

MIKE GODDARD COMPUTER SUPPORT

NEW AND USED COMPUTER EQUIPMENT BO

TI-99/4A SPEC

SPARES

COMPONENTS

RESIST

BATTERIES

POV

POWER CABLES

PRINTERS

CAPACITORS

DISK DRIVE CABLES

FOR LATEST LIST CONTACT:

M.G.C.S. "SARNIA", CEMETERY ROAD, RHOS, WREXHAM  
CLWYD, LL14 2BY

Tel: (0978)843547.

MBX: 022212529

# TI\*MES

## TI99/4A USER'S GROUP (U.K.) CONTACTS.

Vice Chairman, Journal Exchange, & Disk Librarian: Stephen Shaw Tel. 061 432 6097(8pm+)  
 (Letters preferred) 10, Alstone Rd., STOCKPORT, Cheshire. SK4 5AH  
 General Secretary: Jim Ballinger Tel. 0332 772612  
 5 Offerton Ave. DERBY. DE3 8DU  
 Membership Secretary Telecoms & TI\*MES Back Nos.: Peter Walker Tel. 0707 873778  
 24 Bacons Drive. CUFFLEY, Herts. EN6 4DU  
 Treasurer: Alan Rutherford Tel. 0625 524642  
 13 The Circuit, WILMSLOW, Cheshire. SK9 6DA  
 Publicity Officer: Phil Trotter Tel. 0642 817356  
 80 Martonburn Rd., MIDDLESBROUGH, Cleveland. TS4 2TH  
 Programming: Mark Wills Tel. 0743 64177  
 37 Abbots Rd., Monkmoor Estate, SHREWSBURY. SY2 5PZ  
 TI\*MES Editor & Distribution: Alan Bailey Tel. 081 508 1053  
 14, Shelley Grove, LOUGHTON, Essex. IG10 1BY  
 Hardware & Projects: Mike Goddard Tel. 0978 843547  
 Sarnia, Cemetery Rd., Rhos., WREXHAM, Clwd. LL14 2BY  
 Librarians: Cassette: Nicky Goddard Tel. etc. as above.  
 Modules: Edward Shaw Tel. 0538 360382(fpm to 8pm.)  
 Crow Holt Farm, Basford, LEEK, Staffs. ST13 7DU  
 Publications: Mike Curtis Tel. 0209 219051  
 21, Trelliske Rd., Roseland Gdns., REDRUTH, Cornwall TR15 1QE

## MAGAZINE CONTENTS

IFC Editorial, disclaimer, copy date.

- p1 Membership News by Peter Walker.
- 2 Module Library Report by Edward Shaw.
- 3 Disk Library Report by Stephen Shaw.
- 5 More Telecom Tips by Peter Walker.
- 6 For Sale
- 7 TI LOGO Database by Peter Walker.
- 13 On the Hardware Front by Mike Goddard.
- 14 Cassette Reviews by Nicky Goddard.
- 15 RTTY for the 99/4A by Wilson & Crawford.
- 21 Plotting from University of Dallas(MG)
- 24 Truchet Tiles by Stephen Shaw.
- 25 TI World News by Jim Peterson(SJS).
- 26 Machine Code Tutorial by Mack McCormick(SJS).
- 31 Module Review by Ashley Tilling(SJS).
- 33 Rambles by Stephen Shaw.
- 44 Reviews by Stephen Shaw.
- 48 Answers to Readers by Stephen Shaw.
- 49 XB Tutorial by Tony McGovern(SJS)
- 54 Tips from Tigercub by Him Peterson(SJS).
- 57 Additional News by Jim Peterson(SJS).
- 61 TI Bits by Jim Swedlow(SJS).
- 66 Pattern Program by Stephen Shaw.
- 68 Graphics Program by Stephen Shaw.

#### EDITORIAL

First of all an explanation and apology. In the last issue I included a contribution which contained a program in the form of a print-out not reproducible by the copying machine at the magazine printers. This led to the necessity of changing 10 pages of text already at the printers and updating the contents page, which had already been printed. Hence the separate contents sheet and a late despatch. Contributors! please look very critically at your copy. I no longer have access to free, to the group, copying, so I do not necessarily spot such unsatisfactory copy before it gets to the printer.

This issue includes the offending article itself but the program is available from the disk library.

The committee has decided to offer two issues each year enlarged to accommodate the wealth of material at present coming forward. These will be the Autumn and Winter ones, and this issue is 76 pages including covers, the maximum conveniently possibly with the normal staples used by our printers.

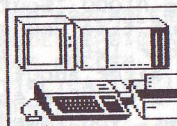
I would particularly like to draw your attention to the description of the Continental scene in Jim Peterson's "Additional News". Although prices seem to be high to us, there is every evidence of strong, continued, support there for our computer!

#### DISCLAIMER

Views expressed in this magazine are those of the contributor him or herself and not necessarily supported by the Committee.

#### COPY DATE FOR NEXT ISSUE

Please let me have any copy for the next issue by 1st. February.



## MEMBERSHIP NEWS

from Peter Walker

Membership Secretary

Since issue 30, we welcome the following new members: Chris Stacey, Victor Watson and Harry Turkington. Last time's announcement concerning back issues brought a small flurry of sales. Would members note that since the last issue postage has increased again, so please add enough for your return package.

I've had 9 responses to my request for information from printer owning members. These are still trickling in, so I'll put off my survey until next time, in the hope that a few more will respond. If you do own a printer, please write to me!

In the last issue I gave details of the Southern Area User Show & Workshop at my home village of Cuffley in Herts. This is still on for Saturday 26th January 1991, 11am to 5pm, entrance £2. This is the ideal opportunity to meet other 99ers and see the TI99/4A in action. A map of Cuffley and approach roads was printed on the back cover of issue 30 and further details are given below. Could all members who intend bringing their systems to demonstrate, or intend to sell items at the Show, please confirm with me as soon as possible. I hope all members in easy reach of the London area will come along. We enjoy meeting all our members as this gives us both information and encouragement to continue our task of directing the group and the contents of TI\*MES.

#### Directions to Cuffley.

I expect most car drivers to approach Cuffley from the M25. If coming from the West, exit at Junction 24 and take the first roundabout exit to Potters Bar. After half a mile, take the right filter at the first traffic lights. After another half a mile, take the right filter at the next set of lights (ignoring the pelican crossing lights). At this junction you need to take The Causeway which is "at one o'clock" almost ahead of you, rather than the side road at hard right. Follow this winding road for 3 miles or so, passing through the village of Northaw on the way. After entering Cuffley, Station Road is about one third of a mile on the right at a triangular junction on a distinct slope, opposite St Andrews Church on the left. This junction is marked "Goffs Oak".

If coming from the East, exit the M25 at Junction 25 and take the third exit, the A10 North. At the next roundabout junction, take the B198 to Cuffley. At the next roundabout, turn left on the B156 to Cuffley. Follow this road for about 2 miles, passing through Goffs Oak on the way. Cuffley is immediately entered past the railway overbridge.

If approaching from the North via the A1, exit the A1M at Junction 3 at Hatfield, just after the motorway tunnel. Take the first exit on

the roundabout and go straight over the mini-roundabout which immediately follows it. At the next roundabout, turn left; at the next go straight over. Carrying straight on you join the A1000 South. At the next roundabout, carry straight on: after a short time you see the large BBC radio transmitter at Brookmans Park on your left. Take the second junction left, marked to Cuffley, which is after about half a mile. This is Shepherds Way, which you follow for a mile or so until you reach a sharp right hand bend. Round this corner, you immediately turn left down The Ridgeway. This is real "switchback" road; you continue on this road until you enter Cuffley, down Plough Hill. Station Road is on the left at the bottom.

For those coming from the North via the M1, you can either carry on to the M25 and proceed as described above, or exit earlier at the M10 Junction 7 and continue along the A414 until you reach the A1 junction 3, from where you reach Cuffley as described in the last paragraph. Any people coming South via the A10 should turn right at the first traffic lights at Cheshunt (the first for many miles save for the pelican lights just 100 yds before this). Go left at the first mini-roundabout and right at the next and straight over the next main roundabout and on to Cuffley via Goffs Oak as described above.

If travelling by train, these leave Kings Cross (bound for Hertford) at 15 and 45 minutes past the hour, usually from the separate suburban platforms 9 to 11 to the left of the main train shed. Return trains are at 24 and 54 minutes past the hour.

When you reach Cuffley, Cuffley Hall is reached from the upper walkway, above the Library along Maynard Place. For those bringing equipment, this can be brought into the Hall via the back entrance from the car park at the rear, though the latter gets very full on a Saturday morning. The hall should be open from 9am for those wanting time to set up. There are two other car parks in Cuffley, one at the station, which is free on Saturdays, and one almost opposite the station in Sopers Road.

I look forward to seeing as many of you as possible on 26th January.  
Peter Walker

**MODULES MODULES MODULES MODULES**

**APPEAL TO ALL DISK DRIVE OWNERS.....**

Do you have any modules that you would consider selling or donating to the module library. Reasonable prices paid. For more information please contact me at the address given below.

The latest list of modules available for purchase follows; please note that cheques should be made payable to "E.H.SHAW". Also members are advised to contact me about the modules that they are seeking as the stock is constantly changing.

ADDITION AND SUBTRACTION 1	2.00	PARSEC	3.50
ADVENTURE and PIRATE TAPE	3.00	HOUSEHOLD BUDGET MAN.	3.00
		PROTECTOR	4.50
		SHAMUS	3.50
BEGINNING GRAMMAR	3.00	DEFENDER	4.00
		TUNNELS OF DOOM	4.00
CHISHOLM TRAIL	3.50	OTHELLO	4.00
CONNECT 4	3.50	THE ATTACK	3.50
		TI INVADERS	3.50

* DISK MANAGER II	4.00	STATISTICS (NO DOCS)	3.00
EARLY READING	2.50	* TI LOGO I + CASSETTE	
EARLY LEARNING FUN	2.50	+ MANUAL 32K REQD.....	12.50
* EDITOR ASSEMBLER + MANUAL	22.50	* TI MULTIPLAN/DISK/MANUAL	25.00
* EXTENDED BASIC AND MANUAL	22.50	* TI WRITER NO MANUAL	5.00
INDOOR SOCCER	4.00	TOMBSTONE CITY	2.50
MULTIPLICATION 1	4.50	VIDEO CHESS	4.50
MUNCHMAN	4.00	VIDEO GAMES 1	3.50
MINIMEMORY + MANUAL	10.00	YAHTZEE	4.00
NUMBER MAGIC	3.50		
DEMONSTRATION	3.50		

Cassettes for adventure module (please write) 50p each.

\* MODULES MARKED WITH AN ASTERISK REQUIRE DISKS OR 32K RAM OR BOTH. ALSO PLEASE NOTE THAT EARLY READING NEEDS A SPEECH SYNTH TO RUN.

**PURCHASING MODULES FROM THE LIBRARY**

You may return any module purchased within four weeks and be refunded the purchase price less postage which will be charged at the rate of 40 pence per module.

**Application to loan/purchase modules.**

Name:..... Modules required:.....  
 Address:.....  
 .....  
 .....

I enclose cheque/PO for £.....(as indicated on the list) & post to  
**PLEASE MAKE CHEQUES PAYABLE TO E.H.SHAW.** MR. E.H. SHAW  
 Foreign orders can only be accepted if a CROW HOLT FARM  
**BANKERS DRAFT is enclosed drawn in STERLING** BASFORD  
 on a LONDON bank. It also helps if a little LEEK  
 extra is added on for postage overseas. STAFFS. ST13 7DU

**DISK LIBRARY REPORT...**

Stephen Shaw. 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

**NEW DISKS ADDED since SEPT 21st 1990:**

>JIM PETERSON. All new contents! Two programs for TE2 and J/S for those who find speech impossible or difficult, including a version with large display for those with fading eyesight; programs to work math out to an (almost) unlimited number of digits-lots more than 13 anyway! ; and articles written by Jim in 1990

>USER GROUP LIST BY JIM PETERSON is now updated to August 1990, with all groups not heard of for 18 months deleted, and groups known to have disbanded so marked. Only TIUGUK is listed for the UK with a 1990 entry.

>CRYPTOGRAM-SONG VERSES. One hundred verses to be encrypted (replacement code) and decoded by you. From Jim Peterson.

>CRYPTOGRAM-BIBLE VERSES. As above but maybe easier? as the contents may be more familiar!

>GEORGE TELLS YOUR FORTUNE by Leo W DuBry. The computer selects ten playing cards and tells your fortune with them. Prints out to PIO and has an Epson screen dump of the cards. This program is absolutely for fun only- complete with spelling errors! The print out is about two pages long, depending on the length of selected phrases.

>MULTIPLAN TUTORIAL (TWO DISKS). These disks contain the text and spreadsheet sheets from a good Multiplan book published by Sams, with the advantage that the text has received the little alterations required for the TI, making the disk better than the book! While the spreadsheets are provided, you will learn more by keying them in yourself! Drop a line to Herbert Schlesinger to thank him for his hard work! A valuable addition now that Multiplan books are hard to find.

>FUNLWEB VN 4.31. 40 column VERSION (SPECIFY!) (THREE DISKS!).

The main change this time is to officially drop DM1000 and DISKPATCH (unofficially included in archived form) and replace them with a rather more powerful form of DISKREVIEW.

>FUNLWEB 4.31 80 column SUPPLEMENT (TWO DISKS) extra files and docs for owners of 9938 video chips! No significant changes in this release.

>GEORGE 8 now has ADDITION AND SUBTRACTION 1, EARLY LEARNING FUN, AND NUYMBER MAGIC.

>TI DISK 1. Four ancient disk packages written a long time ago for American use, of archival interest mostly! Checkbook Manager; Personal Financial Aids; Programming Aids 1; AC Circuit Analysis. This lot would have cost a fortune back when... US\$85 I make it!

>GAMES TIs PLAY DISK 1 from the book of the same name, files on this disk are: Golf, Matchkey, Mubblechase; Nim; Numaway; Reverser; Robotchase; Schmoos; Sci-Fi; Stranded; Target; Tictactoe; Transition; Twinky; Word Scramble. Nothing too fantastic but if you have the book now you dont need to key them in...

>GAMES TIs PLAY DISK 2. Files are: Artist; Barrel; Biorhythm; Block-Em Braintease; Connect5; Craps; Digits; Faucet; Flim Em; Gruestew; Hangman; IRSMAN (eg tax collector man); Itche; Kingdom; Learner; Magic Squares.

>ODDMOD4...OLD modules (1978) written for Americans- Home Financial Decisions; Tax/Investment Record Keeping; Personal Real Estate. At launch prices these would have cost US\$170!!!

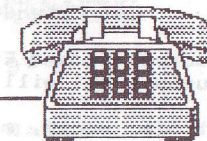
>UTIL26. CadMaster Vn 1.9 by John Miller-a graphics program with no print facility and non standard disk storage; XB2PP by Ed Johnson which will translate an XB screen to PagePro format; Ramtest for 32k by E L Wilkerson-requires ram at >6000; MCFLPRINT a MacFlix printing program in XB offering a variety of print proportions using [ESC]\*\*(M)(N1)(N2) with M from 4 to 6, offering 72dpi,80dpi or 90dpi.

>24PIN. An up to date offering of programs which will print pictures using 24 pin printers- produces pictures from files which have been printed to disk by MacFlix or PagePro -not from their usual picture files. Disk includes a 265 sector picture of MM ready to print. Uses [ESC]\*\*(39)(n1)(n2) format.

All disks are SSSD and there is a copying fee of one pound per disk plus one pound post and packing, eg you send 4 disks and five pounds for a four disk order.

Stephen

MODEM



## MORE TELECON TIPS

Peter Walker describes the TE2 protocols

In the last issue, I mentioned some of the features of the Terminal Emulator II module (TE2). The full TE2 protocol manual is very detailed and contains much information which as a user you don't need to know. Its original purpose was to provide information for those designing host systems for such things as bulletin boards that TE2 users might access.

In this article I will describe the control and escape codes used by TE2 or at least the most simple ones. The more complex "Extended Writes" which transfer to the connected TI99/4A such things as character definitions, colours, sound, speech etc will have to wait to a later issue, as will the special coding used by Extended Writes for encoding binary data. It was an example of an Extended Write that I included in the last issue. For those who didn't try it out, the string when received by the TE2 causes the TI99/4A to transmit the sound of telephone ringing tone - continuously until another Extended Write cancels the sound.

Here are the control codes which are active when the TE2 is in the default 40 column Text Mode.

---CHARACTER---

ASCII CODE	CTRL	FUNCTION	COMMENTS	
1	SOH	A	Home Cursor	Cursor goes to Top Left. All text destroyed when next character is received.
7	BEL	G	Beep	
8	BS	H	Backspace	Destructive backspace
10	LF	J	Line Feed	Cursor advances one row in the same column.
12	FF	L	Clear Screen	Leaves cursor at Top Left
13	CR	M	Carriage Return	Cursor returned to left of current row, but is invisible until next character received.

The following Escape Codes are active in both Text mode and the 32 column Graphics mode. In each case the code should be preceded by the Escape character ASCII 27, which is Control . (fullstop)

Y <C> <R>	Locate Cursor	C = Column + 32	R = Row + 32
H	Home Cursor	Leaves cursor at Top Left but invisible, screen is not cleared.	
y	Enter Graphics mode (from Text mode)		
z	Enter Text mode (from Graphics mode)		

: Lock Keyboard  
; Unlock Keyboard  
G Extended Write Details in a later article maybe!

Escape 9 and Escape 8 are also used by TE2 (System Reset and Read Buffer) but will not be dealt with here.

It is worth mentioning that the Control codes are not active in Graphics mode; characters 1-31 are graphics characters, indeed characters 1 to 9 (CTRL A to I) are the 9 components of the Texas Instruments symbol normally seen on the Title Screen. In Graphics mode there is no visible cursor.

By way of explanation of the Locate Cursor escape code, this is how you would display the TI symbol in the centre of the screen. The system must already be in the Graphics Mode.

```
<Esc>Y.+<Ctrl A><Ctrl B><Ctrl C>  
<Esc>Y.,<Ctrl D><Ctrl E><Ctrl F>  
<Esc>Y.-<Ctrl G><Ctrl H><Ctrl I>
```

When you change from Text mode to Graphics mode and vice versa, you must allow at least 90ms before any other characters can be received. For those unfamiliar with these two modes, Text mode allows 40 columns, but only the background and foreground colours can be changed. In Graphics mode, there are 32 columns and, as in Basic, each set of 16 characters can have different colours and all characters are redefinable. However, all these colour and character changes need Extended Writes to activate them, which as I said above, I hope to cover in later articles if there is sufficient interest. Do let me know!

Peter Walker

#### FOR SALE

Modules:(with instructions) Household Budget,PRK,Tombstone City,Bigfoot,Music Maker, Honey Hunt,Beginning Grammar,Jungle Hunt,Shamus,ProtectorII,King of the Castle,eaf2.50  
Disks:TI-Count(full accounts suite)£45(cost£80),99 WriterII£2.50,Bitmac(art&grs)£2.50  
Better Banners & Banner Graphics £2.50  
Books:TI Games£4, TI99/4aSound&Graphics£4,Mastering the TI£1.50,Kids & the TI£1.50  
Your first TI99/4a Program£2.50. H'ware:Speech Synthesiser£8.C.Mehew Phone0469 8404

Modules:XB,Music Maker,Mini Mem,Chess.ea.£15(all£50)Tombstone City,Speech Editor, Attack,Beginning Grammar,Soccer,TE,PRG,PRK,Household Budget,Typing Tutor,eaf5(all£40)  
Disks:18 TIdisks & manuals(phone for details)£10. TI Logo2(Disk cassette manuals£250)  
Books:Assembly Language Guide,Multiplan Workbook,99/4a Favourite Programs Explained, Getting Started With the 99/4a,Dynamic Games,Tantalising Games.All mint.All£10.  
20 copies of TI\*MES 4-20,24-27.All mint.Offer.  
Hardware: TI Silent 700 80column printer/terminal. £25.  
M.Kitchen.55,Carey Way,Olney,Bucks.MK46 4DR. Phone.0234 712340

Modules:A-Maz-Ing£3.50,Blastof£3,PRK£4,Early Reading(Spch req)£2,Adventure£4.50, Alpiner£4  
Cassette:TI99/4a Calc(Spreadsheet)£4.50,Teach Yourself XB+others£4  
Books:Compute's 1st Book of TI Games£4.50,Guide to TI99/4a(regena)£4.50,Mastering the TI-99(Brooks)£2.50,9900Family Data Book£4.50,Layflat Users Guide£4, XB ManualOnly£4,  
Hardware:Double Cassette Lead £2.50. All prices include postage & are negotiable.  
Wanted Games Modules(esp.3rd. party),Cheap hardware. Exchanges.L.Watson0245 72572(6pm+)



## TI-LOGO DATABASE

Stretching the LOGO language  
to the limits...

by Peter Walker

I last wrote about LOGO in Issue 24. I will not repeat what I said then, but suffice to say that LOGO aside from being an educational language for children, is also a fascinating interactive, procedural, recursive, list-processing language. However, LOGO cannot interact with disk files or with the printer (save for listing procedures) so is hardly an obvious choice of language to construct a database from. On the other hand, given these difficulties, a LOGO database is a challenge I couldn't resist.

In issue 24 I described the main features of a database. My aim was to provide as many of the essential features as possible, except for printing. The main features are: Review, Display, Add, Edit, Delete, Sort and Print. The key to creating and then saving data for later use is that when LOGO saves procedures, it actually saves the entire LOGO workspace. Thus global variables, as well as the procedures (the program) are saved by the SAVE command. When the database is RECALLED, so are all the data.

The method used is to make each data record a list of fields and each field a list itself: either a list with only one word or several. The entry for my name and address is structured as follows:

```
MAKE "WALKER [ [WALKER ] [PETER ] [24 BACONS DRIVE ] [CUFFLEY ] [HERTS  
] [EN6 4DU ] ]
```

By associating the list with its keyword, in this case the Surname, we also create a content addressable database, though I chose to make most of the functions menu driven. However I structured the program so that one could use it from the menu or from command mode, and I provided two forms of the display function: a menu driven DISPLAY command and a content addressable SHOW.ME command. Thus the command SHOW.ME :WALKER lists the entire list associated with the name.

I avoided making the database a single entity (which would become a list of lists of lists) since it would almost certainly exceed the total length of 127. What is not well documented is that, while you can save a data entity longer than 127, the file always crashes when RECALLED. However, in order to keep a track of what global variables are part of the database and what are part of the general procedures, I defined a list called "KEYLIST which holds a list of the keywords (ie surnames). The maximum amount of data which this database can hold is thus constrained by:

No record can be over 127 in length  
The Keyword list must be less than 127

Also by virtue of the way I wrote the program (but could be altered) the total number of records needs to be less than 22 in order that the Review list fits on one screen. (This is a mini-database: the principle is more important than the size!)

The diagram below shows the procedures map of the whole database. A description of each procedure follows.

HELP a standalone procedure that lists what commands may be used from the command level.

MENU is the top level menu routine. The line RUN SELECT (CHARNUM READCHAR - 48)... runs the selected program from the list, thus avoiding lots of separate lines of the form IF READCHAR = 3 THEN ADD.

REVIEW displays the keyword list. REVU is a recursive routine that displays a single line.

DISPLAY displays the contents of a record by selection from a review list. SHOW is a recursive routine that displays each field. The SHOW.ME :XXX command can display directly from the command mode.

ADD allows a new record to be created. IP.DATA is the recursive routine that enters each field item. CHECK ensures that a keyword is unique.

CHANGE allows a record to be edited. First the record is selected from a REVIEW list, then the field to be changed is selected from a list displayed by SHOWFIELDS. ALTER is a recursive routine that recreates the new record list from the old inserting the newly defined field item at the right point.

DELETE allows a record to be deleted. DELKEY is a recursive routine that creates a new keyword list from the old omitting the deleted keyword. There seems to be no way within a procedure of deleting the data itself.

SORT performs a bubble sort on the keyword list. SORT recurses until no change is detected in the sorted list. SCAN is a recursive routine that compares adjacent list entries and swaps them into the appropriate order using the test routine BIGGER?. BIGGER? is necessary since in LOGO you can't use the "greater than" operator > on word strings as you can in Basic. If BIGGER? doesn't separate the strings by the initial letter, it recurses down the string until it does separate them.

Printing the data is the largest challenge, since the only interaction with the printer is through the PRINTOUT command which lists all procedures to the printer or other device. The trick is achieved by using the DEFINE command to create a new procedure called DATA, which is unrunnable but is listable. This procedure contains one record on each program line. The other procedures are then erased. The last two procedures are erased and the final PRINTOUT are performed in immediate mode. It is essential therefore to save all changed data before the PRINT.DATA command is used. The ability in LOGO to create new procedures from other procedures is one of its more powerful and unusual features.

Overall, the program with 13 records consumes some 60% of the 3677 nodes available in LOGO, so there is scope for a larger database. The procedures used give good examples of list-processing in LOGO. If you expected graphics and turtles you will be disappointed!

Now for the listings. For those not wishing to type it all in, this program and the Roman numeral program from issue 24 are in the group disk library. I have added a comment line to each procedure indicating its function.

## PROCEDURES

```
TO HELP ☐  
CS ☐  
PRINT [COMMAND LEVEL PROCEDU  
RES ARE:] ☐  
PRINT [ ] ☐  
PRINT [REVIEW ] ☐  
PRINT [DISPLAY OR SHOW.ME :X  
XXX ] ☐  
PRINT [ADD ] ☐  
PRINT [CHANGE ] ☐  
PRINT [DELETE ] ☐  
PRINT [SORT ] ☐  
PRINT [PRINT.DATA ] ☐  
PRINT [MENU ] ☐  
END☐
```

```
TO MENU ☐  
CS ☐  
PRINT [ _____ TI - LOGO _____  
] ☐  
PRINT [ _____ - - - -  
] ☐  
PRINT [ _____ DATABASE _____  
] ☐  
PRINT [ _____  
] ☐  
PRINT [ ] ☐  
PRINT [PRESS ___ FOR ] ☐  
PRINT [ ] ☐  
PRINT [1 _____ REVIEW ] ☐  
PRINT [ ] ☐  
PRINT [2 _____ DISPLAY ] ☐  
PRINT [ ] ☐  
PRINT [3 _____ ADD RECORD ]  
☐
```

```
PRINT [ ] ☐  
PRINT [4 _____ CHANGE RECORD  
] ☐  
PRINT [ ] ☐  
PRINT [5 _____ DELETE RECORD  
] ☐  
PRINT [ ] ☐  
PRINT [6 _____ SORT ] ☐  
PRINT [ ] ☐  
PRINT [7 _____ PRINT DATA ]  
☐  
PRINT [ ] ☐  
PRINT [BACK _____ COMMAND MODE  
] ☐  
PRINT [ ] ☐  
RUN SELECT ( CHARNUM READCHA  
R - 48 ) [(REVIEW ) [(DISPLAY  
] [(ADD ) [(CHANGE ) [(DELETE  
] [(SORT ) [(PRINT.DATA ) ] ] ☐
```

```
IF OUTPUT "FALSE ☐  
END☐  
  
TO PRINT.DATA ☐  
; PRINTS DATA ☐  
CS ☐  
PRINT [HAVE YOU SAVED YOUR D  
ATA? ] PRINT [Y / N ] ☐  
IF NOT READCHAR = "Y THEN ST  
OP ☐  
MAKE "L HARDCOPY :KEYLIST [  
] ☐  
MAKE "L FPUT [ ] :L ☐  
DEFINE "DATA :L ☐  
ERASE "L ☐  
ERASE SHOW.ME ☐  
ERASE CHECK ☐  
ERASE BIGGER? ☐  
ERASE HELP ☐  
ERASE ALTER ☐  
ERASE SHOWFIELDS ☐  
ERASE SCAN ☐  
ERASE HARDCOPY ☐  
ERASE SORT ☐  
ERASE DELETE ☐  
ERASE CHANGE ☐  
ERASE ADD ☐  
ERASE DISPLAY ☐  
ERASE REVIEW ☐  
ERASE DELKEY ☐  
ERASE SELECT ☐  
ERASE SHOW ☐  
ERASE IP.DATA ☐  
ERASE REVU ☐  
PRINT [ERASE MENU PRINT.DA  
TA ] PRINT [IN COMMAND MODE  
] PRINT [THEN PRINTOUT ] ☐  
END☐
```

```
TO HARDCOPY :K :L ☐  
; CREATES LIST L FOR DEFININ  
G DATA PROCEDURE ☐  
IF :K = [ ] THEN OUTPUT :L ☐  
MAKE "L LPUT THING FIRST :K  
:L ☐  
OUTPUT HARDCOPY BUTFIRST :K  
:L ☐  
END☐
```

```
PRINT [ ] PRINT [PRESS ANY KE  
Y ]  
MAKE "AN READCHAR  
MENU  
END
```

```
TO REVIEW  
; PRINTS A SURNAME LIST  
CS  
REVV :KEYLIST 1  
END
```

```
TO REVU :K :N  
; LISTS ONE KEYWORD  
IF :K = [ ] THEN STOP  
TYPE :N  
TYPE [ ]  
IF :N < 10 THEN TYPE [ ]  
PRINT FIRST :K  
REVV BUTFIRST :K :N + 1  
END
```

```
TO DISPLAY  
; DISPLAYS A RECORD  
REVIEW  
PRINT [ ]  
PRINT [ENTER CHOICE ]  
MAKE "AN READLINE CS  
MAKE "TITLES [SURNAME:___ FOR  
ENAME:___ STREET:___ TOWN:___  
___ COUNTY:___ POSTCODE:___ ]  
SHOW :TITLES THING SELECT FI  
RST :AN :KEYLIST  
END
```

```
TO SHOW :T :D  
; DISPLAYS A FIELD  
IF :T = [ ] THEN STOP  
TYPE FIRST :T  
PRINT FIRST :D PRINT [ ]  
SHOW BUTFIRST :T BUTFIRST :D  
END
```

```
TO SHOW.ME :D  
; DISPLAYS A QUOTED RECORD  
CS  
SHOW :TITLES :D  
END
```

```
TO SELECT :N :K  
; OUTPUTS NTH ITEM FROM LIST  
:K  
IF :N = 1 THEN OUTPUT FIRST  
:K  
OUTPUT SELECT :N - 1 BUTFIR  
T :K  
END
```

```
TO ADD  
; ADDS A RECORD  
CS  
PRINT [ADD A RECORD ] PRINT  
[ ]  
MAKE "PROMPTS [SURNAME FOREN  
AME STREET TOWN COUNTY POSTC  
ODE ]  
MAKE "E IP.DATA :PROMPTS [ ]  
IF CHECK FIRST FIRST :E :KEY  
LIST = "TRUE THEN PRINT [SUR  
NAME ALREADY EXISTS ] WAIT 2  
00 ADD  
MAKE FIRST FIRST :E :E  
MAKE "KEYLIST LPUT FIRST FIR  
ST :E :KEYLIST  
CS SHOW :TITLES :E  
END
```

```
TO IP.DATA :P :E  
; INPUTS A FIELD  
IF :P = [ ] THEN OUTPUT :E  
PRINT FIRST :P  
MAKE "E LPUT READLINE :E  
OUTPUT IP.DATA BUTFIRST :P :  
E  
END
```

```
TO CHECK :E :L  
; CHECKS IF :E IS IN LIST :L  
IF :L = [ ] THEN OUTPUT "FAL  
SE  
IF FIRST :L = :E THEN OUTPUT  
"TRUE  
OUTPUT CHECK :E BUTFIRST :L  
END
```

```
TO CHANGE  
; EDITS A RECORD  
REVIEW  
PRINT [ ]  
PRINT [ENTER CHOICE ]  
MAKE "N FIRST READLINE  
MAKE "PROMPTS [SURNAME FOREN  
AME STREET TOWN COUNTY POSTC  
ODE ]  
CS MAKE "NM SELECT :N :KEYLI  
ST  
PRINT :NM PRINT [ ]  
SHOWFIELDS :TITLES 1 THING :  
NM  
PRINT [ ]  
PRINT [ENTER FIELD TO CHANGE  
 ]  
MAKE "M FIRST READLINE
```

```
CS TYPE :NM TYPE [ ] PRINT  
SELECT :M :PROMPTS  
PRINT [ ] PRINT [PRESENT VAL  
UE IS: ] PRINT [ ]  
PRINT SELECT :M THING :NM PR  
INT [ ]  
PRINT [CHANGE IT TO ] PRINT  
[ ]  
MAKE "NL ALTER THING :NM :M  
[ ] READLINE 1  
CS  
IF :M = 1 THEN IF CHECK FIRS  
T FIRST :NL :KEYLIST = "TRUE  
THEN PRINT [SURNAME ALREADY  
EXISTS ] WAIT 200 CHANGE  
MAKE FIRST FIRST :NL :NL  
IF :M = 1 THEN MAKE "KEYLIST  
ALTER :KEYLIST :N [ ] FIRST  
FIRST :NL 1 PRINT [LIST MAY  
NEED SORTING ]  
CS SHOW :TITLES :NL  
END
```

```
TO ALTER :L :NO :NEL :VL :X  
; CREATES NEWLIST NEL FROM L  
SUBSTITUTING :VL AT ITEM NO  
IF :L = [ ] THEN OUTPUT :NEL  
MAKE "IT FIRST :L  
TEST :NO = :X  
IFF MAKE "NEL LPUT :IT :NEL  
IFT MAKE "NEL LPUT :VL :NEL  
OUTPUT ALTER BUTFIRST :L :NO  
:NEL :VL :X + 1  
END
```

```
TO SHOWFIELDS :P :N :D  
; DISPLAYS FIELDS FOR CHANGE  
SELECTION  
IF :P = [ ] THEN STOP  
TYPE :N TYPE [ ] TYPE FIRS  
T :P PRINT FIRST :D  
SHOWFIELDS BUTFIRST :P :N +  
1 BUTFIRST :D  
END
```

```
TO DELETE  
; DELETES A RECORD  
REVIEW PRINT [ ] PRINT [ENTE  
R CHOICE ]  
MAKE "N FIRST READLINE  
MAKE "AN SELECT :N :KEYLIST  
MAKE "KEYLIST DELKEY :AN :KE
```

```
YLIST [ ]  
MAKE :AN [ ]  
REVIEW  
END
```

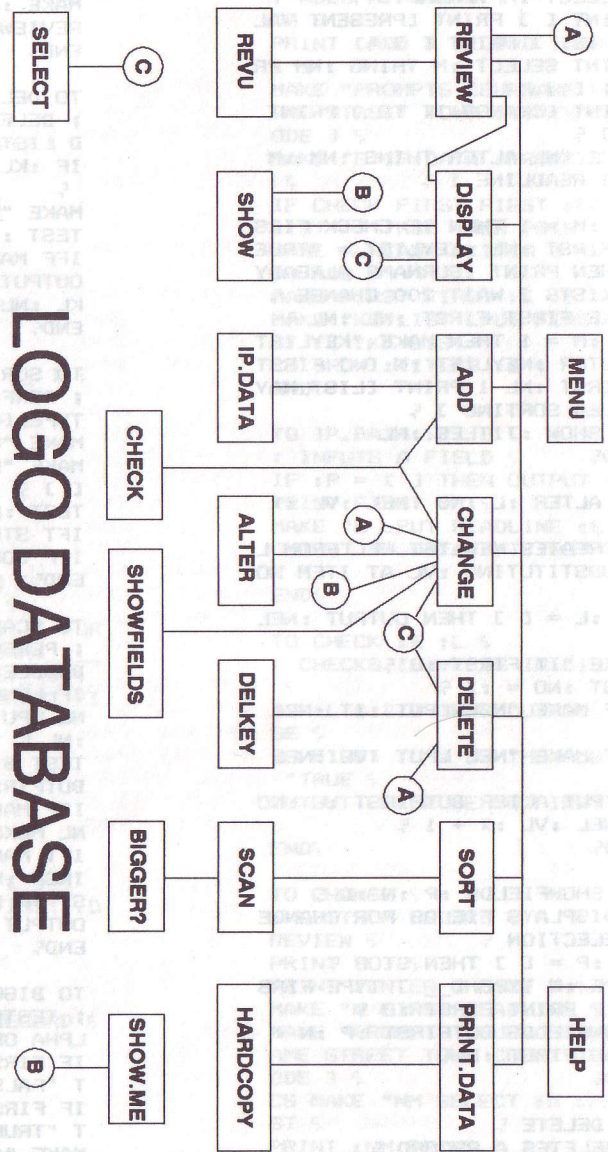
```
TO DELKEY :AN :KL :NL  
; DELETES A NAME FROM KEYWOR  
D LIST  
IF :KL = [ ] THEN OUTPUT :NL  
MAKE "NM FIRST :KL  
TEST :AN = :NM  
IFF MAKE "NL LPUT :NM :NL  
OUTPUT DELKEY :AN BUTFIRST :  
KL :NL  
END
```

```
TO SORT  
; PERFORMS BUBBLE SORT  
TYPE [WAIT.. ]  
MAKE "OLDLIST :KEYLIST  
MAKE "KEYLIST SCAN :KEYLIST  
[ ]  
TEST :OLDLIST = :KEYLIST  
IFT STOP  
IFF SORT  
END
```

```
TO SCAN :K :NL  
; PERFORMS ITEM EXCHANGE IN  
BUBBLE SORT  
IF LENGTH :K = 1 THEN MAKE "  
NL LPUT FIRST :K :NL OUTPUT  
:NL  
TEST BIGGER? FIRST :K FIRST  
BUTFIRST :K = "TRUE  
IFF MAKE "NL LPUT FIRST :K :  
NL MAKE "K BUTFIRST :K  
IFT MAKE "NL LPUT FIRST BUTF  
IRST :K :NL MAKE "K FPUT FIR  
ST :K BUTFIRST BUTFIRST :K  
OUTPUT SCAN :K :NL  
END
```

```
TO BIGGER? :A :B  
; TESTS IF A IS ABOVE B IN A  
LPHA ORDER  
IF FIRST :A = [ ] THEN OUTPU  
T "FALSE  
IF FIRST :B = [ ] THEN OUTPU  
T "TRUE  
MAKE "AN CHARNUM FIRST :A  
MAKE "BN CHARNUM FIRST :B  
TEST :AN > :BN  
IFT OUTPUT "TRUE  
TEST :AN = :BN  
IFT OUTPUT BIGGER? BUTFIRST  
:A BUTFIRST :B
```

# LOGO DATABASE



ON THE HARDWARE FRONT..... Mike Goddard

Quite a few things happening lately there has been a hefty turn over of equipment lately with quite a few members acquiring expansion systems. At the time of writing I have just taken delivery of a large selection of stand alone items RS232 and Disk interfaces plus a couple of 32K memory expansions. The RS232 interfaces are useful for printers and communications with TE11.

There have been no new developments as exciting as the ZENO boards of late but a lot of members are busy putting ZENO boards together and quite enjoying themselves (I think).

As from January M.G.C.S will not be supplying any full height disk drives as the last ones were made about five years ago and now are proving to be too unreliable I will still accept "trade in" drives but where a half height is supplied to replace a full height then an adaptor kit will be supplied to fill the gap.

The January Sale, M.G.C.S is holding its first January Sale and although lists will be available from the end of December nothing will be allocated or sold before the 7th of January keep in touch there might just be what you've always been looking for.

During 1990 M.G.C.S attended several Radio and Computer Rallies at various venues around the North West of the country as this was very successful it will be repeated again during 1991 there will be a list of venues published in the next issue of TIME if we happen to be nearby pop along for a chat it's always nice to see members in person.

A quick tip to those who keep a spare console for emergencies it isn't a good idea to let the spare stand idle as moisture will make its way into the chips and next time you use it there is every chance of the spare blowing a chip or chips. It is better to keep cycling the consoles say once a week or so this should keep them fairly active and if one does go down the other should keep you going.

I would like to wish all friends and acquaintances a happy and prosperous new year.

*Mike*



All of the games reviewed here are available from the group cassette library at the current library terms.

## STAR RATING GUIDE.

One star = terrible, Two stars = "OK", Three stars = quite good, Four stars = very good, Five stars = Brilliant.....

## GOBLIN CAVES

The object of the game is to explore a complex of caves to try to find 'The Monster', but there are 3 hazards.

1. Pits of boiling lava, which turn all adjacent caves red.
2. Goblins, which turn all the adjacent caves green.
3. The Monster, which changes all the adjacent caves blue.

You can kill the goblins and the monster by firing 1 of 6 arrows that you carry around with you. You fire by pressing the space bar, which fires an arrow into the room facing you. If you miss a goblin or 'The Monster' they move into a totally different room and that room could be yours.....

You change direction (and only direction) by pressing 'N' for North, 'S' for South, 'W' for West and 'E' for East. The direction you are facing is shown on the right side of the screen.

You move by pressing 'F' which moves you through the door facing you.

The game ends when you have either run out of arrows or have been killed by any of the 3 hazards.

A good Extended Basic game.

STAR RATING \*\*\*

## NOTEWORTHY

In this game you are a little round thing which has to collect music notes.

The screen is split into 4 long sections and on one section there is a 'Nasty' which could be either:

- 1 A Robot
- 2 A Slug Creature
- 3 Old man
- 4 Mad Dog

They patrol each section they are on by moving along it from side to side.

There are also stationary arrow type things which are placed at certain points above you. These arrow things can squash you and you can lose a life if you are not careful.

To move you use 'B' for moving towards the left and 'D' for moving towards the right.

Above you along certain areas are little square boxes. The biggest you can jump through by pressing 'P' to get to the next section. The others you use if a 'Nasty' approaches you, because by pressing 'P' you jump up and hit it but you come spinning slowly down, thus giving the 'Nasty' time to pass under you. Don't use them all the time if you can help it though because as soon as they have been used they disappear.

A very good Extended Basic keyboard game.

STAR RATING \*\*\*\*

This article was originally published in TISHUG September 1987.

The aim of this project is to enable the reception of Radio Teletype (RTTY) transmissions on the TI-99/4A computer with the minimum of external hardware.

Requirements for the system are a standard 4A console and minimemory module along with the appropriate software. Externally we need a radio receiver with good sensativity and high signal to noise ratio along with a demodulator capable of decoding F.S.K (frequency shift keyed) analogue signals and converting them to digital signals which are then inverted and applied to pin 4 of the joystick port.

The Baudot or Murray code is the 5 bit international teleprinter code and at first appears to have a maximum of 32 combinations for a character set. As there are 26 letters in the alphabet plus numerals and punctuation space is a problem. Cunningly the code is shifted by two control characters called LETTERS and FIGURES achieving a possible 60 combinations. The actual number used is 58, as the NUL character or all BITS Zero is not used.

The baud is the shortest single unit in a code and can be expressed as the reciprocal of the time of that unit e.g if the shortest unit is 20ms the speed of the signal will be 1 over .02 secs or 50 baud. Amateur Radio speeds are 45.5 or 50 baud with the former being the most common. Commercial RTTY transmissions often use 75 baud along with encrypted code to ensure privacy. The speed used in this project is 45.5 baud or approx 22ms and the value for this speed is used in the full and half bit delay routines documented in the software. Other speeds may be obtained by altering the value loaded into REGISTER 3 and can be accurately monitored at pin 7 of the joystick port with an oscilloscope. The data displayed on the screen is also stored in VDP RAM from >1000 upwards.

INSTRUCTIONS. Select the DEBUG option from M/MEM menu using the M-Modify command and enter the source code from the listing starting at address 71FC and ending at 75FE. After the program has been checked for errors you can then store it on cassette tape using the S-SAVE command. This will make reloading the program easier if it is lost due to battery failure in the M/MEM module.

The output of the RTTY demodulator is applied to pin 4 of the joystick port through a simple transistor inverter as this point is a normal -INT3 keyboard line and is active low. There is no chassis at this port and will have to be obtained elsewhere e.g cassette port pin 3.

To run the program use the E-EXECUTE command from 71FC and then via a switch turn on the data input to pin 4. NOTE if the data is turned on before the program is running the data will appear on the screen as garbage. A simple single pole switch at pin 4 that is enabled after the program is running will stop this problem. To escape from this program at any time all you have to do is press the FCTN key and disable the data switch and the program will

return you to DEBUG.

Testing the operation of the complete receive system requires a known error free signal. Murphys law says there will not be any RTTY stations on the air just when you need one. A standard audio cassette recorder can be used to record signals from your headphone socket and can then be played many times thus making monitoring and setup less tedious. For those of you who have an FM receiver e.g a scanner you will find RTTY transmissions on some amateur repeaters located in major cities. The demodulator we used was a simple PLL type called the ETI.733. Alternatively any RTTY demodulator may be pressed into service and this is left to the experimenters choice.

In conclusion I hope this little project will lead to more experimenting using the 99/4A as other fields and applications for its use are limited only by imagination.

\*\*\*\*\*  
 BAUDOT TO ASCII CONVERSION FOR RTTY DECODER ON 99/4A USING MINIMEM AND JOYSTICK PORT  
 \*\*\*\*\*

DATA COMES IN ON PIN 4 OF THE JOYSTICK PORT IS CONVERTED TO ASCII IN A LOOK UP TABLE AND IS DISPLAYED ON THE SCREEN ALSO IT IS STORED IN VDP RAM FROM >1000 FOR LATER DUMPING TO A PRINTER.

REGISTERS USED:-

- R0 CURRENT VDP ADDRESS
- R1 DISPLAYED BYTE IN ASCII
- R2 HIGHEST VDP ADDRESS TO DISPLAY
- R3 TIMER COUNT VALUE
- R4 SCRATCH LOCATION
- R5 BYTE BEING READ AS BAUDOT
- R6 SHIFT COUNT
- R7 LETTER OR FIGURE SHIFT
- R8 HIGHER VDP ADDRESS (NOT VISIBLE)
- R10 TEMPORARY COUNTER STORAGE
- R11 PC LINK
- R12 CRU BASE ADDRESS
- R13 PC
- R14 WORKSPACE POINTER
- R15 STATUS

LOCN	DATA	LABEL	MNEMONIC
INITIAL REGISTER SETUP			
71FC	0208	LI R8,>1000	STARTING HIGH VDP
71FE	1000		
7200	020C	LI R12,>0006	CRU ADDRESS
7202	0006		
7204	C29B	MOV R11,R10	SAVE RETURN LINK

7206	0200	LI R0,0002	FIRST SCREEN ADDR
7208	0002		
720A	0202	LI R2,>0240	MAX SCREEN ADDR
720C	0200		
720E	0207	LI R7,0000	"LETTERS" DEFAULT
7210	0000		

DO START BIT AND WAIT FOR CHARACTER OR CHECK FOR "FCTN" KEY

7212	0206	START	LI R6,0005	SHIFT COUNT
7214	0005			
7261	0205		LI R5,0000	CLEAR CHARACTER
7218	0000			
721A	1F00	PIN4	TB 0	TEST PIN 4
721C	1604		JNE DELI	JUMP IF LOW
721E	1F04		TB4	TEST "FCTN" KEY
7220	13FC		JEQ PIN4	JUMP IF HIGH
7222	C2CA		MOV R10,R11	RESTORE LINK
7224	045B		B *R11	RETURN TO EASYBUC

WAIT ONE AND A HALF-BIT TIMES AND SAMPLE FIVE TIMES.

7226	06A0	DELI	BL @DELAY	DO HALF DELAY
7228	7300			
722A	06A0	SAMPLE	BL @DELAY	DO DELAY
722C	7380			
722E	1F00		TB 0	TEST PIN 4
7230	1602		JNE +2	JUMP IF LOW
7232	0225		AI R5,>0000	SET MSB TO 1
7234	8000			
7236	0606		DEC R6	DEC SHIFT
7238	1302		JEQ +2	JUMP IF R6=0
723A	0915		SRL R5	SHIFT R5 RIGHT 1
723C	10F6		JMP SAMPLE	
723E	09B5		SRL R5,>0B	RIGHT JUSTIFY

CONVERT BAUDOT TO ASCII. DISPLAY BYTE, ADJUST R0

7240	C105		MOV R5,R4	SAVE R5
7242	06A0		BL @BASCII	CONVERT TO ASCII
7244	7400			
7246	0420		BLWP @VSBW	DISPLAY R1
7248	6024			
724A	06A0		BL @RZERO	TREAT R0
724C	7480			
724E	06A0		BL @HDELAY	
7250	7380			
7252	C100		MOV R0,R4	SAVE R0
7254	C008		MOV R8,R0	HIGH VDP RAM
7256	0420		BLWP @VSBW	PUT INTO VDP
7258	6024			
725A	C200		MOV R0,R8	SAVE HIGH VDP ADDR
725C	C004		MOV R4,R0	RESTORE SCREEN
725E	0588		INC R8	ADDR
7260	0288		CI R8,>3F00	MAX VDP ADDRESS
7262	3F00			

```

7264 130A JEQ QUIT EXIT
PRINT SPACES TO PREVIOUS PRINT
7266 C100 MOV R0,R4 SAVE R0
7268 0220 AI R0,>0020
726A 0020
726C 0201 LI R1,>002E ASCII FULL STOP
726E 002E
7270 0420 BLWP @VSBW DISP FULL STOP
7272 6024 ERASE
7274 C004 MOV R4,R0 RESTORE R0
7276 1000 JMP 0 CONTINUE
7278 10CC JMP START
727A 045B QUIT B *R11 RETURN TO E/BUG

```

SUBROUTINES.....

DELAY FOR HALF OF A BIT TIME

```

7300 020C LI R12,>0024 CRU FOR OUTPUT
7302 0024
7304 1D00 SBO 0 SET P2 HIGH
7306 1D01 SBO 1 SET P3 HIGH
7308 1D02 SBO 2 SET P4 HIGH
730A 1000 JMP 0 SHORT DELAY
730C 1000 JMP 0
730E 1E02 SBZ 2 RESET P4
7310 1E01 SBZ 1 RESET P3
7312 1E00 SBZ 0 RESET P2
7314 020C LI R12,6 CRU FOR PIN 4
7316 0006
7318 0203 LI R3,>0490 DELAY FOR 45.5 BAUD
731A 0490
731C 0603 DEC R3
731E 16FE JNE-1 11mS DELAY
7320 045B B *R11

```

DELAY FOR WHOLE BIT TIME THEN PULSE OUT ON PIN 7 AT TIME OF SAMPLING

```

7380 0203 LI R3,>0920 45.5 BAUD VALUE
7382 0920
7384 0603 DEC R3
7386 16FE JNE -1 22mS DELAY
7388 020C LI R12,>0024 CRU FOR OUTPUT
738A 0024
738C 1001 SBO 1 SET P3
738E 1D02 SBO 2 SET P4
7390 1000 JMP 0 SHORT DELAY
7392 1000 JMP 0
7394 1E01 SBZ 1 RESET P3
7396 1E02 SBZ 2 RESET P4
7398 020C LI R12,6 CRU FOR PIN 4
739A 0006
739C 045B B *11

```

CONVERT BAUDOT VALUE IN R5 TO ASCII AND PUT IN R1

```

7400 0285 BASCII LI R3,>0920 FIGURE SHIFT?
7402 001B
7404 1604 JNE LETTER
7406 0207 LI R7,0002 OFFSET VALUE OF 2
7408 0002
740A 1000 JMP 0
740C 1006 JMP LOOK
740E 0285 LETTER CI R5,>001F LETTERS SHIFT
7410 001F
7412 1603 JNE LOOK
7414 0207 LI R7,0000 NO OFFSET IF LETTERS
7416 0000
7418 1000 JMP 0
741A 0A25 LOOK SLA R5,2 SHIFT R5 LEFT TWO
741C A147 A R7,R5 ADD OFFSET
741E C065 MOV (R5),R1 VALUE INDEXED R5
7420 7580 BASE ADDR TABLE
7422 045B B *R11

```

INCREASE R0, AND PUT IN LEFT MARGIN

```

7480 C144 MOV R4,R5
7482 0285 CI R5,>001B LETTERS SHIFT?
7484 001B
7486 1304 JEQ +4
7488 0285 CI R5,>001F FIGURE SHIFT?
748A 001F
748C 1301 JEQ +1
748E 0580 INC R0 INCREASE CURSOR
7490 C100 MOV R0,R4
7492 0244 ANDI R4,>001F MASK
7494 001F
7496 0284 CI R4,>001F R0 AT RIGHT MASK
7498 001F
749A 1602 JNE +2 JUMP IF NOT
749C 0220 AI +3 MAKE LEFT MARGIN
749E 0003
74A0 0280 CI R0,>0282 MAX SCREEN ADDR
74A4 1602 JNE +2
74A6 0200 LI R0,2 BACK, TOP OF SCREEN
74A8 0002
74AA 045B B *B11

```

LOOK UP TABLE

```

7580 2000 SPACE
7582 2000 SPACE
7584 4500 E
7586 3300 3
7588 0A00 L/F
758A 0A00 L/F
758C 4100 A
758E 3D00
7590 2000 SPACE
7592 2000 SPACE

```

7494	5300	!
7596	2100	I
7598	4900	B
759A	3800	U
759C	5500	7
75A0	0000	C/RET
75A2	0000	C/RET
75A4	4400	D
75A6	2A00	*
75A8	5200	R
75AA	3400	4
75AC	4A00	J
75AE	0700	BELL
75B0	4E00	N
75B2	2C00	.
75B4	4600	F
75B6	2400	\$
75B8	4300	C
75BA	3A00	:
75BC	4B00	K
75BE	2800	(
75C0	5400	T
75C2	3500	5
75C4	5A00	Z
75C6	2200	"
75C8	4C00	L
75CA	2900	)
75CC	5700	W
75CE	3200	2
75D0	4800	H
75D2	2300	f
75D4	5900	Y
75D6	3600	6
75D8	5000	P
75DA	3000	O
75DC	5100	Q
75DE	3100	1
75E0	4F00	0
75E2	3900	9
75E4	4200	B
75E6	3F00	?
75E8	4700	G
75EA	2600	&
75EC	2000	SPACE
75EE	2000	SPACE
75F0	4D00	M
75F2	2E00	.
75F4	5800	X
75F6	2F00	/
75F8	5600	V
75FA	3B00	:
75FC	2000	SPACE
75FE	2000	SPACE

**\*\*\*PLOTTING\*\*\***  
by University of Dallas  
Edited by T.STEVENS

Way back in 1982 I saw an article in the December issue of 99er Magazine in which one could plot in TI BASIC in BIT MAP MODE... Yes what I said was correct. So I thought it would be a nice idea to bring it back to our group for all those that missed it, seeing the 99er is now finished.

The program only works in minimemory via Editor Assembler, as this program uses call LINKS to do the access into bit map.

If you program wisely you can plot curves, draw axes and even objects in full perspective. The minimum requirement is the Minimemory and a cassette recorder plus 32k. However you will not be able to enter this source program into Minimemory as it writes over the LBLA. Don't worry those with only the basic configuration as I have provided Nicky GODDARD, the Cassette Tape Librarian, a copy of the program for you to use if you send for a copy. Those other lucky people who have a full system can enter the source code by hand from the listing provided and create their own Object Code. They can however obtain a copy of the program on a disk, if they send for a copy from the Disk library held by Stephen SHAW.

The routines act as a basic plotter, in black and white only. There is an imaginary pen which can be moved in the up or down position. If it tries to draw off the screen area it stops and the pen remains stationary. The screen area is over to the left hand side of the screen in a area of 192 pixels vertically and 255 horizontally. The routines allow you to clear the screen, scale the screen, draw X and Y axes, output text, change pen positions and draw lines.

When running the program takes up about 12k of space in the VDF ram this leaves very little room for your basic program. Therefore a buffer was created in the 32k ram to do just this. Because this buffer is not in the VDF RAM the plot cannot be seen until the GRAPH command is issued. Also once the command is issued you lose your basic program, so **SAVE IT BEFORE RUNNING**. As a positive side effect, after the plot has come up on the screen you can push 'Q' and the screen will return to the power title screen, but leaving the plot unchanged in the memory buffer. This will exist until the GCLEAR routine is used, or the machine turned off. Therefore it is possible to add data to a plot even after looking at it several times.

**THE ROUTINES AND HOW TO USE THEM**

CALL LINK("GCLEAR")

This initializes the graphics package, clears the screen and the pen set to lower left most pixel. Scale is set so that X axis has value of zero and maximum of 255. Y to minimum zero, and max of 191. The lower left hand pixel has the coordinate value of (0,0).

CALL LINK("SCALE", Xmin, Xmax, Ymin, Ymax)

This command lets you set the minimum and maximum values of the screen area. The values can be real or integer. They may be passed as numeric expressions or numeric variables. The minimum must be less than the max, else \*BAD ARGUMENT IN ... will be issued. You can't pass variables through numeric arrays, as these are not supported. You can give the SCALE command at any time without affecting screen buffer data, and the pen location.

CALL LINK("DRAW", Xpos, Ypos)

The Draw command will move the pen from its current location to the x,y position given. The program knows where the pen was so it only needs the new position. This command is like pen down in LOGO. If you go off the screen, then no problem, except maybe a numeric overflow if it was that bad.

CALL LINK("MOVE", Xpos, Ypos)

This command is the reverse of Draw, it is pen up. However the pen is automatically put down on reaching the new position required.

CALL LINK("XAXIS", Xmin, Xmax, Yintercept)

This command draws a horizontal axis specified by a minimum and maximum along the X axis. The axis will intersect the Y axis at the user defined point. The pen position remains unchanged. Arrays are not allowed, but variables and numeric expressions are.

CALL LINK("YAXIS", Ymin, Ymax, Xintercept)

This is the counterpart to XAXIS.

CALL LINK("LABEL", variable) CALL LINK("LABEL", "String")

This allows you to print on the screen with characters from 32 to 127, above and below will equate to first or last character. ie 31=80 . 129=127 In this case also a space. Due to the nature of the character definitions, they will be output to the character blocks, starting at the block designated by the current position of the pen.

CALL LINK("GRAPH")

This command destroys the Basic Program and then brings the plot up onto the screen. You leave with the 'Q' command already discussed. To add to your graph reload your program or type in your new command, but do not GCLEAR.

## LOADING ROUTINE INTO MINIMEMORY

Once the machine routine is loaded into the Minimemory, it is not necessary to reload it unless the routine has been destroyed for some reason, such as loading another program into it. Do not, once loaded, CALL INIT. Also make sure locations 2800 to 3000 in the 32k expansion memory are not loaded with data.

To load the routines you can do it two ways:-

### METHOD 1

- 1) load E-A into machine. (Cartridge or BEA).
- 2) Load EDITOR and enter by hand (first time only) Listings given. Bit 1 first and create File DSK1.BIT1
- 3) Create disk file DSK1.BIT2
- 4) Create disk file DSK1.BIT3
- 5) Create disk file DSK1.BIT4
- 6) Create disk file DSK1.SOURCE
- 7) Execute Assembler on disk file SOURCE. Direct OBJECT code to disk file GRAPHICS. Use Option R.
- 8) Remove E-A insert MM.
- 9) Put disk containing the file graphics into disk 1.
- 10) Select Mini-Memory from the menu.
- 11) Select Re-Initialize then PROCEED.
- 12) Select Load Run option.
- 13) Under file type DSK1.GRAPHICS
- 14) When asked for another file select QUIT.

The routine is now loaded. All later loads start at 8.

### METHOD 2

Using Tape source code.

- 1) Insert Minimemory.
- 2) Re-initialize as 10 to 11 above.
- 3) Quit.
- 4) Select Easy Bug.
- 5) Press any key.
- 6) Follow instructions on screen.
- 7) Press Quit when ? appears at bottom of screen.

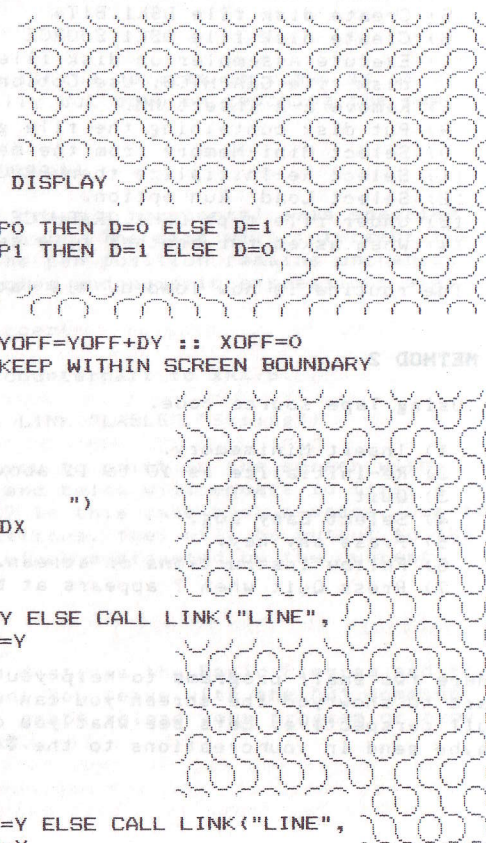
Here two small programs to help you on your way. The first is a test card to show you the screen you can work in. The other is a 3d Box in full perspective. Lets see what you out there can come up with and maybe send in your creations to the TI\*MES for others to share.

## TRUCHET TILES

How to describe this graphics program? We are if you like preparing a potato (or lino) cut pattern composed of two quarter circles. These can be printed in two possible ways, one being a 90 degree rotation of the other. The computer program places them randomly to produce some interesting patterns. The program given uses RND to select numbers between 0 and 1, which are then converted to 0 and 1 values. There are two fairly obvious shapes created- circles and dumbbells (two circles connected by a thin lobe). If our RND truly is random, we may expect that if we count the total number of tiles then there will be one eightieth that number of dumbbells, and a trifle over one twentieth that number of stand alone circles. Instead of using RND we could say PEEK into memory locations and convert the values we find to binary (011100 etc) and use those 1's and 0's to produce a quite different pattern! Or try various uses of RND eg RESULT=(RND+RND/2) or RESULT=RND\*RND and so on. Send print outs to me if you find anything pleasing!!

```
100 ! TRUCHET TILE GENERATION      from Pickover p.333
110 ! for XB+TML
120 ! S SHAW   OCT 1990
130 ! easily converted to  any other pixel graphic
      language or utility.
140 !
150 RANDOMIZE
160 XOFF,YOFF=0.0
170 DX,DY=12 :: R=DX/2
180 JUMP=180/DX :: OLDDATA=0
190 P1,P0=0.5000 ! 50/50
200 REM
210 FOR J=1 TO 100*100 ! 100X100 DISPLAY
220 RESULT=RND
230 IF OLDDATA=0 THEN IF RESULT<P0 THEN D=0 ELSE D=1
240 IF OLDDATA=1 THEN IF RESULT<P1 THEN D=1 ELSE D=0
250 IF D=1 THEN GOSUB 340 ! UP
260 IF D=0 THEN GOSUB 510 ! DOWN
270 XOFF=XOFF+DX
280 IF (J/JUMP)=INT(J/JUMP) THEN YOFF=YOFF+DY :: XOFF=0
290 IF YOFF>200 THEN GOTO 290 ! KEEP WITHIN SCREEN BOUNDARY

300 OLDDATA=D
310 NEXT J
320 STOP
330 REM
340 CALL LINK("PRINT",180,20,"UP  ")
350 IF OLDDATA=0 THEN XOFF=XOFF+DX
360 FOR I=90 TO 180 STEP 10
370 X=R*COS(I/180*PI)+XOFF
380 Y=R*SIN(I/180*PI)+YOFF
390 IF I=90 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",
OLDX,OLDY,X,Y) :: OLDX=X :: OLDY=Y
400 NEXT I
410 REM
420 REM
430 XC=XOFF-DX :: YC=YOFF+DY
440 FOR I=270 TO 360 STEP 10
450 X=R*COS(I/180*PI)+XC
460 Y=R*SIN(I/180*PI)+YC
470 IF I=270 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",
OLDX,OLDY,X,Y) :: OLDX=X :: OLDY=Y
```



```
480 NEXT I
490 RETURN
500 REM
510 CALL LINK("PRINT",180,20,"DOWN")
520 IF OLDDATA=1 THEN XOFF=XOFF-DX
530 FOR I=0 TO 90 STEP 10
540 X=R*COS(I/180*PI)+XOFF
550 Y=R*SIN(I/180*PI)+YOFF
560 IF I=0 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",
OLDX,OLDY,X,Y) :: OLDX=X :: OLDY=Y
570 NEXT I
580 REM
590 XC=XOFF+DX :: YC=YOFF+DY
600 FOR I=180 TO 270 STEP 10
610 X=R*COS(I/180*PI)+XC
620 Y=R*SIN(I/180*PI)+YC
630 IF I=180 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",
OLDX,OLDY,X,Y) :: OLDX=X :: OLDY=Y
640 NEXT I
650 RETURN
660 END
```

Truchet you ask? (OK you dont ask...). This idea dates back to... 1704!

It is fun to save this as TI ARTIST pics and then use the FILL to prove that this is a two colour pattern only. Also, by its nature, you can always align one printout with another, top or bottom or sides, to make your own very unique wallpaper, gift wrap, book covering etc. The pattern is also suitable for an all-over fabric print and can be used with either very strongly contrasting colours or with two close shades of the same colour for a more subtle effect. Have fun!

TI WORLD NEWS by Jim Peterson  
AUGUST 1990

The Spring 1990 catalog of All Electronics Corp. lists many brand new original TI parts at remarkably low prices, especially on quantity purchases - TI keyboards for \$3.50, R/F modulators for \$5, heat sinks at 3 for \$1, etc. The address is P.O. Box 567, Van Nuys CA 91408, phone 800-826-5432.

Computer Buyer's Guide is now called Vulcan's Computer Monthly. It contains zillions of ads for computer products, and a monthly TI column by Barry Traver - which was missing in August, probably because he didn't get it to them on time. It is still not available on very many newsstands, but the annual subscription price of \$12 (delivery in US) is a steal.

However, I pity the mail carrier - if he ever had to deliver that magazine plus Computer Shopper and the MicroCenter catalog on the same day, he would surely have a hernia. I have suggested to Vulcan's that they buy a computer for their bookkeeping department - they double-billed me for my subscription, and then for my ad!

\*\*\*\*\*  
DIJIT is no longer taking orders for their AVPC 80-column card,  
but will continue to support those they have sold.

\*\*\*\*\*  
Another XBasic programming environment, called Multi-Mode XB, is  
available from Disk Only Software. It was written by Jean  
Marleau and, like the others, consists of CALL LINKs to  
assembly. However, it offers some features not found in any  
other. For a complete review, see the article by Barry Traver in  
Vulcan's Computer Monthly for September 1990.

\*\*\*\*\*  
Paul Scheidemantle is publishing the P&A Express, which appears  
to be a short newsletter with news about new graphics utilities.  
For a free copy, send a SASE to him at 2762 Lovington, Troy MI  
48083.

\*\*\*\*\*  
Corcomp has been taken over by International Diversified  
Technologies Inc., 2211 E. Winston Rd, Suite G, Anaheim CA  
92806, phone (714) 635-1815). They will continue to manufacture  
and repair Corcomp products. For repairs, call (714) 965-4450 to  
get a return material authorization. This is a tape, so have  
your information ready.

~~~~~

## MACHINE CODE TUTORIAL 2

by Mack McCormick.

### Using SBUG to DEBUG your Assembly Programs

This tutorial will describe how to use the TI SBUG program to debug  
your assembly language programs. Although specifically written for SBUG  
the principles apply to the other debuggers on the market. As I am  
sure you have already discovered, the familiar error messages and  
helps are not available from Assembler that you are accustomed to from  
BASIC. I know many of you have typed in my first tutorial or used one  
of the beginning assembly language books on the market and had  
difficulty getting the program to run properly. Take heart, we all  
had the same problems when we were learning and until you master the  
use of one of the debugging programs your you can pull your hair out  
and may eventually give up.

As I have already mentioned there are several very good debug  
programs available for the TI. Here they are briefly. DEBUG is the  
program supplied with your editor assembler cartridge. It has many  
helpful features particularly the ones that allow you to set CRU bits,  
move memory, and compare memory. My main criticism of DEBUG is it's  
inability to single step through instructions. BUGOUT written by  
Gregg Wonderly is also very good. It offers multiple fields of  
information on the screen at the same time. Many advanced features  
like dumping to disk.

Without a doubt the most advanced that I have used. My main  
criticism is that it maps the VDP to text mode which interferes with  
VDP mapping for the screen image table and color table if you need to  
inspect these locations in your program. The one I use the most is  
SBUG. This is the super debugger released by TI to users groups after  
the pull out. If you need a copy [from disk library. ss]. SBUG  
allows you to operate from graphics or bit map mode, single step, and  
output to printer or disk. Let's get down to how to use it.

First you should print out the instructions supplied with SBUG on  
the disk. Since this is a display/variable 80 file you may print it  
out with TI-Writer or Editor/Assembler editor. This help file covers  
all of the instructions available. I intend to cover only the ones of  
greatest interest to us.

It's important to know where your assembly program loads into  
memory. With some few exceptions which won't be discussed now, your  
program will load into high memory expansion at >A000. That's the  
first address in high memory. You can make your program load in other  
places but that is beyond the scope of this tutorial. SBUG is what we  
call relocatable object code. It loads on top (not over) of the last  
program entered. In other words, if your program is 500 bytes long it  
would load from >A000 to >A500. SBUG when loaded will load beginning  
at >A502.

I've included a brief program in this tutorial which puts  
characters on the screen. We'll use it to illustrate how to use SBUG.  
Go ahead at this time and type it in using the editor, save it, load  
the assembler except this time at the LIST FILE NAME prompt enter your  
printer description. Remember if you are using a PIO printer it should  
be entered with a period following PIO. At the OPTIONS prompt enter  
RLS.

R to tell the assembler that you used R in front of your registers  
in the source code, L to give you a source listing to your printer [if  
using PIO type in PIO. -a full stop after! ], and S to give you a  
table of the symbols you used. This source list is necessary for using  
SBUG effectively.

Let's examine the source listing columns for a moment. Here's an example  
line from the source listing:

```
0049 006C 0201      LI R1,>5500
      006E 5500
```

Here's what that tells us. 0049 means this statement is number 0049  
in our source code. That means if we received the message invalid  
register in line 0049 when we were assembling the code we could look  
at line 0049 and see what was wrong with R1. The next field over is  
the most important one to us when using SBUG. This gives us the exact  
location in memory where this instruction resides. If this was the  
first program you loaded into memory then this instruction would  
reside at >A06C. Remember I told you that the program loads at >A000  
by default? In other words >A000 plus >006C=>A06C. We'll come back to  
this field shortly.

The next field >0201, is the machine language mnemonic for Load  
Immediate (LI). Look on page 5 of your blue editor/assembler card. See  
the opcode >0200? This tells us the instruction for this OPCODE is LI.  
The 1 in >0201 tells us to load immediate R1 with the value in the  
next word of memory. In this case Load Immediate R1 with >5500. See  
how easy that is? For a detailed description see section 15.4 in the  
E/A manual.

We're ready to get down to business. First load the object code  
for the program I have given you into the computer. You should  
\*always\* load your program before you load SBUG. Next before you press  
enter after you have loaded the demo program load SBUG. One note here.  
As long as you are not using the X/B cartridge to load with, it is  
much faster to load the compressed version of SBUG or SBUGC as it is  
listed on your disk. Both do the exact same job. Press enter and for  
the program name enter SBUG. The SBUG title screen should come up. The  
message on the screen will ask you if you are using a bit map screen?  
Enter N at this prompt.

Next you will be asked for your list device. Enter your printer  
description here (you could enter DSK1.FILENAME instead if you wanted  
output to disk). The input prompt is a ' ' .

At the prompt enter L to turn off the list device. Our program runs  
from SBUG's control so we need to set up SBUG to run our program so we  
may interact with it. To do this we must tell SBUG where the first  
executable instruction (entry point) to our program is and what  
address we are using for a workspace.

You may remember there are three hardware registers used by the 9900 CPU in the TI-99. These are the program counter, workspace register, and status register. The program counter always points to the next instruction to be executed in memory. The workspace register points to the current 32 bytes (16 words) of memory we are using for workspace. The status register contains the current status of the computer as a result of the last instruction executed. Given that, we now need to find the entry point to our program.

There are two basic ways to do this. Perhaps the easiest is to look on the source listing at the label (in the example program START) we placed in the DEF statement at the beginning of the program and get the address directly from the source listing. The alternate way is to use the M command of SBUG to examine memory.

The memory we need to examine is the Reference/Definition (REF/DEF) table approximately from >3F30 to >4000. This table lists the entry point to the program and the utilities we referenced. Here's how to examine this table.

```
M 3F30,3FFF <enter>
```

This command will scroll the contents of these memory locations to the screen. In this case we are looking for the program name which is START. When you see START press any key to stop the scroll. You should see a line of memory which looks like:

```
3F30=5354 4152 5420  START
3F36=A058  ....  ....
```

Press FCTN X to abort the listing or any command. Since labels may be up to 6 characters (bytes) long, the seventh and eighth bytes contain the program entry point. In this case >A058. This should be the same address on the source listing. \*Remember\* SBUG always displays and only accepts HEX numbers.

Next we need the workspace address. We obtain that from the source listing. In this case it is >A05C. We get this by finding the label for our workspace, in this case WS. We look at the second column and there is the beginning of the workspace. We're now ready to set up the hardware registers to run our program. We do this by entering a R at the prompt. SBUG responds with W=0000. W is the workspace address. In this case type A05C and press the <space bar>. That calls up the prompt P=0000 for program counter address which in this case is A058. If you press space again you could set the status register but this is unnecessary in this case so just press enter. We've now told SBUG where our program resides in memory and where the workspace is.

To make it easy to figure the offset of our instructions we have a bias command. Press X and the computer displays 0000 press A000 <enter>. This sets the value of X to >A000. Now let's get down to examining our program in memory. To do this we set breakpoints (just like BASIC) at the instruction where we want to stop.

Ready to set one? At the prompt press B 68X <enter>. This automatically adds the A000 offset to the instruction at 0068. You could have just entered it as B A068 if you wanted to add the offset yourself. This tells SBUG to interrupt the program before it executes LI R0,>390. See where I got that?

Look at the second column on your source listing and go down it until you find memory location 0068. Now if you want to see where your current breakpoints are press B <enter>. Now to execute the program up to that instruction press E <enter>. The screen will show three addresses across the screen. From left to right they are the workspace

address, program counter address, and status register. Now we're ready to single step through some instructions. Enter S for single step at the prompt.

```
This displays the following:
A068=0200 LI R0,>0390
```

A068 is the memory address for the instruction. Remember what 0200 is? LI R0: Then there is the plain text instruction. What value should be in R0? >0390 of course. To check this press W 0 <enter>. This shows the current contents of R0.

If you ever have the need you may change this value by entering another value before pressing enter. If you press the space bar you see the next register R1 and so on.

One important point. Never try to single step thru VDP, GROM, or KSCAN routines. There is a good chance you will lock up the computer. When you encounter one of these instructions just set a breakpoint (B) on the other side of it and execute (E) around it. Now press B BOX or B A0B0 <enter> at the prompt. This sets the breakpoint at the INC R0 instruction. Press E <enter>.

What's the value in R0? It should be >383 from instruction >A0B2. Now press S. The INC R0 instruction appears. Again check the value in R0 (remember W0), it should be >384. See how you may interact directly with the computer? Press S again and you see the CI instruction. Press S again and you get: JNE \$+>00FA

```
JMP TO A07C
```

This says to jump to A07C (LOOP2) if R3 is not equal to zero. Now set B A0CE. Press E <enter> to execute. The program stops at the MOV @SAVRTN,R11 instruction. Let's examine some VDP RAM. To do this we use the M command again but proceed the address with a V for VDP access. Press M V168,1AF .

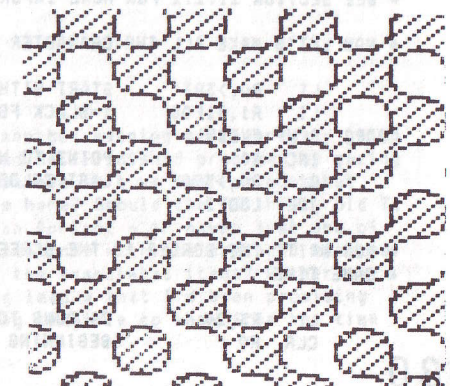
This shows the screen image table from decimal 360 to 431 where we put our text and ball. Examine the bytes carefully and you should see the border characters.

Note: The screen is not exactly as we had it because SBUG shares the same screen with us and we're seeing some of SBUG's screen also. Now press M VBF8,C00 <enter>. This is the pattern descriptor table where the shape of our ball is stored. Compare it with PATRN from the source listing and then press M A022,A028 to see the pattern description as it resides in CPU RAM.

Well I've gone long again but there is so much to cover. You really need to get the hang of using SBUG, it allows you to interact with your program to see exactly what is going wrong when you have a problem. It will save you many hours of grief.

Mack

```
*****
*
* This program accompanies the tutorial *
* on using SBUG. It also shows you *
* how to redefine characters, place *
* them on the screen, and change the *
* color of each character set *
* Entry Point -> START *
* Must be run from E/A or MM Load & Run *
* R0-R2 Used for general VDP access *
* R3 Used as a general purpose counter *
* R11 contains the return address on *
* entry. *
* by Mack McCormick 74206,1522 *
*
*****
```





```

DEF START      DEFINE THE ENTRY POINT OF THE PROGRAM
REF VMBW,VSBW,VWTR SYSTEM UTILITIES WE WILL USE

```

\* DATA STATEMENTS \*

```

SAVRTN DATA 0      SET ASIDE A WORD OF MEMORY TO SAVE THE
                    * RETURN ADDRESS
BORDER DATA >8080,>2020,>2020,>2020 CHAR >80 WILL BE OUR BORDER
DATA >2020,>2020,>2020,>2020
DATA >2020,>2020,>2020,>2020
DATA >2020,>2020,>2020,>8080
PATRN DATA >3C7E,>FFFF,>FFFF,>7E3C

```

```

STATUS EQU >B37C    LOCATION OF THE GPL STATUS BYTE
*                  THIS EQUATES THE LABEL STATUS TO
*                  * THIS ADDRESS
WS BSS >20          SET ASIDE 32 BYTES OF MEMORY FOR OUR
*                  * WORKSPACE

```

```

MSG TEXT 'THIS IS A TEST' 14 BYTES LONG
EVEN FORCES THE PROGRAM COUNTER TO AN EVEN
* MEMORY ADDRESS.
* AS A GENERAL RULE ALWAYS USE EVEN AFTER THE LAST BYTE, BSS, OR
* TEXT OPCODE.

```

```

START MOV R11,@SAVRTN MOV THE ADDR IN R11 TO THE WORD OF
* MEMORY AT SAVRTN
LWPI WS TELL THE HARDWARE WORKSPACE REGISTER
* WHERE YOUR WS IS

```

\*--PUT A BLUE BORDER AROUND THE SCREEN--\*

```

LI R0,>0705         07 IS VDP REG 7 OR THE SCREEN
* BACKGROUND COLOR BLWP @VWTR 05 IS THE
BACKGROUND COLOR. (LT BLUE)
* THIS SETS THE TOP AND BOTTOM OF THE SCREEN TO LT BLUE.
* SEE SECTION 16.1 E/A MANUAL FOR MORE INFORMATION ON VWTR.

```

```

* NOW SET CHAR >80 (128 decimal) TO A BLUE ON BLUE SQUARE
LI R0,>390          THIS IS THE POS IN VDP RAM COLOR TABLE
* FOR CHARS >80->87
LI R1,>5500         BLUE FOREGROUND/BACKGROUND IN MSB OF R1
BLWP @VSBW         REMEMBER R0 ALWAYS IS THE ADDR IN VDP. R1
* ALWAYS CPU.

```

\* SEE SECTION 21.2.2 FOR MORE INFORMATION ON THE COLOR TABLE

\* NOW LET'S MAKE ALL THE CHARACTER SETS BLACK ON WHITE

```

LI R0,>383          START WITH CHAR SET >18
LI R1,>1F00         1=BLACK FG, F=WHITE BKGND
LOOP2 BLWP @VSBW
INC R0             POINT TO NEXT VDP RAM COLOR TABLE ADDRESS
CI R0,>390         LAST COLOR TABLE ADDRESS (CHARS >78->7F)
JNE LOOP2

```

\* NOW WRITE THE BORDER TO THE SCREEN AND CLEAR THE SCREEN AT THE  
\* SAME TIME

```

LI R3,24          24 ROWS TO WRITE
CLR R0            BEGINNING OF SCREEN IMAGE TABLE

```

```

LI R1,BORDER     ADDRESS OF ONE ROW OF BORDER DATA
LI R2,32         32 BYTES TO WRITE
LOOP BLWP @VMBW  WRITE A ROW
AI R0,32         POINT TO BEGINNING OF NEXT ROW
DEC R3
JNE LOOP

```

\* NOW WE'LL PUT THE MESSAGE ON THE SCREEN CENTERED  
\* REMEMBER THAT THE SCREEN IMAGE TABLE IS FROM 0 TO 767 IN VDP RAM  
\* TO DETERMINE THE ADDRESS FROM ROW AND COLUMN WE USE THE  
\* FOLLOWING FORMULA

\* ADDR=((ROW-1)\*32)+(COLUMN-1) IF WE WANT OUR MESSAGE AT  
\* ROW 12 COLUMN 9 THE VDP  
\* ADDRESS WOULD BE 360

```

LI R0,360        VDP ADDRESS IN SCREEN IMAGE TABLE
LI R1,MSG        ADDRESS OF THE DATA IN CPU RAM
LI R2,14         14 BYTES LONG
BLWP @VMBW       WRITE IT TO THE SCREEN

```

\* NOW SUPPOSE WE WANT TO MAKE CHAR >7F A BALL SHAPE AND PLACE  
\* IT UNDER THE TEXT THE PATTERN TABLE IN E/A IS LOCATED AT >8000.  
\* TO CALCULATE THE LOCATION OF A PARTICULAR CHARACTER MULTIPLY ITS  
\* HEX VALUE BY 8 AND ADD THE RESULT TO >800.

```

LI R0,>0BF8      (>7F*8)+>800=>BFB
LI R1,PATRN     PATTERN TO DEFINE >81 TO FROM CPU RAM
LI R2,8         PATTERNS ALWAYS 8 BYTES
BLWP @VMBW     >81 IS NOW THE SHAPE OF A BALL

```

```

LI R0,431       SCREEN IMAGE TABLE TWO ROWS BELOW TEXT
* CENTERED
LI R1,>7F00     WRITE >81 (BALL) TO THE SCREEN
BLWP @VSBW     PUT IT UP

```

\* EXPERIMENT WITH THIS PROGRAM UNTIL YOU ARE COMFORTABLE WITH  
\* VDP RAM ACCESS

```

MOV @SAVRTN,R11 RESTORE R11 TO THE ADDRESS YOU WISH TO
* RETURN
LIMI 2          ENABLE INTERRUPTS SO QUIT KEY WILL WORK
JMP $           LOCK UP THE COMPUTER (SAME AS 100 GOTO 100)
RT
END

```

=====
WANT MORE FROM MACK? TELL ME! STEPHEN
=====

MODULE REVIEW - WORDWRITER +

=====

by Ashley Tilling.

How many of us have come to the end of another evening of cheerfully going through those familiar cassettes of games and half-finished programs by gazing fondly at our unexpanded system (even that phrase left us feeling somewhat emasculated) and thought: where do I go from here? Should the faithful, old TI be abandoned in favour of an Atari, a Mac, an Apricot - or maybe just one of those nice and cheap Amstrads? Or should I continue with the orphaned machine by splashing out vast sums on a PEB and all the associated items? Either way I'd be able to move into the word processing league that I'd been promising myself for all those years and have something tangible to show from the time spent tapping away in the spare room!

Being the financially cautious sort (if I wasn't I wouldn't have a TI!) I decided first to consult Francis 'Parco' Parrish to ask for advice on the costs on expanding the TI. The long and the short of my investigations were that a couple of years ago my interest in the TI was re-kindled by buying a module that Parco called MINI-WRITER III, but is labelled and documented as WORDWRITER+.

For around #80 I received the module (of the usual cartridge type), which has a built-in output port, and a made-up cable to connect directly to the parallel input of a printer. It is, in fact, made by DataBioTics Inc. (c)1987 with a program written by Todd Kaplan. Not having seen TI WRITER, the word processing capabilities of this device did nevertheless seem pretty close to the TIW text editor facilities I'd read about.

The first thrill is to enter the world of the 40 column text mode. By using the 40x24row windowing technique you can create your documents up to 80 columns wide.

On first entering the program you are presented with a prompt line consisting of:

Edit Tabs Quit Files Search Lines Purge

with the next line as the Command line for your abbreviated choice. To start on your new page you may change the preset tabs and left/right margins, otherwise simply press E for Edit. The arrow keys are 'live' to move the cursor around the screen (ie. function E,S,D,X) and the window is moved around the text by using CTRL E or FCTN 6(window up), CTRL X or FCTN 4(window down), CTRL D or FCTN 5(window right) and CTRL S or FCTN 5(twice) (window left).

Text can be inserted using fixed or word-wrap (CTRL O), and the text closed up after an insertion by reformatting either relative to the left margin (CTRL 2) or relative to the cursor with CTRL R. FCTN 8 inserts a blank line whilst CTRL 8 starts a new paragraph by placing a Carriage return and a Line Feed on the next line. Inevitably characters, lines and blocks of text can also be deleted.

Particularly impressive are the facilities which enable the user to search for a nominated string, ie. a word or phrase, and to replace strings with different ones at all or selected occurrences. Lines of text can be copied and can be easily moved around to different positions on the page.

Text can be saved to cassette or disk. Using the SaveFile -SF- option will store your outpourings in the usual Dis/Var (ie text) mode and by specifying start and finishing line numbers it will save only that part of the file. An interesting feature is that files can also be saved using SaveMem -SM- or SavefiX -SX- options. SM saves in memory image, ie program, mode thereby reducing the space needed, whilst SX stores in Dis/Fix format. Similarly files (or parts of files) can be loaded using LF,LM, or LX.

If you do have the facilities to save to disk another useful capability would be that of Show Directory (SD). This displays all files held on the specified disk, giving details of the stored format, sectors used and size of each file.

Finally, of course, you can print your file through the module interface or through to any other printer you may have connected to your computer. Printer control characters can be entered (escape sequences) to give access to your printer's special commands, eg. underlining or changing fonts.

It should also be noted that for most of the functions obtained by going through the command line, there is also the option of, what is termed in the perfectly adequate accompanying handbook, the Expert Mode. For example, instead of FNCTN 9 & PF to get to PrintFile, this can be entered merely by keying CTRL P.

So, in summary, if you've been looking for a way to get a lot more out of your dusty, old TI by expanding into the word-processing field, but have shied away from a fully expanded system, this comprehensive module may be just the thing for you.

[Ashley now has a PEB and TI Writer. In order to load the Wordwriter+ DV80 files into TI Writer it is necessary to use the short conversion program in TI\*MES #21, or to use VIEW to discover the last line number and then LOAD all lines EXCEPT that one- both these tricks divert TI Writer from the non-compatible Wordwriter+ control codes in the final record.]

## RAMBLES

by Stephen Shaw  
for TI\*MES January 1991

Welcome to another issue of RAMBLES and a happy new year to you all. Your comments and letters are always welcome, and your queries too -please help out with an SAE if you would like a direct response! My address, the same as the groups disk library, is:  
10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.

I have been asked to write some summary type comments on the various XBs and also on what a brand new disk owner should be looking for- in his last case my comments are additional and supplemental to Peter Walkers discussion in the last issue.

Firstly, all versions of XB are based on TI XB, and are merely enhancements to it. All versions other than TI seem to have added some form of bit-map graphics- where you can draw in hi resolution.

TI VERSION 100 is very rare, being the first, and contains a number of bugs, the more serious (for TI) being the incredible ease with which a non-expanded owner can remove the proprietorial "PROTECTION" feature which you can have with XB which prevents programs being saved or listed. Version 100 also worked on the basis that you always had 28 sprites in action, and was consequently rather slow unless (being expanded) you told it otherwise with a CALL LOAD.

TI VERSION 110 is the more common version and forms the standard for which almost all XB programs are written. The most serious omission being the lack of bit map graphics.

then came MECHATRONICS XB from Germany, a heavy module which tended to get a little hot. It had a very powerful but extremely complicated set up for bit map graphics, which, like TI Logo, was NOT bit map graphics but only redefined characters, consequently you could quickly run out of ink.

Next on the scene was MYARC XB which has been around in a number of versions, the latest and definitive version being Vn 2.12 - this has the disadvantage of REQUIRING the Myarc 512k ram card, as the ExBas itself resides in RAM. This does allow for some increase in speed. The Myarc XB has good true bit map graphics, variable screen windowing with both 32 and 40 column text modes.

Myarc XB is in the form of a module which contains a loader plus ram, a disk, and a chip for the ram card.

Added extras include integer and real variables- with integer variables processing a little faster than real ones. You may set a default disk drive and subsequently omit "DSKL" when using OLD and SAVE or RUN. Because more VDP is available, Myarc XB can load AND run those very long TI Basic programs that will NOT load in TI XB, AND the TI Basic extra character sets ARE available in Myarc XB.

CALL CHAR is available from 0 to 255! and you may have 32 sprites. There is a full range of bit map graphics commands- draw, drawto, circle, point, rectangle, fill, write, dcolor, and uniquely you can determine if a pixel is on or off.

You may RUN many program format machine code programs directly. DF80 files written for TI XB may require you to use a utility program to reset VDP registers or may not function at all.

Myarc XB allows you to run many memory image files which otherwise would require a SuperCart (EdAs with 8k ram) as the Myarc module contains ram in the correct location.

Myarc has several unique abilities but does require the Myarc ram card- which is pretty handy anyway!

The disk library has some utilities for Myarc XB.

=====  
The latest version was TRITON SUPER EXTENDED BASIC which amalgamated a number of TI XB mods prepared for Gram device owners. Thus you have such useful new commands as CALL CHIMES, CALL ALOCK, CALL ALL, a very useful CALL CAT, CALL CLOCK (runs slow in the UK), GOSUB and GOTO with variables, but perhaps its most important features would be the enhanced editing features, which allow you to move, copy, delete or resequence PORTIONS of an XB program quite rapidly- a major use of my module- and also it has included on ROM a commercial machine code bit map graphics program called DRAW N PLOT. This one allows you to plot bit map graphics, but only off screen- only when completed can you SHOW them, although there is a cursor on screen when you SHOW which allows direct editing. Graphics can be saved and printed and the format is compatible with TI ARTIST. There is a major conflict in use of VDP between the graphics and an XB program and hang ups may be frequent, especially if you PRINT the pic. Triton XB requires 32k ram for the graphics feature.

There is a graphics program available (CLASS) for Triton XB.

=====  
With all this talk of graphics it is worth adding that the program THE MISSING LINK will work with all versions except Myarc to add true bit map mode graphics, with plotting in or out of windows. It does not bat an eye if you draw a line to a point off screen, and can dump a screen to printer at any time. Picture files on disk are compatible with TI Artist. TML is incompatible with the CLOCK command of Triton XB.

TML is now my language of first choice for graphics, but suffers in not being able to tell me if a pixel is on or off.

+++++  
NEW DISK DRIVE OWNER?

The world is your oyster, with many programs available at very low cost which compare well with PC programs costing LOTSD more.

THE standard graphics program is TI ARTIST at about US\$25, well supported with many disks of graphics, compatible with Triton XB, The Missing Link, RLE and so on.

THE MISSING LINK is well worth having especially if you have any interest in Bit Map Graphics. It too is US\$25 and a demo disk is available from the Group Disk Library.

TI BASE is THE database for the TI, a fully featured database at US\$25 which would cost lots more for any other computer. It is very close to dBase II. You can use it very simply or get stuck into its command language, micros, and so on. NOT easy, if you want something easy stick to the limited format PERSONAL RECORD KEEPING module!

The only significant word processor is TI Writer, which is a line orientated text processor with mail merge capability. It has been much enhanced in the form of FUNLWEB, available from the disk library, which includes also the TI Assembler plus a disk manager including sector editor. If you have a disk drive you must get Funlweb, and print out all the docs. You may not use all its features! but they are there.

Funlweb provides one "environment" for operating in- another is BOOT (disk library) which gives you an easy to configure menu for your disk, as well as the ability to catalogue and print from the menu screen.

All disk owners should have MCOPI (disk library) which will organise your disks in a more efficient manner, making disks with more than 16 files operate more quickly and with less wear and tear.

After that the choice is yours and depends on what YOUR interests (and storage space or pocket money) will run to. The group has an extensive disk library which is well worth perusing- disks are inexpensive, and the catalogue even cheaper- just send three disks and return postage!

Perhaps a quick list of the programs I have on my main two utility disks may help- remember I have a strong interest in graphics! In addition to the usual Funlweb package I have:

T-SHELL, an XB utility which gives you disk catalog and file read functions in the background while you are programming in XB. Available from the disk library.

SQUEEZER a graphics utility which makes 4 tiny TI Artist pics from one full screen one, with different density levels for you to pick the best. (Disk library).

ARCHIVER- packs and unpacks backup disks for more storage on less disks. Must unpack to use! (Disk library).

UNBASHER- makes sense of XB programs with multistatement lines taken to excess, helps to unbug them! (Library).

VDP- allows you to use those early TI Basic programs which use Character Sets 15 and 16, in ExBas (library).

EXTRACTOR- allows you to quickly take out a part of an XB program for use elsewhere (Triton XB can also be used instead)(library).

RLE- graphics display and print program, TI Artist compatible.

ARTDISK and PHOTO from Harry Brashear (commercial) to print TI Artist pics.

TEXTLOADER allows you to take a program in DV80 text format and load it directly into the computer, no editing required. (Library).

MCPPIX and PIXPRO are commercial graphics programs that enable you to see and print pics in McFlix format.

~ ~ ~ ~ ~

Referring to Peters article, the following programs are in the disk library:

Funlweb; TI Writer Vn 4.5; Multiplan Vn 4 (module rqd); PR Base; Fast-Term; EASLoader & Textloader; Ace; System; Archiver; Star; JBM103; Neatlist; Paragon EDP;

Full up to date catalogue in return for three disks and return postage only!

Commercial offerings -which you must buy!- are:

TI BASE (Texaments or Inscebot US\$25); TI ARTIST PLUS! (same suppliers and price); Graphx (available?); Page Pro 99 (Asgard); Oak Tree DEP (available?); The Missing Link (Texaments US\$25);

Semi commercial offering:

TELCO. US\$20 +overseas mail (say US\$25 to 30 altogether):

Charles Earl, 34 McLeod Str, OTTAWA, Ontario, CANADA, K2P 0Z5.

Anyone interested in videotaped tuition? My old tv set, bought specifically for use with my TI, lasted not as long, and developed a very NASTY fault which blew two ExBas modules on me- well, wouldnt be a cheap module would it! Courtesy of Mike Goddard I now use an attractive mono monitor which has the added feature of an extra socket to link to a video recorder...

In theory I could feed the computer into a composite video link on a video, with a mike into the audio in socket, and help you all out with some of those complex programs we now have- seeing it happen in front of you is a great deal easier to take in than a letter or telephone call!

So if there is an interest - AND someone has a battered VHS video recorder which has composite video and separate sound in connections they do not want!!!! - maybe we can be of help! At present I do not have a video with composite video in!

=====

If you do not have a complete set of TI\*MES, back issues are well worth having for reviews of old programs which are still available and which you wont know about unless you get those back issues, some of which are in rare supply (some indeed sold out) so contact Peter Walker today!!!

ISSUE 8 contains a TI Basic program which demonstrates the power of MiniMemory from Basic including constant music; TI Writer tips; first lesson in TI Forth;

ISSUE 9 contains a second article on TI Forth; an important article on VDP Registers for use with TI Basic and mini memory (also now applicable to XB using VPOKE utility); single pixel drawing program in TI Basic for mini memory module.

ISSUE 10: Call Sound with TI Forth; An important FASTER ExBas version of the unduly slow SIN function good for 0 to 1.7 radians (say 97 degrees); Speech with TI Forth; Faster trig for TI Forth;

ISSUE 11: Tutorials on cassette data files and extended basic and TI Forth; connecting a printer to TI PIO socket; and changing a VDP register using only TI Basic and a cassette recorder- no modules required at all! including use of SPRITES (without automation) in good old TI Basic using CALL CHAR...

ISSUE 14: c99for beginners; map of TI Basic program in VDP RAM and on disk; DIY expansion box;

ISSUE 16: Editing a CHARA1 disk file; new CALL COINC to see if sprite is near an "on" dot in usual 32 column screen; illustration of speeding up machine code by using scratch pad ram; TI Writer graphics;

=====

And if you would prefer me not to use a 9 pin dot matrix printer for Rambles, a donated daisy wheel may improve matters - I just don't need one for my personal use! A serial interface and cable would make it easier for me to use side by side with my trusty Epson...

=====

#### IS TI EXPANSION EXPENSIVE?

Certainly it can cost quite a lot of money to chase down an expansion box, with prices in the region of 150-200 for a populated box. However, you must also consider the cost of the programs which you may then purchase, and compare the cost of those programs to similar (or less efficient!) programs for a new computer such as a PC. A PC may be faster and may have more memory, but do you need it? Often all the extra memory is fully occupied by programs which need the memory due to being inefficiently written and/or containing a multitude of features you will never need!

Consider the software when you consider the hardware!!!

The TI99/4A is considerably easier to use and especially to program for than the PC.

#### WHAT IS EXPANSION???

Here in the UK TI never really tried to sell anything other than the console, and extra hardware was almost never seen in the retailers. With many of our present members in the position of having console only, a brief word of explanation may be handy!

There are two forms of extra hardware- "stand alone" and "peripheral cards". The latter require that you have the Peripheral Expansion Box, or PEB, a large and very heavy steel box which has a power supply and room for up to 7 cards and a full height disk drive. Many members use two half height drives but you have to watch the power requirements if you do that!

At the "low end" you may purchase modules which have plugs coming out of them which plug directly into a printer (parallel port) for word processing or a spreadsheet.

The Speech Synthesiser is a small device that plugs into the right hand socket and allows speech with a suitable module- TE2 or XB. There are "stand alone" devices which plug into the right hand socket and allow you to connect a printer (usually these are "parallel" devices, the other standard to RS232).

You may find "stand alone" 32k memory expansion- and if you only add 32k ram, we can supply several machine code programs on cassette for you.

The cards for the PEB offer a wide variety, including the fairly standard printer interface card (RS232 and parallel on board).

The disk drive controller card, to which you attach up to 3 single sided or double sided single density (TI controller) or single or double density (Myarc and Corcomp controllers) 40 track disk drives. The Myarc controller may also use 80 track drives. It is possible indeed to interface with almost any standard of drive, but the standard required for interchange is single density (90k single sided, 180k double sided).

The older stand alone TI disk controller will only function single sided single density.

Ram cards allow you to use RAM chips as though they were disk drives, allowing much faster access- and some may allow you to store the contents even when you switch the computer off, using battery power supplies. Power drain may be high, and there is always a risk of corruption with these- always keep a disk backup! Ram cards may also allow you to use them as printer buffers (you fill the ram quickly with text and this is then sent to the printer as required while you carry on with something else). Myarc ram card allows you to use Myarc Extended Basic as well.

There is a USCD Pascal card (very slow language!) and even a FORTI music card allowing more sound channels. GRAM cards are similar to RAM cards but are configured for use with the special TI internal language GPL.

You can even buy a hard disk controller to operate with a hard disk or two! A hard disk is an expensive storage medium which has much more room than a floppy disk, typically these days 20 million bytes is the minimum compared to 90 thousand bytes for a single sided single density floppy disk! The cost per byte is low but when a hard disk becomes inoperable you of course lose rather more data than when a floppy disk decides not to work!

A MOUSE is a drawing/pointing device which requires special disk software and may -depending on make- plug into the joystick port or RS232 printer port. A TRACKBALL was once made for the joystick port, this was a special version for the TI only made by Wico.

There was the SUPERSKETCH drawing tablet, which connected a drawing arm through two variable resistances to a drawing program in the module- it was possible to cut off the tablet and attach a mouse instead.

The MBX UNIT was a complex box of tricks which would have allowed use of a more complex (three plane) joystick, and for a few modules allowed rather primitive speech control. It also had a touch tablet face for switching, and offered speech without the speech synth, but only with the special MBX modules.

Consoles may also be modified with the addition of a LOAD INTERRUPT switch which causes the console to tear off to a specified memory address when pressed to do a program located there. You need to put the program there and tell the console where it is! The auto-reset when you insert a module can be switched by a manual switch- you can also prevent a module resetting the console. Ask Mike Goddard for details, which are in his book of things to do to a console (10% ok Mike?).

\*\*\*\*\*

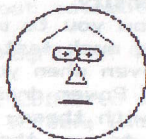
One item of hardware mods omitted from Mikes excellent compendium (plug) is a Load Interrupt Switch- this appeared on page 46 of issue 12. As I now have some information on a program many of you have which can use it! here once more are the details....

At any time you can by means of a hardware switch tell the computer to stop what it is doing and go to a specific memory location and run the program it finds there. This has been used for example for dumping screen displays to printer, although it is also done (better) using a software interrupt, whereby the computer runs the main program and keeps taking time off to say scan the keyboard for a command key- effectivly doing two things at once.

TI have provided us with a program which can use a load interrupt switch, although they did not tell us about it! It is the DEBUG program that comes with editor assembler, and this is how you utilise it...

Insert the Editor Assembler module, and have the disk with DEBUG on it in drive one. Choose TI Basic and type in and run this program:

```
100 CALL INIT
109 REM SEE NOTES BELOW RE LINE 110!
110 CALL LOAD(8228,96,0)
120 CALL LOAD("DSK1.DEBUG")
130 CALL LOAD(-4,131,224,112,190)
140 CALL LOAD(8228,160,0)
150 PRINT "PRESS Q THEN ENTER"
160 CALL LINK("DEBUG")
170 END
```



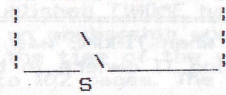
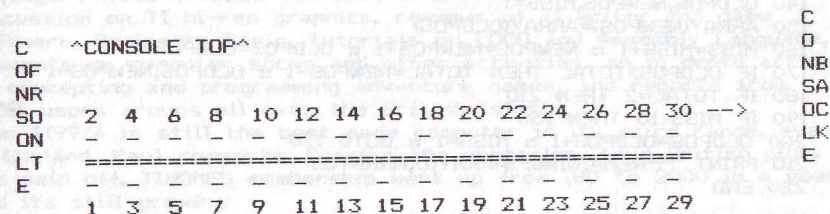
This loads DEBUG. Enter BYE and select an option of Editor Assembler- eg LOAD AND RUN. While the LOAD AND RUN (D/F 80) program you have selected is running- say a game! - press your load interrupt switch and you will be in DEBUG and can have a look around!

For the technically minded, pressing the switch causes the computer to do a BLWP to >FFFC where >FFFC contains the WS pointer and >FFFE contains the Program Counter. Line 130 above puts the addresses into these locations.

Now, if you have DEBUG in memory, then load a program, where is debug going to be? Hmm? The listing above places DEBUG out of the way into ram area >6000 - which you need to provide! Either by modifying an editor assembler module by adding 8k ram or by use of a SUPER SPACE module or similar. Or omit lines 110 and 140, use only relocatable code, and hope there are no clashes.... Line 110 resets FFAH (First Free Address in High Memory) to >6000 while line 140 changes it back to >A000.

OK... that switch! You need to connect a simple switch between to tracks on the right hand port - using a speech synth is best as you dont then need to take the console apart. First, just a simple switch, which due to what is called switch bounce, may send multiple signals to the CPU causing a crash (a good switch will work OK around 60% of the time!) then a slightly improved circuit.

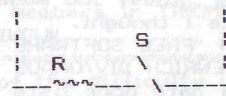
Looking at the edge of the card, at the upper right of the console, pin 13 (LOAD) is the seventh pin from the left, on the bottom. This is one end of your switch. Inside the speech synth you will find that on the SAME side there are four tracks connected- 11,12,13 and 14 from the left - this is the other end of your switch. For improved operation connect a 0.1mF bypass capacitor across the switch and a 2.2k resistor in series with it.



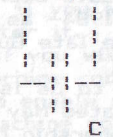
CONSOLE BOTTOM



Better version:



=====
   
RIGHT HAND PORT
   
VIEWED FROM RIGHT
   
=====



As ever, any hardware modifications are at YOUR risk !!!!

WordPerfect is an expensive word processor for some other computer, which has a Spell Checker. The manufacturer has sent out a newsletter with -on one page- the following interesting words: IMNAGES. SIMILARE. DESIGNE. Word processors may have spell checkers but who uses them? Not many folk. (New Scientist 6 Oct 90).

REMEMBER WHEN.... A year ago, back in issue 27 (page 20) I set a couple of puzzles. No response still to TEST 4 but here is a response to TEST 5 which - as you have forgotten it! - is to set up a circle of things and count them off removing every tenth one. If you start at position 1 and the first you remove is number ten, what are the numbers of the last two left?

```
100 ! JOHN SEAGER
110 OPTION BASE 1 :: DIM ARRAY(100)
120 CALL CLEAR :: INPUT "NUMBER OF ITEMS 3>100:";TOTAL :: IF
(TOTAL>2)+(TOTAL<10 1)<-2 THEN 120
130 FOR I=1 TO TOTAL :: ARRAY(I)=I :: NEXT I
140 OLDPOS,NEWPOS,MISS=1
150 ARRAY(NEWPOS)=ARRAY(OLDPOS)
160 MISS=MISS+1 :: NEWPOS=NEWPOS+1 :: OLDPOS=OLDPOS+1
170 IF OLDPOS>TOTAL THEN TOTAL=NEWPOS-1 :: OLDPOS,NEWPOS=1
180 IF TOTAL=2 THEN 210
190 IF MISS<10 THEN 150
200 OLDPOS=OLDPOS+1 :: MISS=1 :: GOTO 170
210 PRINT "":REMAINING:";ARRAY(1),ARRAY(2)
220 END
```

REMEMBER 1983? Back when TIHOME was extinguished just BEFORE TI pulled the plug, leaving TI owners in the UK with NOTHING to fall back on?

I was interested in reading a review of US computer magazines by Bill Gaskill to see that amidst the likes of COMPUTE!, ENTHUSIAST 99, 99ER MAGAZINE, and THE SMART PROGRAMMER, there was TIDINGS - apart from the rather glossy IUG magazine, the ONLY UG publication to be listed. How nice I thought.

Now I have a copy of "FREE SOFTWARE FOR YOUR TI99/4A" published sometime in 1984 by ENRICH DIV/OHAUS in the US of A.

This now very out of date book lists the IUG - which produced its last magazine in June 1984 - as having 80,000 members (!!!!!) - all TI owners - and gives the late Guy Stefan-Romanos San Francisco address as the IUG Library.

Of much more interest to us is the entry for the UK, where the last issue of TIHOME was dated May 1983! There is an interview with Paul Dicks, indicating a membership of 2000. The article makes remarkable reading for those of us in from the start and is quite interesting history for you newer owners!

"Although there are now twenty to thirty thousand TI owners in the British Isles, Paul's group is 'virtually the only resource that TI owners have in England'

"TI users face several obstacles, Paul explains. 'Although the price of the TI home computer is reasonable, the government encourages the BBC Acorn and won't give schools subsidies to purchase TI 99/4 computers. Also, the price of peripherals is almost prohibitive - we're really working on trying to improve this situation. At this time, there are probably only 12 disk drive owners in the entire British Isles! [me-ss here- I got my disk drive

in Autumn 1981]. Third party software is just beginning to appear [in Nov 82 I was selling software by Not Polyoptics, PS Software, Norton Software, Pawterware, and others! By May 83 to this was added Roach Software, Oak Tree Software, FFF Software, Kuhl Software, Maple Leaf Microware - all advertised in TIDINGS - 5 page ad in May 83 issue!!! and there were also ads by UK houses Lantern and Apex!!! - but it's still difficult to even get a copy of 99er magazine from any source other than our service [never mentioned this service in TIDINGS - first I have heard of it!!!ss].

"To help British TI users, Paul founded a by-mail support group that offers serices and helps TI owners contact others in their area to form local users groups.

"TIHOME offers members a resource for public domain software, some computer supplies, publications, local contacts, and technical information.

"TIHOME publishes a 100-page magazine six times yearly, which members receive as part of their annual dues. The magazine offers editorials, comments, and advertisements by members; detailed software and book reviews; and in depth tutorials and program listings. A recent issue features LINE DRAWING ROUTINE as part of a discussion on TI hi-res graphics, reviews of TI Writer and game software, Beginners Basic, tutorials on LOGO and Assembly Language, comments on computer shows and other activities, an in depth article on conceping and programming adventure games, and reports from local users groups all over the British Isles.

"The TI99/4 is still the best made computer in its price range, even in England, Paul comments. Despite the obstacles, true British pluck has paid off. TIHOMES membership went up from 180 to 2000 in a year, and its still growing!

[By the time the book was published TIHOME had been dead for at least 7 months! The comment on membership would place the interview at around April 1983 - about the time of the very last magazine. Only the last two magazines ran to 102 pages. The LINE DRAWING ROUTINE mentioned as being in a "recent issue" actually appeared in the last one, May 83.]

A curious interview especially in regard to its unfortunate timing!

In April 1982 there were 180 members in TIHOME. In August 1990, there were 158 members in TIUGUK.

We have previously advised of the difficulty of using the Editor Assembler module which insists that if you wish to use LIST DEVICE as PIO you must put a full stop after it.

Mike Poskitt has reminded me that if you wish to RUN PROGRAM and type CSI to load a memory image machine code program then also you must add a full stop at the end, to enable multi-file programs to be loaded (some memory image format machine code programs may be on three four five or more files!).

And it is appropriate to remind you that if you have a cassette recorder, extended basic, and 32k ram, but no disk system, you can obtain a number of machine code games ON TAPE from our cassette librarian, and if the demand is there, more can be made available. It is possible to load maybe 70% of machine code programs from cassette, including about 90% of games.

The Southern User Show on Jan 26th is one I shall TRY to get to - weather and rail maintenance permitting. It would help a great deal if anyone travelling by car past Stockport could give a lift, as the rail interchange at London makes my journey into a rather long five hour one (each way!). So all you Southerners, this is your chance to come and ask awkward questions and say what you want in Rambles!

Anyone notice the pic on page 56 of last issue was a mirror image? Just checking to see if you were awake! Apparently not.... hee hee hee!

ATTENTION VIDEO OWNERS: Does anyone have -or have access to- a film with the late Lee Marvin called THE TERROR WITHIN? Please would you take a look at it, with your camera at the ready, to record a cameo appearance towards the end of a..... TI99/4A. And send a pic in please!

Asgard News V2 No2 -undated!!! but after July 89 - gives the following version numbers and suggests return of master disk to Asgard with a small fee (typically \$3 or \$4 but add overseas mail): High Gravity(2.3); Picasso 2 (2); Page Pro (1.5); TOD Editor(3.0) There is a list of famous 4Aers including a Belgian, a German, an Italian, and an Australian. You may believe that there are no UK names listed, but under what circumstances could Tony McGovern have been omitted??? There are the usual many complaints of piracy. I would be grateful if anyone can tell me the name or even country of the "Large dealer of TI software in northern Europe" who is involved in large scale commercial piracy of Geneve (or TI) software. Bobbitt is talking about international court action- which could only be justified by REALLY large scale piracy. It seems to have passed me by...

#### BASIC PROGRAMMING

Nice bit of code here for you to look at. Let's suppose we have CAT=6 and DOG=9, how do we change those round? We could set up a temporary variable like this:

```
10 CAT=6 :: DOG=9
20 PET=CAT :: CAT=DOG :: DOG=PET
30 PRINT CAT;DOG
```

But we can save a little variable storage space and produce more interesting code like this...

```
10 CAT=6 :: DOG=9
20 CAT=CAT+DOG :: DOG=CAT-DOG :: CAT=CAT-DOG
30 PRINT CAT;DOG
```

Try it- it works! And saving variable names is of great value if you are programming for THE MISSINK LINK, which only has limited VDP space for variable storage.

Purely in the interests of science, ExBas programmers can tackle this another way, but it is slower:

```
10 CAT=6 :: DOG=9
20 CAT=CAT XOR DOG :: DOG=CAT XOR DOG :: CAT=CAT XOR DOG
30 PRINT CAT;DOG
```

SEE PAGES 42 TO 44 OF YOUR EXBAS MANUAL and note that this use of XOR will only work if both variables are less than 32767!

MICROpendium will be ending its seventh year of publication with the January 1991 issue, and starting its eighth year with February 1991 making it one of the longest running TI-based publications around. A few user groups have a continuous print record of longer duration (TI\*MES is one, but only just, with this our 31st quarterly issue!) but MICROpendium is the clear winner of "commercial" publications. MICROpendium had a paid circulation of about 3600 at the start of 1990, and if you do not yet subscribe, additional UK subscribers are welcome! The dollar exchange rate at the time of writing makes it a good buy. The seamail subscription is US\$30 per year (12 issues, 40 pages each) but seamail may take six weeks, and you may receive one issue before the one that preceded it! Airmail is only US\$42 per year. P O BOX 1343, ROUND ROCK, TX, USA, 78680

And for those of you not subscribing (fie) the following news was in the October issue:

With both TENEX and TRITON withdrawing from the TI supply scene, only TEXAMENTS and TEXCOMP are left apart from direct-supply producers such as Asgard.

LOU PHILLIPS is said to have another job and a new baby to take care of, leaving little time for MYARC work- reporting on Delphi, Beery Miller reported from a telephone call with Lou that there are to be no more major soft or hardware items Geneve related, that there are virtually no Geneves available for sale, no news on the HFDC, and HFDC and Geneve cards are being sent out for repair to third parties. RAY KAZMER and KEN GILLILAND are said to have joined in a new software venture under the name NOTUNG SOFTWARE (any Wagnerites out there?).

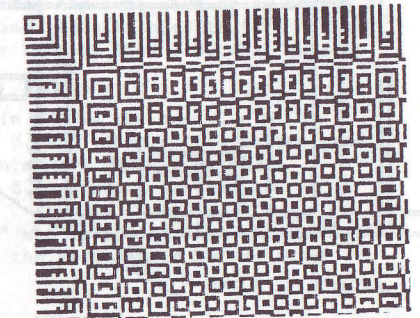
MCCANN SOFTWARE (The Printers Apprentice) have moved to: 4411 North 93rd Street, OMAHA, NE, USA, 68134.

McCann are producing no new TI software; TPA-MDOS version is not compatible with MDOS .97H and no amendments are to be made to TPA-MDOS to make it compatible with any further revisions of MDOS.

Had you subscribed you would have known all this is October!

TI\*MES #30 page 53 No 2:

```
100 REM TI*MES COMPETITION
110 REM ISSUE 30 P53
120 REM PETER HUTCHINSON
130 MILES=30
140 RENT=3
150 GALLON=2
160 CHEAP=9999
170 SPEED=0
180 FOR SP=1 TO 117
190 X=SP/3
200 MPG=40-X
210 COST=MILES/SP*RENT
220 FUEL=MILES/MPG*GALLON
230 TOTAL=COST+FUEL
240 IF TOTAL<=CHEAP THEN 270
250 CHEAP=TOTAL
260 SPEED=SP
270 NEXT SP
280 PRINT "CHEAPEST SPEED IS"
290 PRINT STR$(SPEED);"MPH AT A COST OF"
300 PRINT "$"&STR$(CHEAP)
310 END
```



## CASSETTE REVIEWS.

HUNCHBACK HAVOC. TI BASIC. 5.00 from Database.

This game comes from Lantern Software, a UK publisher of good quality games. In this one you move your man around the screen firstly putting out fires- you may need to unlock some doors, and can only carry EITHER a key OR a fire extinguisher! Once all the fires are out, some gems appear for you to collect and then beat a hasty retreat to screen right before time runs out. You start with 300 time units. For increased game speed this is updated only once in each screen, when you have put all the fires out.

The various screens include handy things like lifts to go up and down in, and even speedy little cars! Graphics are good and game play is excellent.

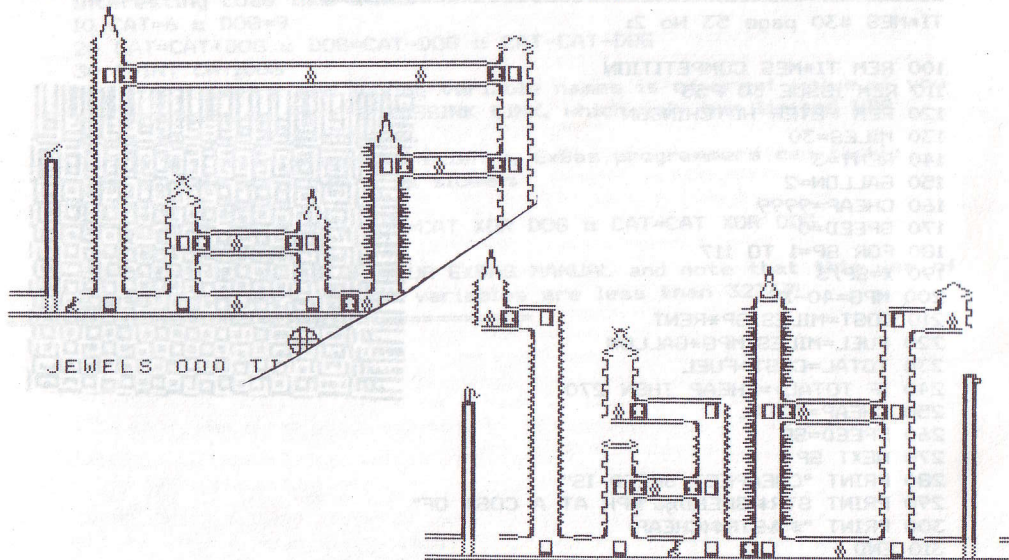
No aliens to zap here. Just a good race game in which you must plan your moves carefully for a fast exit. The fires and gems are randomly placed so each play will offer a different challenge.

This is a game I still play from time to time, and I can readily recommend it to you. Disk owners will have problems moving it to disk due to its length!

This is a quote from the HCW review by "CE" who gave it four stars out of five "A very attractive presentation in which you collect jewels hidden in a nuclear powered castle. Guide Egor through the castle unlocking doors and tackling fires or other hazards in order to turn on the reactor which makes the jewels visible.

"There is a very short time limit before the reactor explodes ending the game. The main difficulty is Egor cannot carry both key and the fire extinguisher so he must scuttle back and forth.

"Four screens show different parts of the castle. Hazards come in six forms including electrical faults and gas leaks, dealt with in a similar fashion to fires. This does not add more difficulty but introduces some graphic variety. Since time is very limited I found it impossible to progress beyond screen four [well done, I havent seen screen four yet! ss], so I cheated by increasing the time limit so as to view all 24 screens- the same four repeated for six types of hazards. Play might become repetitive IF you get that far- not a problem for me though... a well designed game with excellent graphics".



HANG GLIDER PILOT. Extended Basic. From Database 7.50  
From the Canadian software house MAPLE LEAF MICRO WARE.

Formerly sold by Stainless Software in the UK, this excellent game was reviewed in Home Computing Weekly, a now departed UK magazine which reviewed software and printed listings. It earned the maximum award of five stars.

An interesting change to the usual flight simulators, you here are in charge of a hang glider, which you guide around an on-screen map. There is a sea to provide off-shore breezes, hills to bump into, forests to crash land into... you can steer and control rate of ascent/descent, and from a mountain top launch you have three aims in life: i. to stay alive- to land somewhere in one piece! ii. To stay in the air as long as possible. iii. To land in a small target area marked on screen. I have yet to master the last one, although I have managed to come quite close to it!

This is not an especially fast game, although quite fast enough as the mountain or the ground looms ever closer. Definitely challenging, and very well put together with lots of on screen information, including a rising and setting sun for time of day. The eagle shows you where there are thermals for gaining height!

The following description is from the UNISOURCE catalogue of 1984, a good resource of what was available!

"A hang gliding game and trainer... in which time and distance flown must be maximised and concluded with a safe precision landing. Use ridge lift generated by a morning coastal sea breeze (which becomes a land breeze in the evening) and thermals during mid-day. Wind and thermal strengths build and subside in a realistic manner. Various sized clouds and a soaring eagle help in locating the strongest thermals. Very realistic flight characteristics including stalling behaviour and flying and landing techniques."

AIRLINE from Database 7.50 Extended Basic.

An ADVENTURE INTERNATIONAL game- best known as Scott Adams! who wrote the databases for the Adventure Module. This unusual game of strategy is quite different and well worth getting hold of while you can! Never widely available- it was released after the pull-out. It is a sort of trading game in which you operate a fleet of aircraft around the States, represented by a grid of squares on screen. Each landing point has certain trading characteristics which are to be taken into account, and each type of plane you can operate has its own costs and benefits. Hard to describe without repeating the docs, but it is a game of strategy OK? The computer plays fairly well and you will need to think to beat it. Not a shoot em up! Not a flight simulator! I play this one, what more do you want!

Other cassette games from Database reviewed in HCW in the past...

Funpac 2, reviewed by "DB" earned 4 out of 5, and contains one arcade game and two text only adventure games. "entertaining and good fun".

Funpac 3, reviewed by "DB" earned 4 out of 5, and was described as entertaining an good fun.

Pilot, by Apex, earned 4 out of 5 from "MH" who indicated "overall fun and ver addictive at the start. Later on however the program did become a bit boring after I had mastered landing".

STEPHEN SHAW HERE:

I would like to obtain copies of the following games, released in the UK on cassette, but apparently lost without trace. Does anyone have a copy I could borrow/buy?

Soccer Supremo/Dromeda; and Napoleon(card game)/MarKat  
Atarisoft modules which had numbers allocated and were advertised but no-one seems to know of them-anyone have them on module or disk?:

Stargate ATA108; Robotron 2084 ATA109; Joust ATA 113.



REVIEW: TI BASE UTILITIES:  
COMMAND FILE EDITOR and  
EVENTS CALENDAR  
by WILLIAM GASKILL  
2310 Cypress Ct, GRAND JUNCTION, Co, USA, 81506

I received these - as a surprise! - on disk, together with issue V1NS of THE TI-BASE USER, a newsletter on TI Base published by Bill, for a mere US\$22 per year to UK subscribers (payable in US funds). You may well feel that the \$22 is worth it for these utilities alone. Seeing how Bill puts his command files together is a real education, and may inspire you to bigger and better things with your copy of TI BASE (commercial database published by Inscebot and also available from Texements).

In the last issue of TI\*MES you may have seen mention of a utility to merge two command files together- the on board editor only allows a limited length command file, even though TI Base can in fact handle longer. As the 40 column editor allows the use of inverse characters, not available in DV80 command files, the MERGE utility allowed you to prepare several short files and then stick them together for use!

Having stuck them together how to amend them though! Too long to load into the on board 40 column editor!!!! CFE to the rescue, a well-featured editor especially to amend those merged files- though it will not allow insertion of reverse video characters, it will not delete them either!

The CFE utility is in EXTENDED BASIC and uses a 40 column machine code utility by Brad Snyder. There are the usual editing facilities, including disk catalogue, save, load, insert/delete line [ a little slow those two! ], insert/delete character etc. You can dip back to XB command mode then type RUN again, should that be necessary though you will need to reload your text. Very nicely written and very useful.

EVENTS CALENDAR is intended as a demonstration of using TI Base and the command files are well worth printing and examining. It is also a simple but effective diary system, with three fields- date (MM/DD/YY), day (MON), and event (65 chars). You can check to see if anything is listed against the date you entered when TI Base was loaded, or against any date, or day, and there is a FIND utility for the EVENT field, which can find HOLIDAY if you search for OLI! There is also a QUERY EDITOR which is worth a close look as it is something you may wish to copy in your own creations.

If you have TI base, the newsletter (with these utilities) has to be an essential purchase, and if you do not have TI Base, why not consider it!

=====

REVIEW- REMIND ME.

JP SOFTWARE. US\$15.00 plus p&p.

My personal life is so complex that my appointments for the year easily fit onto a single sheet of paper, but if I ever needed to keep a diary, this program is one I might consider, certainly if I did not have TI Base and Events Calendar!

The basis of the program is as follows: You select a month/year and that month is displayed on screen in usual calendar format. You may move around to select a specific day, and check to see what is listed for the day, or edit the entry for the day, which is virtually a full screen of data.

You may print out the entries for any day or series of days within the displayed month- print outs are in two columns of 40 characters. And there is a splendid FIND routine which operates quite quickly. Lets say you wanted to find every day in the month with the word LUNCH in the data. Type in the word and if LUNCH appears anywhere in the data for a day, a tick will appear in the calendar box, and you can move the cursor to the box and check the full data. If the date has any data there is a blob in the box, empty boxes have no data.

A fairly simple utility- useful utilities generally are simple!- and if you need computer assistance to keep track of yourself, this one is well worth looking at.

REVIEW- BOOK- COMPUTERS PATTERN CHAOS AND BEAUTY

Clifford A Pickover.

Hardback. 391pages. ISBN 0-86299-792-5

PUBLISHER: ALLAN SUTTON PUBLISHING. Around 25.00

Not a cheap book this, but a thoroughly splendid read for anyone interested in the odd graphics programs I have offered from time to time in TI\*MES, including some in this issue from this book.

The 27 pages of references are really useful, especially the details of the very many newsletters and magazines you can look to for more graphics experimentation. Although you are invited to write to the author for a free copy of his magazine, his address is omitted- it is suggested you write to his US publisher, but THEIR address is also omitted- anyone interested, contact me! Addresses ARE given for other publications- so many!

There are 17 chapters, and sub headings include:

Bach Beethoven The Beatles; Cartoon Faces in Education; FM Synthesis of Speech; Image Processing of the Shroud of Turing; Is there a double smoothly undulating integer?; Wild Monopodial Tendril Plant Growth; A note on oculatory packing....

Yes there are long words in there! but used with skill and even if you failed math and cannot follow the equations, there is much of general interest. The "recipes" for programs are very very basic program descriptions, quite adequate to work up into graphics programs you can play with. And there are some beautiful illustrations, in mono and colour (and an address for poster versions!- one appeared in a Daily Telegraph supplement in August).

This IS a serious book, one worth looking at again and again, which raises all sorts of questions and ideas of a computer related type. If you are not afraid to wet your feet, it is a good book to start with in exploring serious and semi-serious graphics, and maybe to explore the possibilities of computers a little more.

My copy contained an error in Pseudocode 10.1 on page 153.

theta=theta+angle(i) should read:

theta=theta+angle(j)

Recommended for the adventurous!

=====

SOFTWARE REVIEW- ROCK RUNNER by Eric LaFortune

Distributed by Asgard Software

P O Box 10306, Rockville, MD, USA, 20849

Disk US\$12.95 plus post and packing US\$3 air mail.

Credit cards accepted.

REQUIRES: Disk drive, 32k ram, EDITOR ASSEMBLER MODULE (no other module works!)

This bit is written by Asgard:

" You are about to experience one of the most remarkable arcade games ever written for the TI99/4A computer, and certainly the best new game for the TI in years. Not only is Rock Runner a great game, it is also a technical feat that pushes the 4a to the limits. Written on a cassette system with the minimemory module, Rock Runner utilizes a graphics mode never before seen in any other TI-99/4A or Geneve program- game or otherwise. Dubbed "half bitmap" this new graphics mode mixes the color capabilities of bit map mode with the speed and flexibility of pattern mode graphics to allow an unprecedented amount of full colour animated graphics! Combine this with excellent sounds, beautiful animation, and fantastic action, and you get a game that puts to shame anything on the Atari, Nintendo, Commodore or anything on any other home computer. Rock Runner is a herculean effort, a stunning introduction to the work of this young author."

This next bit is written by Mike Poskitt:

....WOW!



Superlatives aside this IS indeed a very enjoyable game.

The object of Rock Runner is to move your character around the playing area using joystick 1 or 2, picking up a specific number of diamonds to enable you to move to the next level. The playing area scrolls in four directions and the total area is about nine times the visible screen area.

As the character moves he digs a tunnel through which various monsters (from level two upwards) can pursue him. His progress is also impeded by rocks which may fall on him if he tunnels beneath them. In later levels other obstacles are encountered (such as expanding acid pools) and you need to drop bombs and lay traps to destroy monsters which will then turn into the diamonds needed to progress to subsequent levels.

If this isn't enough to keep the adrenaline flowing, you are also fighting against a time factor.

There are 15 levels and you may start at any level, which saves the boredom of repeating easier levels at future attempts.

The accompanying instruction leaflet is more than adequate and even offers advice on strategy.

All in all this is an excellent machine code game.  
\*\*\*\*\*

## ANSWERS TO READERS...

COMPUSERVE, GENie, Delphi and The Source are all United States based information networks, accessed by computer users using suitable software plus a modem plus a normal telephone line. You pay a subscription plus a charge based on actual usage plus the charge made by your telecom supplier for line time. These services are much better developed in the USA than Europe, most likely because Europe has telecom providers who ensure much higher charges than our US cousins face! In the UK, consider how many private citizens subscribe to Prestel. The machine code tutorials from Mack McCormick were originally "uploaded" (put onto) the Compuserve Network, and any subscriber could then "download" (get) the text through their own computer. They were placed onto disk for our disk library. Many disks in the library have come indirectly from the networks. Macks reference to DL3 on page 23 of issue 30 refer to the format and filename used on the original Compuserve files and can be ignored!

The reference to the length of a Basic program line being in the LINE NUMBER TABLE- issue 30 page 43- was incorrect. Let's look at the program at the top of page 44 and see what it is doing...

First PEEK four bytes from -31952 for the top and bottom addresses of the line number table.

Then step through these four bytes at a time- the first two bytes are the line number, then the next two which we are going to use are the address at which the program line starts.

We look at the address where the program line starts LESS ONE because this initial byte is the actual line length. It is NOT used to execute a line, which is in tokenised form, but it is required when we LIST a program as the tokens have to be "undone" and it is easier to put a length indicator in than install a more intelligent undertokeniser.

EXTENDED BASIC TUTORIAL (c) by Tony McGovern of Funnelweb Software PART TWO. This article is a revised version of one which originally appeared in the newsletter of the Australian group TISHUG.

## III. SUBPROGRAM PARAMETER LISTS

In the last chapter we saw how subprograms fitted into the overall workings of Extended Basic. In this chapter we are going to go into the details of writing subprograms. Most of the fiddly detail here concerns the construction of the parameter lists attached to CALL and SUB statements, and some of the little traps you can fall into.

Any information can be transmitted from the CALLING program to the CALLED subprogram via the parameter list, and anything not transmitted this way remains private for each program, with the exception of the DATA pool which is equally accessible to all. If something is mentioned in the parameter list then it is a two-way channel unless special precautions, provided for in XB, are taken. In this case the CALLING program can inform the subprogram of the value of a variable, but not allow the CALLED program to change the value of the variable as it exists in the CALLING program. Arrays however, numeric or string, can't be protected from the follies of subprograms once their existence has been made known to the subprogram through the parameter list.

Let's for starters take a very simple but useful example, where a program needs to invoke a delay at various points. Now some BASICs (and TI LOGO) have a built-in function called WAIT. XB doesn't have this command so you have to program it. It can be done by a couple of CALL SOUNDS or with a FOR-NEXT loop. Let's use an empty loop to generate the delay, about 4 millisec. each time around the loop, and place the loop in a subprogram.

```
230 CALL DELAY(200)
.
670 CALL DELAY(200/D)
.
990 CALL DELAY(T)
.
3000 SUB DELAY(A):: FOR I=1 TO A :: NEXT I ::SUBEND
```

This is easier to follow when editing your program than using a GOSUB, and you would need to enter the subroutine in every subprogram since GOSUBbing or GOTOing out of a subprogram is verboten. Also it's less messy than writing the delay loop every time. The example shows several different CALLs to DELAY. The first supplies a number, and when DELAY is CALLED, the corresponding variable in the SUB list, A, is set to 200. This is a particular example of the kind of CALL from line 670 where the expression 200/D is first evaluated before being passed to DELAY to be assigned to A. Variable D might for instance represent the level of difficulty in a game. The CALL from line 990 invokes a numeric variable T, and A in the subprogram is set to the value of T in the CALLING program at the time when the CALL is executed.

Nothing untoward happens to T in this example, as the DELAY subprogram does nothing to change A. Now it may not matter in this instance if T did not retain its value after the subprogram CALL. Suppose instead the delay was to be called out in seconds. Then a subprogram on the same lines DELAYSEC might go

```

.
230 CALL DELAYSEC(2)
.
990 CALL DELAYSEC(T)
.
4000 SUB DELAYSEC(A):: A=A*250
4010 FOR I= 1 TO A :: NEXT I :: SUBEND

```

Now after DELAYSEC has been executed with the CALL from 990, T will have value 250 times its value before the CALL. This won't be a bother if you don't use T again for its previous value. If the CALLing program specifies a numeric constant as in line 230, or a numeric expression, the change in A in the subprogram has no effect on the main program. Suppose you can't tolerate T being changed in line 990 (and this kind of thing can be a source of program bugs). You will find that XB allows for forcing T to be treated as though it were an expression, thus isolating T from alteration by the subprogram, if T is enclosed in brackets in the CALL (not SUB) list. Suppose DELAYSEC is also called from line

```
970 CALL DELAYSEC((T))
```

If this CALL in line 970 is followed by the CALL from line 990, T not having been altered in the meanwhile, the same delay will be obtained, but if the order of CALLs were reversed the second delay would be ~250 times the first. In the language of XB this is known as "passing by value" as distinct from "passing by reference". This can only be done for single variables or particular array elements, which behave like simple variables in CALL lists. Whole arrays cannot be passed by value, but only by reference. Expressions and constants can only be passed by value, and it's hard to see what else could be done with them. In the example as written, a different variable name was used in the SUB, but if you remember the little experiment in the last chapter you'll see that it wouldn't make any difference if T had been used in the SUB list instead of A.

Now let's complicate things a little by flashing up a message on the bottom line of the screen during the delay interval.

```

.
200 CALL MESSAGE(300," YOUR TURN NOW")
.
270 CALL MESSAGE(T,A$)
.
3000 SUB MESSAGE(A,A$):: DISPLAY AT(24,1):A$
3010 FOR I=1 TO A :: NEXT I :: DISPLAY AT(24,1):""
3020 SUBEND

```

The SUB parameter list now contains a numeric variable and a string variable in that order. Any CALL to this subprogram must supply a numeric value or numeric variable reference, and a string value or string variable reference, in precisely the same order as they occur in the SUB list. In the little program segment above, line 200 passes constants by value and line 270 passes variable references. There is no reason why one cannot be by value and one by reference if so desired.

This process can be extended to any number of entries in the parameter list, provided the corresponding entries in the SUB and CALL lists match up entry by entry, numeric for numeric, string for string. The XB manual does not say so explicitly, but it appears that there is no limit apart from the usual line length problems, on the number of entries in the list. This is the only apparent difference between the parameter list in XB subprograms and the argument lists for CALL LINK("xxxxxx", , ... ) to machine code routines in XB, and Minimemory and E/A Basics.

One little freedom associated with built-in subprograms is not available with user defined subprograms. Some built-ins, such as CALL SPRITE permit a variable number of items in the CALLing list. Parameter lists in user defined subprograms must match exactly the list established by the SUB list or an error "INCORRECT ARGUMENT LIST in ..." will be issued. To compensate for this inflexibility user defined CALLs allow whole arrays, numeric or string, to be passed to a subprogram. Complete arrays may be passed by reference only. Individual array elements may be used as if they were simple variables and may be protected from alteration by bracketing in the CALL list.

An array is indicated in the parameter list by the presence of brackets around the array index positions. Only the presence of each index need be indicated as in A0. MATCH(,,) indicates a three-dimensional array MATCH previously dimensioned as such, explicitly or implicitly. Don't leave spaces in the list. If the subprogram needs to know the dimensions of the array these must be passed separately (or as predetermined elements of the array). TI Basics are weaker than some others in that they do not permit implicit operations on an array as a whole, a very annoying deficiency.

Arrays may be DIMensioned within subprograms. This will introduce a new array name to the program, and an array or variable name from the SUB parameter list can't be used or an error message will result. In the following code the main program passes, among other things, an array SC to subprogram BOARD (perhaps a scoreboard writing routine in a game).

```

.
100 DIM SC(2,5) :: ....
.
450 CALL BOARD(P,A$0,SC(,))
.
4000 SUB BOARD(P,A$0,S(,)):: DIM AY(5):: ....
.
. .... :: CALL REF(P,AY0,S(,))
.
4080 SUBEND
5000 SUB REF(V,A0,B(,)):: .... :: SUBEND

```

BOARD generates internally an array AYO which is passed to another subprogram REF (maybe this resolves ties) along with SC(,), which BOARD knows as S(,), and REF in its turn as B(,) -- the same name could have used in all places. There is however no way that the main program or any subprogram whose chain of CALLS doesn't come from BOARD can know about the array AYO. This would hold equally well for any variable or array, string or numeric, first defined within BOARD and whose value has not been communicated back to the CALLING program via some other variable mentioned in BOARD's parameter list.

By following this line of reasoning you can see that there is no way for a subprogram whose chain of CALLS does not come through BOARD to know about array AYO. The only way around this is for AYO to be DIMensioned in the main program (even if this is its only appearance there) and the message passed down all necessary CALL-SUB chains.

This idea of DIMensioning an array only within a subprogram is particularly useful if the array is to READ its values from DATA statements and to be used in the subprogram. This could be done again from any other subprogram needing the same data, without having to pass its name up and down CALL-SUB chains. Remember that DATA statements act as a common pool from which all subprograms can READ. If the array values are the results of computations then these values must be passed through the CALL parameter lists.

For completeness note that, although the XB manual has nothing to say about it, IMAGE statements for formatting PRINT output are accessible from any part of a program in the same way as DATA statements and not confined to the subprograms in which they occur as are DEF entries.

It is not necessary to have any parameters in the list at all. Subprograms used this way can be very helpful in breaking up a long program into more manageable hunks for ease of editing. We shall also see in later chapters that there can be other benefits as well.

One more XB statement for subprograms remains, the SUBEXIT. This is not strictly necessary as it is always possible to write SUBEND on a separate line and to GOTO that line if a condition calling for an abrupt exit is satisfied. Like a lot of the little luxuries of life however, it is very nice to have and makes programs much easier to read and edit. It does not replace SUBEND which is a signal to the XB pre-scan to mark the end of a subprogram. SUBEXIT merely provides a gracious and obvious exit from a subprogram (awkward in some Pascals for instance). The next chapter will demonstrate typical examples of its use.

---

Note inserted by Stephen: Please do read these articles very carefully. Tony is an excellent ExBas programmer, and these notes should be of assistance to every ExBas owner. Tony wrote this as a result of what he considered to be the appalling quality of some ExBas programs he has seen! Now back to Tony....

#### IV. USEFUL SUBPROGRAM EXAMPLES

In the previous chapter we used as an example a DELAY subprogram which could, with a little refinement, be used to substitute for the WAIT command available in some other languages. You can extend this idea to build up for yourself a library of handy-dandy subprograms which you can use in programs to provide your own extension of the collection of subprograms that XB offers.

For our first example let's take one of the more frustrating things that TI did in choosing the set of built-in subprograms. If you have Minimemory or E/A you know that the keyscan routine, KSCAN, returns keyboard and joystick information simultaneously, while XB forces you to make separate subprogram CALLS, KEY and JOYST, to dig it out. Since these GPL routines are slow it is difficult to write a fast paced game in XB that treats keyboard and joysticks on an equal footing as is done by many cartridge games. On the other hand in games where planning and not arcade reaction is of the essence there is no reason why the player(s) should be forced to make a once-and-for-all choice and not be able to use either at any stage of the game.

The subprogrammers approach to this problem, once he realised that it can be done (and we have commercial XB games where the writers haven't) is to write the game using joysticks, but replacing JOYST by a user defined sub-program JOY which returns the same values as JOYST even when keys are used.

The first step in telling whether keys or joysticks are being used is to check the keys, and if none have been pressed then to check the joysticks. If a key has been pressed then its return, K, has to be processed so that the direction pads embedded in the keyboard split-scan return the corresponding JOYST value. A subprogram along the lines of the one used in TEX-BOUNCE does just this.

```
900 SUB JOY(PL,X,Y):: CALL KEY(PL,K,ST):: IF ST=0
      THEN CALL JOYST(PL,X,Y):: SUBEXIT
910 X=4*((K=4 OR K=2 OR K=15)-(K=6 OR K=3 OR K=14))
920 Y=4*((K=15 OR K=14 OR K=0)-(K=4 OR K=5 OR K=6))
930 SUBEND
```

PL is the player (left or right joystick or side of the split keyboard) number and is unaltered by the procedure. The simple-minded approach for converting K to (X,Y) values by using the XB logic operators (one of the more annoying omissions from console Basic) seems to work as well as any. The subprogram as written checks the keys first but balances this out by putting the processing load on the key return.

This is as good a time as any to sharpen your own skills by working out alternative versions of this procedure, and also by writing one for mocking up a substitute CALL KEY routine to return direction pad values even if a joystick is used.

[ TEX BOUNCE aka TXB is available from our Disk Library! ]

## TIPS FROM THE TIGERCUB

#57

Tigercub Software  
156 Collingwood Ave.  
Columbus OH 43213

The contents of the first 52 issues of this news letter are available as ready-to-run programs on 5 Tips Disks at \$10 each.

And my three Nuts & Bolts Disk, \$15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.

\*\*\*\*\*

### TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an auto loader by full program name on each disk. These are available as a copying service for just \$1.50 post paid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or \$1, refundable for 9-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.

\*\*\*\*\*

I like little programs that load quickly and do just what I want to do at the moment. And one of the things I wanted to do quickly was to find phone numbers. So, I used FUNLWEB to create a little file -

```
SMITH,JOHN (999) 111-2222
BUSH, GEO. (000) 123-1234
GHADDAFI, O. (666)66-6666
```

and all my other frequently called numbers. I SAVED it as DSK1.PHONELIST and wrote this little routine to use it ---> --->

```
100 CALL CLEAR
110 OPEN #1:"DSK1.PHONELIST"
,INPUT
120 DISPLAY AT(12,1):"LAST N
AME?" :: ACCEPT AT(14,1):N$
130 LINPUT #1:M$ :: IF POS(M
$,N$,1)<>0 THEN DISPLAY AT(1
6,1):M$ :: RESTORE #1 :: GOT
O 120
140 IF EOF(1)<>1 THEN 130
150 DISPLAY AT(16,1):"NAME N
OT FOUND" :: RESTORE #1 :: G
OTO 120
```

Now actually, that was all I needed, (even though it did take several seconds to find a name at the end of the file), and it was easy enough to load the file into FUNLWEB when it needed updating. But, programmers are never satisfied, so I decided to write a self-contained program -

```
100 CALL CLEAR
200 DATA "ALDA, ALAN 888-999
9"
201 !@P-
300 DATA "BUSH, GEORGE 111-1
111"
400 DATA "PRESLEY, ELVIS 000
-0000"
499 !@P+
500 DISPLAY AT(12,1):"LAST N
AME?" :: ACCEPT AT(14,1):N$
600 READ M$ :: IF POS(M$,N$,
1)<>0 THEN DISPLAY AT(16,1):
M$ :: RESTORE 200 :: GOTO 50
0
700 ON ERROR 800 :: GOTO 600
800 DISPLAY AT(16,1):"NAME N
OT FOUND" :: RESTORE 200 ::
GOTO 500
```

That funny thing in line 201 turns off the prescan and speeds up initialization. This routine is no faster than the last, but can be updated by editing the program itself. It is limited to about 500 records due to the least-known and greatest weakness of the TI, that string storage is limited to console memory. ---->

But, computer users are paranoid about speed, so I decided to put my data into a pre-loaded array with self incrementing subscript numbers, and find the data by a binary search.

```
100 !QUICKFINDER by Jim Pete
rson
200 DIM D$(50):: GOTO 300 ::
D$(0),X :: !@P-
300 X=X+1 :: D$(X)="ALDA, AL
AN (999) 666-1234"
400 X=X+1 :: D$(X)="BUSH, GE
ORGE (111) 111-1111"
500 X=X+1 :: D$(X)="GHADDAFI
, OMAR (999) 456-1234567"
600 X=X+1 :: D$(X)="KHOMEINI
, AYATOLLAH (666) 666-6666"
700 !@P+
800 INPUT "NAME? ":M$
900 IF M$>D$(X) THEN PRINT "N
OT FOUND": "CLOSEST IS":D$(X)
:: GOTO 800
000 IF M$<D$(1) THEN PRINT "
NOT FOUND": "CLOSEST IS":D$(1
):: GOTO 800
1100 H=X :: S=INT(X/2)
1200 S=D$(S):: IF POS(S$,M$,
1)=1 THEN 1700
1300 S=D$(S+1):: IF POS(S$,
M$,1)=1 THEN S=S+1 :: GOTO 1
700
1400 IF S>M$ THEN H=S :: S=
INT(H/2):: GOTO 1600
1500 S=S+INT((H-S)/2)
1600 IF S=S2 THEN 1800 ELSE
S2=S :: GOTO 1200
1700 PRINT D$(S):: GOTO 800
1800 PRINT "NOT FOUND": "CLOS
EST ARE"
1900 IF D$(S2)>M$ THEN PRINT
D$(S2-1):D$(S2+1):: GOTO 80
0
2000 PRINT D$(S2+1):D$(S2+2)
:: GOTO 800
```

Note that in this case the records must be in alphabetical sequence. New records can be inserted in intermediate line numbers, in alphabetic sequence, always preceded by X=X+1 :: D\$(X)= . Obsolete records can be deleted, and records can be corrected in place if the correction does not change the alphabetic sequence. This idea did not work out as well as I hoped. The maximum number of

records is less than 300, for the reason mentioned above, and this leaves so little free memory that even a binary search is slow. However, for a smaller file this is perhaps the best method. For a large file, the best method is certainly a fixed sequential disk file, accessed by a binary search routine. But, that requires other routines to delete, add or change records, and had best be the subject of another Tips.

-----  
There is apparently a mistaken belief that sprites cannot be used together with my BXB routine. Not so - you can use all 28 of them! However, you cannot change their color with CALL COLOR(#,N). The only other limitations of BXB that I can think of, are that a single CALL COLOR cannot be used for multiple character sets and a single CALL CHAR can only reidentify one character. CALL CHARPAT cannot return the hex code of an ASCII above 143 because those ASCII's were not supposed to be available in Extended Basic. I have used BXB on hundreds of Basic-only programs and have had only two rare problems. If the program contains multiple line feed colons :::::, the computer may rearrange them into pairs of double colons :: :: and lock up. Or, if the colons are before the text, as in PRINT "something" you may get a puzzling error message.

Also on rare occasions you might get an error message indicating the subprogram was called from a line containing a CALL CHAR, if the programmer had inadvertently put more than 16 characters in the hex code. Basic just ignores any extra characters, and XBasic uses them to reidentify the following ASCII, but BXB crashes. \*BxB is in TI\*MES issues 15 and 17.

-----

From the TI\*MES news- letter from England, here is an extremely useful bit of assembly which should be assembled as ALPHA/O and placed on the disk of every joystick program, or imbedded in it with ALSAVE.

```

DEF ALPHA
* save old R12
ALPHA MOV R12,@>FFFC
* 9900 CRU base=0
CLR R12
* signal alphalock key line
SBZ 21
* check alphalock other side
TB 7
* jump if state=on
JNE STATE
* state=off
SETO @>FFFE
* as off skip next line
JMP JUMPA
* state=on
STATE CLR @>FFFE
* stop sending to alpha key
JUMPA SBO 21
* restore R12
MOV @>FFFC,R12
* standard XB return now
* clear error for basic
SB @>837C,@>837C
* return to calling program
B @>0070
END ALPHA

```

Now, put this in the first lines of the joystick program -

```

1 ! by M. Bikow, Andover
MA August 1988
2 ! used with ALPHA/O,
will detect whether
Alpha Lock is up (A=
255) or down (A=0)
3 CALL CLEAR :: CALL INIT ::
CALL LOAD("DSK1.ALPHA/O")
4 CALL LINK("ALPHA"):: CALL
PEEK(-1,A):: IF A=0 THEN DIS
PLAY AT(12,1):"RELEASE ALPHA
LOCK" :: GOTO 4 ELSE CALL CL

```

I published this one in the C.D.N.N.I. newsletter. Barry Traver picked it up and put it in the TI Forum in Computer Shopper, but their typesetter garbled it, so here is how it was supposed to be -

According to the TI-Writer Reference Guide, page 77, when you select the Printf command, then type C and space once and then the device name, any control characters with ASCII less than 32 are removed before the file is printed.

With Funlweb, at least, this is not quite true. A carriage return character, ASCII 13, or a line feed character, ASCII 10, at the end of a line is actually not deleted but is changed to the space bar character, ASCII 32. This can be proved by running this little routine -

```

100 OPEN #1:"DSK1.(filename)
",INPUT
110 LINPUT #1:M$ :: PRINT M$
:: LEN(M$):: IF LEN(M$)>0 THEN
PRINT ASC(SEG$(M$,LEN(M$),1
))
120 CALL KEY(O,K,S):: IF S=0
THEN 120 ELSE 110

```

Therefore, when a file is Filled/Adjusted and the line feed characters are stripped with the C option, the lines are one character longer than they appear to be. An apparently blank line also contains ASCII 32.

Since these characters are blank, they normally do no harm. However, they can create problems when records are read into programs for multiple column printing or concatenation of strings. In these cases, this routine can be used to strip out any ASCII below 33 at the ends of records.

```

100 DATA INPUT,OUTPUT
110 FOR J=1 TO 2 :: READ J$
:: DISPLAY AT(12,1)ERASE ALL
:: J$&" FILENAME?":"DSK" :: AC
CEPT AT(13,4):F$(J):: OPEN #
J:"DSK"&F$(J),UPDATE :: NEXT
J
120 LINPUT #1:M$
130 IF ASC(SEG$(M$,LEN(M$),1
))<33 THEN M$=SEG$(M$,1,LEN(
M$)-1):: IF LEN(M$)>0 THEN 1
30
140 PRINT #2:M$ :: IF EOF(1)
<>1 THEN 120 :: CLOSE #1 ::
CLOSE #2

```

Memory almost full,

Jim Peterson

Additional news from Jim Peterson...

On 19 Oct., Triton informed me that they are no longer supplying products for the TI-99/4A. They referred me to TM Direct Product Marketing, and stated that this firm would have a toll free number 1-800-336-9966 effective 29 October.

T&J Software (515 Alma Real Drive, Pacific Palisades CA 90272) has released Hardback for both the 99/4A and Geneve 9640 as a backup program for Myarc hard disk systems. The price is \$15.

Texaments (53 Center St., Patchogue NY 11772) has released GIF Mania, described as the first program able to DISPLAY standard GIF graphics on an ordinary TI-99/4A. Features include ability to control the overall appearance of images, and to save back to disk in TI-Artist format. GIF is one of the most popular graphics formats for the IBM, and many files are available for downloading. The program costs \$14.95 plus \$3 shipping and requires XB or E/A, 32k and disk system. It operates on the Geneve but does not utilize the Geneve's superior graphics abilities.

Beery Miller reports, in a file on Delphi, that he attended the TI International Fair in Weisbaden, West Germany, and was very impressed. He is of the opinion that the German TI users are more technically advanced than those in the U.S. He saw 7 or more different GRAM devices, including plug-in GRAM modules with 200k + working flawlessly. Programs being demonstrated included the very advanced Copy-C track copier and an 80-column Fractal program for the 4A and Geneve with many advanced features including an animation.

New programs from German authors, which will be released by Miller, include a very powerful stock market analysis program for the Geneve, called \$\$\$CRASH\$\$\$, which Beery will demonstrate at Chicago this year, and a very remarkable program to scroll My-Art graphics.

A CONFERENCE ON TIFORUM  
 ABOUT THE EUROPEAN SHOW IN OCTOBER ON THE EVE OF GERMAN UNIFICATION  
 BY CHRIS BOBBITT, WHO ATTENDED.  
 THIS FILE CAN BE FREELY DISTRIBUTED ON ALL NETS, USER GROUP  
 PUBLICATIONS, COMMERCIAL PUBLICATIONS, OR OTHER MEDIA IF CREDIT TO  
 TIFORUM IS PROVIDED.  
 (Edited for publication by Jim Peterson & Stephen Shaw)

I had the pleasure to attend the Fall '90 All-European TI Show in Weisbaden Germany. I along with Beery Miller who also attended, were the first vendors from North America to make it. It was a pleasant experience. Jim Reiss invited me here tonight to tell about what I saw and did. Tho, honestly, I'm still trying to put everything in order in my mind. A 3-day show is an overwhelming experience. In other words, I'm a bit burned-out still. In any case, I will endeavor. Well, after 8 hours by Pan Am I arrived in Germany tired, and was picked up by my host, Dee Turner, and Jim Fetzner.

Dee's club in the Frankfurt area (where Weisbaden is sort of a "suburb") was sponsoring the show. In any case, they whisked me from the airport directly to the show, which was held at the Burgerhaus (sort of the town hall of, believe it or not, a suburb of the suburb). The first day was set aside for "setup", though there were about 100 people in the hall - many of whom were playing with computers and talking - the actual setup had started on Thursday.

Here, for those who've been to a TI faire, things were immediately different. Instead of the traditional layout over here with user groups selling stuff from booths and local, regional and national vendors, the get-togethers in Germany are just that - informal get togethers. The chairs were on the OUTSIDE of the booths, and massive tables were set up to hold the most bizarre collection of equipment I've ever seen. User groups set up multiple systems and showed off group projects; individuals showed off their own work; and the 2-3 dealers of TI stuff were consigned to the flea market area.

Beery and I, as honored overseas guests, got tables roughly in the center of the hall.

As I said above, the equipment was bizarre. As Jim Fetzner aptly put it, every one in Europe has a "mutant" system. No two were alike. Few peripherals used over here were in evidence and most of the software in use was unrecognizable. Virtually everything in use other than the console was highly personalized, including the software. This presented some problems in demoing stuff on the system provided to me. It was tantamount to taking every unusual piece of hardware shown at all the fairs in the States and throwing it into one room.

Listing it all would take too long, so I'll just hit the highlights.

There are very few 9640's in Germany (they are really P'd about supply, far more than in the States) but everyone else had a Mechatronics 80-column card. Some were highly modified - additional ports, etc. The EPROM in them, owners over here may be interested to know, is about 3 versions beyond what's here.

Most people had IBM keyboards sticking off (literally) their consoles. Evidently there are 5-6 different interfaces floating around over there for attaching them - many of them quite elegant and cheap. (I'm working on importing one in particular).

Most people had GRAM devices or "super cartridges", not traditional 8k or 32k supercards, mind you, but cartridges with dozens of modules and several banks of GROM, or GRAM devices smaller than a cartridge. I picked up a 40k GRAM, 8k RAM device that emulates a GramCracker for about \$75.

There is a wide variety of software for manipulating that sort of thing, including a truly fascinating memory manager utility, a universal GROM loader utility, etc. Some of these I intend to upload in the coming weeks (copyrights respecting).

Barry Traver would be in literal ecstasy at all the flavors of XB in attendance. There was an "XB 3" widely in use, and versions of 99/8 Basic for the 4A, etc. Some of the Basics were quite rich, many were huge programs taking up 40-50k in GRAM.

In that area, I caused quite a stir with the 99/8 I brought. Evidently it was the first one ever seen in Europe, and by the end of the 2nd day it was up and running (I forgot the power supply - which isn't a standard unit) and everything of value had been sucked out of it - including the 32k ROM and the 16k Pascal ROM.

I expect to see it running on a 9640 any day now - at least that is what they told me they were going to do with it. Amazing, the similarity between the 99/8 XB and the Myarc 2 XB (knowing smile).

While it was an all-Europe show (attendees from Holland, Belgium, Austria, Switzerland, Denmark, France and Germany) every country had its own style. TI had maintained a laboratory in Holland for 4A development and the Dutch, being the pirates at heart they sometimes are, took every scrap of technical information not nailed down after Oct. '83. They did the same in the U.S. but the difference is that here the stuff was sat upon by its owners.

The Dutch decided to spread it all over Europe.

In essence, they had their hands on all these technical docs 2 years before it really started to get out over here. It has also endowed their software development with a substantial head start, and shows in their projects. We aren't just talking technical manuals, mind you, but detailed copies, commented, of original TI source code for everything. The Intern manual seems a bit primitive by comparison.

I tried to pick up a bit, but I had to settle for promises from owners. It was considered so "old hat" that people didn't bring extra copies of anything; everyone already had them (for several years).

Other strange hardware included custom P-boxes. It seems I saw only 4 TI P-boxes in the whole place. The most unusual was a 6x2x3 FOOT steel case housing an ungodly number of disk drives, 14 or so slots, and a power supply big enough to power Berlin.

The widely-rumored TI-IBM card interface didn't make it there, but I saw a preliminary design that "almost worked".

I saw several hardware MIDI interfaces, but the demo of Mike Maksimik's largely software interface fascinated them. In fact, I got requests from the builders for enough information to make them software compatible - a promising start.

The software was fascinating. I saw a very well-done stock management package running on a 9640, but I think Beery Miller will be publishing it (he knew the author via 9640 News). I picked up an excellent CAD package from a gentleman who seemed to be looking for a U.S. publisher (we'll see on this). It does work on a 4A with a 192k video RAM card (perhaps updating my 9640 to 192k may make it boot). The most fascinating part of it is that it looked like a Mac program right down to the little TI menu (complete with map of Texas instead of an apple), that when pulled down got you 2 desk accessories and an "About" menu or window.

80-column card owners will appreciate the 2 disks of utilities and programs I will be uploading in the coming week or so. In terms of application software, there is a bit of a shortage over there, and as a result my sales were very good - I more than paid for the trip (sigh of relief), though the much smaller 9640 market hurt Beery's sales (since he primarily sells 9640 stuff).

However, there was tons of systems software, including (hold your breath, everyone) a version of the P-system that doesn't require a P-code card and will run on a stock 4A, or in 80 columns on a 9640 or a 4A properly equipped. Since it only uses the contents of the P-code card, it doesn't seem to be in apparent violation of Pecan's copyrights. As a result, I may be in a position to distribute it soon (at least once I get a final version).

A 9640 version is expected shortly. I guess the Germans are going to save Lou's hide on that score so on. By the way - it runs 6 times faster than it used to, this way. GROM is awfully slow.

Also, I saw a number of highly modified versions of the operating system. Including one gentleman who, in an amazing example of Teutonic patience, ingenuity and perhaps futility, completely rebuilt a 99/4A from scratch (mind you, no faster or better) completely in wire-wrap. He had a custom operating system he called "Proton" running on it that seemed quite ingenious.

He DID have to download it over the joystick port from another machine (at 600 baud) but once up and running it was very intuitive. It DID emit so much RFI noise that he had to house it in a chicken-wire frame. Otherwise the Bundespost (just think of the post office, the phone company, Western Union and the FCC all wrapped up in one) would destroy it for him.

An aside, in Germany you have to register your modem with the government (well, the Bundespost) and speeds higher than 2400 baud are illegal and punishable with steep fines. Also, you have to pay for the specific privilege of using your modem (Southwestern Bell wanted to do that once). This sort of setup is the same in all telecommunications. Until they get rid of that sort of thing, the U.S. has a huge advantage over Germany in this area (and over many other countries in Europe for the same reason).

Oh, well, back to the story. Despite the language barrier, "tech talk" is a universal language. But I spent three days ogling software and hardware. Like U.S. shows, these things tend to be almost stag affairs (rats). But it was fascinating. This is the only TI show I've been to with beer bottles littering the floor and all the tables, and the catering served home-made strudel, espresso and cake, and for lunch a very nice schnitzel.

There were a few Americans present, all U.S. servicemen or dependents. I even met a guy who, until last month, was in my local user group. The morale over there among TI people is very high. The enthusiasm was infectious. I feel much better about the TI community. It certainly improved my morale. There is a sort of "can do" spirit - probably because they were less dependent on TI from the start..

(The SYSOP asked Chris's views on the future of the Geneve and other follow on items) - Well, software-wise I think we are on the verge of a lot of break-throughs on the 9640. Hardware-wise, I'm in a sort of "wait and see" mode. It is widely known that Myarc hasn't shipped anything in months, and it is next to impossible to reach them nowadays. The Germans have written them off and a number of new computers seem to be in mid-development over there.

The 99/8 I brought caused a major commotion in that respect (please remember that they have duplicated every peripheral used in the States and most people own thoroughly non-TI equipment there; as a result, there is a lot of hardware expertise)...they had never SEEN one before. They didn't realize the relative simplicity of the design, and guys took pictures of the boards and counted chips. The existence of another 99/4A compatible in the flesh, I believe, added credence to their own efforts to develop one.

I intend to develop more commercial ties to Germany. The Germans really liked the stuff I brought and were very interested. They view the Americans as very competitive in terms of software design (that is, comparable). In my opinion, they excel at systems related stuff, and North Americans are better at applications software. With the exception of graphics software, in which we are at par.

(The SYSOP - "I understand one fellow known to be a pirate of American stuff did not make the show, Did he hear you were coming?) Well, I really can't comment about that sort of thing. I can say I have two quite busy registered Asgard dealers in Germany now.

Oh - one note about attendance at this show. While it was a 3 day affair and it was harder to count, about 450 people went through it. If the ENGLISH, the Italians and the Swedes showed up it would have topped 600. The reason for the absence of the English is inexplicable considering they promised to come. [Stephen here- TIUGUK made no such promise, they are talking of someone else!]. The Italians and Swedes bowed out due to the distance involved (quite expensive by train and long by car. There is a seeming phobia in Europe about traveling more than 50 miles to a show. In Texas they do that to go to a movie!)

=====

By Jim Swedlow

[This article originally appeared in the User Group of Orange County, California ROM]

[edited by Stephen Shaw]

[FUNLWEB... 4.10 and later..]

The McGoverns have added some new keys to the TI Writer Editor.

<CTRL Q> and <CTRL A> perform a ROLL UP and ROLL DOWN. <CTRL ;> changes whatever is under the cursor to lower case and <CTRL -> to uppercase.

If you start typing with ALPHA LOCK on, just move your cursor back to the first letter and press <CTRL ;>. The letter will be changed to lower case. Better yet, it is a repeater, so if you keep those keys down, everything you typed will be converted. Very nice.

=====

FILE NAMES:

TI Controllers ONLY. [May not apply to]

... MYARC or CORCOMP.

The Disk Controller book says that TI file names can contain any character between ASC 32 and 95 except space and period. Having seen other characters used, I decided to test this. I wrote a simple program to open a file, print something, close the file, open it again, read the text, close it and then delete the file:

```
100 FOR I=0 TO 255 :: ON ERROR 190
110 OPEN #1:"DSK1."&CHR$(I)
120 PRINT #1:STR$(I)
130 CLOSE #1
140 OPEN #1:"DSK1."&CHR$(I)
150 INPUT #1:A$
160 IF A$<>STR$(I) THEN PRINT
    "BAD READ IN";I
170 CLOSE #1:DELETE
180 NEXT I :: STOP
190 ON ERROR 210
200 CLOSE #1:DELETE
210 PRINT "FILE ERROR IN";I
220 RETURN 180
```



Note line 170: CLOSE #1:DELETE. The DELETE command causes your disk controller to delete the file after it is closed. This was necessary as your TI will only allow 127 files per disk and if I didn't delete the files, the limit would have been reached.

So what were the unacceptable file names? Everything over 127 bombed out as did 0, 32 (space) and 46 (period). Everything else worked, including lower case.

The TI Manual recommends against using lower case letters in a file name. You can, but there is a danger of saving a file as "DSK1.myfile", trying to read it as "DSK1.MYFILE" and not finding it. These are two different names to your disk controller.

The point is that everything below 128 (except 0, 32 and 46) can be used in a file name (with a TI controller, anyway).



## FUNNELWEB VERSION 4.11 AND UP...

You may opt to have the User List menu appear first when you load Funnelweb from Editor Assembler. This will let you use Funnelweb as a user defined menu system.

With this off, when you boot Funnelweb from Editor Assembler, the first menu you get is the Editor/Assembler Menu, which always has options for Editor and Assembler. By activating it, you could use Funnelweb as a menu system for a disk of programs of your choice. There are other programs out there, like BOOT, that also serve this function.

This feature can be activated only through the Configuration program. It is called "UL Immediate".

The other major change is a complete rework of DM 1000. Tony McGovern said that the change was fixing a major bug. In fact, the program has been significantly enhanced. Although it is still called Version 3.5, it is very different from the Version 3.5 that came with Funnelweb Version 4.10. Among the changes are:

Option 1 in File Utilities and in Disk Utilities are now identical. You can print a directory AND copy / move / delete / rename / etc. files. The other changes are in this function.

DM 1000 now recognizes lower case letters. Before, if you pressed <c> instead of <C>, nothing happened. Now it works.

If you press the letter <A>, all the <N>'s are replaced with <C>'s. Handy for copying all files on a disk.

When you press <FCTN 6> your command starts immediately. No more pressing <Y> to get it going.

If there is more than one screen of files, when you get to the bottom or top of one, DM 1000 now automatically moves to the next screen.

When you Type a file (that is, display it on your screen), DM 1000 now returns to the file listing. Before, you had to do a new directory after typing a file.

All in all, this version of DM 1000 is a much nicer program to use.

### TI WRITER FOOTERS AND HEADERS

I have been working on a TI Writer Reference Guide. Many of us use TI Writer but do not have the Manual. This guide describes all of the Editor Keys, Editor Commands and Formatter Commands. It should be out by the time this is printed. Here are a few things I found about Footers and Headers.

Most of use do not use Footers and Headers in normal work. However, they can be quite helpful on multi-page documents.

The basic format is:

```
.FO t or  
.HE t
```

Where "t" stands for the actual text. A percent sign (%) in a Header or Footer will be replaced with the current page number. For example:

```
.FO Page %
```

Will print "Page" and the page number on the bottom line of each page. All Headers and Footers start in column 1 and ignore any margin settings you have made. If you are using a left margin of 5, <LM 5>, then you need to put five spaces in front of your text:

```
.FO Page %
```

Headers and Footers will print the text that follows the <.HE> or <.FO> on the same line. If you have your Editor margins set at 1 and 38 (so that you can see everything), a long Header might look like this:

```
.HE User Group of Orange County August,  
1988 ROM, Page %
```

The Formatter will print only "User Group of Orange County August," as the Header. The other line, "1988 ROM, Page %" will print as a regular line of your document.

If you use Begin Page, <.BP>, to start a page, all Headers and Footers must be reset as <.BP> cancels Headers and Footers. Another way to cancel a Header or Footer is the command without any text (<.HE> or <.FO>).

Headers and Footers do not reduce the number of lines you can print per page. They print on the lines that the Formatter skips for the top and bottom page margins (which you cannot change in TI Writer).

Enjoy.

```
=====
X X BBBB # 16
X X B B
X BBBB By
X X B B Jim
X X BBBB Swedlow
```

[This article originally appeared in the User Group of Orange County, California ROM]  
[Edited by Stephen Shaw]

### ON BACK-UPS AND FLIPPIS

If you already back-up your disks faithfully and have decided about flippies, skip this. Otherwise, read on.

BACK-UP's are essential. The first thing you do when you get a new program or disk is back it up. Don't run it, don't modify it, don't catalog it, don't list it - back it up.

Why? Simply put, disks go bad and disk drives eat disks. If your only copy goes bad, it could take days to weeks to get a replacement, depending on the source. Ever read the 'warranty' that comes with some software?

So, make your back-up first. If you buy a program, keep the master (with the maker's label) with your back-ups and use a working copy for every day.

Keep your back-ups and masters in a separate disk box away from your computer. That way you won't use them by mistake.

If a disk does go bad, make sure that your hardware is working by using another disk BEFORE using your master or back-up. Otherwise, you could destroy both copies!

Updating back-ups is vital. If you wrote the program, you probably will revise it more than once. If you get a single program, you will normally add it to an exiting disk. If you don't update your back-up, you will lose your new program if something goes wrong with your working disk.

A FLIPPY is a single sided disk that has been modified to act like two single sided disks. By adding three holes, you can put your disk in your drive upside down and record on the back. Instructions for making flippies can be found in the August, 1984 ROM.

Some folks, like Craig Miller, recommend against flippies. They argue that a disk is designed to turn one way and bad things happen when you flip it over and make it turn the opposite direction.

Others use flippies and claim that they have had no problems. One approach is to use the front as a working copy of one disk and the back as the back-up of another disk.

I compromise by only using flippies for back-up copies, thus reducing the number of back-up disks I need. I don't, however, use them for working disks. I haven't had any problems.

=====

### ASC AND SEG\$

A common coding for determining the ASCII value of the first character of a string is:

```
ASC(SEG$(A$,1,1))
```

This can be simplified by omitting the SEG\$ function:

```
ASC(A$)
```

ASC always returns the ASCII value of the first character of the string. You will get an error if A\$ is a null string (if A\$="").

Suppose you want to lop off the first four characters of a string. You might do this:

```
A$=SEG$(A$,5,LEN(A$)-4)
```

Our 4A's, however, do not compare the third value in the SEG\$ function (the length of the new string) to the length of the old string. Therefore, this works just as well:

```
A$=SEG$(A$,5,255)
```

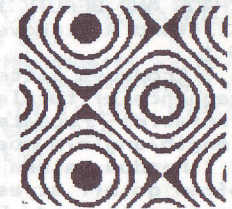
Use 255 as that is the maximum length of a string variable. If the length of A\$ is less than 5 - even if it is zero the new A\$ will be a null string (but NO error).

=====

### Position

One way to ask for a menu selection is to ask the user to input the first letter of his/her choice. Suppose that the options were <C>hange, <P>rint or <Q>uit. You might do this:

```
190 ACCEPT AT(10,10)SIZE(-1)
  VALIDATE("CPQ"):A$ ::
  IF A$="C" THEN 230 ELSE
  IF A$="P" THEN 340 ELSE
  IF A$="Q" THEN 980 ELSE 190
```



A simpler way would be to use POS:

```
190 ACCEPT AT(10,10)SIZE(-1)
  VALIDATE("CPQ"):A$ ::
  ON POS("CPQ",A$,1) GOTO
  230,340,960
```

If your user inputs a null, the POS function will return a value of 1 and control will transfer to the first line in the GOTO list. If this is a problem, do it this way:

```
190 ACCEPT AT(10,10)SIZE(-1)
  VALIDATE("CPQ"):A$ ::
  IF A$="" THEN 190 ELSE
  ON POS("CPQ",A$,1) GOTO
  230,340,960
```

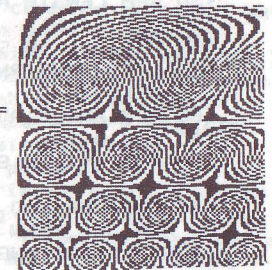
(Source: a Tigerclub program)

=====

### DIM's and SUBPROGRAM's

Will this program work?

```
10 DIM A(5)
20 CALL SETUP(A())
30 SUB SETUP(B())
40 FOR I=1 TO 10
50 B(I)=I
60 NEXT I
70 SUBEND
```



Answer: It will crash in line 50 when I=6. Once A has been DIMentioned in line 10, the process of calling SETUP in line 30 transfers the DIMention to B within SETUP. In essence, there has been a DIM B(5) inside SETUP.

Note that a STOP is not needed at the end of line 20. Your 4A will not execute a SUBPROGRAM unless it is CALLED.

Enjoy!

```

1 ! XB + THE MISSING LINK
2 ! Stephen Shaw August 90
3 !
4 ! from COMPUTERS PATTERN
CHAOS AND BEAUTY 1990 by
Clifford A Pickover.
5 !
6 ! Produces patterns from a
sequence of turtle like
variables- repeats of
length/angle/number of times
7 ! Random values below will
often go off screen as no
scaling is used. Plan and
fill variable arrays with
your own values!
8 !
100 CALL LINK("CLEAR")
110 CT=0
120 RANDOMIZE :: SETS=INT(RN
D*5+1):: CALL LINK("CLEAR")
::CALL LINK("PRINT",180,50,S
TR$(SETS))
130 REM LENGTH ARRAY
140 DIM L(10)
150 FOR I=1 TO SETS
160 L(I)=(INT(RND*8)+18)
170 NEXT I
180 REM ANGLE ARRAY
190 DIM A(10)
200 FOR I=1 TO SETS

210 ANG=RND*360 :: IF ANG<12
OR ANG>350 THEN 210 ELSE IF
(ANG>160)AND(ANG<210)THEN 21
0
220 A(I)=ANG*PI/180

230 NEXT I
240 REM REPEAT ARRAY
250 DIM R(10)
260 FOR I=1 TO SETS

270 R(I)=INT(RND*5+3)
280 CT=CT+R(I)
290 NEXT I
300 OLDX,X=90 :: OLDY,Y=110
310 THETA=0 :: LL=0
320 FOR I=1 TO CT*1.3333
330 REM
340 FOR J=1 TO SETS ! # SETS
350 REM
360 FOR K=1 TO R(J)
370 CALL LINK("PRINT",2,3,ST
R$(I)&"! "&STR$(J)&"! "&STR$(K
))

```

```

380 LL=LL+1 ! # OF VERTICES
390 REM
400 THETA=THETA+A(J):: IF TH
ETA>=2*PI THEN THETA=THETA-2
*PI : 360 IN RADIANS
410 X=L(J)*SIN(THETA)+X
420 Y=L(J)*COS(THETA)+Y
430 CALL LINK("LINE",OLDX,OL
DY,X,Y):: OLDX=X :: OLDY=Y
440 NEXT K
450 NEXT J
460 NEXT I
470 CALL LINK("PRINT",180,30
,"ANY KEY FOR ANOTHER")
480 CALL KEY(S,N,B):: IF B<1
THEN 480 ELSE RANDOMIZE :: G
OTO 100
490 END

```

```

-----
1 ! XB + THE MISSING LINK
2 ! STEPHEN SHAW AUG 1990
3 !
4 ! How random is a sequence
of numbers? This program
uses the RND function but
can take any list of values,
ranged accordingly.
5 !
6 ! from Computers Pattern
Chaos and Beauty 1990 by C A
Pickover.
7 !
8 ! If series is by chance
pattern should stay on
screen, preferably with in
central circle (oval!).
9 !
100 RANDOMIZE
110 SEED=INT(RND*256):: RAND
OMIZE SEED
120 CALL LINK("PRINT",3,200,
STR$(SEED))
130 CALL LINK("BOX",1,1,192,
240):: CALL LINK("CIRCLE",96
,120,35)
140 X,OLDX=96 :: Y,OLDY=120
150 CALL LINK("PIXEL",X,Y)
160 REM
170 C=INT(RND*8)
180 ON C+1 GOSUB 230,240,250
,260,270,280,290,300,310,320
190 CALL LINK("LINE",OLDX,OL
DY,X,Y):: OLDX=X :: OLDY=Y
200 GOTO 170
210 REM
220 STOP
230 X=X-3 :: Y=Y+3 :: RETURN

```

```

240 Y=Y+3 :: RETURN
250 X=X+3 :: Y=Y+3 :: RETURN

260 X=X-3 :: RETURN
270 X=X+3 :: RETURN
280 X=X-3 :: Y=Y-3 :: RETURN

290 Y=Y-3 :: RETURN
300 X=X+3 :: Y=Y-3 :: RETURN

310 RETURN
320 RETURN
330 END

```

If you enjoyed the speed of the bifurcation plot already given, try this one, which has similarities to the affine plots we have given in earlier issues- it is slow! It is based on a code in Clifford Pickovers recent book reviewed elsewhere in this issue.

```

100 ! BIFURCATION PLOT
GENERATOR - PICKOVER
110 CALL LINK("CLEAR"):: RAN
DOMIZE
120 MN=56 :: MX=74 ! FULL-IS
H PIC WOULD BE 0 TO 125-
CHAOS RULES FROM 59 UP.
130 BETA=5 ! LOW VALUE 3
MAKES CHAOS FARTHER AWAY
HIGHER VALUE INCREASES CHAOS
140 RS=180 ! PLOT RESOLUTION

150 X0=1.95 ! START VALUE FO
R Xt [t=0]
160 N=90 ! ITERATION COUNT
USE HIGHER FOR MORE CHAOS
MORE PATIENCE NEEDED
170 RSC=RS/(MX-MN):: CS=48
180 FOR LA=MN TO MX STEP (MX
-MN)/RS
190 CALL LINK("PRINT",1,120,
STR$(LA))
200 X=X0
210 FOR I=1 TO N+10
220 X=LA*X*(1+X)^(-BETA)
230 IF I<11 THEN 250
240 CALL LINK("PIXEL", (LA-MN
)*RSC+11,X*CS)
250 NEXT I
260 NEXT LA
270 CALL LINK("PRINT",1,1,"M
N ")::CALL LINK("PRINT",1,18

```

```

,SEG$(STR$(MN),1,3))
280 CALL LINK("PRINT",1,48,"
MX "):: CALL LINK("PRINT",1,
75,SEG$(STR$(MX),1,3) )
290 CALL LINK("PRINT",1,110,
"B "):: CALL LINK("PRINT",1,
120,SEG$(STR$(BETA),1,3) )
300 CALL LINK("PRINT",1,160,
"X0 "):: CALL LINK("PRINT",1
,180,SEG$(STR$(X0),1,4) )
310 CALL KEY(S,A,B):: IF B<1
THEN 310
320 MN=RND*150 :: MX=MN+RND*
100 :: BETA=3+RND*3 :: X0=RN
D*3+.01 :: CALL LINK("CLEAR"
):: GOTO 160
330 END

```

An earlier program printed patterns of circles using a simple modulus method of deciding if a pixel was on or not. The following program can produce much more complex patterns but again uses a modulus decision making method. The second simpler form uses variable values which produce a very regular pleasing pattern. This program is to experiment with. Instead of using SIN(X)+SIN(Y) try other formulae! And let me have any good patterns - formulae and variable values. By all means set R higher, it just takes longer! And you can plot "to the right" or "to the bottom" by using say FOR I=50 TO R+50 and then when LINKING to PIXEL instead of I use I-49 for the plot. I especially like the patterns when B1=-5, B2=-10, G=66, A=9, and M=2. What happens to a pattern when you halve or double the value of G and leave other variables unchanged?

To see how amending G or M varies the pattern, try say putting in a new line:  
 205 A=INT(I/12)+2 or  
 205 M=INT(I/20)+2

Have fun, experiment, and share your discoveries!

If you don't have the Missing Link (commercial) program, these programs are easily adopted for ANY program that allows BIT MAP (PIXEL) plotting!

95 ! PATTERN MAKING PROGRAM IN TI XB REQUIRES THE MISSING LINK. Can be re-written for other bit map languages/utilities.

96 ! from egg tile generator page 242 of Computers Chaos Pattern and Beauty by Clifford A Pickover.

97 ! for ti by s shaw 1990  
 98 !

100 RANDOMIZE  
 109 ! R= number of pixels down and across of pic.  
 110 R=110

118 ! do not have to use INT function except for M.

119 ! b1,b2=phase shift of sine wave, minor adjustments  
 120 B1=INT(RND\*5)+6\*-1  
 130 B2=INT(RND\*9)+12\*-1  
 139 ! g= frequency of sine wave, main pattern determinant.

140 G=INT(RND\*140)+60  
 149 ! a=degree of disorder, low a=good order  
 150 A=INT(RND\*6)+4  
 159 ! m=modulus, degree of pattern density- m low=more pixels on. Also affects pattern.

160 M=INT(RND\*2)+2  
 170 CALL LINK("PRINT",20,160,STR\$(B1)):: CALL LINK("PRINT",40,160,STR\$(B2)):: CALL LINK("PRINT",60,160,STR\$(G))  
 180 CALL LINK("PRINT",80,160,STR\$(A))::CALL LINK("PRINT",100,160,STR\$(M))  
 190 REM  
 200 FOR I=1 TO R :: FOR J=1 TO R

```

210 X=B1+(G*I)
220 Y=B2+(G*J)
230 Z=A*(SIN(X)+SIN(Y))
240 C=INT(Z)
250 IF C/M=INT(C/M)THEN CALL LINK("PIXEL",I+10,J+10)
260 NEXT J :: NEXT I
270 CALL LINK("PRINT",172,20,"ANY KEY")
280 CALL KEY(5,Z,X):: IF X<1 THEN 280
290 GOTO 100
=====

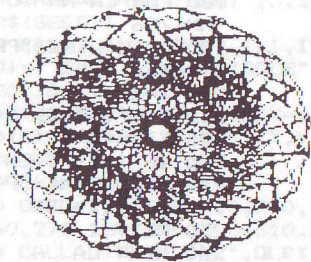
```

Or in a much simpler form:

```

110 R=110
120 B1=-6
130 B2=6
140 G=314
150 A=5
160 M=3
170 !
180 !
190 !
200 FOR I=1 TO R :: FOR J=1 TO R
210 X=B1+(G*I)
220 Y=B2+(G*J)
230 Z=A*(SIN(X)+SIN(Y))
240 C=INT(Z)
250 IF C/M=INT(C/M)THEN CALL LINK("PIXEL",I+10,J+10)
260 NEXT J :: NEXT I
270 CALL LINK("PRINT",172,20,"ANY KEY")
280 CALL KEY(5,Z,X):: IF X<1 THEN 280
290 GOTO 100

```



Well yes, this IS a graphics program, but it is NOT a Fractal program, not even a chaotic program! (Shock, horror!).

What this program does is draw a face! The facial features are set by ten parameters.

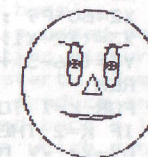
The program is written in XB for use with The Missing Link but it can easily be modified for use with other graphics programs that allow pixel addressing. The image is small enough to be used with the TI BASIC listing given in an earlier issue.

Notice how much work the computer is doing to draw a smiling or frowning mouth!

```

90 RANDOMIZE
99 Z=150
100 ! PICKOVER B3 P327 FACES
110 RANDOMIZE
120 FOR I=1 TO 10
130 P(I)=5+INT(RND*6)-INT(RND*6):: CALL LINK("PRINT",I*10+40,120,STR$(P(I)))
140 NEXT I
150 CALL HEAD(P(1))
160 CALL EYE(P(2),P(7),P(8))
170 CALL PUPIL(P(3),P(7))
180 CALL EYEBROW(P(4))
190 CALL NOSE(P(5))
200 CALL MOUTH(P(9),P(6),P(10))
210 CALL LINK("DUMP") ! to printer
220 FOR I=1 TO 500 :: NEXT I :: CALL LINK("CLEAR"):: RUN
400 SUB HEAD(P1)
401 Z=150
404 DEF CD(X)=COS(X/180*PI)
407 DEF SD(X)=SIN(X/180*PI)
410 EX,EY=0 :: R=30
420 IF P1>5 THEN EX=(P1-5)*2 ELSE IF P1<5 THEN EY=ABS((P1-5)*2)
430 FOR T=1 TO 360
440 X=(R+EX)*CD(T)+50
450 Y=(R+EY)*SD(T)+50
460 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
470 IF T<>1 THEN CALL LINK("LINE",Z-OY,OX,Z-Y,X)
480 OX=X :: OY=Y
490 NEXT T
500 SUBEND
510 SUB EYE(P2,P7,P8)
511 Z=150
514 DEF CD(X)=COS(X/180*PI)
517 DEF SD(X)=SIN(X/180*PI)
520 EX,EY=0 :: R=5
530 IF P2>5 THEN EX=(P2-5)*2 ELSE IF P2<5 THEN EY=ABS((P2-5)*2)
540 P7=P7-5
550 P8=(P8-5)/2
560 FOR T=1 TO 360
570 X=(R+P8+EX)*CD(T)+40-P7
580 Y=(R+P8+EY)*SD(T)+60
590 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
600 IF T<>1 THEN CALL LINK("LINE",Z-OY,OX,Z-Y,X)

```



```

610 DX=X :: OY=Y
620 NEXT T
630 FOR T=1 TO 360
640 X=(R+P8+EX)*CD(T)+60+P7
650 Y=(R+P8+EY)*SD(T)+60
660 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
670 IF T<>1 THEN CALL LINK("LINE",Z-OY,DX,Z-Y,X)
680 DX=X :: OY=Y
685 NEXT T
690 SUBEND
700 SUB PUPIL(P3,P7)
701 Z=150
710 PS=P3/5 :: IF PS=0 THEN PS=.1
720 FOR K=PS TO 0 STEP -.2
730 CALL LINK("CIRCLE",Z-60,40-P7,K):: CALL LINK("CIRCLE",Z-
60,60+P7,K):: NEXT K SUBEND
800 SUB EYEBROW(P4)
801 Z=150
810 Y1,Y2=70
820 Y1=(P4-5)+Y1
830 Y2=Y2-(P4-5)
840 CALL LINK("PIXEL",Z-Y1,35)
850 CALL LINK("LINE",Z-Y1,35,Z-Y2,45)
860 CALL LINK("PIXEL",Z-Y2,55)
870 CALL LINK("LINE",Z-Y2,55,Z-Y1,65)
880 SUBEND
900 SUB NOSE(P5)
901 Z=150
910 P5=(P5-5)/2
920 CALL LINK("PIXEL",Z-55,50)
930 CALL LINK("LINE",Z-55,50,Z-45-P5,46)
940 CALL LINK("LINE",Z-45-P5,46,Z-45-P5,54)
950 CALL LINK("LINE",Z-45-P5,54,Z-55,50)
960 SUBEND
1000 SUB MOUTH(P9,P6,P10)
1001 Z=150
1010 P9=P9-5
1020 X1=40-P9 :: Y1=35
1030 X2=60+P9 :: Y2=35
1040 X3=(X2-X1)/2+X1
1050 Y3=(P6-5)+35
1060 REM
1070 FOR K=1 TO 2
1080 IF K=2 THEN Y3=Y3+P10/2
1090 DX=X1 :: OY=Y1
1100 D=X1^2*(X2-X3)+X1*(X3^2-X2^2)+X2^2*X3-X3^2*X2
1110 A=(Y1*(X2-X3)+X1*(Y3-Y2)+Y2*X3-Y3*X2)/D
1120 BB=(X1^2*(Y2-Y3)+Y1*(X3^2-X2^2)+X2^2*Y3-X3^2*Y2)/D
1130 C=(X1^2*(X2*Y3-X3*Y2)+X1*(X3^2*Y2-X2^2*Y3)+Y1*(X2^2*X3-X3^2*X2))/D
1140 REM
1150 FOR I=X1 TO X2
1160 X=I :: Y=A*I^2+BB*I+C :: CALL LINK("LINE",Z-OY,DX,Z-Y,X)::
DX=X :: OY=Y :: NEXT I
1170 NEXT K
1180 FOR K=1 TO 400 :: NEXT K
1190 SUBEND
1200 END

```



#### GRAPHICS PROGRAMS:

While these are all presented ready to run with Extended Basic plus the disk utility THE MISSING LINK (\$25 + \$8 post from Texaments) the programs can be readily converted for use with any other utility that allows bit map graphics such as Triton XB, Myarc XB or the disk utility JBM103 (from disk library).

The first program although one of the smaller ones, produces some most interesting results. I have used a multiplier of 1.2, but at some stages it pays to take it a little more slowly. In effect you are backing away from a wall which is covered with circles, and as you back away, there is a repetitive effect to the patterns. The circular patterns are technically known as aliases, as the computer is not really plotting circles at all....

```

1 ! XB + THE MISSING LINK
2 !
100 REM CIRCLES
110 REM JE CONNETT/PWH
    MOON/S SHAW 1990
120 SIDE=20 ! 10>2000 value
130 REM
140 CALL LINK("CLEAR")
150 FOR I=1 TO 150 :: FOR J=
1 TO 150
160 X=I*SIDE/150 :: Y=J*SIDE
/150:: C=INT(X*X+Y*Y):: D=C/
2 :: IF D-INT(D)>.1 THEN 180
170 CALL LINK("PIXEL",I+20,J
+20)
180 NEXT J :: NEXT I
187 !
188 ! next bit just stores
pictures! omit if required
189 !
190 PIC=PIC+1 :: A$="DSK2."&
STR$(PIC)
200 CALL LINK("SAVER",A$)
201 !
202 !
203 ! lets look from a
little farther away...
204 !
210 SIDE=SIDE*1.2 :: GOTO 14
0
220 END

```

The following program produces a random dot pattern on the screen and then first enlarges it AND rotates it slightly, and then just rotates it. See what effect these minor manipulations have. There are possibilities for some interesting animations here!

The program to be found in the right hand column is based on work by M Feigenbaum in 1979 and produces some interesting pictures, depending on the limits you set when the program starts. Sometimes looking VERY closely (maximum magnification) will produce an interesting result, at other times it pays to stand back a little. Towards the right of the overall plot is chaos, on which can be found some overlying non-chaotic detail.

The number of "invisible" plots can be varied- failing to plot the first few dots concentrates the display on certain features. Reducing the invisible range will produce a little more detail- if you wish to look at all dots (invisible=0) then try using the 16 colour mode and using a different plot colour for the first 14 plots say- or use all the colours at different plotting stages in the inner loop. More detail will be seen! Have fun! This program is a true fractal as the same shapes keep reappearing.

```

1 ! XB + THE MISSING LINK
Stephen Shaw Aug 1990
2 ! from COMPUTERS PATTERN
CHAOS AND BEAUTY by Clifford
A Pickover
3 !

```

```

100 REM RANDOM DOT PATTERNS-
MOIRE DOT PATTERN
110 REM AFTER PICKOVER.
120 SEED=4 :: RANDOMIZE SEED
130 ! ANGLE OF ROTATION IN
DEGREES:
140 AN=2 :: AN=AN*PI/180

```

```

150 ! SIZE OF PLOT IN PIXELS
:-
160 SZ=140 :: MD=SZ/2

```

```

170 FOR I=1 TO 3000

```

```

180 RANDX=RND*SZ :: RANDY=RN
D*SZ
190 CALL LINK("PIXEL",RANDX+
5,RANDY+5)
200 REM

```

```

210 RANDXX=1.11*((RANDX-MD)*
COS(AN)+(RANDY-MD)*SIN(AN))+
MD
220 RANDY=1.11*((RANDY-MD)*C
OS(AN)-(RANDX-MD)*SIN(AN))+M
D
230 RANDX=RANDXX
240 CALL LINK("PIXEL",RANDX+
5,RANDY+5)
250 NEXT I
260 REM
270 CALL LINK("PRINT",180,20
,"ANY KEY FOR NEXT")
280 CALL KEY(5,A,B):: IF B<1
THEN 280 ELSE CALL LINK("C
LEAR")
290 REM

```

```

300 ! different pattern
310 ! this time we have no
enlargement, just rotation.
320 SEED=4 :: RANDOMIZE SEED
330 ! ANGLE OF ROTATION IN
DEGREES:
340 AN=2 :: AN=AN*PI/180
350 ! SIZE OF PLOT IN PIXELS
:-
360 SZ=140 :: MD=SZ/2

```

```

-----
1 ! XB+ THE MISSING LINK
STEPHEN SHAW AUG 1990
2 !

```

```

3 ! FEIGENBAUM DIAGRAM
Becker & Dorfler / Leon
Heller :-
FRACTAL REPORT #10 AUG90
4 !
100 CALL LINK("CLEAR")
110 CALL LINK("PRINT",1,25,"
VERTICAL AXIS FROM: TO:"):
CALL LINK("PRINT",21,25,"HOR
IZONTAL AXIS FROM: TO:"):
120 A1$="1.140" :: A2$="1.17
1" :: A3$="2.530" :: A4$="2.
586" :: CALL LINK("INPUT",1,
104,BOTTOM,7,A1$)
130 CALL LINK("INPUT",11,147
,TOP,7,A2$):: CALL LINK("INP
UT",31,103,LEFT,7,A3$)::CALL
LINK("INPUT",31,147,RIGHT,7,
A4$):: CALL LINK("CLEAR")
138 !

```

```

139 ! Follow these lines:-

```

```

140 IF RIGHT-LEFT<.0000001 T
HEN LEFT=LEFT+RIGHT :: RIGHT
=LEFT-RIGHT :: LEFT=LEFT-RIG
HT
150 IF TOP-BOTTOM<.0000001 T
HEN TOP=TOP+BOTTOM :: BOTTOM
=TOP-BOTTOM :: TOP=TOP-BOTTO
M
160 IF (TOP=BOTTOM)OR(LEFT=R
IGHT)THEN 100
170 CALL LINK("LINE",1,1,1,2
40):: CALL LINK("LINE",1,24
0,192,240):: CALL LINK("LINE
",192,240,192,1):: CALL LINK
("LINE",192,1,1,1)
180 WIDE=240 :: DEEP=190

```

```

190 CALL LINK("PRINT",3,12,"
<"&STR$(LEFT)&" ^BTM="&STR$(
BOTTOM):: CALL LINK("PRINT"
,2,170,">"&STR$(RIGHT))
200 CALL LINK("PRINT",180,20
,"TOP="&STR$(TOP))
210 VISIBLE=50 :: INVISIBLE=
50
220 SCALE=(RIGHT-LEFT)/WIDE

```

```

230 REM

```

```

370 FOR I=1 TO 3000

```

```

380 RANDX=RND*SZ :: RANDY=RN
D*SZ
390 CALL LINK("PIXEL",RANDX+
5,RANDY+5)
400 REM

```

```

410 RANDXX=1.00*((RANDX-MD)*
COS(AN)+(RANDY-MD)*SIN(AN))+
MD
420 RANDY=1.00*((RANDY-MD)*C
OS(AN)-(RANDX-MD)*SIN(AN))+M
D
430 RANDX=RANDXX

```

```

440 CALL LINK("PIXEL",RANDX+
5,RANDY+5)
450 NEXT I

```

```

460 REM

```

```

470 CALL LINK("PRINT",180,20
,"ANY KEY TO END")
480 CALL KEY(5,A,B):: IF B<1
THEN 480 ELSE STOP
490 END

```

```

240 FOR RANGE=1 TO WIDE

```

```

250 K=LEFT+RANGE*SCALE

```

```

260 P=0.3

```

```

270 FOR I=0 TO INVISIBLE ::
P=P+K*P*(1-P):: NEXT I
280 FOR I=0 TO VISIBLE

```

```

290 ROW=(P-BOTTOM)*DEEP/(TOP
-BOTTOM)
300 COL=(K-LEFT)*WIDE/(RIGHT
-LEFT)
310 CALL LINK("PIXEL",ROW,CO
L)
320 P=P+K*P*(1-P)

```

```

330 NEXT I

```

```

340 NEXT RANGE

```

```

350 GOTO 350

```

```

360 END

```

