CALCULTR	2 I 13
CAR1	2 I 13	CAR2	4 I 13
CASSETTE	3 I 13	CHIP	3 I 13
CLOWN1	6 I 13	CLOWN2	6 I 13
COMMODORE	2 I 13	D1LRSGN	2 I 13
DOSDISK	4 I 13	DRUMMER	7 I 13
DSKDRV	3 I 13	FEMALE	3 I 13
FIFER	7 I 13	GASPUMP	12 I 13
HELICPTR	12 I 13	HGWELLS	12 I 13
IBM	3 I 13	INDIAN	9 I 13
JET	2 I 13	JOYSTCK	3 I 13
MALE	3 I 13	MCKYDIP	10 I 13
MODULE	3 I 13	MONALISA	22 I 13
MOUSE	13 I 13	NEWSPAPR	3 I 13
NINJA	15 I 13	OLDPHONE	2 I 13
PAPRCLIP	2 I 13	PBOX	6 I 13
PHONE	2 I 13	PORSCH	13 I 13
PRINTER	13 I 13	RTANGLE	4 I 13
SAW	3 I 13	SCISSORS	2 I 13
SCREW1	6 I 13	SCREW2	6 I 13
SCRWDRVR	2 I 13	SOCRRBLL	2 I 13
SPADE	2 I 13	STAR	3 I 13
TEACHER	13 I 13	THUMBSUP	4 I 13
TORCH	2 I 13	TREE	16 I 13
TREE2	2 I 13	TRIANGLE	3 I 13
WATCH	4 I 13	WIZARD	9 I 13
WOMAN	20 I 13	WORLDMAP	3 I 13
WRENCH	2 I 13		

E048

Used= 178 Free= 180

COLORS	3*i 80	DOCUMENT	44*d 80
GREEK	46*Prog	LETTERNAME	3*i 80
LOAD	5*Prog	1TRSOUNDS	4*i 80
NTGG1	19*i 80	NTGG2	17*i 80
NUMBERS	8*i 80	TOBE	3*i 80
VERBS	26*i 80		

E049

Used= 230 Free= 128

ADDWORDS	12*Prog	CATALOG	4*Prog
COLORS	2*i 80	DAYS	3*i 80
DOCADD	3*d 80	DOCEDIT	6*d 80
DOCFW	15*d 80	DOCINTRO	7*d 80
DOCJOIN	5*d 80	DOCMAKE	14*d 80
DOCMC	6*d 80	DOCSPLIT	4*d 80
DOCTF	5*d 80	DOCUMENT	9*d 80
EDITFILE	12*Prog	FOOD	6*i 80
HANDY	6*i 80	JOINFILES	6*Prog
LOAD	5*Prog	LOOKDOCS	9*Prog
MAKEFILE	11*Prog	MCQUIZ	17*Prog
MENU	14*Prog	MONTHS	4*i 80
NUMBERS	4*i 80	ROADSIGNS	5*i 80
SIGNS	4*i 80	SPLITFILES	8*Prog
TFQUIZ	19*Prog	WEATHER	5*i 80

E050

Language Disk

Used= 132 Free= 226

BOTHSAMPLE	4*i 80	CATALOG	4*Prog
CZECH	37*Prog	DOC2	11*d 80
DOC3	9*d 80	DOCUMENT	31*d 80
FIXER	6*Prog	HARDCOPY	5*Prog
JOINFILES	7*Prog	LOAD	6*Prog
LOOKDOCS	7*Prog	SAMPLE1	3*i 80
SAMPLE2	2*i 80		

E051

Language Disk

Used= 144 Free= 214

CAT	4*Prog	CATALOG	4*Prog
CZECHO1	17*Prog	CZECHO2	25*Prog
CZECHO3	25*Prog	CZECHO4	19*Prog
CZECHBASE	10*Prog	DOCUMENT	26*d 80
LOAD	7*Prog	LOOKDOCS	7*Prog

E052

Language Disk

Used= 128 Free= 230

BOTHSAMPLE	3*i 80	CATALOG	4*Prog
DOC2	11*d 80	DOC3	9*d 80
DOCUMENT	31*d 80	FIXER	6*Prog
GERMAN	34*Prog	HARDCOPY	5*Prog
JOINFILES	7*Prog	LOAD	6*Prog
LOOKDOCS	7*Prog	SAMPLE	3*i 80
SAMPLE2	2*i 80		

E053

Language Disk

Used= 154 Free= 204

CATALOG	4*Prog	DOC2	11*d 80
DOC3	9*d 80	DOCUMENT	31*d 80
FIXER	6*Prog	FRENCH	37*Prog
HARDCOPY	5*Prog	JOINFILES	7*Prog
LOAD	6*Prog	LOOKDOCS	7*Prog
SAMPLE	31*i 80		

E054

Language Disk

Used= 194 Free= 164

CATALOG	4*Prog	CYRILLIC	40*Prog
DOCUMENT	27*d 80	DONOTTOUCH	3*d 80
LOAD	4*Prog	LOOKDOC	4*Prog
LOOKDOCS	4*Prog	RUSSIANI	46*Prog
RUSSIAN2	45*Prog	SHOWDOCS	4*Prog
SUNSHINE	13*Prog		

E055

Language Disk

Used= 259 Free= 99

ADJECTIVES	4 i 80	ANIMALS	4*i 80
BODY	4 i 80	DAYSOFWEEK	3*i 80
DEFKANA1	5*I 17	DEFKANA2	5*I 17
DOCUMENT	50*d 80	DRINKING	5*i 80
EATING	4*i 80	HGFOO1	17*i 80
HGFOO2	6*i 80	KANAFILER	27*Prog
KKFOO1	20*i 80	KKUSGEOG	11*i 80
LOAD	18*Prog	MCKANAFILER	24*Prog
MISC1	5 i 80	OLDNUMBERS	3*i 80
SPEAK	24*Prog	VERBFILE	20*i 80

EO56

language Disk

Used= 130 Free= 228

BEGIN/SPAN	33 Prog	LOAD	5 Prog
MATH/SPAN	12 Prog	SPAN/ALPH	41 Prog
SPAN/VERB	11 Prog	SPANISH	28 Prog

END OF ARTICLE

LEARNING to KNOW YOUR TI

LESSON 8
with Percy Harrison

I must admit that these lessons are still somewhat boring but if you keep persevering I assure you that they will capture your attention, but better still you are gradually getting to know how to use your faithful TI and will soon be able to write better and more interesting programs so keep working through these lessons each month and do not hesitate to ask questions of me, or other members of your club, if you do not understand any of the features included in these lessons.

This month we are going to look at the GOTO statement. The GOTO command allows a "dump" loop that goes on forever. It also helps in the flow of commands in later programs after the IF statement is introduced. It provides a slow and easy entrance into the idea that the flow of commands needs not just go down the list of numbered lines. For now the main use of the GOTO command will be to let programs run on for a reasonable length of time, in later lessons we will introduce other commands to be used in conjunction with the GOTO command which will allow greater flexibility in programming.

LESSON 8 THE GOTO STATEMENT

THE KEYBOARD OVERLAY

The strip overlay is a narrow piece of plastic which has words printed along it:
DEL INS ERASE CLEAR PROC'D AID RED O BACK QUIT

It belongs in the groove above the top row of keys.

DEL	...1...	lesson 4
INS	lesson 6
CLEAR	this lesson
QUIT	this lesson

Fill in which key is below each word.

The rest of the words are not used in TI BASIC.

JUMPING AROUND IN YOUR PROGRAM

```

Try this program: 10 CALL CLEAR
                  20 PRINT "YOUR NAME?"
                  25 INPUT N$
                  30 PRINT N$
                  35 PRINT
                  40 GOTO 30

```

RUN this program. It never stops by itself! This is what we call a LOOPED program. To stop your name from whizzing past your eyes:

hold down the FCTN key

and press the 4.

FCTN 4 means FCTN CLEAR. It stops the program and clears the screen.

line 40 uses the GOTO command. It is like "GO TO JAIL" in a game of Monopoly. Every time the computer reaches line 40, it is instructed to go back to line 30 and prints your name again.

We will use GOTO in a lot of programs in following lessons.

When you use FCTN CLEAR to stop the program it does not erase the program from memory, if you type RUN and press ENTER the program will start again.

WARNING!

You may be tempted to press FCTN QUIT to escape from a looped program by holding down the FCTN key and pressing the "equal sign" key. This will certainly stop the program but it will also ERASE the program from memory and the program will be lost. DO NOT use FCTN QUIT to escape from a looped program unless you want to erase that program from memory. Always use FCTN CLEAR to stop the program from running.

MORE JUMPING

```

Enter: 10 PRINT "SAY SOMETHING"
       30 INPUT S$
       35 PRINT
       40 PRINT "DID YOU SAY '";S$;"'?"
       45 PRINT
       50 GOTO 30

```

Run the program. Type an answer every time you see the "?" and the flashing cursor. Press the FCTN CLEAR keys to end the program.

Notice the arrow from line 50 to line 30. It shows what the GOTO command does. You may want to draw such arrows in your own program listings when you do programming to clarify what the program is doing.

KINDS OF JUMPS

There are only two ways to jump: ahead or back.

Jumping back gives a LOOP.

```

10 PRINT "HI"
20 GOTO 10

```

The path through the program is like this:

```

10 PRINT "HI"
20 GOTO 10

```

The computer goes around and around in this loop. Press the FCTN CLEAR keys to stop.

Jumping ahead lets you skip part of the program. It is not done as often as jumping back.

A CAN OF SPAGHETTI

Look at this program:

```

10 REM ::: SPAGHETTI :::
20 GOTO 70
25 PRINT "A"
26 GOTO 50
30 PRINT "S"
31 GOTO 25
40 PRINT "C"
41 GOTO 90
50 PRINT "U"
51 GOTO 40
70 PRINT "SPAGHETTI"
71 GOTO 30
90 PRINT "E"
99 REM ::: END :::

```

WHEW! This is NOT a good clear program! It is a "spaghetti" program.

Do not write spaghetti programs! Do not jump around too much in your programs.

Assignment 8:

1. Just for practice in understanding the GOTO statement, draw the "road map" for this spaghetti program:

```

10 REM ::: FORKED TONGUE :::
20 GOTO 40
30 PRINT "N"
31 GOTO 60
40 PRINT "S"
41 GOTO 30
50 PRINT "E"
51 GOTO 99
60 PRINT "A"
61 GOTO 90
90 PRINT "K"
91 GOTO 50
99 PRINT "B I T E"

```

2. Write a program which prints "TEEN POWER" over and over.
3. How do you stop your program when it is in a loop?
4. Write another program which prints your name on one line, then a friend's on the next, over and over. Sound a tone as each name is printed. Stop the program with FCTN CLEAR keys.
5. Write a program which uses each of these commands: CALL CLEAR, PRINT, INPUT, LET, GOTO. It also should glue two strings together.

ANSWER TO LESSON 7

Assignment Question 7-2

```

10 REM FEELINGS
15 CALL CLEAR
20 PRINT
22 PRINT
24 PRINT "HOW IS THE WEATHER?"
26 PRINT
28 INPUT W$
30 PRINT "AND HOW DO YOU FEEL?"
32 PRINT
34 INPUT F$
36 PRINT
38 PRINT "YOU MEAN:"
40 PRINT
45 S$ = W$ & " AND " & F$
50 PRINT S$

```

END OF ARTICLE

Vincent's Corner

ADVENTURES and SOLVING them PART I

So, you have bought an adventure module? And you have started on the pirate one? You have absolutely no idea of what to do? You are tearing your hair out after six months?

These things take the most intelligent person in the world one hour to solve, so it may take the rest of us days or weeks or years (no joke, it took me five years to solve Ghost Town!).

But here are some tips from an experienced Adventurer of nine years. Do Not make the mistakes I did!

ADVENTURES:

- 1) Use action words, such as- RUN, GO, LOOK, WEAR, SAY, CLIMB
- 2) Read each and every screen! For example, In Adventure #8 we have an iron statue of a Pharaoh sitting on a throne. Pull the chain and he is standing. Pull the chain again ...)
- 3) Think of every possibility. It may be more complicated than a Sherlock Holmes novel.
- 4) If you cannot work it out, see if your friends can.
- 5) Do not get discouraged. It is a very rewarding experience solving these 'impossibilities'.
- 6) If, after a very long experience of thought, you cannot work it out, keep reading my columns. Adventure is included every month.
- 7) I hope you do play Adventure, especially Adventure #8. It took me a year and a half to solve it all. How long will it take you?

VINCENT MAKER

P.S. In what follows you will find the first of the hints to #2. Happy adventuring.

ADVENTURE #2: PIRATE'S ADVENTURE

- 1) Do not know what to do at the start? What do mountain climbers do? We might need to use additional footwear too.
- 2) Problems in the alcove? Expands the mind! Think about "ABRACADABRA".
- 3) Do not what to do when the tide is coming in? Solution- use, "Look Lagoon".
- 4) Stuck to the shack? Guess who is an alcoholic?
- 5) Cracks in the rocks? Do not like books about Treasure Island?
- 6) Cannot get lumber out of the cavern? look, listen and learn for the moment.
- 7) Cannot get hold of a set of keys? Got a hammer? Then act like a handyman in the flat.
- 8) Have we met the crocodiles yet? Try- "Go Cave" and you will need a bottle of salt water (Do not drink it!) and some fish. Waterwings are handy things in the sea.
- 9) No anchor? You might need to use the shovel on a particular beach.
- 10) Do we have a ship yet? If not, we might find some plans in the treasure chest!
- 11) The ship will not go anywhere? Never sponsor a PIRATE in a readathon. In other words he hates books.
- 12) Snakes snare? You might recall a certain character talking about pieces of eight. "I be liking parrots matey, they be smart."
- 13) Cannot get the pirate back on board? What did we do on the beach back home?
- 14) Have we got *Rare Stamps* yet? We had better re-read out note again...paces...then.
- 15) Box a bore? Well, you will get the hang of it if you re-read some of my hints.
- 16) If all is in order, take the treasures back to the flat and say...

Here is a concluding letter from Vincent.

Dear friends,

If you wish to get in touch with me to ask a question about Adventure or about my column or maybe something you want to put in, please feel free to do so. My phone number is

(02) 873 1321

If I am not at home, please leave your question or request with my family and they will pass it on to me. I will then either include your request in my next column or if it is a question on Adventure I will ring you up or write to you, whichever you prefer. Any Adventure enquiries I receive will be printed in my next column. I will, however, straight away give you the solution over the phone or by letter.

The best time to ring me up is on the weekends, Saturday before 12 m.d. or Sunday before 10 am as I have church at 11 am. On a weekday call between 5-6pm.

Again, my address is:

* Vincent's Corner *
* 7 Thrift Close *
* West Pennant Hills *
* NSW 2125 Australia *

Keep in mind that the best contributions to my column win \$10. Periodically I run Adventure competitions... prizes being \$5.

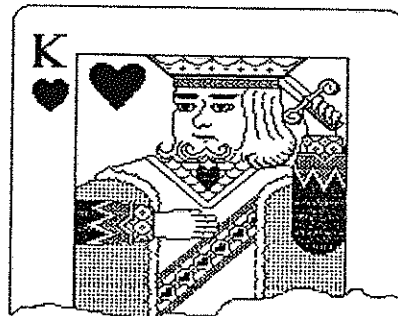
Consolation prizes will be games for the TI99/4A and anything else I can lay my hand on. Every entry will receive a prize, so gets yours in.

Do not miss out! Be sure to included whether you want your games on cassette or disk. As of August I will be on the BBS.

Cheers 4 now,

Vincent Maker

END OF ARTICLE 



TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month	\$ 448.85
Expenditure for previous month ..	\$ 1020.59
Loss for previous month	\$ 571.74
Membership accounted for	\$ 40.00 of Income.
Shop sales	\$408.85 of Income.
The expenditure was made up of the following	
Purchase of stock for Shop	\$ 397.00
Repairs to Club Equipment	\$ 75.00
Printing & Postage of TND	\$ 326.29
Payment to Australian Securities	\$ 162.00
Replacing damaged School item ..	\$ 29.95
Postage & Telephone etc.	\$ 30.35

LETTER HEAD DESIGN

by Ken Burdges
Atlanta Call Newsletter Sept/Oct 85

This is a demonstration of a letterhead designed by bit graphics for the Gemini 10X printer. The entire letterhead is programmed for TI-Writer using the Transliterate command and various printer commands. My experience indicates that a TL command cannot be longer than one line. This appears to be the only limitation of the program. Your patience will get a real work out in building something like this letterhead, but it is a unique capability allowing personalized communication.

The following file was used to build the letterhead:

```
.IF DSK1.LETTERHEAD
.CO Draft file commands, elite,
.CO dbi space,1 1/2 and 1 in margin
.TL 91:27,45,1
.TL 93:27,45,0
.TL 60:27,66,3
.TL 62:18
.TL 123:27,52
.TL 125:27,53
.TL 126:27,66,2
.TL 47:27,85,0
.CO Elite 12 cpi,[Underline],[Italic]
.CO <Condensed>
.CO Personal Logo in bit graphics
.TL 33:27,49,13
.TL 34:27,85,1
.TL 35:27,14
.CO ! Line feed and carriage return, " for
unidirectional, # for enlarged
.TL 49:27,75,15,0,0,0,6,30,48,96,64,128,192,224,254,
254,126,126,58,0
" !
.TL 49:27,75,15,0,0,0,0,6,2,0,128,192,96,96,112,112,
240,224,224
.TL 50:27,75,5,0,226,192,128,0,0
" 12 {#NOVATIVE}!
.TL 49:27,75,15,0,0,0,14,124,240,192,192,70,126,126,120,
112,96,192,0
" !
.TL 49:27,75,15,0,0,0,0,6,30,240,192,0,0,6,30,254,240,
224,128
" 1 {ENGINEERING CO.}!
.TL 49:27,75,15,0,0,0,126,254,0,6,30,126,254,192,0,0,
0,0,0
.TL 50:27,75,15,0,0,2,2,2,2,2,2,2,2,6,6,6,6,6,6
.TL 51:27,75,15,0,6,14,14,14,14,14,12,28,28,28,28,24,
24,56,56
.TL 52:27,75,17,0,56,48,48,48,48,112,112,112,112,96,
96,96,96,96,64,64
.TL 53:27,75,6,0,192,192,192,192,128,128
.TL 54:27,75,5,0,128,128,128,128,128
" 123451
.TL 49:27,75,15,0,0,0,128,224,96,248,248,248,248,248,
248,248,240,240,240
.TL 50:27,75,15,0,240,240,224,224,224,224,192,192,192,
192,128,128,128,128,128
" 12!
.TL 49:27,50
.TL 50:27,85,0
12
.CO Put numbers back to original form
.TL 49:49
.TL 50:50
.TL 51:51
.TL 52:52
.TL 53:53
.TL 54:54
.CO Begin text control
.LM 0;RM 70;FI;IN +5
```

Editor's Comments

by Bob Relyea

I am writing this column greatly encouraged by the fact that we will go into 1994 with an editor. You may recall that I asked for volunteers for the editing job in last month's issue and at the last club meeting we got one! Details of this will be left to the directors to announce. I have done the job for four years now and would like a change to something else. Dick and a couple of other directors and I discussed this at the last meeting and came to a very acceptable arrangement. I am happy that I will be able to continue serving the club in some capacity and also happy that somebody volunteered to take on the very important editing job. I have learned a lot about the club, the computer and word processing in the last few years and look forward to continuing this in coming years in other capacities. Hope you enjoy this issue.

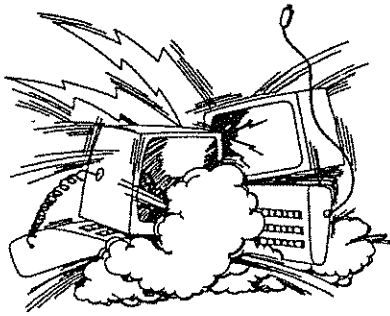


FROM the BULLETIN BOARD

IF anyone is interested in a program that can work on IBM and the TI, i.e. files that are interchangeable please show your interest at the upcoming meeting. The program is basically an address book that stores information in text format, which can then be downloaded to the TI and used with no modifications on an equivalent program which I will write for the TI. Conversely you can upload files from the TI to the IBM and use them with no problems. Note that you are not restricted to an address book program. I just wrote that as an experiment. If you want something else written I will do it, if I can. Claude (Shifty).

END OF ARTICLE

END OF ARTICLE



crash

CO-ORDINATORS REPORT

1994 will mark a new stage in our club's development. At present, membership is declining steadily, and interest in the TI is waning. We continue to meet regularly, to help our members with their computer problems, provide a shop, a library, technical support, and a good monthly magazine.

I envisage some changes later in 1993. Firstly, we will move to new premises for our monthly meeting from November. The October meeting will be the last Saturday meeting at Ryde School. We will be transferring to Meadowbank School in Thistle St. Meadowbank, for all future meetings.

The directors have agreed to this change for a number of reasons. eg.

We wish to encourage those who attend to both bring and use their own systems. At Ryde, there is a shortage of furniture, particularly tables, and appropriate sized chairs, as well as power points. At Meadowbank, we will have the use of the school library, a large staffroom, and a double classroom with tables. The power supply is adequate. The cost to the club will be the same as it is now. Transport is probably even easier to reach Meadowbank School, as it is closer to the railway.

With a wider range of activities, we can separate the different groups of users. I think we will be able to use

other rooms on special occasions if we need to do so. Parking will be as easy or easier than at Ryde.

Secondly, in 1994, we will gradually change our emphasis from using the TI exclusively to include more activities and help for PC users. Hopefully, the September Magazine will include information for IBM users. We have been given permission to publish tips for using the PC printed in "Your Computer". Until our own contributions increase, these tips will be appearing in our supplement. At this stage they will be printed using our normal TI software.

but this will change in time. As our emphasis changes, and we can provide help and service for PC users, we intend to publicise our activities, and recruit a steady number of new members to replace those who have left the club. It would be really good if we attract people who actually know us, either old members, friends or relatives. If YOU use a PC, what about letting your friends know about us. Tell people where you work. If we get sufficient interest, I'm sure that we could run courses in a wide number of popular areas; eg in using software like Windows, Xtree Gold, Lotus 123, Q&A, or Dbase etc. If members are interested, we could bring in experts to talk on all kinds of hardware problems, eg. Hard Drives, Memory Configuration, portables etc. With the new arrangements, I trust some members will bring their portable machines and notebooks with them.

Peter Schubert may soon have a wide range of low priced hardware available, including the new Pentium machines available through the shop. As demand increases, we will approach suppliers and wholesalers, to act as an outlet for computer hardware and peripherals. As a result of these changes, I confidently expect that our membership will grow again, and as a result we will be able to support TI members for a long time.

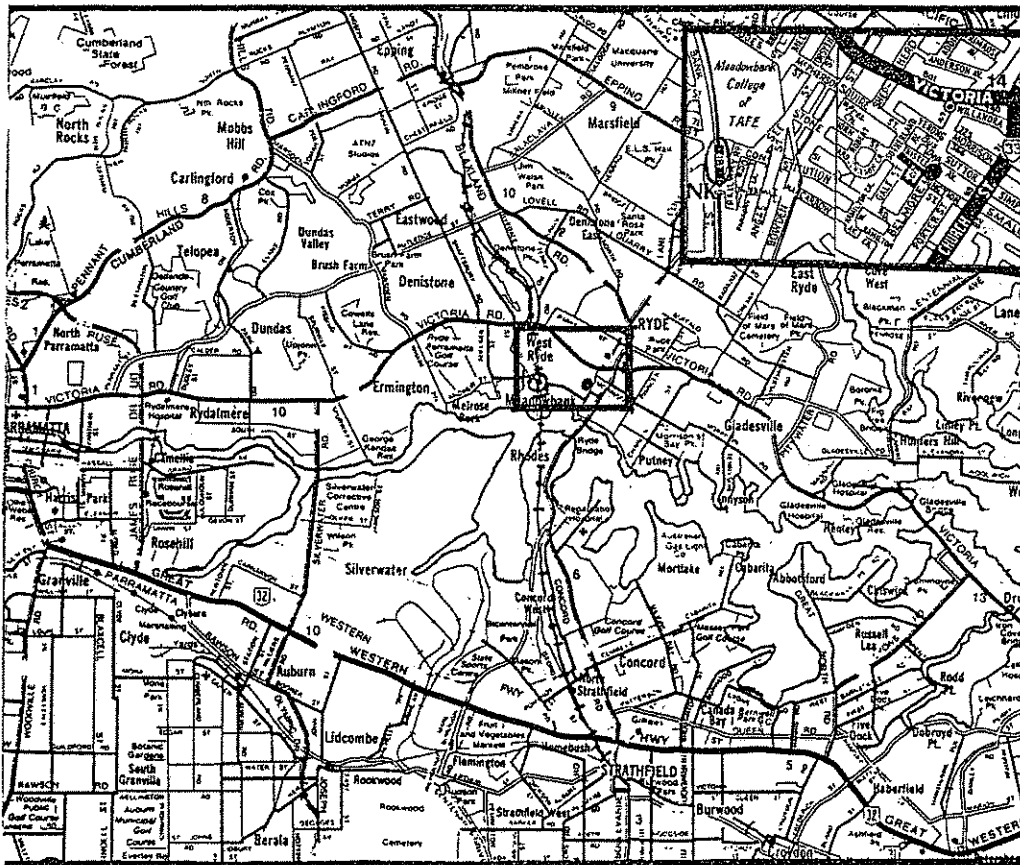
Recently I spent some time with Mike Slattery who still enjoys programming on his TI. He has just completed an excellent wages program in

extended basic. The program is so big that it took over an hour to prescan it. Prescan makes programs run so efficiently. It speeds the operation up considerably. Mike finds that he still gets real enjoyment from programming on his TI. He finds programming on his Macintosh to be much less enjoyable, and much less productive. I continue to prefer to use my TI at work as it is so convenient to do so. The programs available do almost everything I want. One certainly gets spoiled using a ramdisk (when it works well) and it is always a frustration to use the PCs at work and wait a long time for Windows to load.

Sometimes I enjoy running very simple programs. It is far easier to do so on the TI than on a machine where the language environment has to be loaded first. I certainly enjoy both machines, but I can only wonder what heights the TI would have reached if it had been supported like the PC.

I'll see you at the next meeting.

Dick Warburton.



END OF ARTICLE

Nostalgia Time

by Geoff Trott

This series of articles consists of my observations on the contents of the early TNDs. Assuming that you find it interesting, I am continuing with the series this month. Please stop me if you do not want me to continue. I will repeat my general disclaimer in case anyone reading this article gets the wrong idea. I am attempting to describe the look, layout and content of the newsletters without any critical intent. I will try to avoid using any adjectives which could cause offense and if anyone takes offense, that is purely their interpretation of the words and not my intention. I hope that makes my position clear and that no one will be offended.

The June 1984 issue was called the Tutorial Edition. It had 24 pages and a similar layout to the May issue with a white cover which was thicker than the inside pages. The cover picture was of a robot with a TI99/4A as its face and holding a book of programs from TISHUG. The inside cover page contained club information while the inside rear cover contained the cut-up page. The back page contained two advertisements. Most pages were divided into three columns with borders around each column. There was a box across the bottom of each page containing the page numbers, but nothing else.

The Communicators announced the rules for the next meeting, which was a tutorial day. Each person coming to the day had to pay \$2.50 (or \$2.00) and had to have their membership card with them. They would get drinks and a 32 page book for that. Numbers were restricted to 250 people! Shane also mentioned the imminent start of the BBS and that the Perth group had got their BBS up and running. Secretary's Notebook with John Robinson has many photographs on the page of members manning a stand at the Ryde Educational Computer Faire. John reported that he had a telephone answering machine and that the May meeting attracted 280 members who showed great interest in the robots shown by Shane. A technical sub-committee was formed.

The first article from Tony McGovern on getting the best out of Extended BASIC appeared in this issue. I am sure that much of what he wrote is still of great value today. Perhaps a series like this should be re-printed regularly. Jenny's Younger Set had two programs, one in BASIC and the other in Logo. The Regional news contained reports from Blaxland, Maitland, Newcastle, Liverpool, Marrickville and Nepean. In the letters to the editor there were letters from England about connecting a cheap Tandy serial printer to the TI99/4A, using a variable speed cassette to load programs more reliably and a program to allow text to be displayed with graphics. There was also a reply to a letter from New Zealand.

The first TechoTime article appeared to announce the formation of the technical sub-committee. Robert Peverill was the contact person. There was an article on printing out a program listing using a joytalk serial interface from Chris Potts. There was another program by Russell Welham on music, this time the Baby Elephant Walk. There was a short article on using CALL KEY for the input of data. J.P. Ashmore wrote a program called Squares, Garry Christensen (Brisbane) wrote a little program called Spritipede, S. Nielsen (Brisbane) wrote a small program called 5 little Ducks, Chick de Marti (USA) wrote a program called Stickman and Derek Megginson wrote a program to help people learning a foreign language (German in that case).

There was a review of a program called Dow Flight Trainer and mention of a program called Graphics Code Generator. There was also a review of a local program called The Rainbow Pyramid which was a game by John Ashmore.

The July issue had a grey cover (thicker) with a drawing of a rural scene on the front cover and the back cover pages totally taken up with cut-up pages. One of the cut-up pages had a membership application, classified advertisement form, Shop mail order form and a Bankcard authorisation form. The back page itself was a registration form for the BBS which was a free service to TISHUG members. The inside front cover had the club information which gave the committee as: Co-ordinator, Peter Varga; Secretary, John Robinson; Treasurer and Librarian, Terry Phillips; Assistant Librarian, Andrew Nutting; Editor, Shane Andersen; Public Relations, Cris Ryan; Programmers Crisis Line, Graeme Holliss; Music Co-ordinator, Russell Welham. Gone are Paul Mansel and Peter Lynden.

This issue was 28 pages long with the centre 4 pages printed on yellow paper. These four pages formed a liftout section for the new BBS which was running using software written mainly by Robert Crago, who won the contest for this software. The operating instructions were printed here along with a list of the first users. Techotime gave the connections for a monochrome monitor, a parallel printer to PIO and RS232 to several modems. Robert also announced that a 32K memory card was being prepared and appealing for someone to design a stand alone PIO or RS232.

Shane's editorial talked about the contents of the issue and mentioned that the photographs in the issue are from the tutorial day. I noticed that there was a photograph of Bob Montgomery, Michael Bail and Rolf Schreiber appeared in one also. Secretary's Notebook had two programs to find the first 100 prime numbers in BASIC. They had agreed to swap newsletters with the Sydney Forth group (I wonder where they went?). Regional news came from Liverpool, Illawarra, Baulkham Hills, Newcastle who have a President, Joe Wright, Secretary/Treasurer, Peter Coxon, Librarian, Al Lawrence, and the Marrickville Regional group.

There was an article on Forth from Ed York of Cincinnati with a program (word) for a clock. It has the usual problem of programs originating from the USA that it assumes 60 Hz power whereas we have 50 Hz power. This is easy to correct. In fact, I think that it also required an adjustment to the correction factor to keep good time. It used interrupts to keep the clock on the screen while you ran other programs. Tony McGovern's second article on Extended BASIC appeared here along with another music program from Russell Welham. Jenny's Younger Set appeared to be from overseas sources, with a few small programs. There were program listings from: D.A. Lynden, Bat Attack; R.A. Johnsson, a program to calculate the monthly cost of running a motor vehicle; Ben Vang, The Entertainer music program. There were a number of small programs given as hints from various parts of the USA. There was a review of a game called Tex-Bounce written by Tony and Will McGovern. It sounds like a great programming effort and I am sorry that I have not seen it running.

All the articles and listings in this issue were printed on the daisy wheel printer which gave a uniformity of presentation. Some of the articles were in the three column format while others were in two columns. Listings of Forth programs are always a problem as they require 64 characters per line. There were a lot of photographs of people in the issue with a number of these labelled with their first names. This is quite interesting to identify people that are still members. One continuing problem was that spelling and typographical mistakes kept slipping through, sometimes making it hard to understand the meaning of the author. It was also rather difficult at times to determine who was the author of some articles.

END OF ARTICLE

LETTER from AUSTRALIA No. 1

by Tony McGovern Mar/17/92

A while back I made the suggestion to Charlie Good that instead of writing letters in which a lot of the material could well be of general interest as newsletter articles, that I might as well do the relevant parts as a disk file under the name "letter from Australia". No sooner had I made this rash suggestion than Charlie accepted it before I could change my mind and had the idea that it could be a bit like Alastair Cooke's letter from America which is broadcast on the BBC world service. That was indeed the model I had filched for the name - I regularly listen to the program on ABC radio here most Sunday evenings at 7.15, but I do not think it is going to be in the same journalistic league. I always enjoy the mixture of whimsy and often very penetrating comment on the state of the US of A. We will be more concerned with the TI99/4A world, or what remains of it. First for the whimsy.

The question arose here a little while back as to whether security guards would be more effective if they had a solid classical education, particularly in ancient Greek. The incident that set this off was not really anything to do with George Bush's flying visit here on the way to Japan, but it did follow on in a sort of way. Now George's table manners in Australia were impeccable and our newly minted Prime Minister made a truly awful speech, but that was not it either. It was in fact a Trojan Horse that caused the problem. A what? Well, Australian farmers are quite unhappy about the way the US is soaking its own taxpayers to subsidize wheat sales around the world to do battle with the EEC who do even more of it. They do not really care about anyone else's taxes, but they do care about their markets and were demonstrating to make their point to Mr Bush. They rolled out a metal horse made by a former political candidate for campaigning in a previous election. The body of this thing is a metal tank like the ones that get buried in the driveways of gas stations, so you have an idea of the scale of the thing.

After the official visit the metal horse was hauled back to Sydney on a very large truck. In the meanwhile one of the banks, they are no different here, had decided to foreclose on the business of the ex-political candidate. The bank locked everyone out of the factory and posted security guards around it with instructions to let no one in. This was all publicized in the newspapers. A day or two later the owner rolled up in the huge truck and asked the guards if he could park the Trojan Horse in the factory yard - according to reports in literally those words. Of course once it was inside, out piled 30 or 40 of the factory people and retook possession from the bank. The rest of Australia was still laughing a day later and wondering about security men who had never heard of a Trojan horse. There cannot be anyone with an interest in computers who does not have some idea.

It has been summer here, but not one to remember. First it was drought and heat, then rain and the surf conditions ranged between bad and dangerous. So some programming got done and Charlie should have something new and interesting to show at the May festivities. Not much new finds its way to Newcastle these days and I am so busy programming that I do not bother with commercial software, no matter how cheap it is for the TI99/4A, because I never get around to using it. Besides there is really only one game worth playing on a computer - programming it to do the ultimate it is capable of. The commercial software seen here has not been all that inspiring. Al Lawrence bought a spell-checker and from what I have seen and heard of it, while it may be a fine spell-checker it needs major rework to get its disk handling into acceptable form. For my part I could always spell, it is just that my typing is lousy. So I have no direct user interest but I am concerned that there be a viable spell-checker, be it commercial or fairware, to complement the Funnelweb system.

It has not been entirely a drought on the fairware front. On my wish list for some while has been a GIF picture file loader for 80 column cards in the TI99/4A that would also convert to Myart format. For all its problems, stemming from buggy early Myart releases, this format is the easiest and fastest picture format to load on the TI99/4A, even if the disk files show less compression than GIF. Funnelweb's DiskReview has a fast and reliable Myart loader for convenience, but there is no way that a GIF loader could fit in. We were stuck between Geneve software that would not run on a TI99/4A and an older German program that would load GIF very slowly but not convert. Now the wish has come true thanks to a computer student in the Netherlands, Ton Brouwer.

There are some other items on my wish list. Chief among these is an update to the Horizon RamDisk ROS that will allow 800 Kbyte DSQD equivalent disks to be set up. It is only very recently that we have had a HRD big enough for this to be of relevance, but several projects such as DiskReview and the Editor rewrite have reached the point where it is getting difficult to fit source and object files on a 400 Kbyte RAMDisk. I may be a more obsessive commenter of source code than many, but I am sure there must be other programmers out there with big projects running into the same problem. No, I do not regard the Myarc HFDC as viable device unless its maker gives it at least one more serious development stage - from alpha to beta phase. A not very productive direction was taken in the Vn 8.1x HRD ROS development to cater for zillions of small equivalent drives instead of a few decent size ones on larger HRDs. Maybe Bud Mills will have something good in that line in Lima for May. In the meanwhile we may have to redo the ROS for the local Quest RD so its 512 Kbyte can be used as a single drive.

Well, that is enough for now. Until next time, goodbye to all Bits, Bytes & Pixels readers from the Hunter Valley.

END OF ARTICLE

MINI-TOWER PANDA BOX is LOADED WITH EXPANSION POSSIBILITIES

by Bryant C. Pedigo

This review appeared in the newsletter of the Hoosier User Group, and has been retyped from the May, 1993 issue of MICROPENDIUM.

Last May (1992) at the Lima, Ohio, Multi-User Group Conference, I purchased a Panda Expansion Box, which is a modified IBM mini-tower case marketed by Bill Nelson of Panda Computers Products of Garden Grove, California.

The Panda box has several advantages over the TI Peripheral Expansion Box, the foremost of which is a 200-watt power supply capable of running four floppy drives and a hard drive, all installed internally. There are spaces for 2 half-height 5.25-inch and 2 half height 3.5-inch floppy drives, as well as a 3.5-inch hard drive. Additionally, the Panda has a reset switch, a power connector at the back for use with a monitor, a very quiet cooling fan and a compact contemporary design. For Geneve users, there is a keyboard port on the front, and a card protector with a support post to help support the protruding part of the Geneve card.

Although it is designed for use with a Geneve 9640 in mind, it can easily be used with the TI99/4A. Although I purchased it to use with my Geneve, I tried it out with a TI99/4A. Unlike the Rave Expansion Box, you have to use the flex cable interface card if you want to use a TI99/4A console.

The inside of the Panda has been modified to accept TI style cards inserted horizontally. Because the power supply has a regulated +/- 12v output, any cards that use either a 7812 or a 7912 voltage regulator require modification in the form of a jumper to short out the +/- 12 volt regulators. The directions include complete information as to which cards require this modification and how to jumper both types of regulator.

The directions are, for the most part, complete and easy to follow. One item that was not mentioned, and which proved to be a major problem for me, was that the power supply must have a minimum power draw in order to work. Because of a problem that developed with my hard drive, I tried to set up the Panda with a Geneve, One 5.25-inch and two 3.5-inch floppy drives. It would start to power-up and then turn off. With a TI99/4A, there was no problem. Upon contacting Bill Nelson, I shipped the box and my Geneve to him. He checked the system out and found that it was not drawing enough power without a hard drive installed. After the box was returned to me, I installed a second 5.25-inch floppy, thus solving the power draw problem.

I have been using the system for several months without any problem. I recently tested it with both internal and external hard drives and had no problems. I have noticed that cards seem to run as hot in the Panda box as in the TI PEB, therefore, it would probably be a good idea to block off those parts of the ventilation grill at the rear that do not have cards installed so that the air flow inside will be increased.

I have no qualms about recommending the Panda box, especially in view of the excellent support from the vendor.

The Panda Expansion Box is priced at \$200.00. Nelson builds the boxes on request and requires that buyers ship a TI PEB to him to expedite the process. The TI PEB bus and rack is used in the Panda box - Ed.

END OF ARTICLE 

EXTENDED BASIC III

While there have been many revisions to TI Extended Basic over the years, none is so extensive or fundamental as Extended Basic III by Winfried Winkler.

Re-written and revised in the native language of Extended Basic, GPL, Extended Basic III features substantial improvements in the language itself, not just added calls and functions. Unlike other Extended Basic's, this one offers tangible benefits to people who are casual Extended Basic users, as well as hardcore programmers. Why? Because this version of Extended Basic is the only one that will run your entire library of Extended Basic programmes, without modification and up to 50% faster than the original TI Extended Basic. Unlike MYARC Extended Basic - this version of Extended Basic is 100% compatible with all TI Extended Basic programmes and add-on utilities. Additionally, bugs that cause other Extended Basic's to crash (and the infamous graphics screen show that, while interesting, certainly did not help your programmes) have been eliminated. Extended Basic III is rock-solid and dependable- great for anyone running programmes for hours or days at a time (BBS',ect.).

If you are interested in writing new programmes, Extended Basic III offers a wide range of added functions, calls, modifications and conveniences- many of the best offered in other variations along with a number of unique changes. These include:

- * Allows character definitions up to code 159- in fact, Extended Basic III is fully compatible with TI BASIC
- * The GOTO and GOSUB statements will allow you to jump to a variable (as in GOTO X)
- * Enhancements to the IMAGE command to allow more flexible formatting
- * Extensions to the CALL MOTION command that allows you to stop and start all sprites at once
- * The RESTORE command will reverse a RESequence
- * COPY and MOVE commands for copying and moving ranges of lines
- * The assembly LOADER recognizes a wider range of utilities including GPI utilities for manipulating a stack, RADIX-100 math, and so forth for use in your assembly subroutines. The loader will also load Compressed format assembly files with REReferences- and adds a wide range of new REReferences available for called subroutines

* A range of new calls including:

- ALL(N)- Fills the entire screen with the specified ASCII code
- ALOCK(N)- Tests whether the Alpha lock is set- returns a value
- BYE- Now legal within a running programme
- CHAR ALL- For controlling the character set
- CHIMES- Sounds a chime
- CLRS- Clears only the text area (columns 3-30)
- COLOR- Now accepts the colour set "All"
- FINd- Will find the specified string within an array of strings
- GPEEK- GPI Peek command
- GPOKE- GPI Poke command
- HONK- Produces a "honk" sound
- KEYS(L\$,P)- Waits for the keys specified in the string I\$ and returns the value P with the keys position in the string
- MLOAD- Loads a memory-image file into the specified memory location
- MOVE- Move ranges of memory between RAM, VRAM, ROM and GROM
- MSAVE- Save a portion of memory to disk in memory-image file format
- NEW- Now legal within a programme
- PRNTPAT- Print a character definition to an Epson compatible printer

QUIT OFF/ON- Turn on/off the QUIT key

VPEEK- VDP RAM peek

VPOKE- VDP RAM poke

WAIT- Pause programme for a specified time

* A new range of functions have also been added:

- ASC- Will handle an empty string without crashing a running programme
- CLOSE- Will close all open files at once
- DEF- All user defined functions may be used outside of a running programme at the command line
- DATE\$- Returns the current date with a clock card
- HEX\$- Returns the Hexadecimal equivalent of a decimal number
- LWRC\$- Converts a string to lower case
- TIME\$- Returns the current time with a clock card
- UPRC\$- Converts a string to upper case
- VAL- Will now convert a hex string into a numeric string

* Virtually all Maths functions and floating-point Maths routines (COS, SIN, ect.) have been re-written and are typically 100% to 200% faster. Extensions have been made to the RND function to allow ranges and more randomisation

* A range of new commands available at the command line including:

* Displays a catalog of the device in quotes after the function

APPEND- Defines all Control characters in inverse video
 ERASE- Allows you to delete a range of lines
 OUTPUT- Saves the range of line numbers specified to disk in Merge format
 PERMANENT- Allows you to turn off or on the true-lower-case character set as well as de-activate lower case letters
 SAVE- The "Protected" option now truly protects a programme from being listed
 SIZE- Displays the amount of Low Memory space available as well as other space
 USING- Lists all CALL statements in a programme
 VARIABLE- Lists all variables in a programme

While many of the features and all of the speed enhancements in Extended BASIC III are unique to this cartridge, it is compatible to some degree with MECHATRONICS EXTENDED BASIC.

EXTENDED BASIC III is currently available in disk form for those users with a Mechatronics GRAM-KARTE for \$39.95.

A 96K cartridge version that requires only a console and 32K memory expansion is expected to be available by June 15, 1993 for \$74.95.

Finally, available by April 15 1993, the XBIII SUPER MODULE. This 192K module has built into it:

Extended BASIC III
 The TE2 cartridge- allowing you to perform true text-to-speech in XB3 WITHOUT any additional software(!)
 The Editor/Assembler cartridge
 TI/Writer, it's Editor and Formatter
 The Remind-Me! appointment programme
 Mass-Transfer
 A full disk manager
 Various other miscellaneous utilities

For a list price of \$99.95. This cartridge is an entire productivity package all-in-one module-plug it in and you may never have to unplug it again. This version requires 32K and a disk system.

Send all enquiries to:

ASGARD SOFTWARE
 1423 Flagship Dr.
 Woodbridge, VA 22192

END OF ARTICLE

CHAOS and GRAPHICS

by Joe Jacobson
 adapted for Extended Basic & TML
 by Stephen Shaw

Yet another odd magazine to come my way is THE JOURNAL OF CHAOS AND GRAPHICS, edited by Clifford A Pickover. Sample copy available from:

Clifford A Pickover,
 Editor, Journal of Chaos and Graphics,
 IBM T.J. Watson Research Center
 Yorktown Heights, NY. USA. 10598.

This program below is from Volume 3, August 1988 issue, and produces a different sort of pattern. It is an opportunity for me to show you how to use the MOD function on the TI- instead of A=MOD(B,C) on some other computer, I have used a subprogram and the form CALL MOD(B,C,A).

Also I have used the DEF function to prepare COS and SIN functions COSD and SIND which expect to find degree measures instead of the usual radians. It is a good way of finding out how efficient DEF is too- just try replacing the SIND and COSD functions and replacing them with the ordinary COS and SIN functions but amend the variable in accordance with the DEF - eg instead of COS(A) use COS(A/180*PI) in every SIND and COSD line. Any faster?

```

1 REM ANALYTIC COMPUTER ART
2 ! JOE JACOBSON
3 ! APT 1009 675 E STREET RD
4 ! WARMINSTER PA USA 18974
5 ! fluted scallops
6 ! The Journal of Chaos & Graphics August 1988
7 ! For TI99/4A plus ExBas + The Missing Link by
  Stephen Shaw UK Jan '91
8 !
100 DEF SIND(X)=SIN(X/180*PI)
110 DEF COSD(X)=COS(X/180*PI)
120 OS=99 :: RANDOMIZE
130 L=INT(RND)+3
140 CALL LINK("PRINT",180,220,STR$(L))
150 FOR B=16 TO 76 STEP 2
160 FOR A=0 TO 360 STEP 5
170 GOSUB 250
180 IF A>0 THEN 210
190 CALL LINK("PIXEL",OS+X,Y+OS):: OLDX=X :: OLDY=Y
200 GOTO 220
210 CALL LINK("LINE",OS+OLDX,OLDY+OS,OS+X,OS+Y):: OLDX=X
  :: OLDY=Y
220 NEXT A
230 NEXT B
240 GOTO 290
250 R=B*(1+.25*ABS(SIND(L*A)))! this is the function
  plotted:R=f(x):=polar coordinate curve.
260 X=R*COSD(A)
270 Y=R*SIND(A)
280 RETURN
290 CALL LINK("PRINT",180,20,"+")
300 CALL LINK("DUMP")
305 CALL LINK("CLEAR")
310 RUN

```

END OF ARTICLE

FORTH and FILES

by Geoff Trott

Last month in Techo Time, I started to write about transferring Forth system disks to larger formats like double sided double density. I am not sure that I did the topic justice in that short space, so I thought I would start again with an article just on that subject and perhaps complete the topic a bit better. I will repeat some of the information from the previous article so that it is all together and suggest how to change the word which writes a Forth header onto a freshly formatted floppy disk, as well as how you could get Forth to run from a RAMdisk.

I was contemplating how to move the Forth system from the single sided single density floppies on which it came, to double sided double density floppies, which are the norm for most of us now, or indeed to any other size disk. The difficulty is that Forth has its own way of looking at the contents of a floppy disk and this is not necessarily compatible with the way a disk manager expects to see a disk. Apart from the original boot disk, there is no real need for a Forth disk to be recognisable as a formatted disk by disk managers, but it does help to avoid assuming that a Forth disk, which is full of precious words (programs), is empty and not formatted if it responds to the disk managers correctly. let me see if I can explain a bit better.

Forth uses blocks of data called screens, which are 1024 bytes long. Each screen is numbered and when screens are stored on a floppy disk they are put onto the disk in a physical position which depends on their number. This makes accessing of screens a very simple operation but it means that there is little help in finding out what is on a Forth disk and you must remember on which screens are the definitions of your favourite words and even which disks. There is no other way of doing this without running Forth itself and using one of its words to list the first line of each screen. In other words, Forth disks are not very compatible with normal disks used by the TI99/4A disk operating system. They do not have a recognisable file structure and are a bit like DOS disks in that respect.

Let us look at the details more closely. Our disk system divides the disk into sectors each 256 bytes long. Each screen takes up 4 sectors. Screen 0 is stored in sectors 0 to 3. Screen 1 is stored in sectors 4 to 7. Screen 89 is stored in sectors number 356 to 359. Screen 179 is stored in sectors number 716 to 719. Screen 359 is stored in sectors number 1436 to 1439. If you have a single sided single density system, you can only access screens 0 to 89. If you have two disk drives, you can access screens 90 to 179 on a second disk in DSK2. If you have a double sided drive but still in single density you can access screens 0 to 179 on the first drive and if you have a second drive, sectors 180 to 359 on the second drive, and so on. You can have more than one drive and make it part of the Forth system by adjusting the values in some Forth variables which I will talk about later.

Now the problem with this, as far as compatibility with the normal disk operating system is concerned, is that the first sector on each disk needs to have some information stored on it to make it into a recognisable formatted disk. This is called the Volume Information Block (VIB). Also, the second sector on each disk should contain a list of sector numbers (or 0) which point to the File Information Blocks (FIB) for each of the files on the disk. This means that the first screen on each Forth disk should contain information that the normal operating system can use to say that this is a Forth disk and has no free sectors for use by the operating system. The rest of the disk can then be used to store Forth screens. This is quite easy to do by formatting a disk to whatever size you want and giving it a name of FORTH say. Then, using a sector editor (Disk Utilities or DiskReview), set the contents of all the bytes from byte number >38 to >FF to be >FF. This will set all the bits in the bit-map to 1 which says that they are all used. When you catalogue the disk, it will have the name that you specified and the size that you specified but will have no files and no free sectors. A disk set up like this is suitable for storing your screens of programs and can be any size. You must just be careful not to use the first screen on the disk (screen 0 in DSK1 for example, or screen 360 in DSK2 of a DSDD system). It will not help with a boot disk however, as this needs a bit more effort.

If you catalogue the boot disk, you will see that it has 3 files; FORTH, FORTHSAVE and SYS-SCRNS. FORTH is a DF80 file which is the loader from Editor Assembler option 3, FORTHSAVE is a program file which contains the kernel of the Forth system and SYS-SCRNS is a dummy DF128 file to take up all the rest of the space on the disk so that there is no free space. For the original system, this file is 313 sectors long (1 for FIB, 312 of data). For a disk with 720 sectors, this file will need to be 673 (313 + 360) sectors long while for a disk with 1440 sectors the file will need to be 1393 sectors long. As well as getting the size of this dummy file correct, the sectors used in the original boot disk must be the same sectors used in any other disk as the screens must still be in their correct place on the disk. For example, as part of the boot process, screen 3 is loaded into Forth. This is done by reading in sectors 12 to 15 on the disk in DSK1 and these must contain the correct contents of screen 3. If the contents of a disk are

copied by a normal disk manager, it is going to put the files in particular areas and may change them around, particularly if the two disks are of a different size. To do a sector by sector copy requires that the destination disk is formatted the same as the source disk so that method cannot be used. The method I used is as follows.

First format a disk to the size that you want to use. Then change the bit-map on sector 0 to indicated that only the first 360 sectors are free (implying that the rest are bad sectors). Use any disk manager to copy all the files from the Forth boot disk to the new disk. This will set all the sectors as used. Set the size of the SYS-SCREENS file to the size for the size of disk you are using. This last step is not essential as the disk will be full without it, so that it will not be possible to copy anything to it or save anything to it using the normal disk operating system. Forth does not use the bit-map so it will be able to read and write to any screen. Then you will have a bootable disk of the size that you want to use.

The details for doing this are as follows. Take your freshly formatted disk and using a sector editor such as Disk Utilities or DiskReview edit sector 0. You should see the disk name and in sector >38 there will be >03. This is the start of the bit-map and is showing that sectors 0 and 1 are in use. The bit-map uses one bit for each sector so there are 45 bytes used for a SSSD disk (360 = 45 * 8). Move to byte >63 and put >FF in that byte and all bytes to the end of the sector. Write that out to the disk and then catalogue the disk again. You should find that there are now 358 sectors free. If not, check sector 0 again and make sure that you have 44 bytes containing >00 from byte >39 to byte >62 inclusive and the remainder contain >FF. Then you can copy, file by file, the three files of the system disk to the new disk. Since the free space of the new disk is the same as that of the system disk, the files will be put in the correct sectors and the system will boot from the new disk.

The new disk will work without problems as it is but if you are concerned that it looks a bit funny to have a disk with 1440 sectors available and yet is full with only 358 sectors used by files, you can adjust the size of the dummy file to make it look like it fills the whole disk. One advantage of doing this is that you can then copy the disk easily, without any fiddling, although it would probably be quicker to use a disk copy. The changes that need to be made are on sector 4, which is the File Information Block (FIB) for SYS-SCRNS. If you edit sector 4, you will see the file name and other data related to the size and position of the file on the disk. The information that needs to be changed is the file length in bytes 14, 15, number of records in file in bytes 18, 19, size of first segment in bytes 29, 30 and size of the second segment in bytes 32, 33. The following table provides the values for different disk sizes.

Disk size\ Byte	>0E>0F	>12>13	>1C>1D	>1E>1F	>20>21
SSSD	01 38	70 02	4D A0	11 05	70 13
SSDD, DSSD	02 A0	40 05	4D 20	28 05	FO 29
DSDD	05 70	E0 0A	4D 20	55 05	FO 56

You must then change the variables DISK_SIZE and DISK_HI to suit your new disk size and put these changes on screen 3 so they are set on boot up. For example:
 90 DISK_SIZE ! 180 DISK_HI ! (for 2 SSSD drives)
 180 DISK_SIZE ! 180 DISK_HI ! (for 1 DSSD or SSDD drive)
 360 DISK_SIZE ! 1080 DISK_HI ! (for 3 DSDD drives)

There are some changes required in the word DISK-HEAD if you want to use it to put a header on your disks rather than using the method outlined above. These involve setting up sector 0 to look like a VIB for the size of disk you are using, setting up sector 1 to point to one FIB on sector 2 and setting up sector 2 to be a FIB for a file which fills the entire disk. The word which came on the original system disk is not correct but works with the TI disk controller card. It

```

SCR #40
( Write a header for SSSD disks 24JUL93 GWT)
BASE->R HEX ( save original base and change to HEX)
: DISK-HEAD 0 CLEAR 0 BLOCK ( start sector 0)
  DUP !" FORTH      " DUP A + 168 SWAP !
  DUP C + 944 SWAP ! DUP D + !" DSK"
  DUP 10 + 2028 SWAP ! DUP 12 + 101 SWAP !
  DUP 14 + 24 0 FILL DUP 38 + C8 FF FILL ( sector 1)
  100 + DUP 2 SWAP ! DUP 2+ FE 0 FILL ( start sector 2)
  100 + DUP !" SCREENS  " DUP A + 0 SWAP !
  DUP C + 2 SWAP ! DUP E + 165 SWAP !
  DUP 10 + 80 SWAP ! DUP 12 + CA02 SWAP !
  DUP 14 + 8 0 FILL DUP 1C + 2250 SWAP !
  DUP 1E + 1403 SWAP ! DUP 20 + 4016 SWAP !
  22 + DE 0 FILL FLUSH ( write buffer to disk)
; ( end of word definition)
R->BASE ( restore base to original)

```

This puts the name of FORTH as the disk name and sets up a SSSD disk with 360 sectors (>168). The setting up is done by storing bytes and words in a buffer which is first cleared at the start of the definition. The storage parts start with a DUP function, which puts another copy of the address of the buffer on the stack. The first of these DUP words is followed by a word to store a string (!") and the string is a ten character string starting with "FORTH" and finishing with spaces. The next DUP is followed by code to add 10 to the address (>A) and then store the number of sectors (360 = >168). Then the number of sectors per track (9) is stored followed by the string "DSK" which is what the disk operating system uses to confirm that the disk is formatted for use by the TI99/4A. In the byte at address 16 (>10) in the buffer is stored the code for an unprotected disk (>20, it would be >50 for a protected disk), while in address 17 is stored 40 (>28), which is the number of tracks on a side of the disk. The next byte contains the number of sides on this disk and the following byte contains the density code (1 for single density and 2 for double density). The next 36 (>24) bytes are filled with zeros (not used except for time and date information by Disk Utilities and subdirectory information by Myarc HFDCC) and the remainder of the sector is filled with >FF to set all bits in the bit-map to 1 showing that the disk is full.

Sector 1 information is entered in a similar way by first adding 256 (>100) to the buffer address and then storing the sector address of 2 in the first two bytes and zeros in the remaining 254 (>FE) bytes of the sector. Sector 2 information is set up also by first adding 256 to the buffer address and then starting to store the information. This is a bit more complicated than the previous two sectors, but only in interpretation of the data. First a 10 character file name is stored (SCREENS). At bytes 10 and 11 on sector 2, zero is stored followed by zero in byte 12 (>C) which signifies a display fixed data file while the 2 in byte 13 means two records each sector. Bytes 14 and 15 contain 357 (>165) which is the number of sectors in the file and is 3 less than the size of the disk and one less than the size of the file as given in a catalogue of the disk. Byte 16 (>10) is zero and byte 17 is 128 (>80), so the file type is display fixed 128. Bytes 18 and 19 contain the total number of records in the file which should be 714 (2 * 357 which is >2CA) but in reverse byte order so >CA02 is actually stored. The next 8 bytes are not used except to put time and date information by the Myarc disk controllers and then starts the file segment entries.

This file will have two segments if copied by a disk manager program. The first segment will be from sector 34 (>22) to sector 359 (the last segment in the disk) while the second segment will start from sector 3 to sector 33. The first segment is 326 (>146) sectors long and the second segment is a further 31 (>1F) sectors long. The segments entries are then arranged as >22 50 14 03 40 16, which is saying that the first segment starts at sector >022 and the last sector of the first segment is at >022 + >145 while the second segment

starts at sector >003 and the last sector in this segment is at >003 + >164 - >145 - 1. The pattern of data in the segment entry table is that you take the bytes 3 at a time. Split the middle byte into two digits and put the right digit to the left of the first byte and the left digit to the right of the third byte to make two 3 digit numbers and these are the numbers that define the segments. The first three digit number is the first sector of the segment and the second three digit number is one less than the length of the first segment. If there is more than one segment, the second 3 bytes are treated in a similar manner to get another two, three digit numbers. The first of these is the first sector of the segment, as before, and the length of this and subsequent segments is the difference between the second three digit number and that of the second three digit number from the previous segment.

As you can see, the bytes shown in the paragraph above are stored from byte 28 (>1C) to byte 33 (>21). The rest of the sector is filled with zeros and the buffer is written to the disk. To get the header correct for larger disks, a number of changes are required in this function. I will list the headers for DSSD, SSDD and DSDD for you to compare.

```

SCR #40
( Write a header for DSSD disks 24JUL93 GWT)
BASE->R HEX ( save original base and change to HEX)
: DISK-HEAD 0 CLEAR 0 BLOCK ( start sector 0)
  DUP !" FORTH      " DUP A + 2D0 SWAP !
  DUP C + 944 SWAP ! DUP D + !" DSK"
  DUP 10 + 2028 SWAP ! DUP 12 + 201 SWAP !
  DUP 14 + 24 0 FILL DUP 38 + C8 FF FILL ( sector 1)
  100 + DUP 2 SWAP ! DUP 2+ FE 0 FILL ( start sector 2)
  100 + DUP !" SCREENS  " DUP A + 0 SWAP !
  DUP C + 2 SWAP ! DUP E + 2CD SWAP !
  DUP 10 + 80 SWAP ! DUP 12 + 9A05 SWAP !
  DUP 14 + 8 0 FILL DUP 1C + 22D0 SWAP !
  DUP 1E + 2A03 SWAP ! DUP 20 + C02C SWAP !
  22 + DE 0 FILL FLUSH ( write buffer to disk)
; ( end of word definition)
R->BASE ( restore base to original)

```

There is not much difference between a single density double sided disk and a double density single sided disk. They have the same size and only vary in bytes 18 and 19 on sector 0. Here is the whole word, just in case.

```

SCR #40
( Write a header for SSDD disks 24JUL93 GWT)
BASE->R HEX ( save original base and change to HEX)
: DISK-HEAD 0 CLEAR 0 BLOCK ( start sector 0)
  DUP !" FORTH      " DUP A + 2D0 SWAP !
  DUP C + 1244 SWAP ! DUP D + !" DSK"
  DUP 10 + 2028 SWAP ! DUP 12 + 102 SWAP !
  DUP 14 + 24 0 FILL DUP 38 + C8 FF FILL ( sector 1)
  100 + DUP 2 SWAP ! DUP 2+ FE 0 FILL ( start sector 2)
  100 + DUP !" SCREENS  " DUP A + 0 SWAP !
  DUP C + 2 SWAP ! DUP E + 2CD SWAP !
  DUP 10 + 80 SWAP ! DUP 12 + 9A05 SWAP !
  DUP 14 + 8 0 FILL DUP 1C + 22D0 SWAP !
  DUP 1E + 2A03 SWAP ! DUP 20 + C02C SWAP !
  22 + DE 0 FILL FLUSH ( write buffer to disk)
; ( end of word definition)
R->BASE ( restore base to original)

```

```

SCR #40
( Write a header for DSDD disks 24JUL93 GWT)
BASE->R HEX ( save original base and change to HEX)
: DISK-HEAD 0 CLEAR 0 BLOCK ( start sector 0)
  DUP !" FORTH      " DUP A + 5A0 SWAP !
  DUP C + 1244 SWAP ! DUP D + !" DSK"
  DUP 10 + 2028 SWAP ! DUP 12 + 202 SWAP !
  DUP 14 + 24 0 FILL DUP 38 + C8 FF FILL ( sector 1)
  100 + DUP 2 SWAP ! DUP 2+ FE 0 FILL ( start sector 2)
  100 + DUP !" SCREENS  " DUP A + 0 SWAP !
  DUP C + 2 SWAP ! DUP E + 59D SWAP !
  DUP 10 + 80 SWAP ! DUP 12 + 3A0B SWAP !
  DUP 14 + 8 0 FILL DUP 1C + 22D0 SWAP !
  DUP 1E + 5703 SWAP ! DUP 20 + C059 SWAP !
  22 + DE 0 FILL FLUSH ( write buffer to disk)
; ( end of word definition)
R->BASE ( restore base to original)

```

Do you want to run Forth from a RAMdisk? This would give a very fast load but it does have a few problems. The first problem is that Forth requires its booting disk to be DSK1 (unless that part of Forth is re-written) and it goes to sector 12 to find screen 3 for the booting procedure. So the first thing you must do is to change your RAMdisk to be DSK1. This is going to make physical drive DSK1 not usable. You will also have to use CRU address of >1000 to make sure that the RAMdisk DSK1 is the one that gets used. The second problem relates to the fact that the sectors are used in a different order on a RAMdisk compared to a floppy disk. Because there is no time penalty in using any sector on a RAMdisk, the ROS simply puts the FIB of each file in the next available sector and the data of the file follows immediately after that. That means that there is no gap at the start of the disk and that there will be no fragmented file. This means that you cannot use a file copy to move Forth from a floppy disk to the RAMdisk as this would mess up the position of the screen information, but you must use a sector or disk copy. This means that the RAMdisk and the floppy disk must have the same size. If you try it and get it working, I would be most interested in hearing about it.

END OF ARTICLE

BULL RUN

by Stephen Shaw

A program loosely based on a simple strategic war simulation by Tim Hartnell, converted to a one player TI game by Stephen Shaw, 1990.

This strategic game is played out on a 6x6 grid, with rows identified by the letters A/F and columns identified by the number 1 to 6.

At game start the human player allocates a total of 100 men around the 36 squares which may have from 0 to 9 men in each. Once the human player has allocated all his men the computer will allocate his (without referring to the human placement).

The computer, having the handicap of having no brains! will always go first.

Each player in turn selects one square as the centre of a battle, which will rage over all adjacent squares- a maximum of nine squares in the centre, or if the battle is centred on one of the four corner squares, only four squares are involved. Six squares are involved for any other edge square.

The total men for each player is calculated for all these active squares, and the winner has the most! The loser sees all his men in those active squares wiped out.

Play terminates if one player has more than double the number of men left than his opponent, or after 18 battles. The winner controls the most squares at game end, and could do so with less men than the loser has.

It is easy to play for a draw on territory, so as a penalty, in the event of each player controlling the same territory as the other the game is awarded to the computer.

Your playing strategy takes two parts- what is the best arrangement of your 100 men to tackle any computer play? And what is the best pattern of battle- to take territory or to defend territory or to grab the most computer men?

There is a winning strategy which will usually beat the computer. How fast can you find it?

As written winning can be difficult! Various adjustments can be made to make the computer play more intelligently! or less so. I shall leave that to you dear reader! Enjoy.

Basic programmers may wish to note how the board is quickly placed on screen, and how player input is handled. Note the use of RND*RND as the computer places its men- can you see why? You can manipulate distribution patterns by using more than one RND, eg RND*RND or (RND+RND)/2 or even (RND*RND+RND)/2!!!

```

100 REM
110 REM based on FIRST BULL RUN by Tim Hartnell 1984
120 REM for TI99/4A by s shaw october 1990
130 REM winner has most territory at end not most troops
140 REM
150 REM JUST NEEDS EX BAS!
160 REM
170 REM PLACE UP TO 9 TROOP IN EACH TERRITORY
TROOPS IN BLOCK OF ARE COUNTED FOR RESULT
180 REM
190 CALL CLEAR :: GOSUB 290 :: T=0 :: RANDOMIZE
200 GOSUB 350
210 GOSUB 610 :: IF FLAG=1 THEN 1110
220 GOSUB 820
230 GOSUB 610 :: IF FLAG=1 THEN 1110
240 GOSUB 730
250 BAT1=BAT1+1 :: IF BAT1=9 THEN 1110 ELSE 210
260 ,5
270 REM HUMAN SELECTS
280 REM
290 REM SETUP
300 CALL COLOR(11,15,15,12,16,16)
310 A$="ppppzzzzppppzzzzppppzzzz" :: B$="zzzzppppzzzzpp
ppzzzzpppp"
320 FOR T=1 TO 14 STEP 6 :: DISPLAY AT(T,1):A$:A$:A$:B$
:B$:B$ :: NEXT T
330 FOR T=1 TO 6 :: DISPLAY AT(T*3-1,25):CHR$(64+T)::
DISPLAY AT(19,T*4-2):CHR$(
48+T):: NEXT T
340 RETURN
350 REM PLACE
360 DISPLAY AT(20,1):"YOU MUST PLACE 100 MEN" :: TOT=100
370 DISPLAY AT(21,1):"ROW COL MEN"
380 CALL HCHAR(21,7,32):: CALL KEY(5,A,B):: CALL HCHAR
(21,7,30):: IF A<65 OR A>70 THEN 380 ELSE ROW=A-64
:: CALL HCHAR(21,7,A)
390 CALL HCHAR(21,15,32):: CALL KEY(5,A,B):: CALL HCHAR
(21,15,30):: IF A<49 OR >54 THEN 390 ELSE COL=A-48 ::
CALL HCHAR(21,15,A)
400 CALL KEY(5,A,B):: IF A>20 THEN 400
410 CALL HCHAR(21,23,32):: CALL KEY(0,A,B):: CALL HCHAR
(21,23,30):: IF A<48 OR >57 THEN 410 ELSE MEN=A-48
:: CALL HCHAR(21,23,A)
420 IF ARRAY(ROW+1,COL+1)>0 THEN TOT=TOT+ARRAY(ROW+1,COL
+1):: ARRAY(ROW+1,COL+1)=0
430 IF TOT-MEN<0 THEN 410
440 DISPLAY AT(22,1):TOT-MEN;" LEFT TO PLACE" :: TOT=TOT
-MEN :: ARRAY(ROW+1,COL+1)=MEN
450 CALL HCHAR(ROW*3-1,COL*4,A)
460 IF TOT=0 THEN 480
470 GOTO 370
480 COMP=100 :: FOR ROW=1 TO 6
490 FOR COL=1 TO 6
500 A=INT(RNDRND+1):: IF COMP-A<0 THEN 500
510 COMP=COMP-A :: CALL HCHAR(ROW*3-1,COL*4+2,A+48)::
DISPLAY AT(24,1):"COMPUTER HAS ";COMP;" LEFT"
520 GRID(ROW+1,COL+1)=A :: IF COMP=0 THEN 590
530 NEXT COL
540 NEXT ROW
550 IF COMP=0 THEN 590
560 FOR COL=7 TO 2 STEP -1 :: FOR ROW=2 TO 7 :: IF GRID
(ROW,COL)<9 THEN GRID(ROW,COL)=GRID(ROW,COL)+1 ::
COMP=COMP-1
570 CALL HCHAR((ROW-1)*3-1,(COL-1)*4+2,48+GRID(ROW,COL))
:: IF COMP=0 THEN 590
580 NEXT ROW :: NEXT COL
590 CALL HCHAR(20,1,32,160)
600 RETURN
610 REM CHECK END?
620 YOURTOT,COMPTOT,FLAG=0
630 FOR ROW=2 TO 7
640 FOR COL=2 TO 7
650 YOURTOT=YOURTOT+ARRAY(ROW,COL)
660 COMPTOT=COMPTOT+GRID(ROW,COL)
670 NEXT COL :: NEXT ROW
680 IF YOURTOT<16 OR COMPTOT<16 THEN FLAG=1
690 IF YOURTOT<COMPTOT/2 THEN FLAG=1
700 IF COMPTOT<YOURTOT/2 THEN FLAG=1
710 DISPLAY AT(21,1):" ":YOURTOT;" COMP=";
COMPTOT :: CALL DELAY
720 RETURN
730 REM HUMAN SELECTS

```

AIDKEY

(c) By Arto Heino 1987

If you get a copy of this somehow
please send the Author \$10.

ARTO HEINO
TH 35/8 GUERNSEY AVE.,
MINTO , 2566 ,
N.S.W. , AUSTRALIA

* REQUIREMENTS-XB/32K/DISK *
* * *

CALL INIT
CALL LOAD("DSK1,AIDKEY")
CALL LINK("AIDKEY")

The cursor should now be changed, this means Aidkey
is now ready when you need it.

USING AIDKEY

This program is intended for the XB programmer who
needs AID!!

LOADING AIDKEY

You can initiate Aidkey while:

IN COMMAND MODE
IN EDIT MODE
LISTING TO SCREEN
PROGRAM IS RUNNING

Just Press FUNCTION "7"

CAUTION

DO NOT CREATE A CHAR WHILE YOUR PROGRAM IS RUNNING!

IF YOU USE THE CHARS 48-90 IN YOUR PROGRAM, YOU
WILL NOT BE ABLE TO READ THE MENU.

MAKE SURE YOU PUT THE CORRECT DEVICE NAME.

AIDKEY MENU

The Menu is actually a WINDOW you can move around
using FUNC-D,S,X,E

1. CREATING CHAR

This function will put a HEX string 16 chars long
after the cursor in EDIT MODE. To move char window
around use FUNC-D,S,X,E To move cursor use D,S,X,E To
change ON/OFF press SPACE BAR Press ENTER to return to
menu

2. RLE/BIT SCREEN DUMP

This function is useful only if you have the
MAX/RLE disk. It will do a bit screen dump so you can
use your Basic screen with GRAPHX OR TI-ARTIST OR as a
file for another computer.

3. PRINTER SCREEN DUMP

This will bit dump your screen to a EPSON
compatible printer.

4. TURN OFF AIDKEY

It will return your cursor to normal and FUNC-7
will be inactive. To activate TYPE IN COMMAND MODE CALL
LINK("AIDKEY")

5. RETURN TO BASIC

This will return you to where ever you were before
you pressed AID. If you created a char your string of
HEX will be printed on your line.

```
740 DISPLAY AT(22,1):"BATTLE CENTRE: ROW COL": "" ""
750 CALL HCHAR(22,21,30):: CALL KEY(5,A,B):: CALL HCHAR
(22,21,32):: IF A<65 OR >70 THEN 750 ELSE ROW=A-64
:: CALL HCHAR(22,21,A)
760 CALL HCHAR(22,27,30):: CALL KEY(5,A,B):: CALL HCHAR
(22,27,32):: IF A<49 OR >54 THEN 760 ELSE COL=A-48
:: CALL HCHAR(22,27,A)
770 ROW=ROW+1 :: COL=COL+1
780 IF BATTLE(ROW,COL)=1 THEN DISPLAY AT(21,1):"FOUGHT
THERE ALREADY" :: CALL DELAY :: GOTO 740
790 BATTLE(ROW,COL)=1
800 GOSUB 930
810 RETURN
820 REM COMP SELECTS
830 FOR A=1 TO 16 :: ROW=INT(RND*4+3):: COL=INT(RND*4+3)
:: GOSUB 1210
840 IF CT>UT AND BATTLE(ROW,COL)=0 AND UT>3 THEN 900
850 NEXT A
860 FOR ROW=7 TO 2 STEP -1 :: FOR COL=2 TO 7 :: GOSUB
1210
870 IF CT>UT AND UT>0 AND BATTLE(ROW,COL)=0 THEN 900
880 NEXT COL :: NEXT ROW
890 ROW=INT(RND*6+2):: COL=INT(RND*6+2):: IF BATTLE(ROW
,COL)=1 THEN 890
900 DISPLAY AT(22,1):"COMPUTER FIGHTS AT ";CHR$(ROW+63)
;COL-1:" " " "
910 IF BATTLE(ROW,COL)=1 THEN 820
920 BATTLE(ROW,COL)=1 :: CALL DELAY
930 REM LETS COUNT
940 GOSUB 1200
950 DISPLAY AT(21,1):" " "YOU:";UT;" COMPUTER:";CT
960 CALL DELAY
970 IF CT=UT THEN DISPLAY AT(23,1):"-NO BATTLE-" ::
GOTO 1100
980 IF UT>CT THEN DISPLAY AT(23,1):"YOU WIN" ELSE
DISPLAY AT(23,1):"YOU LOSE"
990 IF UT>CT THEN GRID(ROW-1,COL-1),GRID(ROW-1,COL),GRID
(ROW-1,COL+1),GRID(ROW,COL-1),GRID(ROW,COL+1),GRID
(ROW+1,COL),GRID(ROW+1,COL-1)=0
1000 IF UT>CT THEN GRID(ROW,COL),GRID(ROW+1,COL+1)=0
1010 IF CT>UT THEN ARRAY(ROW-1,COL-1),ARRAY(ROW-1,COL),
ARRAY(ROW-1,COL+1),ARRAY(ROW,COL-1),ARRAY
(ROW,COL+1),ARRAY(ROW+1,COL+1)=0
1020 IF CT>UT THEN ARRAY(ROW,COL),ARRAY(ROW+1,COL-1),
ARRAY(ROW+1,COL)=0
1030 CALL DELAY
1040 FOR ROW=2 TO 7 :: FOR COL=2 TO 7
1050 CALL HCHAR((ROW-1)*3-1,(COL-1)*4,ARRAY(ROW,COL)+48)
1060 CALL HCHAR((ROW-1)*3-1,(COL-1)*4+2,GRID(ROW,COL)+
48)
1070 NEXT COL :: NEXT ROW
1080 DISPLAY AT(22,1):"CALCULATED NEW TROOPS": "" :: CALL
DELAY :: CALL HCHAR(22,1,32,32)
1090 RETURN
1100 CALL DELAY :: RETURN
1110 REM FINAL PRINT OUT
1120 DISPLAY AT(21,1):"";"FINAL TROOPS:";"YOU:";YOURTOT
;" COMPUTER";COMPTOT
1130 FOR ROW=2 TO 7 :: FOR COL=2 TO 7
1140 IF ARRAY(ROW,COL)=0 THEN CW=CW+1
1150 IF GRID(ROW,COL)=0 THEN UW=UW+1
1160 NEXT COL :: NEXT ROW
1170 IF UW>CW THEN DISPLAY AT(24,1):"YOU WON MOST
TERRITORY" ELSE DISPLAY AT(24,1):"COMPUTER WON MOST
TERRITORY"
1180 DISPLAY AT(20,1):"PRESS ANY KEY TO RE-RUN" :: CALL
KEY(5,A,B):: DISPLAY AT(20,1):"PRESS ___ KEY TO
RE-RUN" :: IF B<1 THEN 1180 ELSE RUN
1190 CALL KEY(5,A,B):: IF B>0 THEN RUN
1200 REM CHECK COMP CHOICE
1210 UT,CT=0
1220 CT=GRID(ROW-1,COL-1)+GRID(ROW-1,COL)+GRID(ROW-1,COL
+1)+GRID(ROW,COL-1)+GRID(ROW,COL)+GRID(ROW,COL+1)+
GRID(ROW+1,COL+1)+GRID(ROW+1,COL)
1230 CT=CT+GRID(ROW+1,COL-1)
1240 UT=ARRAY(ROW-1,COL-1)+ ARRAY(ROW-1,COL)+ARRAY(ROW-1,
COL+1)+ARRAY(ROW,COL-1)
1250 UT=UT+ARRAY(ROW,COL)+ARRAY(ROW,COL+1)+ARRAY(ROW+1,
COL+1)+ARRAY(ROW+1,COL)+ARRAY(ROW+1,COL-1)
1260 RETURN
1270 SUB DELAY :: FOR T=1 TO 700 :: NEXT T :: SUBEND
```

END OF ARTICLE

END OF ARTICLE

NEW

SECTION for IBM COMPATABLES

All the following articles are reprinted from
"YOUR COMPUTER'S TECH TIPS"
with their permission.

Hard disk maintenance

In Bruce Illiff's 'PC Maintenance' in June, he advises re-formatting a hard disk every six months 'to keep the disk in working condition'. If re-formatting is not carried out, how long can a hard disk be expected to continue reliable operation? If re-formatting is required, shouldn't a low-level re-format be required, rather than just a logical re-format? Digging for information on computers always uncovers more new questions than answers for old. Keep up the good work.

*Ross Byrne
Albany Creek, 21d*

It's true that it is a good idea to re-format your hard disk occasionally, although the reason given in the aforementioned article could be a bit misleading. The physical deterioration of the magnetic coating on the surface of the disk is not the real problem — it happens so slowly that other sources of wear tend to overtake its effects. However, there is a good case for periodically re-formatting a hard disk, to make sure that all the data on the tracks are aligned correctly.

When a hard disk warms up due to use, it starts to expand — the disk platters grow a little, and the head actuator arms get a bit longer. This causes the sectors written to a cold disk to be slightly out of line with those written when the drive is warm. That's why cooling is important — the hotter the drive gets, the more out-of-line the sectors become. Each track eventually starts to look like a higgledy-piggledy mess, and with normal wear and tear on the head actuator, the drive may begin to have trouble aligning the heads with some of the disk sectors.

Voice coil drives

I own an AT clone which includes a voice coil hard disk. I opted for this type of drive because of the extra speed over the stepper motor type. However, some people have told me that they are better than stepper motor drives because they automatically park the heads between read/write operations and when the power is turned off. Is this correct, and are there any other advantages of voice coil drives over steppers?

*Nick Trent
Wford, NSW*

The reason that most voice coil drives can automatically park their heads while stepper motor drives cannot, is related to the mechanical operation of the head actuators in the two types of drives. In the latter, the heads are moved across the disk surface one track at a time by a 'stepper motor', which is so designed that, when pulses are applied to its terminals in the correct sequence, the shaft of the motor rotates a pre-determined amount. So, for the heads in the drive to move inward four tracks, four pulses are applied to the motor; each pulse stepping the head inward one track. Because the amount through which the stepper motor moves when pulsed is known, the position of the heads can be determined by simply adding the steps applied to the motor in the inward direction, and subtracting the steps applied in the other direction.

A voice coil motor, on the other hand, is a type of linear motor, not unlike the voice coil in a loudspeaker — hence the name. This type of drive is an analog system — it is theoretically possible to move the heads from track zero to the

innermost track of the drive in a single action, rather than a large number of discrete steps. Thus, in a voice coil drive, it is a relatively simple matter to include a circuit which detects an oncoming power loss, and moves the heads to their landing zone before the power fails entirely. The power rails from the power supply take a finite time to collapse, thanks to the charge stored in the capacitors in the power supply — more than enough to move the heads of a disk drive to their landing zone.

However, such drives do not usually park the heads between read and write operations — to do so would seriously degrade the access time of the drive. The exception to this is the drives used in some laptops, which park the heads after a pre-determined period of inactivity, to protect them against the occasional jolt that is part and parcel of a laptop's lot. Some machines even power the drive down after 30 seconds or so, to save power, so of course, the heads are parked then.

You can get auto-parking utilities for stepper motor drives, so that if the power fails before you parked the heads, there is a good chance that they will already have been parked, provided you haven't used the hard disk recently. For a voice coil drive, there probably isn't any point, unless it is an odd one which isn't auto-parking.

END OF ARTICLE

IDE Drives

I am considering upgrading the hard disk in my 12MHz AT. While I know all about MFM, RLL, ESDI and SCSI, I am confused by these new drives on the market which are referred to as 'AT-bus' or similar. Are these like hard cards, which plug directly into the expansion bus of the computer?

*N. Wright
Abbeey, WA*

No, they're not hard cards, at least not physically. However, they are similar in that they connect directly to the expansion bus of the computer. But, they are physically mounted in a drive bay, as for any other drive, for mechanical stability, and connected by a 40-way cable. They are similar in concept to ESDI and SCSI drives in that they have the controller mounted on the drive. While SCSI and ESDI drives only have part of the controller built into the drive (and thus still require a controller card), AT drives have the entire controller built into the drive, and for that reason are also commonly known as integrated drive electronics or IDE drives.

Many computers have an IDE connector on the motherboard, or an interface card can be supplied with the drive. The IDE interface is basically an extension of the relevant lines of the AT's I/O bus. This means that the drive doesn't have to conform to any of the existing drive-interface standards since the drive and controller are one unit.

If you want to use an IDE drive on a particular machine, check that the BIOS supports the IDE drives. If it is an old BIOS, it probably won't. Ask your dealer about an upgrade to a version which does support IDE drives, or has a user-definable drive selection (recent AMI and Phoenix BIOS' have this feature). This allows the specific parameters

— number of cylinders, heads, and sectors, and the write-precompensation value — to be entered for drives which are not supported by any of the pre-defined selections.

User-definable drive types avoid the need for driver software, such as Disk Manager or SpeedStor. If you can't upgrade the BIOS, try getting one of these programs from the dealer whom you bought the drive from. This is not as elegant a solution as upgrading the BIOS, but is easier to implement (especially if you don't feel like pulling chips out).

Monochrome and games

I own an XT with a Thomson TTL monochrome monitor and wish to run a number of games programs. As yet, I have not purchased a graphics card, as I do not know which type of card is best for my application. Can you help?

*David Fix
Mangerton, NSW*

The card which is best for you depends on the type of monochrome monitor your Thomson is. There are three major types of monochrome monitors in the PC arena at the moment — the basic MDA- and Hercules-compatible TTL unit, dual-scan TTL monitors which are compatible with both the MDA mono and CGA colour standards, and the much newer monochrome analog VGA monitors. The latter is suitable only for VGA cards, and in that application can display 64 levels of grey. This mono VGA system is the way that the industry is moving at the moment, as it means that the same VGA standard is used for both mono and color applications, and an upgrade from mono to color is performed by simply changing the monitor.

not have dual-scan capability, then there are two options. One is to use a straight Hercules-compatible card, with some form of software emulation for programs which does not support the Hercules standard, such as some games. Problems will arise, however, with software which writes directly to the screen, or games which need to be booted rather than run from the Dos prompt.

The other possibility is to use a card which connects to a standard MDA-compatible TTL monitor, but emulates both the Hercules and CGA cards. However, in either mode, the signal from the card's output port is compatible with the MDA monitor. Since the emulation is in hardware, it is compatible with auto-boot games and software which performs direct screen writes. Although we have all heard about such a card recently, we don't know who is distributing it. Can anyone out there help?

If your monitor is a dual-scan type, then an auto-switching card which supports both the CGA and Hercules standards will provide the solution. The monitor is able to detect which mode the card is currently operating in, and switch over as the card changes mode. Colors from the color card are displayed as shades of grey (or green or amber), and some foreground /background color combinations may be illegible. Since most early laptops emulated the CGA, a lot of software can be configured to provide a readable display on a monochrome screen.

If the monitor in question does

END OF ARTICLE 

Regional Group Reports

Meeting Summary For SEPTEMBER

Banana Coast	12/09/93	Sawtell
Central Coast	11/09/93	Saratoga
Glebe	09/09/93	Glebe
Hunter Valley	11/09/93	
Illawarra	14/09/93	Keiraville
Liverpool	10/09/93	Yagoona West
Northern Suburbs	23/09/93	
Sutherland	17/09/93	Jannali

BANANA COAST Regional Group (Coffs Harbour Environs)

We never miss meeting at Kerry Harrison's residence 15 Scarba St. Coffs Harbour, 2 pm second Sunday of the month. Visitors are most welcome. Contact Kerry 52 3736, Kevin 53 2649, Rex 51 2485 or John 54 1451.

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The meetings are usually held on the second Saturday of each month at members homes starting at 3:15 pm. Check the location with Geoff Phillips on (049) 428 176. Note that after 9:00 pm this number is used for the ZZAP BBS which includes TI-99 information. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Tuesday of each month after the TISHUG Sydney meeting (except January) at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Last month Geoff prepared an interesting lesson on the idea and use of the FORTH programming language. Contact Geoff Trott on (042) 29 6629 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home) After 9.30 PM or at work (02) 708-1987 Liquorland Yagoona for more information.

*** ALL WELCOME ***

10th September 1993*****
34 Colechin St * GOFER *
Yagoona West 2199 * and Utilities *

8th October 1993 *****
34 Colechin St * Pre Scanning *
Yagoona West 2199 * Programs *

Bye for now Larry.
Liverpool Regional Co-Ordinator

NORTHERN SUBURBS Regional Group
Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02) 918 8132. Come and join in our fun.
Dick Warburton.

SUTHERLAND Regional Group

The July meeting was well attended and covered a range of topics. Garry Wilson made a welcome return after a long absence.

The group gave Kevin Taylor some helpful hints with his latest ramdisk project, which is progressing well. Kevin is using eproms in addition to the static ram chips on his ramcard.

We also formatted and printed an article for TISHUG through TI Writer. Some time was later spent using Page Pro fonts from one of the recent shop disks. We later digressed into "Car" talk - filters, clutches and associated matters.

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young

TISHUG in Sydney

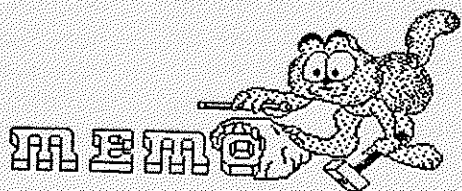
Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

SEPTEMBER MEETING - 4th SEPTEMBER

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

October	12th September
November	10th October

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.



**IT'S TIME
FOR MORE
TND ARTICLES**