



NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 14, Number 7

August, 1995

PRINT POST Approved - PP244099/00016



Sydney, New South Wales, Australia

\$3

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

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

Thomas Marshall (02) 871 7535

Treasurer

Cyril Bohlsen (02) 639 5847

Directors

Percy Harrison (02) 808 3181

Loren West (047) 21 3739

Sub-committees

News Digest Editor

Loren West (047) 21 3739

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

TI Merchandising

Dick Warburton (02) 918 8132

IBM Merchandising

Cyril Bohlsen (02) 639 5847

Software Library

Larry Saunders (02) 844 7377

Technical Co-ordinator

Geoff Trott (042) 29 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Glebe

Mike Slattery (02) 692 8162

Hunter Valley

Geoff Phillips (049) 42 8617

Illawarra

Geoff Trott (042) 29 6629

Liverpool

Larry Saunders (02) 844 7377

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 August Meeting will start at
2.0 pm on the 5th August 1995
at Meadowbank Primary School,
Thistle Street, Meadowbank.

Printed by

Kwik Kopy West Ryde

Title	Description	Author	Page No
240 COLOURS	PROGRAM	DAMON PILLINGER	7
CHEAT FILE FOR INTERNATIONAL WORDS	TIPS	LARRY SAUNDERS	3
EDITOR'S COMMENTS	GENERAL INTEREST	LOREN WEST	4
FOR SALE - PRINTER		TISHUG	18
LEARN TO KNOW YOUR TI LESSON 29	TUTORIAL	PERCY HARRISON	5
MULTIPLAN MACHINATIONS	TUTORIAL	BILL HARMS	15
POTPOURAI	CLUB NEWS	TOM ARNOLD	7
P SYSTEM	TUTORIAL	UNKNOWN	8
RAMBLES	TUTORIAL	STEPHEN SHAW	13
REGIONAL GROUPS	GENERAL INTEREST	VARIOUS	19
RELATIVE FILES	GENERAL INTEREST	KEVIN COX	19
TECHO TIME	TECHNICAL	GEOFF TROTT	9
TISHUG SOFTWARE	CLUB NEWS	LARRY SAUNDERS	2
TREASURERS REPORT	REPORT	CYRIL BOHLSSEN	16
BUYERS GUIDE - MICE	GENERAL INTEREST	JUST NEWS AND REVIEWS	17

I B M I N D E X

TIshUG SOFTWARE FILE
AUGUST 1995
By Larry Saunders

Diskname 80Col:138
 Used= 341 Free= 17

Diskname 80CoL:136
 Used= 328 Free= 30

80 Column pictures By Alf Ruggeri.

80 Column pictures.

16	33 D128	17	33 D128
18	34 D128	19	34 D128
20	35 D128	21	37 D128
22	37 D128	23	40 D128
24	45 D128		

01	11 D128	02	12 D128
03	15 D128	04	17 D128
05	19 D128	06	20 D128
07	21 D128	08	23 D128
09	25 D128	10	25 D128
11	29 D128	12	30 D128
13	30 D128	14	31 D128
15	33 D128		

Diskname 80CoL:137
 Used= 280 Free= 78

Diskname G139
 Used= 333 Free= 25

Yet Another Paint Program (YAPP) and G99

Loading is by SuperCart or P-Gram card.
 Option #5: YAPP (Main program).
 Option #5: HARDCOPY (Printer program).
 OPTION #5: GIFFER1 (GIF Converter).
 I = Interlace (Low and Hi-Res modes) them FCTN 6.
 P = Pencil.
 SPACEBAR brings up Menu at base of screen you can select by pointing pencil. Brushes can be selected with "Q", "W" and "E".
 FCTN 4 = Erase.
 G = Grab Colour.
 A = Air Brush.
 L = Lines.
 V = Frames.
 X = Boxes.
 O = Circles/Ellipses.
 F = Fill.
 M = Copy.
 N = Move.
 C = Capture. (take a snap).
 D = Display Capture.
 Z = Zoom.
 FCTN U = Undo.
 Period (.) toggle between Cursor speeds.
 S = Save, Load, Catalog.
 T = Text mode.
 CTRL L = Crosshairs.
 CTRL S = Change Cursor colour.
 CTRL Q = Toggle NTSC/PAL.
 CTRL . = Quit.

CHARA1	5 Prog	DSR:AM	5 D 80
DSR:MECH	5 D 80	DSR:MM	5 D 80
EXEC	21 Prog	G99	49 D 80
GIF2EXE	25 Prog	HARDCOPY	22 Prog
HCSETUP	10 Prog	LOAD	8 Prog
MMSR/S	11 d 80	PICIND	12 Prog
VRAM192K/A	27 I128	YAPP	32 Prog
YAPPDSR	4 D 80	YAPQ	32 Prog
YAPR	7 Prog		

*** ARROWS ***

A game that you try to get your arrows to the other side of screen without getting them hit. (Has speech).

*** BINGO ***

Bingo game (Has speech).

*** GRIDDER ***

Gridder Loads into Mini Memory only. To load, go into BASIC and Load and Run GRIDDER. It will have the screen go green then go back to title screen. Then select MINI MEMORY Option #2: press enter TWICE.

*** KONG ***

BY David Mercer, UK; adapted for the TI by Jeffrey Brown, Ottawa.

The object of the game is try to get all of your white men to the other side of the board while preventing the computer from getting its black pieces to your side.

MOVING: Move the cursor using the arrow keys and press the <SPACEBAR> to select it. Then move to the desired location to move to, and press <SPACEBAR> again. A move may be cancelled with <BACK> at any time before it is executed. Yellow triangles indicate valid moves.

*** MATCH A PATCH ***

TIGERCUB SOFTWARE TCX-1102

Programmed by Jim Peterson 4/83, XBasic version 7/85
 COPYRIGHT 1983 Tigercub Software 156 Collingwood Ave.,
 Columbus Ohio 43213

Try to match up the pairs of Coloured/Patterned squares. If you make a match, you get another turn. You can play with 1 or 2 players.

*** POWER BALL ***

A type of Q-Bert.

*** TETRIS ***

This is the RUSSIAN version that Alf Ruggeri converted

to operate by JoyStick and I did doc's and colours.

*** T I C - T A C - T O E *** (Philadelphia Style)
An Original Program Released to Public Domain by
B.A. Traver 835 Green Valley Drive

WARNING: This game is really a cream cheese put-on,
for played strictly the 2nd player can't win! for
further information, send S.A.S.E. to B.A. Traver, 835
GREEN VALLEY DR., PHILA., PA 19128

Tic-Tac-Toe is usually a very B-O-R-I-N-G game for
two reasons:

1) The best the second player can ordinarily
hope for is a tie.

2) The game is rather predictable, with the
first player almost always opening in the center
square.

TIC-TAC-TOE (Philadelphia Style) is like standard
Tic-Tac-Toe with two major differences:

1) The first player must win to win (i.e., a
'tie' is a win for the second player!).

2) Neither player can move in the center square
unless that move blocks the opponent from three in a
row (or results in three in a row for the player!).
This entertaining variation was invented to give the
second player a more equal opportunity to win. Try
it!

The computer ask you ("Do you want me to be strict
about your not occupying the center square other than
blocking (or obtaining) three-in-a-row? (By the way, to
give you every chance, I will myself follow that rule
even if I allow you to disregard it. After all, you do
have the handicap of trying to match wits with a
computer!) Your choice (1 for strict, 2 for not so
strict)?)

*** INTERNATIONAL WORD GAME ***

You are given an English word at random, with its
four translations in scrambled form. Pick a language
and enter the unscrambled word. You have three tries.
Points for good answer 2 x length of word + 20 bonus if
you get all four. The four languages are, GERMAN,
FRENCH, ITALIAN, and SPANISH.

The second word game is HANGMAN. You are given a
English word at random and told if the hidden word is
in Italian or Spanish. You then have to figure it out
before getting hang.

Diskname G139
Used= 333 Free= 25

ARROWS	8 Prog	BINGO	38 Prog
CHARA1	9 Prog	GRIDDER	20 Prog
HISCORES	2 i 50	INTHANG	15 Prog
INTMENU	2 Prog	INTNDX	6 D201
INTSCRAM	15 Prog	INTWORDS	27 D 56
KONO	35 Prog	LOAD	5 Prog
MATCHPATCH	18 Prog	POWERBALL	36 Prog
ROOT	28 Prog	TICTAC/PHI	39*Prog
TITRIS*JOY	30 Prog		

CHEAT FILE FOR
INTERNATIONAL WORDS

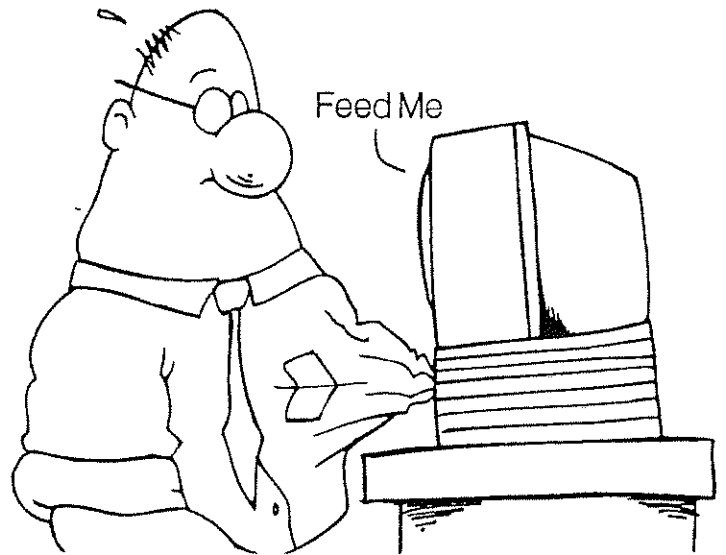
TISHUG Software File
August 1995
By Larry Saunders

Listed below is the English word (Left) and the
translations, German, French, Italian, and Spanish.

ALL	GANZ	TOUT	TUTTO	TODO
APPLE	APFEL	POMME	MELA	MANZANA
ARM	ARM	BRAS	BRACCIO	BRAZO
AUTUMN	HERBST	AUTOMNE	AUTUNNO	*OTON`O
BACK	*RU#CKEN	DOS	DORSO	ESPALDA
BAD	SCHLECHT	MAUVAIS	CATTIVO	MAL
BEAUTIFUL	*SCHO#N	BEAU	BELLO	HERMOSO
BED	BETT	LIT	LETTO	CAMA
BEER	BIER	*BIE`RE	BIRRA	CERVEZA
BOOK	BUCH	LIVRE	LIBRO	LIBRO
BREAD	BROT	PAIN	PANE	PAN
CAR	WAGEN	VOITURE	AUTOMOBILE	COCHE
CAT	KATZ	CHAT	GATTO	GATO
CHAIR	STUHL	CHAISE	SEDDIA	SILLA
CHEESE	*KA#SE	FROMAGE	FORMAGGIO	QUESO
CHILD	KIND	ENFANT	BAMBINO	*NIN`O
CHURCH	KIRCHE	*E'GLISE	CHIESA	IGLESIA
CITY	STADT	VILLE	CITTA	CIUDAD
COAT	MANTEL	MANTEAU	MANTO	ABRIGO
HEAD	KOPF	*TE TE	TESTA	CABEZA
HEART	HERZ	COEUR	CUORE	CORAZON
HOT	HEISS	CHAUD	CALDO	CALIENTE
HOUSE	HAUS	MAISON	CASA	CASA
HUNDRED	HUNDERT	CENT	CENTO	CIEN
KEY	*SCHLU#SSEL	*CLE'	CHIAVE	LLAVE
LEG	BEIN	JAMBE	GAMBA	PIERNA
LEMON	ZITRONE	CITRON	LIMONE	LIMON
MAN	MANN	HOMME	UOMO	HOMBRE
MEAT	FLEISCH	VIANDE	CARNE	CARNE
MILK	MILCH	LAIT	LATTE	LECHE
MONEY	GELD	ARGENT	DANARO	DINERO

MONTH	MCNAT	MOIS	MESE	MESE	EXPENSIVE	TEUER	CHER	CARO	CARO
MOON	MCND	LUNE	LUNA	LUNA	EYE	AUGE	OEIL	OCCHIO	OJO
NAME	NAME	NOM	NOME	NOMBRE	FAMILY	FAMILIE	FAMILLE	FAMIGLIA	FAMILIA
NINE	NEUN	NEUF	NOVE	NUEVE	FIVE	*FU#NF	CINQ	CINQUE	CINCO
NO	NEIN	NON	NO	NO	FLOWER	BLUME	FLEUR	PIORE	FLOR
NOTHING	NICHTS	RIEN	NIENTE	NADA	FOOT	FUSS	PIED	PIEDE	PIE
ONE	EIN	UN	UNO	UNO	FOUR	VIER	QUATRE	QUATTRO	CUATRO
ORANGE	ORANGE	ORANGE	ARANCIA	NARANJA	FRUIT	FRUCHT	FRUIT	FRUTTO	FRUTA
PEPPER	PFEFFER	POIVRE	PEPE	PIMIENTA	GOOD	GUT	BON	BUONO	BUENO
ROAD	STRASSE	ROUTE	STRADA	CARRETERA	HAND	HAND	MAIN	MANO	MANO
ROOM	ZIMMER	CHAMBRE	CAMERA	HABITACION	HAT	HUT	CHAPEAU	CAPPELLO	SOMBRERO
SALAD	SALAT	SALADE	INSALATA	ENSALADA					
SALT	SALZ	SEL	SALE	SAL					
SCHOOL	SCHULE	*E'COLE	SCUOLA	ESCUELA					
SEVEN	SIEBEN	SEPT	SETTE	SIETE					
SHIRT	HEMD	CHEMISE	CAMISIA	CAMISA					
SHOE	SCHUH	SOULIER	SCARPA	ZAPATO					
SIX	SECHS	SIX	SEI	SEIS					
SPRING	*FRU#HLING	PRINTEMPS	PRIMAVERA	PRIMAVERA					
STREET	STRASSE	RUE	VIA	CALLE					
SUGAR	ZUCKER	SUCRE	ZUCCHERO	AZUCAR					
SUMMER	SOMMER	*E'TE'	ESTATE	VERANO					
SUN	SONNE	SOLEIL	SOLE	SOL					
TABLE	TISCH	TABLE	TAVOLA	MESA					
TEA	TEE	*THE'	TE	*TE'					
TEN	ZEHN	DIX	DIRCI	DIEZ					
THOUSAND	TAUSEND	MILLE	MILLE	MIL					
THREE	DREI	TROIS	TRE	TRES					
TO BE	SEIN	*E TRE	ESSERE	SER					
TO BUY	KAUFEN	ACHETER	COMPRARE	COMPRAR					
TO EAT	ESSEN	MANGER	MANGIARE	COMER					
TO GO	GEHEN	ALLER	ANDARE	ANDAR					
TO HAVE	HABEN	AVOIR	VERE	HABER					
TO RUN	LAUFEN	COURIR	CORRERE	CORRER					
TO SLEEP	SCHLAFEN	DORMIR	DORMIRE	DORMIR					
TO TALK	SPRECHEN	PARLER	PARLARE	HABLAR					
TO WORK	ARBEITEN	TRAVAILLER	LAVORARE	TRABAJAR					
TWO	ZWEI	DEUX	DUE	DOS					
UGLY	*HA#SSLICH	LAID	BRUTTO	FEO					
VEGETABLE	*GEMU#SE	*LE'GUME	LEGUME	VERDURA					
VILLAGE	DORF	VILLAGE	VILLAGGIO	PUEBLO					
WATER	WASSER	EAU	ACQUA	AGUA					
WEEK	WOCHE	SEMAINE	SETTIMANA	SEMANA					
WINDOW	FENSTER	*FENE TRE	FINESTRA	VENTANA					
WINE	WEIN	VIN	VINO	VINO					
WINTER	WINTER	HIVER	INVERNO	INVIERNO					
WOMAN	FRAU	FEMME	DONNA	MUJER					
YEAR	JAHR	*ANNE'E	ANNO	*AN'O					
YES	JA	OUI	SI	SI					
COFFEE	KAFFEE	*CAFE'	CAFFE	*CAFE'					
COLD	KALT	FROID	FREDDO	FRIO					
COUNTRY	LAND	PAYS	PARSE	PAIS					
DAY	TAG	JOUR	GIORNO	DIA					
DESSERT	NACHTISCH	DESSERT	FRUTTA	POSTRE					
DOG	HUND	CHIEN	CANE	PERRO					
DOOR	*TU#R	PORTE	PORTA	PUERTA					
DRESS	KLEID	ROBE	ABITO	VESTIDO					
EIGHT	ACHT	HUIT	OTTO	OCHO					

END OF ARTICLE



EDITORS COMMENTS

Well winter is well and truly upon us, this may be one of the reasons why there were low numbers at the last monthly meeting, Wasn't it cold these last few mornings. Anyway I thought what a wonderful time to write an article for your TI computer magazine eaten cardboard boxes and looking for some articles to put in our magazine Well I do really, even though some of these articles I haven't read before (because I am only a new comer compared to some of the other members) however if our regular writers are the only one's to write articles sooner or later they get fed up with the lack of support and feed back, they may leave and go elsewhere. So come on members support your computer club and help out, just a small article will be sufficient, tell us what you have achieved, ask for help with any problems you may have. You could well be surprised with the interest shown in your article. Your magazine also prints advertisements, "For Sale" or "Wanted" free of charge to club members. Hope to see you all at the next meeting 5th August.

END OF ARTICLE

LEARN TO KNOW YOUR TI

LESSON 29

with Percy Harrison

One of the hardest habits to form in creating a program is to impose structure on the program. Structuring has gone by many names such as "structured programming" and "top down programming" and uses various techniques to discipline the programmer.

In this lesson we will outline the program right on the screen. The task is "chunked" into sections by using subroutines. This leads to clarity in the articulation of the program parts and allows testing and debugging of each part separately from the others.

After the outline is done, each subroutine is expanded by writing in ordinary English what needs to be done. Only when the English description is itself sufficiently detailed, does the BASIC programming begin.

Of course, there is always some backing and filling to be done as the program is written. The number of subroutines may change and the tasks performed in each will also change, usually expand.

There are those who advocate performing all planning of the program on paper before starting any of the coding. This may work for some programmers, but children especially are unlikely to adopt this style of work. Besides, if one advocates word processors so that writing text can be done interactively on the screen, it would seem equally appropriate to plan computer programs on the screen.

LESSON 29 LONG PROGRAMS

HOW TO WRITE A LONG PROGRAM

Let's write a hangman game. This is a word game where you draw another part of the hanging person each time you make a wrong guess for a letter.

First make an outline. You can do this on paper or right on the screen.

If you have trouble deciding what to do, then get a friend to play a game with you. Keep careful track on

paper of what is done during the game. Then the program has to do the same things.

The outline could be:

```
10 REM === HANGMAN GAME ===
200 REM INSTRUCTIONS
300 REM GET THE WORD TO GUESS
400 REM MAKE A GUESS
500 REM TEST IF RIGHT
600 REM ADD TO THE DRAWING
700 REM TEST IF GAME IS OVER
800 REM END GAME MESSAGE
```

Save this to tape or disk.

After making this outline, fill in more details. Just write in English what each subroutine needs to do.

```
10 REM === HANGMAN GAME ===
99 REM
100 REM----- MAIN LOOP
101 REM
120 INPUT" NEED INSTRUCTIONS? <Y/N> ": Y$
122 IF Y$="Y" THEN 200
130 REM----- GET WORD
132 REM STOP
135 REM----- MAKE GUESS
136 GOSUB 400
140 REM----- TEST GUESS
141 GOSUB 500
145 REM----- TEST IF GAME IS OVER
146 GOSUB 700
190 REM----- MAKE ANOTHER GUESS
191 GOTO 135
200 REM INSTRUCTIONS
```

--- write the instructions last

```
290 GOTO 130
299 REM
300 REM----- GET THE WORD TO GUESS
301 REM
```

--- use INPUT to get a word from player 1
--- draw dashes for the letters to be guessed

```
390 RETURN
399 REM
400 REM----- MAKE A GUESS
401 REM
```

--- player 2 guesses a letter

```
490 RETURN
```

```

499 REM
500 REM----- TEST IF GUESS IS RIGHT
501 REM

```

```

--- if wrong, GOSUB 600, draw hangman part
--- if right, GOSUB 700, see if game is over

```

```

590 RETURN
599 REM
600 REM----- ADD TO DRAWING
601 REM

```

```

--- add to the hangman drawing
--- test if drawing is done
--- if so, then GOSUB 800

```

```

690 RETURN
699 REM
700 REM----- TEST IF GAME IS OVER
701 REM

```

```

--- see if all letters have been guessed
--- if yes, GOSUB 900

```

```

790 RETURN
799 REM
800 REM----- END GAME MESSAGE
801 REM

```

```

--- message for when guesser loses

```

```

890 RETURN
899 REM
900 REM----- END GAME MESSAGE

```

```

--- message for when guesser wins

```

```

990 RETURN

```

Save this program to tape or disk.

Now is the time to start writing and testing the first part of the program. Go to line 132 and remove the "REM" from the statement "REM STOP" and this will ensure that only the first subroutine will be run. After you have managed to get the first subroutine working correctly put the "REM" back in line 132.

Start by writing the subroutine at 300, GET A WORD. The first step is to write more details, in English, of what the subroutine needs to do. Then start writing the BASIC lines.

Assignment 29

1. Finish the hangman game. This is a long project.

Start by writing the GET A WORD subroutine. Then save it to tape or disk. You may want to write one subroutine each day until the program is finished. (Because of the length of this program an answer will not be published).

ANSWERS TO LESSON 28

Assignment Question 28-1

Due to the fact that I have changed color commands in this Assignment I suggest that you don't try to use the program used in last month's magazine but retype the whole program as follows. This includes the answer to the first question.

```

10 REM MOVE A SPOT
11 REM----- BORDER
12 CALL CLEAR
14 CALL CHAR(91,"FFFFFF0000000000")
15 CALL CHAR(92,"000000000000FFFF")
16 CALL CHAR(93,"COCOCOCOCOCOCOCO")
17 CALL CHAR(94,"0303030303030303")
18 CALL HCHAR(1,2,92,30)
19 CALL HCHAR(24,2,91,30)
20 CALL VCHAR(2,1,94,22)
21 CALL VCHAR(2,32,93,22)
22 REM----- COLOR THE BORDER
23 CALL COLOR(8,6,1)
25 CALL CHAR(42,"FFFFFFFFFFFFFFFF")
27 CALL COLOR(2,7,1)
50 X=16
51 Y=12
59 REM----- LOOK AT JOYSTICK
60 CALL JOYST(1,H,V)
61 DX=H/4
62 DY=V/4
63 REM----- ERASE OLD SPOT
64 CALL HCHAR(Y,X,32)
67 X=X+DX
76 Y=Y-DY
90 REM----- PUT A SPOT ON THE SCREEN
91 CALL HCHAR(Y,X,42)
92 CALL KEY(1,K,S)
93 IF S=0 THEN 99
94 CALL COLOR(2,16,1)
99 GOTO 60

```

Assignment Question 28-2

Add the following lines to the above program:

```

68 IF X>2 THEN 70
69 X=2
70 IF X<31 THEN 76

```

```

71 X=31
75 CALL HCHAR(Y,X,32)
77 IF Y>2 THEN 79
78 Y=2
79 IF Y<23 THEN 85
80 Y=23
88 CALL COLOR(2,7,1)

```

Assignment Question 28-3

Add the following lines to the program resulting from Question 28-2:

```

94 CALL COLOR(2,1,1)
95 FOR T=1 TO 500
96 NEXT T

```

Bye for now.

END OF ARTICLE

240 COLORS

By Damon Pillinger

Here is a great demonstration program, as the name suggest, shows the full 240 colours which your TI99/4A can produce. So type it in and give it a try.

```

10 REM **** 240 COLORS ****
20 REM
30 REM **** BY DAMON ****
40 REM *** PILLINGER ***
50 CALL CLEAR
60 INPUT "SPEED (1-300)":SP
70 CALL CHAR(97,"AA55AA55AA55AA55")
80 CALL HCHAR(1,1,97,(32)
)
90 FOR T=2 TO 16
100 FOR J=1 TO 16
110 FOR G=1 TO SP
120 NEXT G
130 CALL COLOR(9,T,J)
140 NEXT J
150 NEXT T
160 CALL CLEAR
170 PRINT "240 COLOR S"

```

END OF ARTICLE

POTPOURAI .

by Tom Arnold.

Retyped from Clubline, Hamilton Canada Users Group Newsletter, February, 1988 by John Ryan of TISHUG for TEXPAC BBS. Edited by Ross Mudie.

Not much in the TI news world at this time - there is a rumor that TI is about to introduce a new home computer to the market. It is supposed to be TI99/4A compatible. I would doubt that there is any truth to this.

Clint Pulley has released his latest release of c99. This is a supplemental disk which you can get from him or from our (Hamilton) library Rave 99 has released it's Memory Enhancement Card system, which allows up to two megabytes of backed-up RAM memory for the TI.

Rave 99 also has released it's Speech Synthesizer card for the PEB. This allows you to add the speech synthesizer to the PEB, a must for those with the Geneve.

The latest release for the Geneve is 1.0 of DOS, 99.8 for the GPL loader and 1.0 for MyWord. No word yet on Advanced Basic, C-Compiler or the Pascal System. I forgot, there is also a later release of MultiPlan but I don't remember if there is a version number involved.

Innovative Programming of Rohnert Park, CA is going out of the TI business. Don't send them any orders.

Ryte Data of Haliburton has not been answering their mail. Be very cautious about sending them any money.

Micropendium is your best bet on getting news of the TI. It's address P.O. Box 1343. Round Rock, TX 78680. Check for prices.

PC-Transfer is a new program that transfers TEXT files from the TI to an IBM via DISK. Yes you read right, it allows you to format an IBM disk on your TI then you can copy your files onto it in IBM format. You must have a Myarc or CorComp disk controller (you need double density) a TI or Geneve. This program was written by Mike Dodd but I don't know where it is available yet.

On the subject of mail order, Computer Shopper has many good deals. It also has a TI column by Ron Albright. I ordered a chip upgrade for my Epson called Dots-perfect that has been the best money I have spent in a long time. Computer Shopper is available at most newsstands. (Canada).

If you order by mail, never, never send cash or cheque. Always order by Visa or Master Card. If they don't deliver you can have the charges removed from your statement. This is very good protection and when ordering from the U.S.A. You do not have to worry what the exchange is, they will bill you the current rate.

END OF ARTICLE

P - SYSTEM

USING THE P-SYSTEM (TI PASCAL)

Several notes can be made about using the p-System on the TI99/4a. First, you cannot use the TI LOGO cartridge on a system with the p-Code card in place. The p-System will work fine; LOGO will, in certain circumstances, lose control to the p-Code card, destroying what you were doing with LOGO. If your p-Code card has an on/off switch, all will be well if you turn the card off before using LOGO.

When a 99/4 is powered up with the p-Code card in place (and turned on) the p-System initializes and then displays the p-System greeting. The p-System remains active until the user stops it; thereafter it will not restart without using the technique below (or turning off the system). To get from the p-System to the BASIC environment is simple - use the p-System 'H' command. To go from BASIC to the p-System, you need to have one of the following cartridges in place on the console:

1. Extended BASIC
2. Editor/Assembler
3. Mini Memory

From console or Extended basic, execute the following command:

```
CALL LOAD(14586,0,0)
```

and then type `fcfn=` (Quit) or the BASIC command 'BYE'. The p-System will then operate as if you had turned the system off and then on again. Some programs and cartridges seem to have the same effect as the CALL LOAD listed above. Exiting TI-WRITER, MULTIPLAN, or the COMPANION word processor will lead to restart of the p-System. Likewise, some of the commands in the DISK MANAGER will reset the p-System so that it will become active again upon leaving this module.

Speed is an issue of interest to all potential users of the p-System. If computing speed is the reason for the TI BASIC programmer to consider the p-System, the following facts apply: Because the p-System is interpreted, it is relatively slow compared to compiled PASCALS (such as TURBO PASCAL on MS-DOS and CP/M systems). Compared to other machines, the p-System on the 99/4 is quite slow - I believe it is the slowest implementation of the p-System around. TI BASIC, however, can be benchmarked with a sundial, so it is easy for the p-System to beat. Number crunching

programs PASCAL can run up to 60 times faster than their BASIC counterparts, especially if put in RAM (see SETLTYPE). Programs doing mostly screen I/O can be slower than BASIC, however. Disk I/O seems a little faster with the p-System -- perhaps because of the simpler file system, leading to less searches for the next block, etc. I have no experience with FORTH, but know that it too is faster than BASIC. The programmer desiring speed above all else should compare the p-System to FORTH.

On running the V command in the FILER program, you will notice that there is a device #14 called 'OS:'. If you do not have a disk in drive one (the usual root device), one will find that the 'PREFIX' and 'ROOT' devices are defined as 'OS'. This device includes files that are in ROM and are used for booting the system. You can read and examine these files but not, of course, modify them. To change the PREFIX definition, use the FILER P command; to change the ROOT definition you must use the system I (warm start) command.

Compatibility with other Machines

The p-System on the 99/4 is very compatible with other p-System implementations. Source code developed on the 99/4 can and does run without modification on most other machines if you avoid using the TEXAS INSTRUMENTS UNITS included with the PASCAL Compiler. The 99/4 is less able to run programs from other machines due to its relatively small RAM size. If the program is properly segmented, however, there will be no problem. The PASCAL compiler, for instance, is 99 blocks long -- 49.5K. It runs on the 99/4 (with a lot of disk swapping). The only other problem in running programs from other machines on the TI is the presence of a 40 (rather than 80) column display. The TI will display all 80 columns with the SCREEN RIGHT and LEFT keys, but you will probably want to modify the display (if possible) to fit the screen. In the file SCREENOPS.CODE on device #14 (see above) is a description of the TI screen size. This file can be read by a program. This file tells the program in the case of the TI that the screen is 40 columns. Clever programs will adapt themselves to the size of the screen.

A larger problem is physically transferring programs to and from the 99/4a. The TI p-System lacks any functional communications program at this time. Files can be exported from the 99/4a easily by using the FILER to Transfer to REMOUT:.

There is good but not complete compatibility with other PASCALS. I have programs that I wrote on the TI p-System that I ported to an MS-DOS (IBM PC or compatible) machine and converted to TURBO PASCAL.

Little modification was necessary.

Techo-Time

from Geoff Trott

Software for the TI p-System

I know of no commercial software for the TI p-System. I suspect that if enough interest could be shown, then the whole spectrum of existing p-System software would be made available.

USUS maintains a large library of programs, most of which are not machine dependent. These volumes are available to USUS members for a small fee - in TI format. Most programs have not been modified to run on the TI. If you modify a USUS program to run on the 99/4, you are requested to send a copy back to the USUS TI distributor:

Wish List

The p-System on the TI 99/4a is powerful but remains largely undeveloped. What follows are my suggestions for software development which would greatly aid the use of the system.

FORTRAN: could someone port this compiler to the TI? There would surely be great interest in this language on the 99/4. Likewise, many would welcome MODULA-2.

Terminal Emulation: There have been programmers who have had TE programs "almost working". This is probably the highest priority for the p-System. A TE program would allow porting of other programs to the TI without re-keying the source code.

A good Data Base: there are no high quality data base programs available for the 99 in any form. Several excellent programs are available on the p-System. Could one be ported to the 99?

HELP!

The information in this guide is essentially all from my own neager experiences. Like most p-System users on the 99/4a, I doubt that I have ever laid eyes on another user. This guide has been written, therefore, in a vacuum. Its purpose is to catalyze discussion, debate, and the creation of a better guide to the TI p-System.

The GIF file format

I am looking at a GIF display program to see if it could be enhanced in a few ways and Rolf obtained for me a document which details the GIF file format. This is a copyright document by CompuServe so I thought I would share its contents with you by examining a GIF file and explaining what is contained there. I have chosen a file with skin tones, as that seems to be a hard set of colours to produce accurately, and one of the improvements I was hoping to do for the 9958 video processor was to use its enhanced colour display. I have used Disk Utilities to get the data from the file in hexadecimal format which is listed below in blocks of 16 rows of 16 bytes for each sector on the disk. I have listed the first 7 sectors and the last 3 sectors. GIF files on the TI99/4A are Internal Fixed 128 format which means that all 256 bytes of the sector are used for data.

Each GIF file starts with a header block which consists of a signature, screen descriptor, and global colour map, if required. This is then followed by an image descriptor, local colour map and raster data, which is encoded. This last group of data can be repeated as many times as necessary. The file is finished with a terminator character.

The header block starts with the 6 characters "GIF87a" whose ASCII codes can be seen as 47, 49, 46, 38, 37, 61 at the start of sector 1. There is another possible signature "GIF89a" which has a slightly different coding. Following the signature are four bytes which give the screen width in pixels and screen height in pixels. These can be seen to be 0100 and 00C8 (least significant byte first in each case) for this file. So this file is for a screen whose size is 256 pixels wide by 200 pixels high which is the dimension of our display. The next byte contains bit information and has a bit pattern of 10100011. The most significant bit being 1 means that a global colour map follows the descriptor. The next 3 bits (010=2) means there are 3 bits of colour resolution which is the resolution of the 9938 processor. The next bit is always 0 and the last 3 bits (011=3) means there are 4 bits per pixel in the image which means there are a maximum of 16 colours in each pixel. The next byte is 00 and is the index into the global colour map for the background colour. The next byte is always 00 and is the end of the screen descriptor.

END OF ARTICLE

The global colour map follows, as that bit was set in the screen descriptor. The length of the colour map is determined by the number of bits per pixel. In this case that was 4 so there are 16 (2 to the power of 4) entries for each of the three colours giving a total of 48 bytes for the global colour table. These are laid out as follows:

index	red	green	blue	colour
0	0	0	0	black
1	255	255	255	white
2	72	0	0	dark red
3	72	36	36	
4	109	36	0	
5	145	72	36	
6	182	109	72	
7	218	145	109	
8	255	182	145	
9	255	218	182	
10	255	218	218	light pink
11	255	255	255	white
12	255	255	255	white
13	255	255	255	white
14	255	255	255	white
15	255	255	255	white

This uses all but the last 3 bytes of the first 4 rows of the listing of the first sector. Black is represented by a (0,0,0) colour set while white is represented by (255,255,255). This set of colours is not fully used and is dominated by the red colour, which means these colours are all pinks of some sort. There are also a number of whites defined so not all the codes are used. The rest of the file is taken up with the image description.

The image description starts with an image separator character ",," (2C) and then 4 words to define the top left start of the image and the width and height of the image. In this case, the start of the left side is 0000 and the start of the image at the top is 0000. The width of the image is 0140 (320) and the height of the image is 00C8 (200). This image is wider than the screen size. The next byte is 00 and is a bit specifier with the most significant bit specifying that the global colour map be used rather than an optional local colour map. The next most significant bit used to specify whether the image is in sequential (0) order or interlaced (1) order. The next 3 bits are always 0 while the last 3 bits are used if there is a local colour map as the number of bits per pixel. The next byte is the start of the raster data and has the following format.

The first byte is a code size which in this case is 04 and is the minimum number of bits necessary to represent the set of actual data bits. The next byte (FE=254) is the block byte count and so there will be 254 data bytes following this byte. These are

compressed using the LZW algorithm and packaged into the 254 bytes. The next byte (>10) is an addition to the LZW coding as it is a clear code which resets all compression parameters, decompression parameters and tables to a start-up state. There is one other code called the end of information code (>11 in this case) which is used to indicate the end of the image data stream. The LZW compression is a zero error compression algorithm and provides a series of variable length codes from 3 to 12 bits in length. These are packaged in series into 8 bit bytes without any wasted bits. The byte count will be between 0 and 255 which means a total number of bytes in each block of 256 (one length byte). A length byte of 0 finishes the raster data stream for this image. If you are following the details of this article up to now, you will see that the first block of data is 254 bytes long (>FE). The next block is also 254 bytes long as are the first four blocks. The fifth block is 255 bytes long, while the next two are back to 254 bytes long. That takes us to the seventh sector listed here. I have included the last 3 sectors so you can see how the file ends. In the first of these sectors, the length byte is FE on the 4th row, 3 bytes from the right hand end. This leads to the next length byte in the next sector which is also FE and one byte position to the left, relatively speaking. So in the last sector, the length byte is the one position to the left and is 75. This means the end of the file ends with a zero byte count at the start of the twelfth row.

So where is there a problem? The problem with the colours is that the 9938 chip only uses 3 bits for each of the three colours. There are 2 useful modes for displaying colour pictures in the 9938. These are graphics mode 6 and mode 7. In mode 6, a colour palette is used where up to 16 colours can be defined from a choice of 512 (8 values for each of 3 colours, red, green and blue means 8 cubed colours) different colours. Each pixel is defined by 4 bits (the index into the colour palette) and two pixels are in each byte. The picture is built up by these groups of 4 bits stored in the VDP memory. In mode 7, 8 bits are used for each pixel to define the colour of the point and each pixel requires one byte of storage. This means that only 256 different colours are possible but all colours can appear in the same picture while for mode 6 only 16 different colours can appear in the picture at once. To obtain the colour of a point in mode 7, 3 of the 8 bits are used for the red colour, 3 are used for the green colour, but only 2 are used for the blue colour. If the most significant 3 bits of the global colour table above are taken, for mode 6 the colour palette is going to be:

index	red	green	blue	colour
0	0	0	0	black
1	7	7	7	white
2	1	0	0	dark red

3	2	1	1
4	3	1	0
5	4	2	1
6	5	3	2
7	6	4	3
8	7	5	4
9	7	6	5
10	7	6	6 light pink
11	7	7	7 white
12	7	7	7 white
13	7	7	7 white
14	7	7	7 white
15	7	7	7 white

Although there is not much difference in the general colours, a few bits of precision have been lost. In mode 7 the blue colour would have even less variation than the table above, going from 0 to 3. In the 9958 processor, there is another version of mode 7 when the colour signal is coded into YJK signals. Then the colours for each group of 4 adjacent pixels have the same J and K values (6 bits each) while the Y value (5 bits) for each pixel can be different. The values for RGB are calculated from the YJK values by the 9958 to give a 5 bit value for each colour, which gives 4 times as many variations in the colours red, green and blue over the palette of colours in mode 6. The total number of different colours which can be displayed over the whole picture is now 32 cubed or 32,768 colours. This should give much finer change of colours and help produce good skin tones. The formulae used are:

$$Y = 0.5*B + 0.25*R + 0.125*G; J = R - Y; K = G - Y$$

$$R = Y + J; G = Y + K; B = 1.25*Y - 0.5*J - 0.25*K$$

Clearly, using this facility will require more calculation for each point. For the technically minded, Y is always positive and so is represented as an unsigned integer in 5 bits. J and K can be negative and so these are represented as 6 bit signed integers, which means they have a 5 bit magnitude. Of course the conversion from YJK to RGB must result in positive numbers between 0 and 31 (for 5 bits) and this depends on the YJK to RGB conversion table, which is built into the processor. I have processed the values for the colour table of this article, assuming truncation of numbers in the arithmetic to integers, to give the following equivalent palette of colours.

index	red	green	blue	colour
0	0	0	0	black
1	31	31	30	white
2	10	0	0	dark red
3	9	5	4	
4	12	3	0	
5	17	8	3	
6	22	12	8	
7	26	17	13	
8	31	22	17	
9	31	26	22	

10	31	31	27	light pink
11	31	31	30	white
12	31	31	30	white
13	31	31	30	white
14	31	31	30	white
15	31	31	30	white

This has a different look to the previous table, as the extra bits of precision allow for a wider range of values. The next step is to actually program these calculations in and see if the difference is worth the effort. If only I can find an easy place to put the code in the existing program!

```

=====
4749 4638 3761 0001 C800 A300 0000 0000
FFFF FF48 0000 4824 246D 2400 9148 24B6
6D48 DA91 6DFF B691 FFDA B6FF DADA FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FF2C 0000
0000 4001 C800 0004 FE10 C849 ABBD 38EB
CDEB FF93 2080 D848 99A2 4014 0631 OCC2
40CC C661 DF78 6EEC 6E2A C2BE A0EF C5C2
194C A4A4 72C9 6C3A 9FDO 2860 7063 B980
C26C EAA5 1515 6E86 C2B5 40AE E110 39F0
6127 C676 BBAB 3099 20AD DBEF F8BC 3E33
F8CA 5F40 816F 426E 2204 4661 4461 3768
693A 6B6D 836F 7D64 3674 7B98 999A 9B98
2A90 0705 3231 825A 8583 5F60 6104 2A95
8C39 8D88 89A6 923F 716B 039C B9BA BBBC
1B02 7E2B 30C2 B3B4 A566 A9AB 2BA8 8CC7
6935 6CA1 C584 2E04 6148 BDD8 D9DA 786C
sector 0

```

```

=====
5758 C4D2 412F C73B 3C44 A88D B007 68EA
366C 5CR1 5B33 73DB F5F6 F749 3556 A331
F192 8AE5 58B4 50B4 2EDD BA34 B002 F6F0
1763 4635 5C00 62E0 9B48 71E2 1732 80FA
6904 37E8 DB39 32FE 2CA2 F5B1 81A0 E441
57ED D680 0AC5 F18D B216 3360 549C 4973
17AB 3932 8431 1412 F387 0C31 F3DA 8C2C
D82E 65A3 67B2 185E 5446 E65A CDA7 50EF
5052 B530 E34E 9FB5 00BD A046 8364 4958
6049 EAE8 D652 CB33 963B 9C46 5DCB 9644
AD22 D474 062A 5B8A 8B8C 55A3 1C7A FDAA
AE46 CAB1 91FC 3D4B 6600 62DB C388 39FC
9263 63CE 28AD 571B CE25 C8B7 B2CA CB28
1FFD 8937 A548 2817 11D5 261E 9DB8 5A91
3089 BC45 96E7 F327 5193 27D5 1424 EA28
205D 7195 5A44 0B45 BA37 6956 EE62 F6B9
sector 1

```

```

=====
4DAB 0F5E 4087 5ECB 3E53 3965 4031 FE06
C829 72A0 4744 DFD8 D9C6 FOD3 22EB EA21
56B7 5277 B497 2F60 9020 174A 1B09 930D
0AD1 D9E3 DF23 8060 C74A AE3D BF6B FC21
508C C063 CD7D FEE5 CE74 71A4 8711 6706
A061 DC81 5BC8 E720 3E54 DAC7 466A C6E9
F798 173A 9451 4679 0242 E29F 4038 00C5

```

5168 412C 058A 74EE C050 007C OFB6 B889
27A0 9463 CD5D 5645 869C 197F A9F1 5502
BOAD B461 6DEA A540 6229 73E8 6300 8FA1
7CC1 A28B 4CEE 219D 7D2A B1A4 1571 75D9
120B 7908 24C0 6342 3FA6 7256 343E 0C59
4A1A 7D28 B864 9368 E291 9C49 70FD 60A1
0CE5 AC21 6319 5012 A525 6C32 E219 1253
DO09 599C 5709 A8D4 CD99 6916 1AC5 2125
2D72 0095 C585 841E 482F 0C08 E89D 4715
sector 2

51DF 81C2 2983 5717 62FE 5043 020A A844
ODOB 3219 6ARA 1D5F A091 6A75 16D6 72D6
OC5E C9A0 658C 766E E921 2325 C170 CB76
7D72 6A0C A88B CE33 8221 BC9D 6A6C 137D
1C83 4A61 ADFE 6E07 4635 3CDA FO0B A897
D66A 5240 0728 10EA 1103 9AB4 62B2 606A
D1A9 A7D7 D213 8292 C7A6 9BC4 19CF 360B
2722 0904 80C0 0C08 0410 AD0D 5A6E 792D
0F47 CAAB ABAA 7032 580C 897E AC03 9453
84AA ABBO 0454 EC05 4A90 DF25 4BDB A7FA
A261 5F96 3BEA 5970 8C16 BB00 OD5D 32F9
60A5 2A62 2C6C F207 DDF6 B5C2 A65B 30EA
2A87 CE00 4829 C603 8EF7 1C9D 8E89 DOA9
98A1 2923 1DC0 A59E 2C34 07AA 76B3 0A3C
FBC5 F36E 65B6 1AA5 EFCC 7236 3396 4242
5CD7 5157 0325 3CB4 DO09 68E8 10C4 3BC9
sector 3

FODA 8E15 331D A097 624D 9D73 1012 5CCD
5E53 5BC7 8D41 0108 EC99 1ED2 49AF 8763
493C E63B 335F 7D63 8CA7 A099 ODA8 4A98
E38A 33CC 3C90 B829 B7DC 603C A1A1 56D1
ED0D 78BE FF97 FF4D B33B 9839 2327 7443
B60D 07A4 1B8B B1C3 E35B FF52 737A OED9
6808 8082 6789 B9EC 7E97 CDF9 33AE 3813
4D68 56A3 207A 44A8 24F9 0928 A8C7 8DA2
1C71 50C5 72D8 D4B1 C377 EC77 FA9D 7187
B8BF 8208 E8C3 EAFD DEB0 11AD 3C8F 2ADD
7B61 58F1 EA1E 520E ODE6 801D 5DF3 01F2
5DBB E61D 464D DEE7 E1FA 1E26 12BF 8CDA
D825 306A 4D3E 932B B8C5 7F5A 17B1 6504
2870 EF6B DF51 EC83 2323 78C8 30F6 C35F
F670 3207 B4B4 E17F 276B 5861 AAA1 OABC
5DA5 1AAF 785E 736A 4736 7D19 AE4E D75B
sector 4

0589 AEC3 3B21 B5ED 272E 2883 OBOE E03F
OC32 A915 9FA1 8A07 1932 8566 B4EF 693F
B455 D450 E825 EC69 6F67 24BA 8B86 A033
3E1B AAGB 4041 11D1 OE39 8322 E61C 107A
26B1 8C9C FE08 E7C0 41F1 EE8B 3A0B 416B
E084 BC26 3AF1 58C9 D18D 8686 A11F 748C
4D84 CF33 61AA F424 BF98 2522 7B15 E807
OBB7 F018 D2CD EB8C OA4B 6305 79D0 ACE4
E06A 8488 2C0F 2418 581B DB20 AEGA DA13
1950 5ED2 1840 1A4B 6251 63CA D1DE 2416

014D 2F8E 65C3 1323 C933 1DEE 8591 7B89
8BC1 1F7E 569D 0358 F258 9FD3 A470 DE04
AO37 C271 7A42 EC5C DAOC 67C4 EDA1 7274
20B1 C42B 61D9 0D03 A9EF 2AB0 B313 C666
B7A5 COE5 2877 CF51 E12F C5E8 2715 6C4A
89E6 70E1 30D1 C41D F478 6F8A 3B99 A302
sector 5

65F7 43E7 E9B2 8BDD 6861 DBC4 B6C7 9BD0
A18F 0359 C336 D3C4 41A3 CD41 382E D3C2
9AAE 48BB 655A A647 ECF2 5222 4267 BF75
8606 4EB3 B8D1 C38A 354F 2621 C547 8F02
A73F A6FE 604B A6CD AR39 B391 9A58 9082
912F 5AC0 7749 9C81 OFC8 C1D0 86CA 6731
CB72 8737 F339 8983 9473 9CCE 991F 4771
1146 6AAA 252F 3AE1 D522 8E60 D207 0987
1C4C C969 AB12 34CE 666A 716A A4CC 64F6
2498 C7F7 4CE9 65FE A94E 4F5D A42C 4DE5
A459 0040 1422 1538 C49B D94C 50E9 BC9F
360D 9A44 C81C 8205 F509 E025 A62A 9FCB
F827 19F5 9B28 212A 4A4E EAA1 071A 8FDA
D810 7953 D30E B4CC 274C 31C3 A6D8 EA1B
4CDA 46A4 226D 9608 88CA 4F01 D549 4673
B205 EEBC 3856 16E6 7148 665D 096A 2E11
sector 6

.....

DE6A 96D8 DDF7 18BE 4B10 0934 OD66 9E5E
B360 2DE0 1F5E E6AF CD7E ECA1 4769 B61B
31A9 6984 2B81 BA0C DCD6 082F 3862 AF0E
F566 AD91 3C44 530F 5118 9740 A4FE 5F5D
ED53 BF61 54BB 6DC6 9320 C096 1F55 CE22
07A1 C393 FA6E BC6D F452 B02B E9F6 FC36
4D53 080A 005E 4160 6DED 2058 8B55 257F
3528 BC71 D887 2840 0196 A9CD 428B 102E
B09C 4F24 1F7C 411B CD41 F194 23CB CF08
9DA9 4371 67F5 F4A0 F5D3 B6E8 B45F 8F7D
7A11 69AF 0351 DCD5 DE9D 61CC BB00 DECF
6DEC 2AEB F313 5902 D213 4559 BD1D 04C6
9A5F 1D3E E875 919E 7AFA OB08 A01D 3FD4
67FF F7DD CD05 CF2A EF31 C842 08B9 D8FF
E217 3A3C 5941 5CB8 BB9D E89A C7AF F11D
504D F4A3 2009 OC10 005B 08EE 607C 684C
sector last - 2

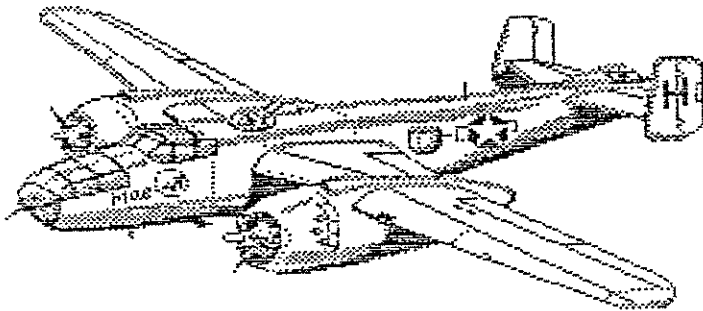
2B30 E7B7 A380 6F61 47FA D2C5 8A27 254D
F1A6 0634 605F D8B6 4004 A195 2771 FD98
6005 E977 8076 5442 5C1F A096 OAD4 EOC1
A358 E583 2204 E067 3A47 3E3D FEE8 6667
5E73 96F2 D4F7 9DA0 A5AB 50D0 F3C0 FC6C
D838 1C9E C20E D833 9D19 56B8 C297 E5A6
5944 E414 CE16 66B9 17E9 0481 E3A9 8053
06D7 43A0 BDCA 7F62 A461 90AE 48BD 2C2A
6A87 E32A 4394 OE21 A9EE 5D66 5F73 9CE2
09EB B4C6 F17C 4004 OD1C 5C2B 8620 OCA9
2904 5A1D 90C7 1DA7 97C7 1664 6C83 2928
8C81 E806 C8DB E42D 9340 C418 126F 7348

44DE 0904 8B04 80CA 1848 B747 1206 6669
 E461 252D 093B 4C16 C676 2000 402B C088
 4660 8630 5284 0966 D1B0 03A3 5496 A655
 D75B 5F0B 4597 AF1F 00E6 72C7 F480 2D6F
 sector last - 1

```

=====
C9B8 3C7A 7121 AE64 4531 BDC9 204D 18A8
9B60 AC13 3293 79BE 14A8 A03E 6163 9F36
A228 A38B A130 0B16 B826 16DB A1B0 3AB1
2F0B A423 A537 81D9 B036 2875 B19C E63C
E79D 6C37 B6F9 2C90 573B 41DD B5EC 1046
B485 2477 D0AC A7C9 2E28 3714 B6F0 0F7C
2807 ODPC 19C8 B369 B3A0 2385 07AA 4AC7
C06C B060 083E 34D7 299F D381 6E55 5469
B3C3 E821 F5D7 CA8E 12F3 A37F 04E7 1746
76CE 259E EF74 AB39 698C 2AB0 D2D5 342B
A212 8DA9 4C9D FA54 A846 55AA 531D 5804
0000 3B33 3333 3333 3333 3333 3333 3333
3333 3333 3333 3333 3333 3333 3333 3333
3333 3333 3333 3333 3333 3333 3333 3333
3333 3333 3333 3333 3333 3333 3333 3333
last sector
  
```

END OF ARTICLE



BBS

JUST A ONELINER

Q. Which hand should you use to stir tea?
 A. Neither. It is better to use a spoon.

RAMBLES .

April 1988, by Stephen Shaw in England.

Greetings once more.

I have been occupied recently grappling with the likes of MS-DOS, and a few other IBM "goodies" such as a data base with the capacity to handle records (records!) of up to 4k. It is a pleasure to get home to a well designed modern computer... and I do mean that!

First an item for Extended Basic users with 32k ram... one of the things Myarc intended to include in their XB was a forced garbage collection, but they seem to have run out of room. It is really quite easy, but what am I talking about? Type in this program and run it:

```

100 CALL CLEAR
110 A$="1234567890"
120 FOR T=1 TO 12
130 CALL SPRITE(#T,42,2,30+T*3,30+T*3,0,34+T)
140 NEXT T
150 CALL PEEK(-31890,I,M)
160 CALL PEEK(-31974,H,L)
170 A$=A$A$A$A$
180 DISPLAY AT(20,15):"FREE:"
190 F=(H6+L)-(I6+M)
200 DISPLAY AT(21,15):F
210 GOTO 140
220 END
  
```

The number you see counting down is the free stack space- it is being filled with redundant definitions of A\$ (line 170). When the number reaches/passes zero, you will see that it starts again from a high value- but at the moment of changeover, all your sprites will halt briefly. During this brief halt, the computer is checking the stack to see what information there is redundant (no longer required) and getting rid of anything it doesnt need- garbage collection.

Garbage collection not only causes sprites to halt but can also interfere with music, causing the odd note to sound for longer. Instead of having garbage collection occur when the stack is full - which can be quite often if the program is long and stack space short- it can be useful to force a garbage collection at a time we choose, when the effect may not be too obvious. Also, if we do a garbage collection before the stack is full, the pause may be quite a bit shorter.

Here's how to do it... insert the following lines into the above program:

```

95 CALL INIT
205 CALL LOAD(-31885,144,"",-31858,81,169,152,0)
  
```

Now run the program again. Notice that the stack space remains constant, as we are forcing a garbage collection after each definition of A\$. How is sprite speed affected?

Try a garbage collection at an intermediate stage:
204 IF F>10000 THEN 210

Any difference? Have fun. Thanks to the Sydney User Group (TISHUG) in Australia for this one, taken from their newsletters for March and July 1987.

LOGO... at last a LOGO query, and a very interesting one too... the LOGO manual contains an ANIMAL program. If you key it in and build up the KNOWLEDGE a little, say by adding three or four extra choices, as the program runs, then save it to disk... you will find that RECALLING the procedure will lock your console up. Hnnnn.

On Page 7 of the Logo manual, you can find the information that input lines are limited to 127 characters.

Here is the clue! Basic program lines are also limited to 127- 127 bytes after tokenisation has occurred, which is why you can sometimes key in less than the 5 screen lines XB allows you and still get a LINE TOO LONG error message.

Back to Logo... as you play ANIMAL, the KNOWLEDGE LIST gets longer and longer and longer... and once it exceeds 127 bytes, it becomes incapable of loading back into the console. The file is just too long for the routine which reads the disk and places the definitions where they belong.

Once you have been playing ANIMAL for a while, if you wish to save it, you must first reduce the LIST attached to KNOWLEDGE to under 127 bytes. You could do this in immediate mode or write a routine to do it- perhaps call it "RESET".

TURBO PASC 99: in the last issue I mentioned Texanents as a supplier of Turbo Pasc 99, a superb language for the TI99/4A, copyright June and August 1986 by an Austrian software house, with rights now owned by a German company.

Texanents advise that they no longer wish to offer this program, and as a result of the time and money I have wasted on this, I now delete Texanents from my accredited list.

Yes, a superb language, now 22 months old, and with NO distribution in the USA, Canada, the UK, Australia, France... available only to a small (very small) fraction of TI users.

If you have an interest in the language, please write to me.

MUSIC MAKER:

I have received several requests on how to use Music

Maker Module with a disk system, and as the original Music Maker had NO disk capacity at all, I assumed that this was the case with all, but not so, we have yet another module appearing in two different guises.

SOME Music Maker modules apparently let you save to disk, but then don't let you load back in, the simple reason being that Music Maker saves a FIXED LENGTH data file. AND that length of data file requires you to use CALL FILES(0) to load it back in. And you can't use CALL FILES(0) can you!

MICROpendium December 1987 gave a complex solution, requiring no less than 14 complex steps! which change the length of the saved file from the former fixed 63 sectors (Display Fixed 128) to 59 sectors, which apparently you can then load in.

One of our members however has a module which saves in the required 59 sectors, but then will only load in 43 sectors.

These are differences of 1k and 4k respectively, and sound like nothing more than sloppy programming by TI - and suggest at least three versions of the module!!!!

You could use a sector editor to change the file length, taking care to see if the data is stored at the start of the file (no problem) or the end of the file (erase early sectors and move header forward,ugh).

It would be very nice if, now you know what is needed, one of our talented machine code programmers were to write a routine to input the original file name, read the disk, and make all the necessary changes for you. I can't do it! Consider yourself challenged!

DISK DRIVES: SECTORS/TRACKS/DENSITIES/SIDES...aaaagh!

To allay a little confusion:

TI Standalone disk controller:

Single Sided 40 track Single Density 360 sector 90k 9 sec/track TI PEB Card: above OR

Double Sided 40 track single density 720 sector 180k 9 sec/track Myarc/Corcomp PEB cards:

above OR:

Double Sided 40 track double density

1440 sec 360k 18 sec/track OR

1280 sec 320k 16 sec/track.

Myarc OPTIONAL extension to Myarc PEB Card:

Double sided 80 track QUAD density (no other data to hand).

COMPATABILITY:

A disk recorded on a system higher in the list can be read on a system lower in the list (except 80 track).

A disk recorded on a system lower in the list may not be read on a system higher in the list.

Except that a single sided system may be able to read SOME files on a double sided disk.

The Corcomp and Myarc controllers can both read and write in single OR double density, so you can still record SSSD on these!

DISK DRIVES: All systems should be capable of working

with any 40 track "standard" disk drive. An 80 track drive is of value only if you have the extended Myarc controller card.

You may need to set small switches in the drives to match the controllers- if disk access time is variable, the TI Controller does not take advantage of faster access times whereas the CorComp and Myarc controllers can- Myarc can use an access time as fast as 6ns. There is no incompatibility unless the disk drive is TOO SLOW, which should be unlikely these days! ALL controllers can work with the same maximum access time as the original TI controller, which is much slower than any modern drive. The TI disk controller has a "hardware" fixed speed. The other controllers can be made to function more efficiently with the faster modern drives.

A letter from a member dated December 1987 advises of the partial delivery of an order sent to PILGRIMS PRIDE in October 1986. Speedy!

MYARC RAM CARD:

Why can you only use 400k of the Myarc ram card for disk emulation? Because the disk usage is BIT MAPPED into an area of the disk which can only map 400k. There is no room to map more- if you want a bigger disk you must use a different mapping system- either increase the area in which the bit map is held, or allow a "floating" bit map on an "as required" basis. That would mean you would need a new disk controller- your ram card would not be strictly compatible with your old software! Myarc were right to stick to 400k. The remainder is available for print spooling, and for your operating system to use as part of the normal computer memory map area.

compatible with your old software! Myarc were right to stick to 400k. The remainder is available for print spooling, and for your operating system to use as part of the normal computer memory map area.

FULL language, and is not to be confused with the ANSI 77 SUBSET.

"The beauty and speed of the TI99/4A is brought out by the fortran program. There are of course limitations such as: having to write a lot of subroutines because the compiler cannot handle large programs; requiring a deep pocket to stock up with printer paper; having sufficient interest to find out how the IBM mathematicians brains work, etc. It is perhaps worth mentioning that assembly language subroutines may be called on as well as Fortran subroutines, thus widening the applications field considerably." Thanks George.

Overwhelming vote in favour of reprinting details of the extra calls available when PRK or Stats modules are inserted (or are inserted into memory from disk). The first part of the article is submitted for this issue.

EXTENDED BASIC PROGRAMMING TIP:

Courtesy John Seager:

Using ACCEPT AT for a numeric variable, how do you idiot proof it so the program will not bomb? You can insert a default input value, and use a negative size, and also use VALIDATE.

However, the user CAN blank the default variable with

CLEAR, and if the input variable is a numeric variable, trying to input a blank will cause an error condition.

I have frequently seen authors input all numbers into a string variable. Inputting a blank does not cause an error, and you can test for a nul input and go back if required.

You dont have to do it that way. John points out that using ON WARNING NEXT will test for the nul input and go back for you. Try it.

eg. 100 ON WARNING NEXT

110 ACCEPT AT(4,5)ERASE ALL VALIDATE(DIGIT):A

120 GOTO 110

John also mentions that if you program:

100 ON ERROR 600

110 RUN "DSK1.NOFILE"

600 ON ERROR 600 :: RETURN

the RETURN will fail as the failed RUN seems to remove the internal pointers.

This is a deliberate ploy by TI to avoid the "accidental" removal of the List Protection flag- which happens with Version 100 of ExBas. You need to use the format RETURN XXXX where XXXX is a line number to go to which will RUN your original program again- you could use RUN 110 or something if required. The second RUN will of course reset all required pointers.

END OF ARTICLE

MULTIPLAN MACHINATIONS

By BILL HARMS - ROM - AUG. '86

In this article I will introduce you to a method to transfer data from a basic program to multiplan. I use Multiplan to keep my budget and to estimate income taxes. I have a spreadsheet with 18 columns: 12 months, yearly total, year-to-date, weekly average, monthly average and two for taxes. Those last two have formulas to get various numbers from the spreadsheet. The rows include: pay, interest, expenses, loans and other. You can really do "WHAT-IFing" and "WHY-NOTing" with Multiplan.

I use a nice fast (I mean fast) XB program I wrote to add all my transactions by category. Then I use a SYLK creator to quickly and correctly prepare them for loading into my Multiplan spreadsheet. SYLK (Symbolic Link) files are a little known feature of Multiplan. They can be written to disk by a basic program and read by Multiplan.

This material is based on a program I got from TI, a series of articles in the May (and later) 1985 SUPER 99 MONTHLY (now called THE SMART PROGRAMMER) and the Multiplan manual.

This bare bones program is based on the one I received from TI in 1984. The disclaimer was bigger than the program! It writes a disk file with a one cell spreadsheet that can be read by Multiplan.


```

100 OPEN #1:"DSK1.SYLKF",DISPLAY,OUTPUT,FIXED 128
110 CALL CLEAR
120 INPUT "ROW NUMBER: ";R$
130 INPUT "COLUMN NUMBER: ";C$
140 INPUT "CELL CONTENT: ";A$
150 FOR Q=1 TO 27-LEN(A$)
160 W$=W$&CHR$(Q)
170 NEXT Q
180 X$=CHR$(34)&A$&CHR$(34) ! SURROUNDS CONTENTS
        WITH QUOTES
190 Z$=CHR$(13)&CHR$(10) !CARRIAGE RETURN AND LINE
        FEED
200 Y$="D:PMP"&Z$&"F:D606B"&Z$&"B;Y"&R$&"X"&C$&Z$&
        "C;K"&X$&Z$&"W;N1;A1 1"&Z$&"D"&Z$&W$
( This monster of a line has the symbolics needed
for Multiplan to read the file. See page 205 of the
Multiplan manual for explanations ).
210 PRINT #1:Y$
220 CLOSE #1
230 END

```

If you enter and run this program, you will find a file on your disk called "SYLKF". Before you can load this file, you must change it. It may seem a bit odd, but the file must be written as DISPLAY, FIXED 128 and then changed to INTERNAL, FIXED 128 in the file header. In other words, the file must use DISPLAY notation but must look like an internal file. There are two ways to do this. You will find an application of Barry Traver's RAW (read and write) in SUPER 99 MONTHLY. Or you can use ADVANCED DIAGNOSTICS to change the four hex characters of the first line of the file header to (0202)

Once you have done this you can load your file. First, boot Multiplan. Press (T)ransfer and then (O)ptions. Next press (S)ymbolic and then ENTER. Now press (T)ransfer again and this time (L)oad your file.

Here is the Multiplan spreadsheet

```

-----
          1
1 HARMS
-----

```

This is what the data looks like on disk using Millars Graphics great ADVANCED DIAGNOSTICS

```

Drive : 2 Track   : 3
Side  : 1 Sector  : 34
Byte  : 0 Display : Ascii

```

```

ID;PMP$F;DGOG
B$B;Y1;1$C;
K"HARMS"$W;W1
;A1 1$E$E$E$E$E

```

Most of the \$\$'s stand for CR/LF (Z\$ in line 190) There are many ways you could input data besides the single INPUT in line 140. You could read data from DATA statements or from a disk file. That disk file could be created by almost anything: TI-Writer, RS232, another module or a Multiplan Print File.

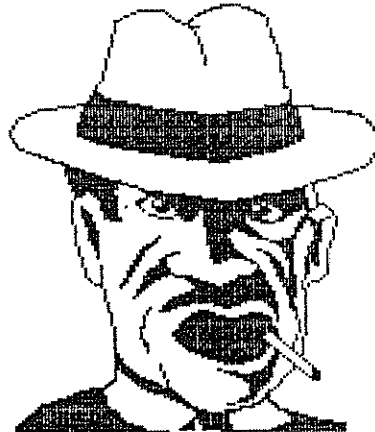
You can create data in Basic and then "dump" it into a spreadsheet en masse' instead of just keyboarding it. You can transmit the outputted SYLK file of your Multiplan Spreadsheet to others via RS232. The DIF (Data Interchange Format) used by Lotus 1-2-3 and Visicalc only accommodates the cell content, not the sheet parameters.

This is only a taste (BAD?) of what you can create to load data into Multiplan. It really opens up Multiplan to other software.

(Courtesy of SPIRIT OF 99 JAN. '88 p.7)

Retyped for TEXPAC BBS by Alistair Leslie of TISHUG.

END OF ARTICLE



TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month	\$ 2205.70
Expenditure for previous month	\$ 2153.49
Profit for previous month	\$ 52.21
Membership accounted for \$ 105.00 of income	
Shop sales	\$ 2100.70 of income

The expenditure was made up of the following :-

Administration	\$ 80.00
Printing and posting of TND	\$ 250.39
Shop purchases	\$ 1823.00

You might be interested to know :-

Membership dues received in the last financial year ...	\$ 2460.00
Printing, posting & stationary for TND cost.....	\$ 3194.00

BUYERS GUIDE: MICE

Mice make working on your PC easier by helping you to point at things you want to. Buying a mouse seems to be left to last, but the mouse probably performs almost 40% of all your input tasks. Getting the wrong one can be at the least very annoying. We give you a few tips on what to look for.

So buying a mouse is just a case of selecting the make, the right plug on the end of the wire and going for it, right?

Wrong. Mice can be small, irritating, irrational and dirty creatures. Just like the mammalian sort. So choose carefully and you can look forward to a long and happy life with your electrical pet. First, a few basic rules.

Rule 1) Cheapest is not best.

Rule 2) Most expensive is not best.

Rule 3) What you need is best.

So before you buy a mouse, check what is best for you. Mice come in a truly staggering array of types and families. First the families: Remote (that includes radio, infra red, ultrasonic); Two-button; Three button; Opto electronic; Mechanical; Ball, and finally Sensor types.

Remote mice use no wires (ordinary mice are wired directly into the PC) but need batteries to power the signals they send to a

receiver plugged into your PC. They are more expensive, usually a bit better made, very much more expensive, but really impress people no end. So if you are tired of wires all over your desk, get a remote mouse.

HOW MANY BUTTONS?

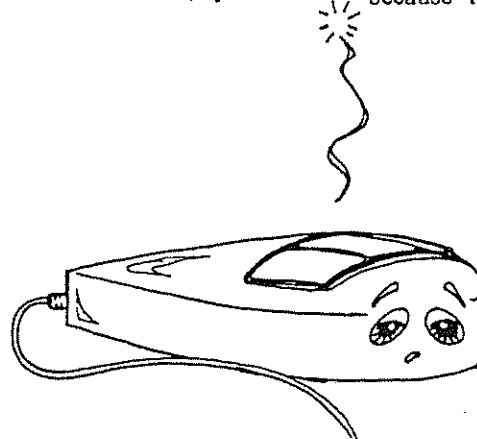
Choosing between two and three button mice is not too tricky.

All software uses at least one button (usually the left one although this can usually be altered if you want to). More and more software is beginning to use two buttons for extra features like context related Help, sub-menus and so on. So the chances are the third button will come in handy in the near future. Even if you don't use the third one, it won't do you any harm.

The choice between mechanical and opto-mechanical is a bit harder. Mechanical mice are usually more robust and resistant to dust and grime, but generally have less

sensitivity and accuracy than the more sophisticated optical types. Some new products - for example the new Honeywell mouse - have a completely sealed base to avoid dirt getting in. Well worth a look (Honeywell is on 02 699 0155).

Sensor mice, pens and



other similar devices have to be used on a special surface or pad, but can give excellent control and accuracy. Unfortunately the special pad usually puts up the price too.

Another option is the trackball. This is a sort of upside down mouse, which can be used either on a desk, clipped to a notebook keyboard, or used on a

train if you want to. Again, a problem with trackballs is dust and dirt getting into the works.

AVERAGE USERS

So decide what sort of mouse you want - for most average desktop users it will probably be a three button, opto-mechanical wired mouse. If you are doing high quality design or CAD (computer Aided Design) work, try a high resolution mouse, or possibly a sensor type and dedicated pad. For notebook and mobile users,

trackballs are the best bet because they can be used anywhere with out trailing wires all over the place or having to rub the mouse on your trouser leg. Once you know what you are looking for, check out the prices.

They can vary a lot - brand name mice are often made by the same people as less expensive ones, so see if you can find a brand name clone. They do exist.

Other things to look out for are a smooth, heavy ball, positive click buttons, removable section and good quality wire. These things all cause untold trouble to

cont....

POINTERS:

- 1) Look for a good price - \$50 is about average.
- 2) Check switches and ball for quality.
- 3) Ask about extra software or accessories.
- 4) If you are left handed, ensure you will be able to work OK.
- 5) Quality of wire connections is important.
- 6) Ease of cleaning is important.
- 7) Make sure it has the right connector for your PC.
- 8) Will it suit the kind of work you do?
- 9) And most of all, does it feel good?

cont from previous page

.....mouse users. A smooth ball is essential for non-jerky operation, and the heavier the better to keep it firmly on the mouse mat.

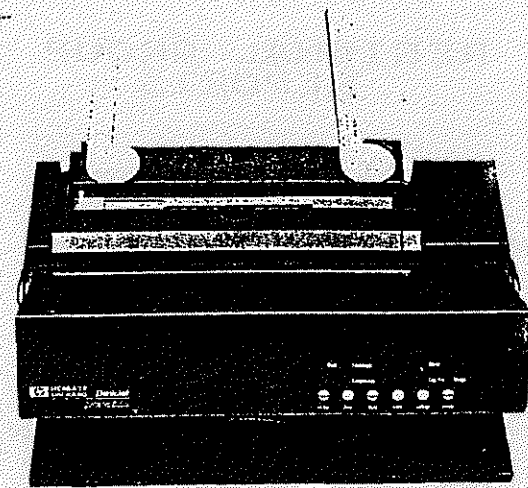
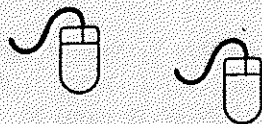
The buttons are usually the parts that get most action, so ensure they give

firm positive click. Weak and feeble clicks spell poor quality switches which often fail early. Any mouse that uses a ball should always have a removable section on the bottom to allow cleaning. Completely sealed units are a no-no - dust will soon render them useless.

The wire that comes out of the mouse should be supple yet firm. Thin, easily kinked wire, poorly joined plugs and little or no wire support where it joins the mouse and plug all point to early failure and bye bye mouse. And last, make sure the plug on the end of the wire fits your PC. Three types (PS/2, 9-pin and 25-pin) are all currently available.

EXTRAS

Most mice come with some basic software to make them work, some have fancy paint packages thrown in, and some have specials like mouse pockets and mouse mats in with the bundle. All these things add to the value of the product, so shop around for the best deal. Happy mousing. ☐



HP Deskjet Portable

FOR SALE

DESKJET PORTABLE PRINTER

For just \$250.00 you could own this superb Hewlett Packard portable printer, it comes with a cut-sheet feeder which connects easily to the printer and provides no-hands sheet feeding. Holding up to 50 sheets of 16 lb.(60 g/m²) paper or 40 sheets of 20lb.(90g/m²) paper, it gives you more convenience, especially when printing multi-page documents. Other types of MEDIA you can use are: transparency film, labels, coloured photocopy, bond, letterhead paper and preprinted forms

With this printer you are able to print in PORTRAIT mode or LANDSCAPE mode

Print speed: Letter quality mode 167 cps at 10 cpi

Draft quality mode 240 cps at 10 cpi

The very easy to follow user's guide which is filled with 14 chapters of usefull information

Power supply and all interface cables

Plus a spare print head

Rechargeable battery installs easily in the printer and last for up to 100 pages of printing. Great for when you print away from the office or home, It charges in the printer. The battery is not supplied.

This printer is currently on sale in one of the leading electronics stores for \$499.00

I am sure that the first person to see this deal will buy it, so bring your money and see the TISHUG shop for an on the spot demo.

**** FOR SALE ****

Owing to upgrading I have these items to sell. 1 486SX-25 MotherBoard, Intel CPU, AMI Bios, No RAM \$135.00. Floppy and AD Controller \$15.00

All in good going condition when removed. Contact Gordon (02)874-1445 A/H if interested or leave a message here.

REGIONAL GROUP REPORTS

Meeting Summary For AUGUST

Central Coast 12/08/95 Saratoga
 Glebe 10/08/95 Glebe
 Hunter Valley 13/08 20/08/95
 Illawarra 08/08/95 Keiraville
 Liverpool 11/08/95 Yagoona West
 Sutherland 18/08/95 Jannali

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 or third Sunday of each month at members homes starting at 3pm. Check the location with Geoff Phillips by leaving a message on (049) 428 617. Please note that the previous phone number (049) 428 176 is now used exclusively by the ZAP BBS which also has TI support. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the first Tuesday of each month after the TISHUG Sydney meeting at 7.30pm, at the home of Geoff Trott, 20 Robsons Road, Keiraville. A variety of investigations take place at our meetings, including Word Processing, Spreadsheets and hardware repairs. 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 10.30 PM or at work (02)602 3312 Liquorland Liverpool West for more information.

*** ALL WELCOME ***

11th AUGUST 1995 : MY PLACE

8th SEPTEMBER 1995 : MY PLACE

13th OCTOBER 1995 : MY PLACE

Bye for now Larry.

Liverpool Regional Co-Ordinator

SUTHERLAND Regional Group

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 on the first Saturday of the month. They are held at the MEADOWBANK PRIMARY SCHOOL, on the corner of Thistle Street and Belmore Street, Meadowbank. Cars can enter from Gale Street and park in the school grounds. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

AUGUST MEETING - 5th AUGUST

SEPTEMBER MEETING - 2nd SEPTEMBER

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

SEPTEMBER 12th AUGUST

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

RELATIVE FILES.

KEVIN COX SARA CAME UP WITH THIS VARIATION OF SPIRIT OF 99'S IRWIN HOTT'S ONE LINER.

```
10 OPEN #1:"DSK1.",INPUT,INTERNAL, RELATIVE
20 INPUT A::INPUT #1,REC A:A$
30 PRINT A$::GOTO 20
```

*KC SARA'S VERSION

```
100 OPEN #1 : "DSK1.",INTERNAL,RELATIVE
110 FOR FILE=0 TO 127
120 INPUT #1,REC FILE:A$
130 PRINT A$,
140 IF A$="" THEN END
150 NEXT FILE
```