

TI*MES



CONTACT ADDRESSES:

EDITORIAL:

Christina Mehew

Holly Croft, 4 Thornton Lane, ULCEBY, South Humberside, DN39 6SG

MEMBERSHIP

Peter Walker

24 Bacons Drive, CUFFLEY, Herts, EN6 4DU

DISK LIBRARY AND JOURNAL EXCHANGE:

Stephen Shaw

10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

CASSETTE LIBRARY:

Maurice Rymill

231 Bournville Lane, Bournville, BIRMINGHAM, B30 1RA

MODULE LIBRARY:

Edward Shaw

Crow Holt Farm, Basford, LEEK, Staffs, ST13 7DU

GENERAL SECRETARY:

Jim Ballinger

5 Offerton Avenue, DERBY, DE3 8DU

CHAIRMAN: Clive Scally

Other Committee Members:

Tim Anderson, Alan Bailey, Neville Bosworth, Bryan Cloud, Geoffrey Coan, Peter Cross, Mike Goddard, Edward Shaw.

CONTENTS

- 1 & 2 1988 TI 99/4A Users Group AGM & Faire in Derby.
- 3 Invitation from Clive. Press Cuttings by Stephen Shaw.
- 4 & 5 TIUGUK Chairmans Report.
- 6 Renewal Time. Reminder from Peter Walker.
- 7-10 Rambles by Stephen Shaw.
- 11 News from the Disk Library by Stephen Shaw.
- 12-14 XE. Orange County UG. Selected by Stephen Shaw.
- 15 & 16 Enhanced Basic by Stephen Shaw.
- 17 & 18 Music by Peter Gunn from TI*NUG.
- 19-23 Reviews by Stephen Shaw.
- 24-31 FUNLWEB Version 4.0 Tony McGovern reported by Stephen Shaw.
- 32 & 33 Tips from Tigercub.
- 34-27 TI Bits. Orange County U.G. Extracted by Stephen Shaw.
- 38-41 Turbo Pascal 99. Example programme by Stephen Shaw.
- 42 & 43 Extracts from T.I.S.H.U.G. by Christina Mehew.
- 44-46 Spad XIII. Chicago Times.
- 47 Editor/Assembler Owners Manual Corrections.
- 48 Module Library. E.H. Shaw.
- 49-54 RS 232 by Peter Walker.
- 55 & 56 Relational Expressions by Christian Mehew.
- 57-59 D.I.Y. Mini Widget Reset Switch & Alpha Lock, by Mike Goddard.
- 60 Repairs & Special Show Hardware Offer, by Mike Goddard.
- I.B.C. Members Classified Advertisements.

1988 TI99/4A

USER'S GROUP

A.G.M. & Faire

Central Suite,

Exchange Street,

DERBY.

Saturday

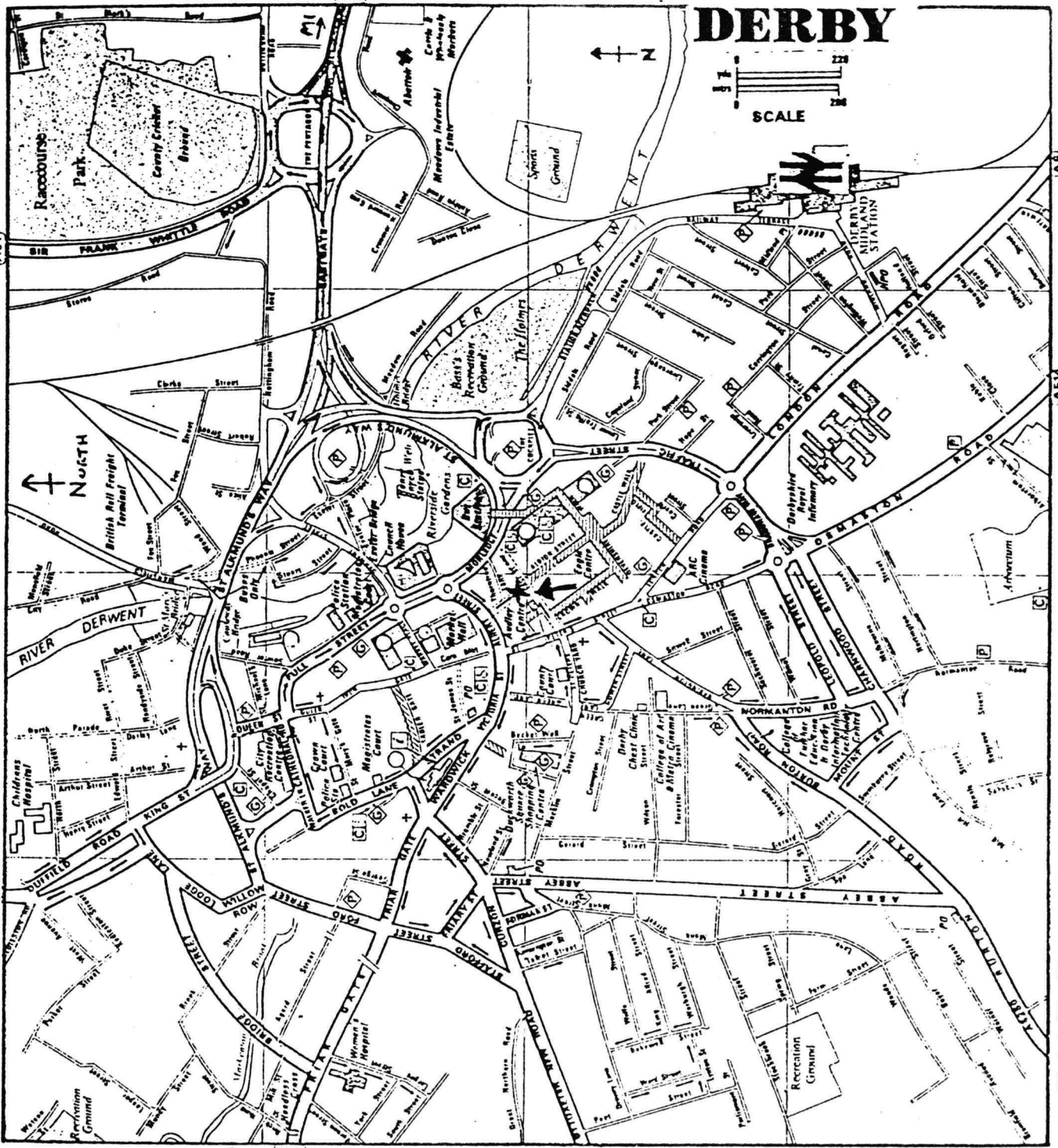
26th May.

Event sponsored by...



TI99/4A USERS GROUP U.K. TUGUK

GRAND TI USERS FAIRE DERBY



Central Suite



* Exchange Street Derby
Central Hall

Entrance in Exchange Street,
between Homestyle
and Albion Travel.
Please take the lift or stairs to the
second floor

- Official Car Park Free
- Multi-storey Car Park
- Parking Available On Payment
- One-way Street
- Convenience

Dear Fellow TI99er User,

This is to cordially invite you to visit T.I.U.G.U.K. second official Annual General Meeting followed in the afternoon by TI Users Groups and representatives SUPER TI FAIRE.

We have been collecting lots of goodies to dispose of. Come and join the BRING AND BUY Sale, Used Hardware/Software, books demonstrations, general discussion groups. TIUGUK manned tables.

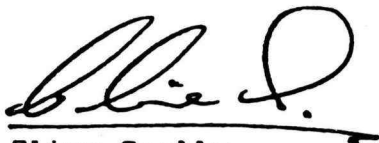
Swap shop, Auction of users goods. Hints and Tips from experts.

See if your own group can attend they will be provided with a table and power free of charge, and of course made very welcome.

The venue is the Central Suite, Exchange Street, Central DERBY. (entrance between Homestyle and Albion Travel. Take the lift or stairs to the second Floor). AGM 11.00am for 11.30am start prompt.

If you are unloading or loading there is a back entrance off Thorntree Lane. (doors open from 9.00am). The location is ideal for families, plenty of facilities. Do come and make this a Great day out.

Every TI99er is looking forward to meeting you again.



Clive Scally.

PRESS CUTTINGS BY STEPHEN SHAW

News that the USCD PASCAL Users Group has dropped to 400 members (this is between ALL computers not just TI)

News of a new PEB card costing US\$500-odd! With an NC4016 with 48k RAM, running CM-FORTH, plus an 8k DSR, and operating speed 5 mips (million instructions per second). Haan. Contact GREAT LAKES SOFTWARE for details.

Comparison of B.data base programs, including CFS and PR BASE: The author of a MICROpendium article loves CFS but places PR BASE higher.

Detail:

On CFS: It offers more features than any other of the programs looked at. For the price and utility offered, CFS is hard to beat.

On PRBase: The best all round application program in my opinion. Fast, flexible, does virtually anything a user would want except number crunching. The most professionally designed application of the group and also the fastest in virtually every respect.

Both CFS and PRBASE are available at low cost from the user group disk library. If you use them, be sure to send a donation to the author(s).

News from Chicago of an incompatibility between "some consoles" and MG's Gram Kracker, and details of the three GK components to replace to fix it- my console is one that refuses to work with the production line GK!.

T. I. U. G. U. K. CHAIRMANS REPORT.

This past year has seen the TEXAS INSTRUMENTS 99/4A USERS GROUP UNITED KINGDOM continuing to develop its activities in all areas.

It might have been thought that after all the euphoria associated with launching T.I.U.G.U.K. that there would come a time when it would settle down to a routine perhaps less exciting existence. That has not happened.

In the shrinking TI99er Worldwide community, this group offers more for the TI99/4a User than ever before. In short we have not stood still.

All this has led to constant reappraisal of our aims, objectives and mode of operation.

Over the last year your T.I.U.G.U.K. committee have been very busy within their sphere of activity.

Your General Secretary, JIM BALLINGER, has maintained a high standard of handling Group affairs, correspondence and magically turning problems into solutions enabling us on the committee to carry out our tasks smoothly.

Your Treasurer, PETER CROSS, has been very prudent in keeping tight strings on your Group funds, with clever investment and tight control we are still here today while others fall by the wayside.

Your Membership Secretary, PETER WALKER, has through his excellent communicative skills keeps us all informed on membership progress. (TIUGUK has grown by 57% over the last year!). PETER W. continues to be the authority on the art of PRESTEL/MICRONET.

Your International representative, "Rambles" Editor and probably the holder of Worlds extensive TI99er Disk Library, STEPHEN SHAW, has ensured that we are kept up to date with anything and everything that is happening on the TI99er scene. Got a problem out there? STEPHEN will try and answer it.

Your TI*MES Editor CHRISTINA MEHEW, has given us editorial that covers most age groups spanning all interests of the TI99/4a. In a male dominated computer world, CHRISTINA has (using her words) given "A future flowering" for our TI99/4a.

Your TI*MES assisting Editor and recently appointed TI*MES printing and distribution co-ordinator, ALAN BAILEY, has helped put a professional face on your magazine, which will be a standard you can expect in future issues.

Your 99er Books/Publications Librarian, BRYAN CLOUD and 99er Module Librarian EDWARD SHAW, both have been seeking out supplies to give us a unique low cost basic alternative to enjoy the TI99/4a to its full.

Your D.I.Y. Projects/Hardware co-ordinator, MIKE GODDARD, has remarkably generated alot of interest in revealing secrets of the TI99/4a. Now MIKE takes on where UK TI Dealers have deserted us. The Worlds first TI99/4a User Group Hardware exchange. If you want to trade in hardware then MIKE is the man to contact. The great thing about this is that all profits go back into the group funds not the private businessman.

Your Cassette Librarian, MAURICE RYMILL, has kept the unexpanded well supplied in programs. TIM ANDERSON takes over this task continuing the good service MAURICE provided.

Your Programming Languages specialist, GEOFFREY COAN, has revealed interesting aspects of TI99/4a language in well written articles, any questions on Assembly to Basic ask GEOFFREY.

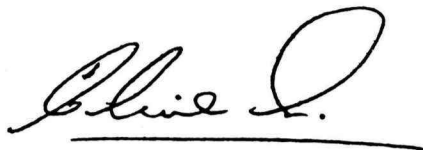
Your Communications/Modems advisor, NEVILLE BOSWORTH, has been keeping the TI99er community happy at his new homebase in the West Midlands.

There is no doubt about it, thanks to your advance support and trust in making an investment, TIUGUK and its dedicated hardworking committe, has seen over the last year sustained growth in activities for the TI99/4a machine, and indeed , in the demand for our services.

To us, it seems encouragaging that where many TI Clubs are a dying breed, there is an undoubted shift towards joining a Nationally owned TI99/4a Group which supports TI99ers without prejudices or political bias.

It is up to us to continue to see improvements in the quality of Texas Instruments 99/4a Users Group U.K. MOST IMPORTANTLY for your committee and its Chairman those elected/ re-elected at the next A.G.M. in DERBY on MAY 28th 1988, TO HAVE YOUR SAME SUPPORT FOR THE COMMING YEAR with the same TRUST you placed in us over the last year. *** PLEASE continue to pledge your support by completing the MEMBERSHIP RENEWAL and forwarding this with your cheque TODAY.

Thanking you in anticipation. Lets continue the good work.



CLIVE SCALLY,
CHAIRMAN
TI99/4a.Users Group United Kingdom.

RENEWAL TIME

For many of you this will be your 4th issue of TI*MES since the group was refounded last May. So for those of you whose subscriptions started with TI*MES No 17, it is now time to renew your subscription for another year. Those of you who joined later in the year, and who started with later issues of TI*MES, you will be reminded about renewal in later issues. For those whose subscriptions do fall due now, you should find enclosed with this issue a renewal form and a SAE (for UK members). You should also find the address label on your envelope is marked "RENEW!". Please send the form back to me as soon as possible, with your remittance, to ensure you receive your next TI*MES.

However, let me remind you that there are plenty of reasons other than the magazine why you should resubscribe. Our Library of Disk and Cassette programs continues to grow, with excellent software being added from all around the world. We now have 2 new libraries, one for the loan of modules and one for the loan of publications. We are now starting up two new ventures: holding a stock of second hand items for you to buy, and a repair service to ensure that your machine will never die. With TI99/4A Users Group (UK) you can stay in touch with all the most knowledgeable experts on the system who are always willing to help you with any problems you may have. Elsewhere in this issue are details of our User Show and AGM - yet another good reason to renew!

In future I hope to print in TI*MES area lists of members so you can meet with other 99ers close to you. I will also be able to put you in touch with any local TI99 groups that already exist. However, not all of you may wish this so I am including in the renewal forms two questions on these points. If you wish to be included on these area lists but are not renewing this quarter, or if you would wish to act as an area coordinator, please write to me separately. I would also like to hear from anyone already running formal or informal local TI groups.

On the renewal form I have left space for you to send us your comments on TI*MES and the group as a whole. I look forward to your renewals and responses and hope you will join us into 1989.

Peter Walker, Membership Secretary.

RAMBLES.

April 1988. by Stephen Shaw

Greetings once more. Your letters of enquiry and requests are always welcome- an SAE for a direct reply please!:

10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

This is also the address for the disk library, with copying costs at just one pound per disk (plus one pound per order to cover post, packing etc etc)- why not send for a list of disk library contents? It comes on two SSSD disks, and all you need to do is send two disks plus return post and packing.

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

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

Type in this program and run it:

```
100 CALL CLEAR
110 A$="1234567890"
120 FOR T=1 TO 12
130 CALL SPRITE(#T,42,2,30+T*3,30+T*3,0,34+T)
140 NEXT T
150 CALL PEEK(-31890,I,M)
160 CALL PEEK(-31974,H,L)
170 A$=A$&A$&A$&A$
180 DISPLAY AT(20,15):"FREE:"
190 F=(H*256+L)-(I*256+M)
200 DISPLAY AT(21,15):F
210 GOTO 140
220 END
```

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

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

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

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

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

Try a garbage collection at an intermediate stage:

```
204 IF F>10000 THEN 210
```

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

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

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

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

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

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

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

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

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

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

MUSIC MAKER:

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

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

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

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

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

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

It would be very nice if, now you know what is needed, one of our talented machine code programmers were to write a routine to input the original file name, read the disk, and make all the necessary changes for you. I cant do it! Consider yourself challenged!

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

To allay a little confusion:

TI Standalone disk controller:

Single Sided 40 track Single Density 360 sector 90k 9 sec/track

TI PEB Card: above OR

Double Sided 40 track single density 720 sector 180k 9 sec/track

Myarc/Corcomp PEB cards:

above OR:

Double Sided 40 track double density 1440 sec 360k 18 sec/track OR

1280 sec 320k 16 sec/track.

Myarc OPTIONAL extension to Myarc PEB Card:

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

COMPATABILITY:

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

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

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

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

DISK DRIVES: All systems should be capable of working with any 40 track "standard" disk drive. An 80 track drive is of value only if you have the extended Myarc controller card.

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

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

.....
MYARC RAM CARD:

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

.....
FORTRAN from Tenex. George Michel reports as follows: " I have the LGMA FORTRAN discs and I have succeeded in producing working object programs.... I concluded that the LGMA discs apply to a very powerful extract of the ANSI 77 FULL language, and is not to be confused with the ANSI 77 SUBSET.

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

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

.....
EXTENDED BASIC PROGRAMMING TIP:

Courtesy John Seager:

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

However, the user CAN blank the default variable with CLEAR, and if the input variable is a numeric variable, trying to input a blank will cause an error condtion.

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

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

```
eg. 100 ON WARNING NEXT
    110 ACCEPT AT(4,5)ERASE ALL VALIDATE(DIGIT):A
    120 GOTO 110
```

 John also mentions that if you program:

```
100 ON ERROR 600
110 RUN "DSK1.NOFIL"
```

```
.....
600 ON ERROR 600 :: RETURN
```

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

This is a deliberate ploy by TI to avoid the "accidental" removal of the List Protection flag- which happens with Version 100 of ExBas.

You need to use the format RETURN XXXX where XXXX is a line number to go to which will RUN your original program again- you could use RUN 110 or something if required. The second RUN will of course reset all required pointers.

TI Cassette software was seen on retail sale in DERBY on February 27th 1988- Virgin Software cassettes at 50p each. There could be other odds and sods around....

| We regret the sad and sudden loss of |
| Ian Martin, who died in February 1988, |
| following a car accident. |
| Ian was the proprietor of TIMELESS SOFTWARE |
| and was responsible for the UK retail sale |
| of DIABLO and other classic XB games. |
| Ian was an avid adventurer and spent much |
time with the Infocom series.

A recent committee meeting of your Group felt that the presentation of Rambles left a little to be desired. Therefore THIS copy is being contributed unpasted, with ALL print in a standard font - ELITE - and double struck for as much density as my printer will allow! If any pieces are MISSING, they have fallen out from the paste up stage, letters to the Editor please.

A.G.M.....A.G.M.....A.G.M.....

In this issue you will see news of the forthcoming AGM (I hope!) - which offers a good chance for TI users to get together and have a chat, AND to express their wishes for the future of THEIR User Group.

Would you like a more professional magazine, either for higher subs, or with less content? Would you like any surplus funds used to make such improvement or for an extended subscription next time or held on to to prolong the group as numbers inevitably drop... WHERE do you wish us to go? Do come and express your views. If you really can't make it, why not drop our Hon Sec a line and tell him what YOU want from the Group!

As a result of comment by non-members, the section dealing with exchange newsletters has been dropped from Rambles. If you feel strongly that I should make a brief list of newsletters received from elsewhere and brief details of what they contain, please write to me. The section takes up a lot of room and unless it is really wanted, we have other (less contentious) material for our Group magazine.

RAMBLES is intended as a friendly letter to a member of the family - the TI family -- YOU. There are PERSONAL comments and feelings and news - and as with ALL articles in TI*MES, these do not claim to be the views of the Group as a whole. I welcome YOUR letters in return, with news, views, feelings, whatever. Remember - this Group is YOURS dear member! If you think Rambles is too long, or should not appear at all, drop a line to our Hon Sec, Jim Ballinger. And to those of you who so kindly write in and advise of your appreciation for Rambles, please accept my sincere thanks. Such letters keep me going.

This issue marks the 6th Anniversary of Rambles, which first appeared in the April 1982 issue of TIDINGS, although my first contribution to TIDINGS actually appeared in issue 2:March 1981 (Pete Brooks got into Issue 1!). When Rambles first hit the streets less than 20 people read it! A lot of words have been used in six years - some of them, on reflection, a trifle shall we say... wrong! but TI never did provide us with much information... the earliest documentation I hold here is dated August 1979!

Stephen

NEWS FROM THE DISK LIBRARY:

For a complete listing on disk, please send TWO disks together with return postage and packing to your group disk librarian, Stephen Shaw.

Ken Gilliland has sent slightly revised versions of his music disks- Startrek, Patsy Cline, Music Man and South Pacific- the changes are minor.

He has also sent along:

DER RING DES NIBELUNGEN by Wagner (please send 2523 disks.... no only kidding!) not the entire Ring, just the Preludes to: Das Rheingold, Act 1; Die Walkure Act 3; Siegfried Act 1, and Goeterdammerung Act 3. Complete with the story in text, and spritely graphics. TWO DISKS REQUIRED.

It may be appropriate to mention that speech is optional on the Patsy Kline, Music Man and South Pacific disks- they work quite well without speech, and indeed you may prefer not to have speech!

Barry Boones ARCHIVER on UTIL 17 has now been updated to Version 2.4, dated 30th January 1988. Several utilities have been added- eg you can now catalogue (on screen) a disk, a packed file, or a compressed file. File utilities have been included- copy file, protect/unprotect file, delete file. It is now possible to use a single drive and two disks. Packing and compressing can often reduce disk usage to 50% of original, and in an extreme case you may even manage to produce a compressed file 20% of the original! Very useful for transmitting files by telephone, or for storing long term backups.

CHARDES on UTIL 18 is updated to Version 5.2, and due to the extra length, the disk contains ONLY Charades! Other files formerly on Util 18 have had to move up to Util 19. The extra facility on Charades is an ANIMATE option. Documentation is now in a separate program file.

NOTE:

>UTIL 13. Withdrawn- John Birdwells DISK UTILITIES has now gone to Vn 4, on a disk of its own. The remaining programs which were on Util 13 have been moved to Util 17 and Util 19.

>TI MATH: This disk contains TI products PHD5006 and PHD5008, Math Routine Library and Electrical Engineering Library. The routines are in EX BAS. Contents include: Bases/Primes/Hyperbolics, Fourier, Function, Ordinary Differential Equations, Simultaneous Equations, Filters, PLL, Root Locus, and Smith Chart calcs. Produced 1980 and 1981.

=====

RAMBLES POST SCRIPT:

Richard Sierakowski reminds me to ask the owners of any MYARC gear to register themselves with him, and Richard will ensure you receive any updates - for instance he tells me that Lou Philips is to produce a Version 2.13 of Myarc XB in the future... if you have 2.12 and want 2.13, write to Richard now!

Tell him WHAT Myarc products you have, and if appropriate, the reference on the EPROM chip - thats the one with a tiny square sliver stuck on it. The Myarc RAM card for instance has an EPROM, and it is important Richard knows the reference number!

Although too short of funds to keep any stock, if you prefer, you may order any Myarc material from Richard, and pay for it in sterling. He will then order it for you and take care of all the customs payments and so on.

RICHARD SIERAKOWSKI, Rusholme, Elcot Lane, MARLBOROUGH, Wilts, SN8 2BA
Telephone (0672) 54975

X.B.

By
Jim
Swedlow

[This article is a summary of XB columns that originally appeared in the User Group of Orange County, California ROM]

[SELECTED BY S SHAW MARCH 1988]

IF THEN

The incompatibility between FOR NEXT and IF THEN statements can cause your program to be awkward. The MAX and MIN statements will often work. For example, instead of this:

```
IF A<6 THEN A=6
```

Try:

```
A=MAX(A,6)
```

TOO MANY GO TO's

I was looking at a text only adventure game and found the following code:

```
370 FOR B=1 TO 58 :: IF A(B)
=L THEN C=C+1 :: GOTO 372
371 NEXT B :: GOTO 374
372 IF C=1 THEN PRINT "You S
ee:"
373 PRINT A$(B) :: GOTO 371
374 ! Program Continues
```

Note that in line 370 the program leaves the loop to print the message and then returns. Then it skips over the print instructions to continue.

This is simpler:

```
370 FOR B=1 TO 58 :: IF A(B)
=L THEN C=C+1 :: IF C=1 THEN
PRINT "You See:":A$(B) ELSE
PRINT A$(B)
371 NEXT B ! Program Continues
```

This saves 25 bytes of memory

MORE IF THEN

Suppose that C\$ can be only "Y" or "N" and that you want to write a routine to change C\$ without knowing its current value.

You might do this:

```
100 IF C$="Y" THEN C$="N" ELSE C$="Y"
```

You don't need to use IF THEN:

```
100 C$=CHR$(167-ASC(C$))
```

TEACH YOURSELF EXTENDED BASIC: This was released by TI to support the XB cartridge. It is available from our Users Group Library for the standard fee.

A working knowledge of BASIC is necessary to understand this material. If you are new to XB or if you have not explored all of XB's features, this is a good tutorial. Even if you are an old hand you might learn something new (see next item!).

The material is clear and presents some information not in the XB book (although most is a repetition). It is primarily text but there are examples, especially for sprites.

On a scale of 0-10 (10 being best), I would rate this at 7.5 - well worth the \$2 and the time to go thru it.

LISTING TO DISK: In the XB book it suggests that you can list a program to a device but the material points you toward a printer. TEACH YOURSELF XB adds that you can list a program to disk. The command is:

```
LIST "DSK1.TEST"
```

The program is now saved on disk exactly as you last saw it on the screen. The file parameters are DISPLAY, VARIABLE 80.

Since those are the parameters for a TI WRITER file you can load the file onto TI WRITER. Why? Well, it could be helpful when doing a newsletter. Also, the FIND STRING command could help you locate something in a long program. Mainly, however, just to see what you could do.

I have not found a way to get the file back to program status. If you could do that there might be some interesting possibilities.

[There are now several ways, including decoding using XLATE or much much faster using TEXTLOADER, both available from the disk library]

NB: This also works in BASIC.

DISK MENU PROGRAM: This month's program will read your disk and display a menu on the screen. After you choose a program, it is loaded and ran. If you save this on your disk as LOAD, it will auto boot when you opt for XB.

This program requires one disk drive and the memory expansion.

When you enter this program, save it to disk BEFORE running it. If you make an error in line 220 or 230, the system could lock up and the program would be lost.

LINES 100-150 comprise the header. This program is based on one published in the Pomona Users Group newsletter.

LINES 160-190 read the programs on the disk and display them on the screen.

LINES 200-210 wait for the user to select a program and then validates the user's selection.

LINES 220-240 change line 240 to have the selected program name rather than "1234567890" and then run that program.

After you get this working, try entering BREAK 240 before running it. When the program stops, LIST 240 to see the change.

```

100 ! DISK MENU PROGRAM
110 ! VERSION XB.1.2
120 ! 29 DEC 84
130 ! FROM THE POMONA (CA)
    99 UG
140 ! MODIFIED BY J. SWEDLOW
150 !
160 DIM A$(18):: OPEN #1:"DS
K1.",RELATIVE,INPUT ,INTERNA
L :: INPUT #1:D$,A,B,C :: DI
SPLAY AT(1,1)ERASE ALL:"DISK
";D$;" * FREE";C: : "Press F
or"
170 INPUT #1:D$,A,B,C :: IF
D$="" THEN 190 ELSE IF ABS(A
)<>5 OR D$="LOAD" THEN 170
180 S=S+1 :: A$(S)=D$ :: IF
S<18 THEN DISPLAY AT(S+4,3):
CHR$(S+64);" ";D$ :: GOTO
170 ELSE DISPLAY AT(22,3):"R
To Continue"
190 DISPLAY AT(24,1)BEEP:"Pr
ess <ERASE> to stop"
200 CALL KEY(3,A,B):: IF A=7
THEN CLOSE #1 :: STOP ELSE
IF A<65 OR A>64+S THEN 200 E
LSE A=A-64
210 IF A=18 AND D$<>"" THEN
CALL HCHAR(3,1,32,32*22):: S
=0 :: GOTO 180 ELSE D$="DSK1
."&A$(A):: DISPLAY AT(24,1)B
EEP:"Loading ";A$(A):: CLOSE
#1
220 CALL INIT :: CALL PEEK(-
31952,A,B):: CALL PEEK(A*256
+B-65534,A,B):: C=A*256+B-65
534 :: CALL LOAD(C,LEN(D$))
230 FOR I=1 TO LEN(D$):: CAL
L LOAD(C+I,ASC(SEG$(D$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
240 RUN "DSKX.1234567890"

```

[EXTRACTED FROM FILES XB01 TO XB05 BY STEPHEN SHAW MARCH 1988]

ENHANCED BASIC.

The original console design concept was that there would be a peripheral device into which you could insert several modules. These would ALL appear on your opening menu screen for selection - that is, you could select modules by SOFTWARE instead of having to use a mechanical switch.

FURTHER, subroutines in each module would be available to you either in machine code OR BASIC.

The peripheral was never built, but you can still sometimes see its shadow with the error message "Review Module Library".

And two modules have routines which you CAN use in TI Basic.

If you insert either the Personal Record Keeping or Statistics modules into the port, and select TI Basic, you have access to additional CALLS which we shall describe here and in forthcoming articles.

If you have a disk system, and run the PRK/Stats modules from DISK (using any suitable loading module or peripheral), after loading the module code, you will find yourself looking at a menu- offering say:

1. TI BASIC
2. EXTENDED BASIC
3. PERSONAL RECORD KEEPING.

If you select 1. TI BASIC, the extra calls ARE still available, just as though the actual module was inserted.

The details of the CALLs seem to have appeared in HOLLAND, via Paul Karis, who wrote an article on CALL A and CALL D for 99er Magazine. We have a printed booklet which seems to be a dump of the TI hard disk archive file "ARCHIVE.PRK.DOC.SUBRLST" and this has been used in putting together this (and preceding) articles.

The archive document indicates also an intention to allow CALL FILES(0), but that was also not implemented.

The extra calls are CALL A (Accept); CALL D (Display); CALL G (Getput); CALL H (Header); CALL L (Load); CALL P (Prep); and CALL S (Save).

These allow you to set aside an area of VDP ram for data storage, move data to and from this area, and to and from the screen, and save and load data stored in the area to an external device IN MEMORY IMAGE FORMAT. This is a very fast way of storing data, especially for cassette users.

We shall look first at the ACCEPT and DISPLAY subroutines as they relate to the SCREEN input and output. Next time round we shall get into the juicy bit of reserving VDP ram and doing something with it.

CALL A...ACCEPT SUBPROGRAM.

There are four formats. The fourth which we shall look at next time, verifies input based on a predefined header.

i and [ii]:

CALL A(ROW,COLUMN,WIDTH,NUMRETVAR,NUMERIC,[LOW,HIGH]).

ROW AND COLUMN are screen positions- 1 to 24 for row, and 1 to 28 for column.

WIDTH is how many screen characters are to be entered. If this takes you past the screen edge, WIDTH is effectively reduced to the room available. You can of course actually enter a shorter input, as trailing spaces will be dropped.

NUMRETVAR must be a numeric variable. Its value is changed according to which key you use to trigger the input:

- NUMRETVAR will be 1 if you use ENTER and the input is valid and non-nul.
- 2 if you use ENTER but the field is blank.

- 3 if you press AID (Fctn 7)
- 4 if you press REDD (Fctn 8)
- 5 if you press PROC'D (Fctn 6)
- 6 if you press BEGIN (Fctn 5)
- 7 if you press BACK (Fctn 9).

- think about how you can use this function!!

NUMERIC is a numeric variable, into which your input is placed.

LOW and HIGH are optional, and can be used to specify a range of acceptable inputs- perhaps you want the user to enter a number between 56 and 121? Not quite the same as the VALIDATE command we are used to in ExBas eh!

iii. CALL A(ROW,COLUMN,WIDTH,NUMRETVAR,STRINGVAR)

is as above except that it is used to input a string variable. There is no range checking available.

Notes: Additional checking is possible using a fourth format which refers to a predefined Header, which we shall cover next time.

Row and Column numbers are MOD(24) and MOD(28) respectively- it wont crash if you use a number over 24 or 28. If you try to use a negative or zero number, the default of 1 is used.

Play around with this a while. Its not too difficult!

Even easier is:

CALL D- DISPLAY SUBPROGRAM.

CALL D(ROW, COLUMN, WIDTH, VALUE,,,,,,,,)

Where ROW and COLUMN are screen positions, MOD(24) and MOD(28) respectively, with zero or negative values read as 1.

WIDTH is interesting...

If WIDTH is POSITIVE, then that number of characters will be replaced with spaces before the display is inserted. If the display is wider than the specified width, the display is truncated.

If WIDTH is NEGATIVE, the display will be inserted up to the specified length, but any excess length is not cleared.

WIDTH is reduced to the distance to the edge of the screen if you use too high a number.

VALUE may be a number, string, or numeric or string variable.

You may use CALL D to display as many items as you can fit into your BASIC program line, specifying the parameters for EACH display.

Try it.

SAMPLES:

```

100 CALL CLEAR
110 CALL D(1,1,28,"Using CALL D and CALL A")
120 CALL D(3,1,28,"PRK/STATS sub programs")
130 FOR T=1 TO 30
140 CALL D(5,1,3,T,5,5,2,-T,5,10,1,T,5,15,4,T/7)
150 NEXT T
160 CALL HCHAR(5,1,32,32)
170 CALL D(6,1,28,"Input a number from 6 to 9:")
180 CALL A(7,20,4,RTN,NBR,6,9)
190 CALL HCHAR(12,1,42,160)
200 CALL D(13,1,28,"DIVIDED BY 12 IS:")
210 CALL D(14,12,-5,NBR/12)
220 CALL D(9,1,28,"PROC'D for next section")
230 IF RTN=5 THEN 240 ELSE 180
240 CALL HCHAR(6,1,1,32,320)
250 CALL D(12,1,28,"PRESS:",13,2,28,"REDD or PROC'D")
260 CALL A(14,12,2,RTN,NUL$)
270 IF RTN=4 THEN 100
280 IF RTN=5 THEN 290 ELSE 260
290 CALL CLEAR
300 CALL D(12,1,28,"End of Demo")
310 END.
```

to be continued=====

Stephen

MUSIC : PETER GUNN

Based on an article by Robert Montgomery in the Feb 87 TISHUB News Digest.

There are lots of ways of programming music- but this program caught my eye because of its unusual method.

I have made a couple of slight alterations to Roberts program- mainly, although a frequency of 30000 is silent, unfortunately a frequency of 30000/4 is not!

DATA is read into a number of arrays.

F(I) and C(I) become frequencies, T(I) is time. M\$(I) and B\$(I) are strings containing the musical score, subsequently broken up.

In LINE 360 two loops are set up- the second loop takes the length of M\$(I) to determine how many times it goes round- the loop is stepped in twos.

In LINE 370 (duplicated in lines 400 and 430), we have a very complex way of making music!

Time:

-T(ASC(SEG\$(M\$(I),J,1))-48):
I is the number used in the outer loop.

J is the number used in the second loop, and is used to indicate the starting position for SEG\$.

M\$(I) is the MELODY line, and contains the TIME and MELODY.

SEG\$(M\$(I),J,1) selects one character from the string M\$(I) beginning at position J.

Because the TIME is negative, the values used for T(I) are of no practical consequence- the negative duration makes the sound start as soon as the computer reaches the CALL SOUND. However, the duration has been included to help us to "read" the music!
You can change the "T" value in the CALL SOUNDS to a straight -60 with hardly any effect- the music just plays

a little bit faster, as the computer has less work to do.

M\$(I) contains the melody in the second character and then every other character- hence the STEP 2 in line 360.

Frequencies are set by using a letter of the alphabet, so that from array F(n), "A"=698, array C(n), "A"=262.

B\$(I) holds the rhythm line- not only is there a dominating line, there is also a secondary counter melody!

The counter melody uses the second voice, using array C(n)... The first character of B\$(I) is a D, with a frequency in array C(n) of 350.

The rhythm using the third voice uses the F(n) array.

For timing, each individual note in the arrays is considered to have a time length of one eighth, so for a note to last one half note, it is repeated in the string four times (4 times one eighth = one half).

WHY program music in such a complex way?

1. It uses up a great deal less memory than any other method!
2. Set up time is quite short compared for example to loading every single note into an array individually.
3. If you manage to follow the programming, you will have learned a lot about Basic!
4. If you key it in without error, you will learn a lot about keying in- and if you key it in WITH errors, you can learn a lot about debugging.

Stephen Shaw.

NOTE: DATA lines 240, 270, 280, 300, 310, 330, do NOT contain any number zero's! They are all LETTER O's, as in Oh dear!

REVIEWS:

EPYX 500XJ JOYSTICK by Konix :

(Sold in the U K for 12 pounds as the KONIX SPEEDKING)

Purchased from Tenex for US\$15.95. They also sell the Wico BOSS Joystick for US\$16.95- the Boss was on sale in Manchester at Christmas time for just £4.95! The Epyx 500XJ joystick is made in the UK! and it requires an Atari to TI adaptor (\$4.95 from Tenex).

(My wife is becoming concerned at my collection of joysticks, but each has its plus features!). Most important note: The Epyx 500XJ joystick is contoured to be held in the hand- but ONLY the left hand. It cannot be used in the right hand. The unique shaping does make it quite easy to hold- I have fairly small hands (7 1/2 for the technically minded!) and suspect a child would not find it so easy.

The joystick operates true microswitches, not the common leaf contacts in so many other joysticks. The action is positive with good loud clicks! Only a small amount of movement is required, and diagonals are fairly easy to get. The trigger button is part of the special contouring and is operated by the middle finger of the left hand (if your hands are larger you could possibly use the first finger!). Trigger action is positive and quite rapid firing is possible, although I found my finger tiring rapidly when faced with long spells of rapid fire! (It is preferable if the program allows you to just hold the fire button down rather than requires separate pushes!).

In general, a well made and comfortable joystick, taking priority over my Prostick II joystick, which is relatively uncomfortable to hold- although the Prostick can be held in either hand. (2021 update: Both named joysticks still in use)

.....

Book: COMPUTE!'S BEGINNERS GUIDE TO ASSEMBLY LANGUAGE ON THE TI99/4A by Peter Lottrup. US\$13.95 from Tenex. 262 pages, paperback. Copyright date is 1985!

Another assembly language book based on the Mini Memory module plus line by line assembler- but also applicable to the main Editor Assembler package. Helpful detail on actually USING the LBLA.- the instructions TI supplied were inadequate to say the least!

Apart from being of inestimable value to anyone without expansion, but with a mini memory module, wanting to write short machine code routines with it, this book is also of great value to anyone learning TI machine code who is using some other assembler, such as the Editor Assembler. Each instruction is very carefully explained and illustrated- except that again there is no explanation of XOP. Does ANYONE have a book or magazine article explaining XOP please!

Addressing modes are described briefly but usefully. The sections on using KSCAN (and Joysticks!) and Sprites are very useful. Moving variables between Basic and Machine Code is explained- but NOT passing Arrays. Sounds and Graphic Modes are explained.

One serious black mark- throughout the book, direct addresses are used to Mini Memory routines. eg BLWP @>6018. No Ed/As equivalents are given in the text, and to save a page, there is no appendix. You will NEED to have the mini memory manual if you wish to understand this book if using Ed/As. And no mention is made of the different environment of Extended Basic- which is also very badly described in the EdAs manual.

Summary: An EXCELLENT book for the mini memory owner, and a good book if you are learning machine code with Editor Assembler: provided you also have the Mini Memory manual!

...

A SHORTCUT THROUGH ADVENTURELAND VOL 2: INFOCOM...ISBN 0-88190-358-2. (c)1984 Paperback, 189 pages, by Lynn,Ashley and Sloan, published by DATAMOST.

Covers the first ten Infocom adventures, from Zork 1 to Infidel. For each adventure there are several sections. First there is a general introduction, which tells you what you are supposed to be doing! including where applicable a list of treasures you need to collect. Then there are general hints, including odd restrictions on what you can do where, which will save you wasting time trying to do impossible things! Also some important problems are solved-eg what do you do to get past the troll in Zork 1, and how do you kill a Cyclops (you can't!).

Next comes ROOM NOTES, a detailed LIST of the rooms, their names and notes on what they are there for, including sometimes detailed hints on problem solving in a particular room. Next item is a detailed HOW TO WIN section- essentially a list of what you must do to win. And finally a room plan.

You may notice when loading Infocom games a REVISION number- my Zork I is revision 75! - as a result of changes to the Infocom adventures, a detailed book such as this one does NOT enable you to zoom through an adventure, as inevitably there have been changes to the detail! Those you must sort out for yourself! This book will give you a good start though and MAY help you with a really difficult problem.

Reading this book convinced me of two things- 1. NO WAY could I possibly complete ANY Infocom adventure. 2- I would not like to be left alone with anyone who had alone and unaided completed an Infocom adventure! The book does however allow you to get the set task out of the way, enjoy the text, and then have a little harmless fun generally exploring and trying to do silly things (though how you could be sillier than some of the intended solutions I do not know!).

My copy was second hand, and I dont know of a source, but watch out for it!

NOTE: Infocom are no longer producing their adventures for the TI, and have run out of stocks of some titles. Contact me for more info.

=====

A CRASH COURSE IN PASCAL. by Donald M Monro. Published by EDWARD ARNOLD. 1985. ISBN 0 7131 3553 0. £7.50 or so. Paperback, 202 pages.

This PASCAL textbook is ideal for anyone trying out the Turbo Pasc 99 package from Texaments. With so many Pascal books now dealing with TurboPasc (the Borland language not the TI one), it is nice to find this book dealing exclusively with ISO PASCAL, the international minimum standard for Pascal.

Commercial versions of Pascal naturally differ from the ISO PASCAL, otherwise no-one would buy them! Typically, features are added, and some things are expanded or made easier. This is true of TI TurboPasc99, but you will find yourself able to use at least half the examples in this excellent book.

I found the text very readable, and with TurboPasc99 loaded and working, easy to sail through. Adjusting for the variations in TP99 was no problem- but I did miss not having TYPE available!

TP99 does not have packed arrays- it has strings instead. Although type is not supported you CAN pass arrays to procedures. In the absence of type, records and files are not available (as types), nor are pointers. Despite these departures from ISO Pascal, at least one half of this inexpensive book on ISO Pascal is directly of use to TP99 users, and it almost certainly represents the best value reference book you can buy to use with TP99.

USCD Pascal is an extension of ISO Pascal, and TI Pascal owners may also find this book of interest. As it is a UK publication, the price is very reasonable and its value is great!

Pascal was developed (in Switzerland) as a language for TEACHING (rather like Logo!) but was then taken up as a serious language in its own right. Because of its origin it is quite a good language to learn after Basic! Once you have an initial grounding in Pascal, you should not find C so hard! And a grounding in Pascal and C will take you very far these days! A formal (eg incomprehensible) description of ISO Pascal can be found by buying a copy of British Standard BS 6192:1982. The BRITISH standard is the definitive one for English versions of Pascal, and derived from it are ANSI/IEEE 770 X3.97-1983 and ISO 7185-1982.

=====

FONT WRITER 2 by J P Hoddie, published by Asgard and widely available from the USA for about US\$25.

Font Writer has already been reviewed. Version 2 goes a long way beyond that! In essence we have a replacement formatter program for TI Writer files, with additional "dot" commands to let us incorporate pictures into the text, and to change fonts. Speed is... SLOW!

Font Writer is not as powerful as The Printer Apprentice, but it is faster and easier to use- eg the products complement rather than compete.

Font Writer 2 can work with most TI ARTIST or CSGD fonts. Graphics can be either TI Artist Pictures (_P) or Instances (_I) or GRAPHX pictures.

Graphics may be inverted and/or framed. Pictures can have a left margin from 0 to 28, while instances can have their own margins. Instances and text can be on the same line- eg text can have ONE instance on the same line- not quite as flexible as TPA! Print density and width- affecting all output- are selected with dot commands. Fonts are also selected by dot commands- allowing each line of text to be a different font (but not mixed fonts on one line). Text can be inverted (eg white on black- an excellent way to get through your ribbon!).

Most Useful Utility: DISK DUMP. If you have a disk with a mixture of CSGD Fonts, TI Artist Fonts, TI Artist Instances and Pictures, Graphx Pictures- this utility will read the disk and prepare a file to run through the Formatter

which will print out all the disk contents! Fonts are printed with as many letters as will fit on a line! File names are given with each item. One problem here: using fanfold paper you may find pictures split by the form feed! Edit the text file to avoid this if you can.

Other Utilities: BANNER which allows you to create huge banners using your font collection- NB: ONLY uses CSGD fonts in /CH and /GR files. However... there is also a utility to transfer TI Artist fonts into CSGD fonts included! The banner may have up to 252 characters. (/GR images are not magnified at all).

Also included is a handy FONT MANAGER to merge fonts. Fonts are stored in memory so some very large fonts may not fit! The FONT MANAGER can convert between _F files and /CH files.

Also we have conversion routines for changing from /GR to _I files, from /DT files to _I files, from _I to /GR files, and from _I to /DT files (eg between TI Artist and CSGD).

There is a FILE MANAGER which is a sort of disk manager- it lets you catalog, delete files, rename files, change file protection, and copy DV80 files.

There is a FONT EDITOR which allows you create your own fonts or modify existing fonts- you may call up and amend individual characters if you are not happy with them! Then use Font Manager to merge the amended character back into the original font file. This is quite a handy font editor to use.

There is also a macro facility for use in the font editor, but it strikes me as being a bit fiddly. All in all, a useful package of programs here, well worth the asking price.

=====

THE PRINTER APPRENTICE TOOLBOX by McCann Software. Widely available in the USA at around US\$22.50

This "companion" to THE PRINTERS APPRENTICE is only JUST a companion, as it can do a great deal of work on its own. You only need the main package to prepare a SCHEDULE for the formatting of a mixed page of different text/ images. This package can, on its own, do the following- items beginning with an asterisk * are really for use with the main package:

*See on screen the page make up of a formatted page- as our screens are not dense enough for WYSIWYG, we have boxes in place of actual text/graphics. I found this item unhelpful and misleading and recommend it is avoided! Go by the numbers provided by the SIZE command and formatter schedules.

FORMAT text- includes a mini editor or can use DF80 files produced with TI Writer. Uses special TPA fonts! (see later!). You can now expand fonts without limit other than printer capability. And expanded fonts do not have to be solid blocks- you may design pixel shapes (nice). Letters can be distorted by magnifying width and height differently. Most useful.

CONVERT TI ARTIST and CSGD fonts to TPA SDSH fonts. Also you can bring in TI ARTIST instances or pictures for use with the TPA Scheduler. Instances and Pictures can be cropped to remove unwanted bits.

*Border Builder creates a border chosen from many designs (design your own!) to put around text or graphics, and the border is placed around the output using the Scheduler from the main package. The Toolbox can however just print out a frame for you.

Create GRAPH PAPER! by placing BOXES of small dimension close to each other! All in all a good package. Essential if you have TPA. If you buy the Toolbox, the main TPA will help you get the most out of it but is not essential.

=====

REVIEWS!!!

ART-CONVERT...Trio+ Software, about US\$10 from Tenex and other US suppliers.

This utility will transfer your TI Artist instances into a disk format which you can print out using TI WRITER Formatter- for instance, you can develop a fancy letter heading to start your letters off!

I was especially interested in this program as I wrote one very similar about a year ago- based upon programs in The Smart Programmer and Super 99 Monthly. My program, ART/FORM, is on Utility 7, in the disk library, and allows you to invert the image if you wish, to choose from three print densities, and to mark the left tab. ART-CONVERT lacks the image-inversion but adds the ability to use an 8-bit high TI Artist font instead of your usual printer font. Maybe I'm biased here! Suffice to say you can get UTIL 7 which has lots of other programs on it as well for a lot less than the cost of ART-CONVERT!

NB: When printing graphics with the Formatter, the printer name must end with .CR, and if you USUALLY use a printer name ending with .LF, then instead you must add a line feed at the end of every line of text you want printed (you

can do this automatically by formatting the text to disk instead of printer as a first pass).

=====

SINGS DATA 1 from TRIO+ Software, includes SINGS-PLAY. US\$10 from TRIO+ and other US Sources.

This program follows on from the FREWARE program TI SINGS, available from the disk library- DATA 1 has NOT been released as freeware! SINGS DATA 1 contains a number of "rock songs" for your TI plus Speech Synth plus TE2 module to sing.

The style is of course a trifle odd, but how many other computers do you know that can sing I HEARD IT THROUGH THE GRAPEVINE or THE HOUSE OF THE RISING SUN! The program SINGS-PLAY enables you to select which songs on the disk you wish to hear, and to select the order of play. Having done this, the songs will play in sequence, rather like a juke-box! SINGS-PLAY will work with any song file produced using the TI SINGS creation program.

If you have TE2 and speech synth, the cost is low. Why not try it!

=====

LEGENDS. by Donn Granros and Ed Johnson. Released through Asgard Software. \$25 from Tenex.

How to describe this program? Hmm. You need a long time... fortunately there is a save option! Most important of all: NEVER use your master disk as a play disk. ALWAYS use backups! The program amends the disk as it goes! Take a party of adventurers out to explore this stricken planet, and if someone is killed, go back to recruit more! This is a graphic adventure on a large scale, with monsters and dungeons.

This is NOT an Infocom type adventure. There is an element of strategy involved, but a great deal of luck. Perseverance and patience are needed. By fighting many monsters you build up your experience and wealth to the point where you can take on the really evil ones! The closest similar program I can think of is SECRET OF KZIRGLA by Kuhl Software, now revamped and improved as THE DUNGEONS OF REMZAK. But LEGENDS is one heck of a program- immense! With an awful lot to explore and do and a great deal of fighting too. If you have a lot of spare hours, and would enjoy lots of random adventuring, you may like this one. I dont think sheer size is a guide to quality, and the advertising is WELL over the top- I repeat, you cannot equate this program with an Infocom adventure (and be sane and honest that is!). It is quite different. If I find myself with many empty hours I may play it...

The price (for two disks) is not bad considering the many many hours that must have gone into preparing this monster, but I wonder about the ability of the program to hold and maintain your interest...

=====

UpDate: SPAD XIII by Not Polyoptics, available widely in the USA:

Version 2 adds a RED BARON option- this is a more intelligent enemy. PLUS the ability to quickly zoom to choice locations- quickly teleport to the German airfield, bomb the hangar, and teleport back home before you get hit by flack! (Cheating isn't it!). You can now see your plane from outside- viewed from the South and slightly above. And at game end you have an option to restart instead of the program just terminating. Plenty of improvements. It is an interesting experience to taxi up to a tree and stop with the tree growing out of your plane!

=====

OLDIE: An old XB game has been mentioned in print several times recently. History first... DIABLO, by Manuel Constantinidis, was released in TI Extended Basic by the US firm of EXTENDED SOFTWARE in 1983, when it was advertised for US\$19.95, and stated to be "1st place winner of the TISHUG Australia Contest". Yes, the author was a member of the Sydney user group.

DIABLO involved the movement of tiles containing various shapes of track, in order to keep a moving ball continuously supplied with track. As it moved over a tile, the track it moved over disappeared, so you were left with less and less track! The object was to clear the screen, so that the ball had moved over every piece. Diabolic! It was sold in the UK by Timeless Software for just £8.95.

DIABLO is NOW on sale again..."Diamond Games, a German software company, marketed in the UK under the Robtek banner. By Manuel Constantinidis, updated to live within the half-a-megabyte machines, with a Commodore version to follow". DIABLO for the Amiga or ST is advertised at £19.95 and £14.95 respectively. The C64 version will be £7.95 on tape or £9.95 on disk. These new versions appear to be exactly like the TI original, apart from a slightly different tile layout, and the ability to pause/save/load.

When a program written FOR the TI99/4A by a user group member makes it onto the Amiga, it is a time for celebration- and pause to consider just how good our old programs really are!

=====

HIGH GRAVITY:

Way way back... December 1982... I enjoyed a game on a Hewlett Packard computer in which you aim a rocket in a chosen direction, with the intention of it reaching a certain point. Its course is affected by the presence of various planets, which pull it off course by gravity...

Well, I can now play this game on my TI! HIGH GRAVITY is a good disk full from Asgard Software (recommend you buy from Tenex or some other reliable source!). You may select how many planets you want on screen, and then solve the "random" screens thrown at you. OR you can set up your own screens- and save them for future solution! The program comes with many such screen files! You may experiment with gravity- and as the rocket leaves a trail behind it, perhaps even use gravity to produce an interesting art form!

After waiting so long I can only recommend the program to you. It is more of a program to experiment with than a game, and you may not play it often after the first week, but it really is an excellent program!

=====

HIDDEN POWERS OF DISK FIXER (book) \$8.95 from Tenex.

Written by Bill Gronos, and intended as the manual for the Navarone Module "Disk Fixer", this book is also available separately, and is of value for use with any disk sector access program (there are several in the User Library).

This book has some interesting tricks to perform with sector access- like making a Basic program unlistable and unresequenceable. How to remove or deal with various types of protection (at sector access level- does not deal with initialisation protection devices). And most important of all, how to recover information from a disk which suddenly becomes uncooperative, just before you make that back up copy you've been meaning to do for the last five years...

A mere 52 pages, with six forming the Disk Fixer Manual, this little book has some interesting information.

It is not however an official manual- the information it contains is gleaned- and may not be totally correct or complete. In particular, the method for putting program lines into line number order is unduly tedious- instead of editing every single line in the program, you can use your console to do the work for you- save in merge format and then type new and merge back in. All program lines now sit in memory in line number order.

Interesting note- although I have seen the book advertised for many years, it contains a copyright date of 1987. It may be that only the format is copyright- placing a copyright date which comes after original publication implies the text is not so protected.

A useful little book to have if you ever need it - and one day you just may!

=====

Stephen

FUNLWEB:

Version 4.0, dated 22.12.88, is now in your Disk Library, and appears trouble free... although I continue to have difficulty with CONFIGURE this is no doubt due to my lack of understanding. I CAN get it to work in two passes!

The following text from Tony goes into a little more detail on FUNLWEB 4.0:

ENTOMOLOGY CORNER #14

~~~~~

Tony McGovern -- Funnelweb Farm

~~~~~

Here we are back writing again after a break of several months of hard programming labor. It has been the Southern Hemisphere winter for the period, not that many readers outside Australia would consider a Newcastle, NSW winter to be a real winter. I have lived through Oregon and Colorado winters so I do know the difference. Still it leaves a lot more time for programming than summer, the funnelwebs are quiescent, and mail from the rest of the world has been very slow. I hope this is just a sign of summertime elsewhere and not of the death throes of the 99/4a.

Can't be that as it is still too good a machine to give up on easily. No sign of the Network Nasties and their backers (see ENTCOR #13) since then either. Hopefully they are now doing something constructive for a change, though I can't say I've been left with a very high opinion of either them or those who went along with them, and sad to see some people intimidated by the virulence of their nonsense.

The main news is that FUNNELWEB Vn 4.0 is now out to old friends as a pre-release test version, and will be generally available by the time you read this. What I propose to do in this chapter is to give some further explanation and background discussion over and above the FWDOC/s which are of necessity refined to eliminate any excess verbiage. So until I run out of space I'll just keep talking about the how and why of FWB 4.0, and continue next time if it overflows a reasonable length.

The pre-release test version had a few file name changes from Vn 3.4 (no 3.5 was ever issued from here), but these have been rescinded after feedback on a DSR EPROM for the Horizon RAMdisk, apparently with some of Vn 3.4's filenames in mind. Now the Horizon and FWB are a perfect match, more so in Vn 4.0, but I don't have one for our Horizons, nor have the producers made any attempt even to contact us. There were good reasons for the changes, but seeing as some FWB users already have the EPROM the names have been restored.

A real change is that UPATCH and ULINSTL have disappeared, and LOAD now contains a warning that it is NOT to be edited. All of the setup procedure is now handled by the single CONFIG program. CONFIG started out as a way of combining the functions of ULINSTL and UPATCH. Then towards the end of the development we were able to get Myarc XBII going after a fashion. The only communication we have ever been able to get out of Lou Phillips was the return of a disk with the Vn 2.12 files which sat around for months because we had no way of running them.

Then Pete Smith upgraded his XBII to Vn 2.12 and passed on his old Vn 2.11 ROM for the Myarc RAMdisk. Not sure what problems the mismatch causes, but we have XBII up and running sort of. The point addressable graphics is what TI should have provided all along and the lack was always the worst failing of XB. Mostly, but not always it is faster, but curiously enough is not fast enough to keep up with the sprite routines in the TXB game program, which work even in console XB only.

The real nuisance factor with the Myarc 128K-OS is that it always writes its boot header and code into cartridge RAM on power-up, even if the 128K-OS is NOT enabled. This destroys anything else you want to keep in your Cache Card, DEBUG for instance or FWB even. This is an atrocious design blunder and users can only hope that they fix it at minimal cost.

So what has all that to do with CONFIG ? The relevance is that once the XB entry code had been adjusted to sense the presence of XBII (same simple trick as used by LDFW - get a character from the screen area in VDP and see whether it has a >60 XBasic screen offset) then it was found that reSAVEing LOAD to disk under Myarc XBII would destroy its integrity. First inclination was to leave it as it was with a warning that it should be edited and reSAVEd only with TI XB. This wasn't really thought good enough by the funnelwebs, or even by the possums that live in the roof, so the last major task in developing Vn 4.0 was to eliminate the need for XB editing of LOAD.

This was handled by upgrading CONFIG so that it also configures the LOAD program as well. There are three aspects to customizing. The first of these is common to both LOAD and UTIL1 and comprizes all the boot specs, color choices, file/device names and numbers. The second is the UL (Option #7) User List function, either to be saved separately or incorporated in UTIL1. The third, relevant to LOAD only is the User List function from the XB level. In previous versions this was handled by the XB editor with program statements. CONFIG writes to the LOAD file but treats it as a generic program file for purposes of fetching it from and saving it back to disk.

All items that can be entered directly are no longer defined in XB statements, but since ANY user option could lead to XB RUN of a program it was necessary to keep all these loading paths as XB statement. So CONFIG actually rewrites the XB statements directly into the tokenized program.

There is no way CONFIG is going to duplicate the complete function of the XB line editor, and it is only just smart enough to do the job, provided the XB program has not been mucked around. Hence the warning about not editing and reSAVEing directly with XB. With XBII it is fatal anyway. The real limitation on direct editing is that all XB lines from #160 up must be left strictly alone and none added after #160.

The sequence of the configuration process establishes the common items first before the LOAD program itself is even loaded. If you skip directly to the UL screen the settings that you are using in LOAD or UTIL1 as used to load CONFIG will be imprinted on LOAD when it is saved back to disk, so if you use several different loading paths it may be a good idea to go through the full procedure each time.

LOAD may be saved back to any drive, as may UTIL1 and UL. I have more ideas on restructuring LOAD/CONFIG but sheer exhaustion has won the day now. Perhaps in some future version ! Will use this idea of treating a FWB type of XB LOAD with embedded assembly code as a generic program file in his DISK-HACKER program.

Also you will find that the exit from DM-1000 now looks for LOAD if it can't find RELOAD/UTIL1 on the system drives. This doesn't restore XB conditions but merely loads LOAD as a generic program file and extracts the FWB code.

So what difference does all this make ? The prime one when you are setting up is that all configuration steps are now prompted or menu driven, even for XB. You don't even have to have XB or XBII in the machine to configure LOAD. No more figuring XB color groups or which XB statement goes with which. When LOAD is RUN initial operation is now much faster, mostly because it is shorter, and it may be re-entered by RUN after return to the XB command level, either by explicit option or because a RUN "DSKx.yy" has failed. This allows recovery from such errors without complete reload of LOAD.

It possible to add to LOAD so that a keypress bypasses boot tracking and installs a fixed drive number, for a deviant device such as the Myarc RAMdisk which might be reset to emulate different drive numbers. Why, I'm not quite sure but here's how to handle it on the fly. This is all despite the warning message against direct editing. Add line #110 to LOAD

```
110 CALL INIT :: CALL KEY(5,
K,ST) :: IF ST=1 THEN CALL L
OAD(-232,K,K,0,0)
```

and remove the CALL INIT from #130. With this addition a key held down as the program starts to execute will be installed as the fixed drive number. See FWDOC/REPT for background information.

In everyday use of FWB the most popular feature will very likely be the newly updated SD in the Editor. The beginnings of this were apparent in late editions of Vn 3.4.

It is now easy while in the Editor to Mark a file in SD for Viewing on the screen while leaving the document in memory unaffected. Very handy for quick checking of the contents of another file without going through the whole save/load/reload rigmarole.

The existing workfile name can be restored by <O>ldFile before exit without explicit reselection. The Print Directory and Program Check functions are unchanged.

Deleting of markable files is also supported, with a cautionary pause, so that memory work is not required as with DF. Files which cannot be marked for loading by the Editor can't be deleted directly this way from SD, but then SD really has no business deleting files that the Editor can't read. The Editor enhancements from late in the Vn 3.4 development cycle are retained, notably right margin beep, crisper screen scrolling, and reduced tendency to lose keystrokes on end of line wordwrap. It is maybe worthy of note that this extra functionality of the Editor and SD does not require any extra hardware over and above a minimum 99/4a system. The same is true of FWB as a whole.

One thing I have found from the pre-release is that there is a pent-up demand out there for a ruler line. I had altered the End of File message to a more elegant (to my eye anyway) form with a discreet EoF and a column number marking across the screen. This immediately set off complaints that it wasn't a full 80 column ruler.

As I'm not under the delusions common to some commercial program writers that their program as handed down is perfect and immutable, I had another look at it and figured how to turn the EoF message line into a 80 column ruler that shifts with horizontal window and line # on/off. While at it I decided to have a go at providing a facility that I have long wanted - upper/lower case conversion without explicit retyping. The major upheavals for EoF on the new ruler line had left a contiguous block of 17 words for the necessary code.

The best I could do for a both way conversion was 19 words of code, but Will saved the day with an improved version that fitted in exactly.

Two of the four unused control keys are used, <ctrl-; > for converting to upper case, and <ctrl-. > just below it for conversion to lower case. If the letter under the cursor is not converted, these just act like a plain right arrow key.

Each of the Central Menu screens has been extended to seven entries. CONFIG allows you to set your own preferred name on the sixth and a two character associated filename. Each of these is set up as an Option 3 program file load

for maximum flexibility. The Option 3 loader itself has been modified so that it will handle a wider range of files.

It still hands over in the E/A USRWSP but now sets this up before loading any files. Previously it used registers and loaded R11 before final handover. So now it is more tolerant of files that load into low memory. It just goes to show how even the most worked over code still can be improved with a fresh look. Not only was the loader improved but some bytes were saved in the process.

Many people do not seem to have appreciated the relative significance of the Option 2 and 3 program file loaders. There is some difficulty with terminology here, but the practical differences are real and significant. Option 2 imitates the RUN PROGRAM FILE function of the E/A module. Why then call it "GPL" ? It's a programmer's view really. The reason is that the E/A function is handled purely by GPL code in the E/A Grom and the E/A assembly utilities are totally ignored.

Also similar conditions are appropriate for other file loads from Grom and for cartridge simulations. Option 3 is very much closer to being a program (memory image) mirror of LOAD and RUN in a way that the E/A module just does not even provide. The E/A utilities are loaded, the E/A USRWSP is set, and FWB makes sure that differences in departure points from GPL (XMLs in LOAD as against LINK) don't matter. The latest modifications just smooth over some of the difficulties that an Option 3 load might cause for program files that really should be Option 2 loads.

If the overhead of loading EA is bothersome and unnecessary in your particular application then the jump table may be altered to give the preferred type of program load. As a practical matter if you are using the E/A module or running with your FWB system files in RAMdisk, then loading of the utilities is so fast that you won't notice it anyway.

The easiest place to find the byte jump table is in UTIL1 where the relevant bytes are presently located at >FFE3 and >FFEA. These are near the end of the second last sector and UTIL1 starts at >E006 so you don't have to fuss about 6 byte offsets when counting sectors and bytes. As issued these are both "I" in ASCII form.

Option 2 loads are "F" in ASCII form and Option 1 are "B". You will have to do it to LOAD also, but you will find the practice on UTIL1 to have been worthwhile. We usually run from a HORIZON RAMdisk and it all zips out so fast that I have to go back to running from physical disk every so often just to see where the program is really getting files from. As always please leave your master disk in its original state for passing on.

Talking of Horizon RAMdisks brings up the first page items in CONFIG. If boot disk tracking is not enabled then you are asked to specify the system drive. Vn 4.0 allows the FWB system files to be spread over two drives. A full complement of system and user list utility files usually overwhelms the capacity of a DSSD disk.

One drive may be specified for system utilities on the TI-Writer side of the Central Menu screen, and a different drive for the system files on the E/A side. The system file loaders look on the specified drive for that side first, but if the file is not found there they check the other drive too before issuing an error.

The only system files not so treated are the CHR1/2 character sets. The Editor files ED/EE are common to both sides. If both primary and secondary disks are RAMdisks then the difference in speed of loading when only one copy of ED/EE is present is hardly noticeable. This feature may also be useful to users of TI original type SS disk systems or TI controllers with DS drives.

A little discovery in assembly coding was that the well known trick for incrementing a register by 4 in a one word instruction, C *R1+,*R1+ can get you into trouble - if R1 happens to point to a memory mapped address as it ended up doing in one path through a first attempt at some tricky code to make the right margin beep.

The next obvious addition in Vn 4.0 is the Low-Loader function. This fills a gap in the facilities that TI made available for the 99/4a. The major utility programs that TI provided, the E/A Editor and Assembler and the TI-Writer Editor (but not the Formatter) have most or all of their functional code in the low-mem BK block, and use hi-mem as a large one piece data buffer. It was easy enough for TI's programmers to prepare these on their minicomputer based development systems, but the nearest the 99/4a came to it was the Minimem module, which provided LOAD and RUN functions only. Helping MM along with more

functions in the cartridge RAM was probably an early educational step for many serious 99/4a programmers.

The decision was made early to duplicate the function of the E/A loader, including the loading speed. This meant that an E/A DSRLNK had to be provided

and by the time that was done there wasn't much more space needed to provide the complete E/A utility set even though most programs will not use them. Low-Load still uses the FWB object loader, but has to alter and restore a bunch of memory values.

Other programs have been adapted to work with Low-Loader as well. Script-Load now recognizes a global LWLD directive which causes it to use the LL utilities rather than the E/A utilities. That wasn't easy. Also the FMSAVE utility automatically recognizes Low-Load and SAVES starting with low memory first in that case. I regularly use this to patch and reSAVE the Editor files. While on loaders, there is now only one off-screen loader entry (#8), and this one cancels autostarting of object files. FWSAVE is there mainly to provide the LSAVE function as used to prepare CONFIG, and SAVE to CS1. There just wasn't enough room to fit these in FMSAVE.

The modified DM1000 files are clearly different from before even if this is not obvious until the exit screens. An invisible change is that the entry code has been altered so that if it is not loaded from FWB it works more or less normally, so that separate versions aren't required. I gather that the new UTIL1 doesn't fit as well in GRAM simulating devices. I don't see quite what the problem is as UTIL1 is still under 8K long, but that's what I'm told. I've never even seen one of these devices, and don't know the details involved, and it just may be the system software for the particular device in question.

This is enough to be going on with so I'll sign off for now and see if I can find some time for playing with high level languages.

NOTE BY STEPHEN:

To return to MYARC XB from FUNLWEB, do not use the RETURN TO XB option as that only works with TI XB. Instead go to DM1000 and choose the CART RAM option- this will cause MYARC XB to be reloaded from disk (do keep the Myarc disk in place for this!) as the Myarc module retains the loading routine all the time.



TIPS FROM THE TIGERCUB

#45

Copyright 1987

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just \$1.00 each, plus \$1.50 per order for cassette or disk and P&H. Cassette programs will not be available after my present stock of blanks is exhausted.

Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus! **MIN ORDER \$10**
TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to \$15.00 postpaid.

NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$15 postpaid.

NUTS & BOLTS #3 is now ready, another full disk of 140 new merge-format utility subprograms, all compatible with the previous. With 11 pages of documentation, \$15 ppd. *****

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to \$10 ppd.

TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$10

TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25 through 32, \$10 postpaid.

TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also \$10 postpaid.

Here is a versatile printer utility which will accept all printer control codes, print in 1 to 5 columns with choice of column separation and margin width, allow alternate margins and pause at end of page to turn paper over, and will load and print a diskfull of

files one after another. It is set up for the Gemini 10X and may require modification for other printers.

```

100 DIM M$(400),F$(50)
110 GOTO 150
120 K,ST,SET,S,P$,P,CL,DW$,S
S$,I$,D$,E$,NC,CW,TC,TA,TX,A
V,CS,S$,LT,A$,LSP,LP,RM,OK$,
QB$,X,F$(1),SL,F,IP,M$(1),T$,F
LAG,J,PP,LT$
130 CALL CLEAR :: CALL KEY :
: CALL COLOR :: CALL SCREEN
:: CALL SOUND
140 !eP-
150 CALL CLEAR :: CALL KEY(3
,K,ST):: ON WARNING NEXT
160 FOR SET=0 TO 14 :: CALL
COLOR(SET,2,8):: NEXT SET ::
CALL SCREEN(5)
170 DISPLAY AT(3,6):"TIGERCU
B PRINTALL": :TAB(7);"Copyri
ght 1987":TAB(6);"Tigercub S
oftware" !programmed by Jim
Peterson
180 DISPLAY AT(12,1):"May be
distributed without":"restr
iction providing that":"no p
rice or copying fee is":"cha
rged."
190 DISPLAY AT(18,7):"TURN P
RINTER ON!"
200 DISPLAY AT(20,8):"PRESS
ANY KEY" :: DISPLAY AT(20,8)
:"press any key" :: CALL KEY
(0,K,S):: IF S=0 THEN 200 EL
SE CALL CLEAR
210 DISPLAY AT(12,1):"PRINTE
R DESIGNATION?" :: ACCEPT AT
(14,1)BEEP:P$ :: IF POS(P$,"
.LF",1)=0 THEN P$=P$&".LF"
220 ON ERROR 230 :: OPEN #1:
P$,VARIABLE 255 :: ON ERROR
STOP :: PRINT #1:CHR$(27);"e
" :: CALL CLEAR :: GOTO 240
230 DISPLAY AT(20,1):"CANNOT
OPEN PRINTER!" :: RETURN 21
0
240 DISPLAY AT(12,1):"PRINT
SIZE?": : (1) PICA": (2)
ELITE": (3) CONDENSED"
250 ACCEPT AT(12,13)VALIDATE
("123")SIZE(1):P :: PRINT #1
:CHR$(27);"B";CHR$(P);
260 !The values 80, 96 and 1
36 in the next line are the
maximum number of pica, elit
e and condensed characters p
er line on Gemini 10X

```

```

270 !Change as necessary for
your printer!
280 CL=(P=1)*80+(P=2)*96+(P=
3)*136 :: CL=ABS(CL)
290 DISPLAY AT(12,1)ERASE AL
L:"DOUBLE-WIDTH? (Y/N) N" ::
ACCEPT AT(12,21)SIZE(-1)VAL
IDATE("YM")BEEP:DW$ :: IF DW
$="Y" THEN PRINT #1:CHR$(27)
;"N";CHR$(1);:: CL=CL/2
300 DISPLAY AT(12,1)ERASE AL
L:"SUPERSCRIPT? (Y/N) N" ::
ACCEPT AT(12,20)SIZE(-1)VALI
DATE("YM")BEEP:SS$ :: IF SS$
="Y" THEN PRINT #1:CHR$(27);
;"S";CHR$(0);
310 DISPLAY AT(12,1)ERASE AL
L:"ITALICS? (Y/N) N" :: ACCE
PT AT(12,16)VALIDATE("YM")SI
ZE(-1)BEEP:I$ :: IF I$="Y" T
HEN PRINT #1:CHR$(27);"4";
320 DISPLAY AT(12,1)ERASE AL
L:"DOUBLE-STRIKE? (Y/N) Y" :
ACCEPT AT(12,22)VALIDATE("
YM")SIZE(-1)BEEP:D$ :: IF D$
="Y" THEN PRINT #1:CHR$(27);
;"6";
330 IF P<>3 AND P<>4 THEN DI
SPLAY AT(12,1):"EMPHASIZED?
(Y/N) Y" :: ACCEPT AT(12,19)
VALIDATE("YM")SIZE(-1)BEEP:E
$ :: IF E$="Y" THEN PRINT #1
:CHR$(27);"E";
340 DISPLAY AT(12,1)ERASE AL
L:"NUMBER OF COLUMNS? (1-5)"
:: ACCEPT AT(12,26)VALIDATE
("12345")SIZE(1)BEEP:NC
350 DISPLAY AT(12,1):"COLUMN
WIDTH (NUMBER OF": : "CHARAC
TERS?" :: ACCEPT AT(14,13)VA
LIDATE(DIGIT)BEEP:CW
360 TC=NC&CW :: TA=CL-TC ::
TX=TC+NC&2-2
370 IF TX<=CL THEN 390 :: DI
SPLAY AT(18,1):STR$(NC)&" co
lumn of "&STR$(CW)&" charac
ters": "plus 2-column spacing
equals"
380 DISPLAY AT(20,1):STR$(TC
)&" characters; maximum": "av
ailable in print size": "sele
cted is "&STR$(CL)&": "!!!!
Please reselect!!!!" :: GOTO
240
390 IF NC=1 THEN 410 :: AV=I
NT(TA/(NC-1)): DISPLAY AT(1
2,1)ERASE ALL:"COLUMN SEPARA
TION?": "MINIMUM 2": "MAXIMUM
"&STR$(AV)&" AVAILABLE ": "2"
400 ACCEPT AT(15,1)VALIDATE(

```

```
DIGIT)SIZE(-2)BEEP:CS :: IF
CS<2 OR CS>AV THEN 400 ELSE
S$=RPT$( " ",CS)
410 TA=TA-CS*(NC-1):: IF TA
2 THEN 450
420 DISPLAY AT(12,1)ERASE AL
L:"LEFT MARGIN WIDTH?": "MA
XIMUM "&STR$(TA)&" AVAILABLE
" :: ACCEPT AT(12,20)VALIDAT
E(DIGIT)BEEP:LT :: IF LT>TA
THEN 420
430 DISPLAY AT(12,1):"ALTERN
ATING LEFT/RIGHT": "MARGIN?
(for pages to be": "later re
produced on both": "sides) (Y
/N) N"
440 ACCEPT AT(1,14)VALIDATE
("YN")SIZE(-1):A$
450 LSP=12 :: DISPLAY AT(10,
1):" " " " "LINES PER PAGE?
60": " " " " " " " :: ACCEP
T AT(12,17)VALIDATE(DIGIT)SI
ZE(-3):LP :: IF LP<70 THEN 4
90
460 DISPLAY AT(12,1):"LINE S
PACING - 72 INCH" :: DISPLAY
AT(11,16):" " " " :: ACCEPT AT
(10,16)VALIDATE(DIGIT)BEEP:L
SP
470 IF LP/(INT(72/LSP))>11.5
THEN DISPLAY AT(20,1):"WON'
T FLT!" :: GOTO 450
480 PRINT #1:CHR$(27);"A";CH
R$(LSP);
490 RM=TA-LT
500 DISPLAY AT(12,1)ERASE AL
L:STR$(NC)&" columns of":STR
$(CW)&"-character width": "le
ft margin of "&STR$(LT)&" sp
aces"
510 DISPLAY AT(15,1):STR$(LP
)&" lines per page": "with "&
STR$(LSP)&" /72 line spacing"
520 DISPLAY AT(17,1):STR$(CS
)&" spaces between columns":
"right margin of "&STR$(RM)&"
spaces": "OK? (Y/N) Y"
530 ACCEPT AT(20,11)VALIDATE
("YN")SIZE(-1)BEEP:OK$ :: IF
OK$="N" THEN 240
540 DISPLAY AT(12,1)ERASE AL
L:"PAUSE AT END OF PAGE? N"
:: ACCEPT AT(12,23)VALIDATE(
"YN")SIZE(-1):QB$
550 DISPLAY AT(1,1)ERASE ALL
:"INPUT FILENAMES TO BE": "PR
INTED.": "PRESS ENTER WHEN DD
NE"
```

```
560 I=X+1 :: DISPLAY AT(X+3,
1):"FILENAME? DSK" :: ACCEPT
AT(X+3,14)SIZE(-12)BEEP:F$(
X)
570 IF F$(X)="" THEN X=X-1 :
: GOTO 600 ELSE F$(X)="DSK"&
F$(X)
580 ON ERROR 590 :: OPEN #2:
F$(X):: CLOSE #2 :: GOTO 560
590 ON ERROR STOP :: CALL SO
UND(1000,110,0,-4,0):: DISPL
AY AT(20,1):"CANNOT OPEN "&F
$(X):: X=X-1 :: RETURN 560
600 SL=1
610 F=F+1 :: IF F>X THEN 700
:: ON ERROR 620 :: OPEN #2:
F$(F),INPUT :: DISPLAY AT(22
,1):"READING ";F$(F):: ON ER
ROR STOP :: GOTO 630
620 CALL SOUND(1000,110,0,-4
,0):: DISPLAY AT(20,1):"COUL
D NOT OPEN "&F$(F):: STOP
630 FOR IP=SL TO LP*NC :: LI
NPUT #2:M$(IP):: IF LEN(M$(I
P))=0 THEN 670 :: IF NC>1 AN
D POS(M$(IP),CHR$(13),1)<>0
THEN M$(IP)=SEG$(M$(IP),1,LE
N(M$(IP))-1)
640 IF ASC(M$(IP))>126 OR AS
C(M$(IP))<32 THEN IP=IP-1 ::
GOTO 680
650 IF LEN(M$(IP))<=CW THEN
670 :: T$=SEG$(M$(IP),1,CW):
: CALL SOUND(1000,110,0,-4,0
):: DISPLAY AT(12,1):M$(IP);
" OVER";CW;"CHARACTERS": "TRU
NCATED TO ";T$;"OK?"
660 CALL KEY(3,K,S):: IF S=0
THEN 660 ELSE IF K<>89 THEN
STOP ELSE M$(IP)=T$
670 M$(IP)=M$(IP)&RPT$( " ",C
W-LEN(M$(IP)))
680 IF EOF(2)=1 THEN CLOSE #
2 :: SL=IP+1 :: GOTO 610
690 NEXT IP :: IF EOF(2)=1 T
HEN CLOSE #2 :: GOTO 720 ELS
E GOTO 720
700 ON ERROR 710 :: FLAG=1 :
: FOR J=IP+1 TO NC:LP :: M$(
J)="" :: NEXT J :: GOTO 720
710 STOP
720 PP=PP+1 :: IF PP/2=INT(P
P/2)AND A$="Y" THEN LT$=RPT$(
" ",RM)ELSE LT$=RPT$( " ",LT
)
730 FOR J=1 TO LP :: ON NC 6
OSUB 750,760,770,780,790 ::
NEXT J :: PRINT #1:CHR$(12):
```

```
: SL=1 :: IF F>X THEN STOP E
LSE IF QB$="N" THEN 630
740 DISPLAY AT(24,1)BEEP:"PR
ESS ANY KEY TO CONTINUE" ::
CALL KEY(0,K,S):: IF S=0 THE
N 740 ELSE DISPLAY AT(24,1):
" " :: GOTO 630
750 PRINT #1:LT$&M$(J)&CHR$(
10):: RETURN
760 PRINT #1:LT$&M$(J)&S$&M$(
J+LP)&CHR$(10):: RETURN
770 PRINT #1:LT$&M$(J)&S$&M$(
J+LP)&S$&M$(J+LP*2)&CHR$(10
):: RETURN
780 PRINT #1:LT$&M$(J)&S$&M$(
J+LP)&S$&M$(J+LP*2)&S$&M$(J
+LP*3)&CHR$(10):: RETURN
790 PRINT #1:LT$&M$(J)&S$&M$(
J+LP)&S$&M$(J+LP*2)&S$&M$(J
+LP*3)&S$&M$(J+LP*4)&CHR$(10
):: RETURN
```

This is an improved version of the math program in Tips #36.

```
100 CALL CLEAR :: RANDOMIZE
110 B=INT(5*RND+2):: IF B=B2
THEN 110 ELSE B2=B
120 F=INT(5*RND+2):: IF F=F2
THEN 120 ELSE F2=F
130 D=INT(5*RND+2):: IF D=D2
THEN 130 ELSE D2=D
140 X=F+BB*D
150 BB=INT(5*RND+2):: IF BB=
BB2 OR BB=B THEN 150 ELSE BB
2=BB
160 DD=INT(5*RND+2):: IF DD=
DD2 OR DD=D THEN 160 ELSE DD
2=DD
170 F=F+BB*DD
180 DISPLAY AT(3,1)ERASE ALL
:"IF";B;"BOYS CAN CATCH";X;"
FROGS IN";D;"DAYS,"
190 DISPLAY AT(6,1):"HOW MAN
Y FROGS CAN";BB;"BOYS": "CATC
H IN";DD;"DAYS?"
210 ACCEPT AT(7,19):Q
220 IF Q=F THEN DISPLAY AT(9
,1):"THAT'S RIGHT!" :: GOTO
110
230 DISPLAY AT(9,1):"NO, THA
T'S WRONG."
240 DISPLAY AT(11,1):"IF";B;"
BOYS CAN CATCH";X;"FROGS IN
";D;"DAYS"
250 DISPLAY AT(13,1):"THEN O
NE BOY CAN CATCH";X/B;"FROGS
```

```
IN";D;"DAYS"
260 DISPLAY AT(15,1):"AND ON
E BOY CAN CATCH";X/B/D;"FROG
S IN ONE DAY."
270 DISPLAY AT(17,1):"SO, IF
ONE BOY CAN CATCH";X/B/D;"F
ROGS IN ONE DAY,"
280 DISPLAY AT(19,1):"THEN";
BB;"BOYS CAN CATCH";X/B/D*BB
;"FROGS IN ONE DAY"
290 DISPLAY AT(21,1):"AND";B
B;"BOYS CAN CATCH";X/B/D*BB*
DD;"FROGS IN";DD;"DAYS."
300 DISPLAY AT(24,1):"PRESS
ANY KEY" :: CALL KEY(0,K,S):
: IF S=0 THEN 300 ELSE 110
```

Here's an idea for an unusual title screen -

```
100 CALL CLEAR :: FOR SET=1
TO 8 :: CALL COLOR(SET,1,1):
: NEXT SET :: CALL CHAR(100,
"0",101,"0")
110 X$(0)="4043241818244202"
:: X$(1)="4021261818648402"
:: X$(2)="2020131C38C80404"
:: X$(3)="1010101FF8080808"
:: X$(4)="081010907E111020"
120 X$(5)="080808FB1F101010"
:: X$(6)="0404C8381C132020"
:: X$(7)="0284641818262140"
130 A$=RPT$(CHR$(100)&CHR$(1
01),13):: FOR R=1 TO 24 :: C
=C+1+(C=2)*2 :: DISPLAY AT(R
,C):A$ :: NEXT R
140 CALL VCHAR(1,29,1,168)
150 CALL SCREEN(2):: CALL CO
LOR(9,5,16):: FOR S=1 TO 8 :
: CALL COLOR(S,16,2):: NEXT
S
160 DISPLAY AT(5,5):" TIGERC
UB SOFTWARE ";; DISPLAY AT(
8,6):" SQUIRMY SCREEN ";
170 FOR J=0 TO 7 :: CALL CHA
R(100,X$(J)):: CALL CHAR(101
,X$(7-J)):: NEXT J
180 CALL KEY(0,K,S):: IF S=0
THEN 170
```

MEMORY FULL

Jim Peterson



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

[A montage created from several articles, selected by Stephen Shaw]

The computer is no better than its program.

--Elting Elmore Morison, 1966

DID YOU KNOW?

When Disk Manager 2 formats a disk it verifies each sector (you knew that). What was new to me was that if it finds a bad sector, DM2 locks it off. This allows you to use the disk, with slightly less storage space. (Source: ^ a letter in MICROpendium - personally verified)

[DM1000 can do the same if you answer Y to the final question before it initialises a disk. ss.] [DSKU can verify the sectors on a disk with files on it, without wiping the files out, giving you the option to map out any bad sectors. DM1000 and DSKU are available from the club disk library, as indeed is DM2. ss]

THE BEST FREEMWARE

COMPUTE did a survey on Compuserve of the best free programs. The five TI winners were:

FAST-TERM (terminal emulator)
 DISK MANAGER 1000
 FUNL-WRITER
 NEATLIST
 MASS-COPY

All of these are available from our UG library.

BOOK REVIEW

The Orphan Chronicles by Ronald G. Albright, Jr. M.D., 172 pages, published by Millers Graphics

I planned to buy this book but never got around to it. Then I won a copy. I started reading it that night and finished it two days later. I just couldn't put it down.

Dr. Albright chronicles the TI Home Computer from the 99/4 (no A) thru TI's self-destructive marketing techniques to Black Friday (October 28, 1983 - the day TI announced that they were dropping the 99/4A) and beyond. Along the way he covers the International Users Group (a "commercial" user group - an oxymoron unique to the 4A world) and publications including Home Computer Magazine (and tells us who Regina really is).

Fully half of the book is devoted to the current status of our orphan. There are chapters on telecommunications, freeware, the future and on current 4A supporters.

An interesting chapter is the survival guide. In ten steps, Dr. Albright suggests practical strategies that we can take to help ensure our computers continued existence.

This informative book is clearly written. The material is comprehensive and logically organized. It will help you understand the history that led to today and will give you food for thought about tomorrow.

All in all, this is a "must have" for the serious 4A owner. It will help put the joys and sorrows of having an orphan into perspective.

[The Orphan Chronicles is available in the User Group publications library.]

When you have eliminated the impossible, whatever remains, however improbable, must be the truth.

Sir Arthur Conan Doyle, The Sign of the Four (1890)

Well, as you can tell, life in the Texas Instruments 99/4A orphanage is active and never dull.

---G. Albright in The Orphan Chronicles (1985)

The sellouts were a sight to behold. Rivalled only by the crunch to buy Cabbage Patch dolls the following year, when a J. C. Penney's or Sears or Montgomery Wards sold out TI, the buyers were lined up before the doors opened . . . Needless to say, stocks were easily sold out. Quickly and efficiently.

---Ibid

QUESTION

Do you remember where you were when you heard that TI was stopping production of the 4A?

FIRST LOOK: THE LOST HITS

Tenex is selling a disk with three games: Computer War, Submarine Commander and River Rescue. According to their catalog, Thorn EMI wrote these for TI just before Black Friday and then they disappeared. Tenex 'rediscovered' them [with more than a little assistance from our former supporter Howard Greenburg] and is selling them for \$29.95 (plus shipping). Requirements are a disk drive, 32K and XB, EA or Mini-Memory.

Arcade games for \$30? Well, my 14 year old twisted my arm until I sent for them. I have not had a chance to try them because I can't get the disk away from him. From over his shoulder I can tell you that they have excellent graphics, changing scenery or multi-screens (no Munch Man here) and require more than just a good hand with the joy stick. You have to think.

My son gives them his highest rating ("not bad" -- which is much better than "OK"). If you are into games, I would say that they are worth the \$30.

SOME THOUGHTS ON BLACK FRIDAY PLUS 5

As the 5th anniversary of Black Friday nears, it may be meet to ponder the future of our computer. You probably noticed that this column deals with this issue from a number of perspectives. Reading The Orphan Chronicles does make you think.

When TI opted out of the home computer market, I figured that the 4A would last for another two or three years. I decided to keep my system as I had too much time and money invested to dump it and move to another computer.

I never expected that, five years later, the support for the 4A would be as diverse, extensive, and strong as it is. The products available today far exceed those that TI gave us. Compare DM1000 to DM2, FAST-TERM and 4A TALK to TEII, TI ARTIST and GRAPHX to VIDEO GRAPHICS.

Look at the products that CorComp and Myarc continue to put on the market. Look at the quality fairware that keeps coming out.

The 4A will not live forever, but I believe that its useful lifespan is far from over.

There remains one key question: Now that we have done so well, what can we do to keep the 4A viable?

I do not think that there is any one answer. I do think, however, that it is time to stop and think about this issue and to think deeply. Time indeed.

THE INPUT-OUTPUT BUFFER

When you send data to your printer or to disk, your TI stores information in the input-output buffer. Generally it will keep data until it sees the end of a record.

To illustrate, consider this program that demonstrates the graphics abilities of Epson and compatible printers:

```
10 OPEN #1:"PIO.CR"  
20 PRINT #1:CHR$(27);"L";CHR$(127); CHR$(0)  
30 FOR I=1 TO 127  
40 PRINT #1:CHR$(I)  
50 NEXT I  
60 PRINT #1:CHR$(13)  
70 CLOSE #1
```

Line 10 opens your printer and tells your 4A NOT to send a carriage return and a line feed every 80 characters. Line 20 puts your printer in graphics mode and tells it to expect 127 graphics characters. The loop in lines 30 thru 50 send the entire range of graphics characters. Line 60 sends a carriage return to clear the printers buffer.

Since there is no print separator after the CHR\$(I) in line 40, each character is taken as a record and sent to the printer. If you added a semi-colon after the CHR\$(I), all 127 graphics codes would be held in the input-output buffer until line 60 executed.

The difference is speed. Without the semi-colon, it took about 10.2 seconds for this program to run. When the print separator is added, run time dropped to 7.5 seconds.

TRACKBALL TIP

The following article was on a disk someone sent me and was written by Rick Kellogg. I do not know where he can be reached.

A few months ago I purchased a trackball for my computer. I planned to use it with a few of the games that I had and with 'TI ARTIST'. Being somewhat lazy, I soon grew tired of switching the trackball and joystick plugs. Then an idea came to me. I pulled out my old (ahem!) ATARI 2600 and took the "Y" adapter. Lo-and-behold, when this adapter was hooked up to the TI joystick port, I could plug in both my TI joysticks and the trackball at the same time. Not only was this convenient, but at any time I could switch from joystick 1 to the trackball and not even loose a life in a game. For those of you who use ATARI compatible joysticks, all you have to do is plug your TI "Y" adapter into the ATARI "Y" adapter and you are all set. You 'TI ARTIST' fans will love it.

[The WICO trackball for the TI has been seen advertised by L L Conner in the USA and Computer Download in Canada. You will require to buy a power supply in the UK. ss]

The best laid schemes o' mice and men
Gang aft a-gley
---Robert Burns 1759-1796

But it does move!
---Attributed to Galileo Galilei
1564-1642

ON SUBPROGRAMS

SUBprograms, if you remember, use different variables from the main program. As a refresher, consider this:

```
10 A=3 :: CALL TEST :: PRINT
A
20 SUB TEST :: A=10 :: SUBEN
D
```

If you run this, you will get the number 3 on your screen because the variable A in line 10 is a different variable from that in SUB TEST.

I wondered about how DATA strings and DEFINations worked in SUBprograms. So I ran some experiments.

It turns out that a XB program can read a DATA statement anywhere. It works if the DATA statement is inside a SUBprogram and the READ command is in the main program or vice-versa. In other words, for purposes of READING DATA, the special rules about SUBprograms do not apply.

Not so with DEFINations. If you DEFINE A in the main program, it does NOT carry over into the SUB program. For example:

```
10 DEF A=10 :: C=A :: PRINT
C
20 CALL TEST(C) :: PRINT C
30 SUB TEST(C) :: C=A :: SUB
END
```

This little program will first print 10 from line 10 where C is set equal to A, which is DEFINed to be 10. It will then print 0 as inside SUBprogram TEST, A is not DEFINed so it is zero. In the same manner, if you DEFINE something inside a SUBprogram, that DEFINation does not operate outside of that SUB.

[Culled from files TI01 to TI03 by sjs. March 1988]

TURBO PASC 99 PROGRAM:

Here is a lengthy sample of a program written in TP99, to enable you to see how it appears, and for those of you with TI Pascal, you can see how the syntax varies. Professional Pascal users may be interested to see how arrays are passed to procedures!

A letter from TEXAMENTS, who have advertised TP99, advises that the English manual is not written, and that they will not be selling TP99. If you like the look of this code, please write to me as I may have heard of an alternate supply by the time this reaches you. TP99 appears to run some 13 times faster than TI USCD Pascal.

This is the first ever Pascal program written by me, and is based on a TI Basic program in the book "Dynamic Games for your TI99/4A" by Scott Vincent. I have wanted a flexible LIFE simulation in machine code, but not found any suitable, so I have taken advantage of Pascal to see what I can do!

In this version, you first set up the LIFE starting position by keying A for life and SPACE for no life, to fill a 12x12 grid. I have not put any cursor control in- it just moves over each line in turn, left to right. The outer frame has fixed status- it will affect the inner 10x10 grid, but no changes will occur in the frame: if you put life there it will stay there! This enables you to experiment with variations on the normal isolated life universe. To help spot the frame, the cursor is a + in the frame and a ? in the 10x10 grid.

You have the option of allowing automatic changes in generations, or keying them. If you select automatic, you may return to the set up section just by pressing a key. If you select keyed progress, to get back to the set up screen involves quickly pressing two different keys, one after the other.

There is one "bug" left in- the counter G is of integer type but there is no test for maxint, so after some 32000 odd generations the program will bomb out!

LIFE patterns may:

- Die out.

- Reach a stable pattern.

- Cycle between patterns- changing between two patterns is the most common, but I have created a 4-pattern cycle.

- As a variant of the above, a pattern has been found in which a "bullet" is regularly produced from a cluster, and moves off screen.

Here is the Pascal code, entire, and as successfully compiled...

< PASCAL CODE =====>

```

PROGRAM life;

VAR k,z:INTEGER;
    m,n:ARRAY[12,12] OF INTEGER;

PROCEDURE rules;
VAR k,status:INTEGER;
BEGIN
  cls;
  writeln("  LIFE");
  writeln("  first set up start position");
  writeln("  Put life forms into a");
  writeln("    10 x 10 grid ");
  writeln("  by pressing A or space");
  writeln(" (The 10x10 grid is in a frame-");
  writeln("  making a 12x12 grid in total:");
  writeln("  If you place life in the frame");
  writeln("  it will never die, and can ");
  writeln("  constantly create new life if");
  writeln("  the proper life conditions are met");
  writeln("  ");
  writeln("  There is no cursor control.");
  writeln("  Generation starts after 144th");
  writeln("  key press");
  writeln("  When prompted indicate if you");
  writeln("  want to prompt each generation");
  writeln("  with a Keypress, or let the");
  writeln("  program go full tilt.");
  writeln("  ");
  writeln("    Press any key to continue");
  writeln("    Stephen Shaw Jan 1988");
  REPEAT
    key(3,k,status)
  UNTIL status=1;
END;

```

```

PROCEDURE draw(VAR m,n:ARRAY[12,12] OF INTEGER);
VAR row,col:INTEGER;
BEGIN
  FOR row:=1 TO 12 DO
  BEGIN
    FOR col:=1 TO 12 DO
    BEGIN
      m[row,col]:=n[row,col];
      cursor(row+2,col+5);
      IF m[row,col]=1 THEN
        write("O")
      ELSE
        write("~");
    END;
  END;
END;

```

```

PROCEDURE think(VAR m,n:ARRAY[12,12] OF INTEGER);
  LABEL rerun;
  VAR c,g,k,s,t,y,z:INTEGER;
  BEGIN
    cls;
    cursor(2,2);
    writeln("  Press K to key generations");
    writeln("  Or any other key for auto");
    writeln("  ");
    writeln("      When all life ceases,");
    write("      Press a key to restart");
    REPEAT
      key(3,y,c)
    UNTIL c=1;

    ( y = 75 TO key prompt )
    cls;
    g:=0;
    REPEAT
      g:=g+1;
      FOR k:=2 TO 11 DO
        BEGIN
          FOR z:=2 TO 11 DO
            BEGIN
              c:=m[k-1,z-1]+m[k-1,z]+m[k-1,z+1];
              c:=c+m[k,z-1]+m[k,z+1];
              c:=c+m[k+1,z-1]+m[k+1,z]+m[k+1,z+1];
              IF (m[k,z]=0) AND (c=3) THEN
                n[k,z]:=1;
              IF (m[k,z]=1) AND ((c=2) OR (c=3)) THEN
                n[k,z]:=1;
              IF (m[k,z]=1) AND ((c>3) OR (c=0)) THEN
                n[k,z]:=0;

              key(3,t,s);
              IF s=1 THEN
                GOTO rerun;
            END;
          END;
        END;
      cursor(24,2);
      write("generation ",g);
      draw(m,n);
      IF y=75 THEN
        REPEAT
          key(3,k,s)
        UNTIL s=1;
      UNTIL false;
    rerun:
  END;

```

```

PROCEDURE setup(VAR k,z:INTEGER;VAR m,n:ARRAY[12,12] OF INTEGER);
  VAR j,jj:INTEGER;
  BEGIN
    REPEAT
      REPEAT
        cursor(k+2,z+5);
        IF (k=1) OR (k=12) OR (z=1) OR (z=12) THEN
          write ("+" )

```

```

ELSE
write("?");
REPEAT
key(3,j,jj);
UNTIL ((j=32) OR (j=65)) AND (jj=1);

IF j=65 THEN
BEGIN
cursor(k+2,z+5);
write("0");
m[k,z]:=1;
n[k,z]:=1;
END
ELSE
BEGIN
cursor(k+2,z+5);
write("^");
m[k,z]:=0;
n[k,z]:=0;
END;

z:=z+1;
UNTIL z=13;
z:=1;
k:=k+1;
UNTIL k=13;

```

END;

{ This is the actual program!:- }

```

BEGIN
cls;
rules;
REPEAT
k:=1;
z:=1;
cls;
text;
setup(k,z,m,n);
think(m,n);
UNTIL false;

```

END.

=====

It takes about a second for each generation- much faster than is possible in TI Basic! It would be simple to amend the size of the life universe, but the larger the universe the longer the creation time!

.....
.....

```

100 ! *****
110 ! * BEAMHEADINGS *
120 ! * AMATEUR RADIO *
130 ! * SEPTEMBER '85 *
140 ! * adapted for *
150 ! * for the TI *
160 ! * by A.Heino *
170 ! *****
180 CALL CHAR(123,"FFFFFFF
FFFFFFF",124,"55AA55AA55AA55
AA")
190 D=1 :: K=111.12 :: M=57.
2957795 :: N=60 :: S=69.86
200 DISPLAY AT(1,7)ERASE ALL
:"++++++":TAB(4);"++
++GREAT CIRCLE++++":
++DISTANCE & BEARING++:" ++
++++"
210 DISPLAY AT(6,1):"ENTER V
ALUES IN WHOLE":"DEGREES AND
DECIMALS.":"USE MINUS VALUE
S FOR SOUTH":"LATITUTES & EA
ST LONGITUTES"
220 IF D<>1 THEN 280
230 A=-34.28 ! YOUR LAT
240 A=A/M
250 L1=-150.29 ! YOUR LONG
260 DISPLAY AT(11,1):"ENTER
DX QTH"
270 ACCEPT AT(11,15):A$
280 DISPLAY AT(14,1):"LATITU
TE" :: ACCEPT AT(14,12):B ::
B=B/M
290 DISPLAY AT(17,1):"LONGIT
UTE" :: ACCEPT AT(17,12):L2
300 L=(L1-L2)/M
310 E=SIN(A)*SIN(B)+COS(A)*C
OS(B)*COS(L)
320 D=-ATN(E/SQR(1-E*E))+1.5
7079
330 C=(SIN(B)-SIN(A)*E)/(COS
(A)*SIN(D))
340 IF C>=1 THEN C=0 :: GOTO
360 ELSE IF C<=-1 THEN C=18
0/M :: GOTO 360
350 C=-ATN(C/SQR(1-C*C))+1.5
7079
360 C=INT(C*M)
370 IF SIN(L)<0 THEN C=360-C
380 R=180+C
390 IF R>=360 THEN R=R-360
400 R=INT(R)
410 CALL HCHAR(6,1,32,32*18)
420 LLA=LEN(A$):: LLB=28-LLA
:: DISPLAY AT(6,LLB/2-1):RP
T$("½",LLA+2):: DISPLAY AT(7
,LLB/2-1):"½";A$;"½"
430 DISPLAY AT(8,LLB/2-1):RP
T$("½",LLA+2)
440 DISPLAY AT(10,1):"BEARIN
G";C;"DEG (SHORT PATH)"
450 DISPLAY AT(12,1):"BEARIN
G";R;"DEG (LONG PATH)"

```

```

460 DISPLAY AT(14,1):"DISTAN
CE";INT(N*D*M);"NM"
470 DISPLAY AT(16,1):"DISTAN
CE";INT(S*D*M);"MILES"
480 DISPLAY AT(18,1):"DISTAN
CE";INT(K*D*M);"KM"
490 DISPLAY AT(22,1):"PRESS
ANY KEY TO CONTINUE..."
500 CALL KEY(0,KEY,STAT):: I
F STAT=0 THEN 500
510 GOTO 180
520 END

```

* PICASSO TIPS - Vers 2.0 *

1. Moving large chunks of graphics from one part of the large screen to another part.

One way of achieving the result is to use the Move option, by putting the pic into the bottom RH corner then shifting the screen viewing area until it is at the top LH corner.

This method might be OK if you haven't anything drawn there. The other way is to first save the whole file, now move it to the position without worrying about what gets in the way, now save the current screen using option 4 on the file utilities. Load your original file, then overlay the graphic file using option 3 on the file utilities and voila you done it.

2. Erasing large graphic chunks is easy by using the Window option and pressing the 'U' to toggle pixels off then press 'A' to fill framed area with blanks.

3. Version 2.0 has few extra features.eg.

- ICON DRAW 'H'
- ICON EDITOR '.'
- ICON SAVE '3'
- ICON LOAD '4'
- TRANSPARENT COPY 'Z'(toggle)
- 33 TEXTURES

4. Also I decided to make the slanted lines from the brushes only 1 pixel wide. The Text cursor is now a square to make it easier to locate the 8*8 Text letters. Area fill was added to Ver 1.4.

5. The Icon option is a very versatile addition, you can now have either a large letter 16*16 or if you load the 'ICON-Z****' it gives you a spray can. When drawing with this option the pixels will to step by 3 , just press the joy slow '-' 2 times to draw continuous pixels. You have 5 ICONS available.

```
*****
*          *
*  USING PILOT 99  *
*          *
*****
|by Arto Heino|
```



To write a new language for our 99/4A is not as awe aspiring as you think. The FORTH language was written in such a way that a whole new language can be easily implemented step by step. PILOT was written in such a way by THOMAS P. WEITHOFER.

Your PROGRAM is actually a TEXT FILE that you write using an EDITOR such as E/A, TI-WRITER or even an EXTENDED BASIC program! Once you have written a program and saved it, you must now LOAD you PILOT SYSTEM DISK.

If you have the E/A version and only have XB then LOAD in your XB FORTH DISK and just type 'COLD' with the PILOT DISK in your drive.(also done with FORTI) If you have a RAM DISK you will be very pleased at the new speed of FORTH or PILOT or any FORTH BASED language.

PILOT is a very powerful language for Programmed Inquiry Learning Or Teaching which is what PILOT stands for. Where as LOGO is GRAPHICS based, PILOT is TEXT based, that is, until THOMAS put a whole lot of the advanced features of our TI 99/4A into the language.

The best way to LEARN a new language is to USE IT! I have written a short program that draws a 12 month bar graph of the data you input.

```
R: *****
R: BAR GRAPH by Arto Heino
R: T.I.s.H.U.G.
R: *****
IT:
CH:
R: *****
R: DEFINE CHARS
R: *****
CP: 128,0000,001F,1010,1010
CP: 129,0000,00FF,0000,0000
CP: 130,0000,00FO,1010,1010
CP: 131,1010,1010,1010,1010
CP: 132,1010,101F,0000,0000
CP: 133,1010,10FO,0000,0000
R: *****
R: DISPLAY OPENING SCREEN
R: *****
TC: 12,8
T: PILOT BAR GRAPH
HC: 10,8,128,1
HC: 10,9,129,15
HC: 10,24,130,1
VC: 11,8,131,3
VC: 11,24,131,3
HC: 14,8,132,1
HC: 14,9,129,15
HC: 14,24,133,1
LP: 70
EL:
```

```
R: *****
R: DRAW UP MAIN SCREEN
R: *****
*GO
IG:
TG: 1,20,PILOT BAR GRAPH by Arto Heino
TG: 16,8,Jan Feb Mar Apl May Jun Jul Aug
TG: 16,40,Sep Oct Nov Dec
TG: 2,3,100
TG: 6,3,75
TG: 9,3,50
TG: 12,3,25
TG: 15,3,0
DL: 16,21,118,21
DL: 118,21,118,208
```

```
R: *****
R: SET UP VARIABLES
R: *****
C: #V<-0
C: #N<-8
C: #L<-116
R: *****
R: MAIN LOOP
R: *****
LP: 12
C: #V<-#V+1
C: #N<-#N+16
T: Values Between 0-100
T: Input Data for Month :#V:
A: #A
T:
C: #I<-#L-#A
C: #J<-#N+8
C: #O<-#I-4
C: #P<-#N+4
C: #Q<-#P+8
DR: #I,#N,#L,#J
DL: #I,#N,#O,#P
DL: #O,#P,#O,#Q
DL: #O,#Q,#I,#J
DL: #O,#Q,#L-4,#Q
DL: #L-4,#Q,#L,#J
EL:
R: *****
R: END OF LOOP
R: *****
T: Try Again Y or N ?
A: $A
M: Y
JY: *GO
M: N
JN: @A
E:
```

Commands used in the Program:

```
R: Remark
IT: Initialize Text Mode 32*24
CH: Clear screen and Home cursor 1,1
CP: Same as CALL CHAR
TC: Text Cursor Position Y,X
T: Same as PRINT
HC: Same as CALL HCHAR
VC: Same as CALL VCHAR
LP: Initialize Loop count
EL: End of Loop
*GO Lable postion for a Jump
IG: Initialize Graphics Screen 256*128
TG: Print At Y,X Graphics Screen 64*16
DL: Draw Line Y1,X1,Y2,X2 (BITMAP ONLY)
DR: Draw Rectangle Y1,X1,Y2,X2 "
C: Compute Numeric Variables
A: Same as INPUT
M: Match same as IF A$="Y" THEN TRUE
JY: Jump with added modifier same as IF TRUE THEN JUMP
JN: Jump with added modifier same as IF FALSE THEN JUMP
E: Same as END
```



by Dave Wakely

Before starting this month's Spadventure, I'd like to backtrack a little. The original idea for this column arrived shortly after I purchased Spad XIII last summer. At the same time it also occurred to me that I knew very little about the development of this program. For example, nowhere in the documentation or in the program itself is the author's name mentioned. All I had was the information that it was produced by Not-Polyoptics, a long-time fixture in the II 3rd party software world. My letter to them brought a detailed response from Gene Harter, one of the partners in Not-Polyoptics, with information which you might find interesting.

What follows is presented as an interview. Actually it is the questions contained in my original letter, interspersed with the responses I received:

DW: Who wrote Spad XIII? How long did this programming effort take, and what problems were encountered?

GX: Spad XIII was written by my brother David, with some assistance from me. He was born, by the way, in Chicago, while our family was living in Des Plaines. If you want to write him he lives at 416 High St., Chestertown, MD 21620.

Spad took about 8 months to write. Soon after we released the original version David discovered graphic algorithms that would make Spad about twice as fast (things often happen like this), so we wrote Mark 2, which includes the new algorithms and some extra features, some of which were suggested by users.

DW: The viewing area in Spad is obviously clipped on the edges. Was this done, as it appears, to obtain or preserve "aspect ratio"?

GX: The reason the screen is clipped is to decrease cycling speed; we estimate a 50% faster speed as a result. F-15 Strike Eagle for the PC cuts off half the screen for the same reason.

DW: Both the sky and ground in Spad are blue. Why? Did some constraint of the computer dictate this?

GX: The greatest limitation of the graphics, we found, was that the II doesn't have a true bit-map mode, that is, each bit does not have its own color, but is either on or off in the context of its byte, which can have only two colors. So whenever the program tries to write three different colors on a byte, say a horizon line, a smoke line, and the blue sky, one of the colors is ignored and the image drawn "smears" over that byte. That is why the ground is blue; if we had colored it in green the smearing would have been unacceptably distracting. Of course we used the bit-map mode that is available in order to get the 3D line perspective views, and I don't think going to mono-color would have helped.

DW: I was quite surprised and pleased to receive the Mark 2 version of Spad XIII. Are further upgrades in the works now or planned? If so, is user feedback as to desired features wanted?

GX: As far as another update, there is no improvement planned as dramatic as the faster algorithm and Red Barons in Mark 2, and memory restrictions are getting pretty tight for any more features, but we do have some ideas.

Chicago Times

PAGE 20

I would just urge everyone to send in their registration cards, just in case. (Also) for your information, Spad is not a 32K program - it is actually over 40K, using console memory and UDP RAM as well as the Extended Memory.

DW: Is there a version of Spad XIII planned specifically for the Myarc 9640 computer?

GK: Spad now works in the Extended Basic mode of the Myarc 9640, but we don't plan on a version specifically for that computer unless we see a perfected version of the Operating System and evidence that more are being sold than at present.

DW: One of the attractions of the Microsoft Flight simulator is the "undocumented" features that author Bruce Artwick put into the program. Are there any "undocumented" features of Spad XIII such as uncharted graphics, etc.?

GK: There are no undocumented features that I am at liberty to publicize.

DW: How was the decision made as to what villages to place in the program? For example, why was Soissons included, but Belleau Woods and Chateau Thierry, where many Americans fought and died, left out?

GK: The French villages were chosen just as representative villages in the vicinity of Paris, caught up in the currents of the war.

DW: This one is half question and half suggestion. In the Microsoft FS, a number of "scenery disks" are beginning to appear. That is, the programming for flight is independent of the programming for scenery. Is it possible (or how about) a system whereby new "operational areas" are available, possibly as "add on" disks. The precedent for this exists, for example, with the TOD editor, whereby Tunnels of Doom owners can have new adventures with the same program.

GK: I like your idea for user-created scenery; maybe a user-created airplane too!

DW: Where did the name "Not-Polyoptics" come from, and what does it mean?

GK: We bought a 99/4 in 1980, choosing it over the Apple II, mostly because it had better graphic and sound capabilities. Being programmers and game designers it was a short step to forming our own company (called Not-Polyoptics because one of the partners didn't like the name Polyoptics), and thus we became one of the first companies to support the II. Through the years, we have seen other companies come and go; we have seen advertising possibilities dwindle; and we have seen the II market survive the most virulent blows and continue to survive. We still sell more than twenty games and simulations for the II, though in the past year and a half our sales of older games have dwindled. Some of our newer Extended Basic programs such as Laser Tank, Tower, and NORAD, really show off some of the capabilities of the II, but we have found it very difficult to get out the word to the II Community. Even Spad has not sold as well as we had expected. We appreciate your efforts to publicize Spad, and look forward to seeing future installments of the adventures.

Chicago Times

PAGE 21

Spadventure #3: Test Flight

You will want to read this adventure all the way through first, before proceeding, so that you will know in advance (hopefully) what to do.

Several officials from the Societe Pour Aviation et ses Derives (S.P.A.D.) have just delivered to the airfield their newest machine. The Spad 13.1 is carrying what is purported to be a new, more powerful experimental engine, and they have asked your wing commander to pick out his most accomplished pilot. He wasn't around, so instead you were chosen to field-test it. Congratulations.

As you climb into the cockpit, check out all the instruments and controls, using our usual pre-flight procedure. The plane is brand new, has been freshly painted, and looks great. The engine has been idling and is warm. In fact, the plane even smells good, but no matter, so open it up completely ("S" key) and let's see what this thing can do. When speed reaches 100 MPH, pull back once on the stick and feel the rush of cold air from the backwash of the prop. Better throw the silk scarf around your face one more time.

Staying at full throttle, take the new machine right up to exactly 1,000 feet in an all-out climb. Straight ahead and up. This is great. You can feel the raw power of the engine coursing through the plane, and you can already picture yourself flying circles around the Red Baron. Now the bad news.

Reach up and hit the "O" key - NOW. Your engine has just quit. No, not quit. Died. The specially glazed crepes used as experimental cylinder linings just disintegrated, and several pistons have seized, turning the plane into an expensive rock. If I were you I wouldn't think about this situation too long, however, as your stall indicator just buzzed, and if I were still you I'd start doing something to keep from becoming a grease spot on the French landscape.

You have a fairly limited number of options here. First and foremost is your desire to live long enough to see Spadventure #4.. On the other hand, anyone crazy enough to take on this test is also crazy enough to see if he can now make it back to the airfield as well. That is, from this position, at this altitude, is it possible with no power to actually land back at the airfield? Only a wild and crazy person like you would even attempt it. Oh, there's probably some principle active here about the conservation of energy and whatnot, but perhaps you are not bound by those constraints. You do want to keep in mind that it's not for nothing that they call this a "dead stick" landing.

So, for this game-within-a-game here are the rules: You must take off at full power, pulling back once on the stick at 100 MPH and climbing until the altitude indicator shows exactly 1,000 feet. During the takeoff and ascent you may use no aileron or rudder controls. You must then cut power to "O" and cannot power up again. When the power is off you may use any and all other techniques which help return you to the home airfield (except that no, you Mark 2 users may NOT use the "M" key!).

If any part of your plane comes to a safe stop within the lines which mark the runway, you will be hailed as a Master Aviator and receive the French Legion of Valor.

Chicago Times

PAGE 22

If you land safely close enough to the runway such that the simulator repositions you on it facing North for the next takeoff, you get to keep the pot of money from the bets that your ground crew was making that you wouldn't be back.

If your plane lands safely within sight of the airfield (you will be able to make out the roof of the hangar even from some distance away), consider yourself lucky. As you walk back, carefully compose in your mind what you want to say to the SPAD officials about their new construction techniques.

If you land safely with absolutely no idea where you are, send up a signal flare and consider transferring to the Artillery division.

No matter where you ended up, if you soiled your flight pants during this event, begin walking to the English Channel. Then keep going.

For any other landing, we will come visit you in the field hospital. You can try again when you get out. They're working on this idea of using stale loaves of French bread for propellers...

Good luck.

ADDENDUM

Editor/Assembler Owner's Manual

Please mark the following changes in your manual.

<i>Page</i>	<i>Section</i>	<i>Description</i>
42	3.1.3.1	In the last sentence of the first paragraph, change "least" to "most".
92	6.10	In the second line of the example explanation, change "value of ADDR" to "value in ADDR".
104	6.14.2	In the example, change "MOV *11,1" to "MOV *11+,1".
127	7.20.1	In the next-to-last line, change ">2220" to ">C220".
168	10.5	In the example, change ">2A41" to "@ >2A41" and "Register 3" to "Register 2".
262	16.2.4*	(Add the following.) NOTE: Some devices modify the GROM read address. RS232 and TP are known offenders. If your program accesses these devices, save the current GROM address (see section 16.5.2) before the I/O operation, and restore it (see section 16.5.1) after your program has accessed the device. Otherwise, the program will not be able to return to the Editor/Assembler or BASIC or to perform a BLWP @GPLLNK properly.
289	17.2.6	Change line 130 in the BASIC program to <pre>CALL LOAD("DSK1.BSCSUP","DSK2.STRING")</pre> This assumes that you have entered the source file on the next page by means of the Editor, saved the file as DSK2.STRING, and run the Assembler, using DSK2.STRING for a source file and producing DSK2.STRINGO as an object file.
328	21.1	The default for VDP Register 7 is >07 in TI BASIC and Extended BASIC.
335	21.5.2	In the last paragraph, change ">00 or >04" to ">03 or >07".
335	21.5.3	In the last paragraph, change ">00 or >04" to ">7F or >FF".
415	24.4.8	Change the second instance of GRMRD to "GRMRA EQU >9802".
416	24.4.8	Change the second line to "NUMREF EQU >200C".
420	24.5	(Add the following.) NOTE: A program to be saved using the SAVE utility should not have an entry point defined on the END statement. If you want to save the Tombstone City game in memory image format, you must first change the last line from "END START" to "END" and then reassemble the program. Otherwise, the game starts to run as soon as it is loaded, and you will not have a chance to execute the SAVE utility.
465	Index	VDP Write-Only Registers: add page 267 to references.

MODULE LIBRARY

The following modules are now available for loan:

Addition & Subtraction	Alpiner	Beginning Grammer
Bigfoot	Blackjack & Poker	Blasto
Buck Rodgers	Burgertime	Car Wars
Chisholm Trail	Connect Four	Demonstration Module
Driving Demon	Early Reading	Hangman
Henhouse	Hopper	Household Budget Man
Hunt the Wumpus	Indoor Soccer	Jungle Hunt
Meteor Belt	Moon Patrol	Moonsweeper
Munchman	Music Maker	Number Magic
Othello	Parsec	Personal Report Gen.
Personal Record Keeping	Pole Position	Protector
Q Bert	Sewermania	Space Bandits
Star Trek	Super Demon Attack	Terminal Emulator II
The Attack	TI Invaders	TI Logo I
TI Logo II	Tombstone City	Tunnels of Doom
Video Chess	Video Games	Yahtzee
Zero Zap		

BORROWING MODULES FROM THE LIBRARY

There will be an initial once and for all returnable deposit of £10.00 on joining the library. Borrowing fee will be £1.50 per module per 4 week period. If the module you require is not listed please ask!

Application to Join the Module Library.

Name:..... List modules required:
Address:.....
.....
.....
.....

I enclose cheque/PO for £..... (£10.00 deposit +£1.50 per module)

Please make cheques payable to TIUG(UK).

The address to write to is:

MR. E.H. SHAW
CROW HOLT FARM
BASFORD
LEEK
STAFFS ST13 7DU

LENDING MODULES TO THE LIBRARY

If you have modules not on the above list then would you like to loan them to the library?

Receipts will be issued to owners lending modules to the library and the value of the module must be agreed before the loan. Modules held by the library can be returned to the owners at two weeks notice. Please note that you can keep your module yourself if you prefer until it is requested by the library.

RS232 AND ALL THAT

by Peter Walker

This is not an article for "Boffins". This article is dedicated to those who have the good fortune to own an RS232 card, presumably sitting in the PEB, but may find some frustration in getting it to work with their printer, modem or other communicating equipments, such as another computer. Indeed many of you may have wondered in a fit of anger why it is that a communications interface that calls itself a serial port should apparently have so many parallel connections with confusing abbreviations such as DCD, CTS, and DSR. The problem is that the TI99/4A, like all computers, requires different types of connection depending on the type of connected equipment.

So first things first. The US Electronic Industry Association (EIA) created the RS232C standard for asynchronous serial communications based on the International CCITT Recommendation V24. Don't confuse V24 with other Recommendations defining types and speeds of data communication on the telephone line, such as V21 (for 300 bit/s full duplex), V22 (for 1200 bit/s full duplex) and V23 (for 1200 bit/s half duplex and 1200/75). By the way, "Duplex" means data passing in both directions simultaneously, and "Half-Duplex" data passing in each direction alternately. To return however to Recommendation V24. This defines the interface between the computer or terminal and the modem which is needed to convert the digital data signals into analogue tones suitable for transmission over the telephone network. Terminals and computers are known as Data Terminal Equipments (DTEs) and modems as Data Communications Equipments (DCEs). The complication is that we often use the RS232 port to communicate with printers, plotters and other directly wired computers which all require a different use of the sundry control leads than a modem does, hence the confusion.

Welcome to the 3rd paragraph!! To those who have coped with the jargon in paragraph 2 and even to those who have understood it, we'll now back track a little to consider what we mean by asynchronous serial communications. Serial means that the data bytes are sent one bit after another as opposed to the Parallel Centronics port (PIO) which we also have on our RS232 port. In this latter port, data bytes are presented simultaneously over 8 parallel data lines. For simplicity I won't be going into the details of the parallel port.

Asynchronous means non-synchronous. In a synchronous data stream a computer would send a permanent stream of data bits at a certain speed. Such a system has several complications:-

- 1 The receiving computer needs to know which bit in the stream represents the 1st bit in each byte so that the bytes can be decoded from the stream.
- 2 Using a simple electrical coding such as binary 1 = 8 volts, binary 0 = -8 volts, a continuous stream of either 1s or 0s would mean a continuous electrical voltage level, so it is rather difficult for any receiving computer to maintain synchronism with the incoming bit stream. Synchronisation bytes or a separate clocking stream become essential.

3 Computers and more especially "dumb" terminals fundamentally do not emit continuous bit streams but only do so when a key is pressed. Therefore since the earliest days of telegraphy (remember teleprinters?), terminals have used Asynchronous data transmission.

In Asynchronous transmission, the DTE (note the subtle introduction of jargon here!) keeps the transmit line at binary 1 ("mark") until it wants to transmit a data byte. It then sends a binary 0 ("space"), the START bit, followed by the 8 information bits and then a STOP bit (a binary 1). The STOP bit serves to separate the data from any following START bit and allows time for old fashioned teleprinters to print the character. Sometimes 2 stop bits are used to give these printers more time. Note that the receiving computer must use the start bit to trigger its own timer so that the data bits are properly decoded even if they have several consecutive 1s or 0s.

I mentioned above the 8 data bits. This can comprise an 8 bit byte representing the 256 different combinations used in the computer data or a 7 bit byte plus a parity bit (an error checking bit). Sometimes 7 bit data is used with no parity in which case two stop bits are generally used. Usually "even" parity is used for asynchronous data and "odd" parity for synchronous data.

So much for the serial data itself - now about these confusing control leads. First we need to look at the plug and pin arrangements. Both DTEs and DCEs usually have 25 pin female D plugs and should be interconnected by a "straight" male to male lead, ie pin 1 to pin 1, pin 2 to pin 2 etc. The RS232C standard defines the functions of all 25 pins but these are never all used on personal computers since many of the pins are for very specialised uses such as timing for synchronous transmission, sundry test purposes and a second set of signal and control leads for allowing backward channel working, ie when the DTE and DCE swap functions. The key pins for our purposes are as follows:

- 1 Earth
- 2 Transmitted Data TD
- 3 Received Data RD
- 4 Request to Send RTS
- 5 Clear to Send CTS
- 6 Data Set Ready DSR
- 7 Signal Earth
- 8 Data Carrier Detect DCD
- 20 Data Terminal Ready DTR

Before describing these functions I should stress that this is NOT the wiring scheme of the TI99/4A for two reasons. Firstly, as crazy as it may seem the TI99/4A RS232 port is configured as a DCE (a modem!!) rather than as a DTE (terminal). Therefore it transmits data on pin 3 rather than pin 2. Pin 20 is an input and Pin 6 is an output. I can only imagine that this was done since TI assumed that the RS232 port would most often be used with printers rather than modems. It happens that printers are usually configured as DTEs rather than DCEs so the TI99/4A can be connected to a printer using a "straight" pin-to-pin lead. This tends to be spoilt by many printer manufacturers who use non-standard ways of indicating Buffer Full/Busy such as on Pin 11. All this leads to difficulties and frustration for all and doesn't make interfacing with modems any easier either.

The second reason why the TI99/4A is different is because we have a completely separate second RS232 port on the same plug. Its wiring is totally non-standard.

Now lets look at the functions of the various data and control leads bearing in mind that they are designed for a DTE (a terminal) sending data to line via a DCE (a modem) and will not be applicable for the TI99/4A with its peculiar DCE style port.

Transmitted Data TD This is the data sent by the computer (DTE) to the DCE (modem). Note that it is still known as transmitted data when received by the modem since eventually it will be transmitted to line.

Received Data RD This is the data stream from line via the DCE to the DTE.

Request to Send RTS A signal sent by the DTE to the DCE indicating it wants to send data. There are two ways in which RTS is used. Some terminals have their RTS lead on at all times, others use the "conversation" mode. In this mode, when the DTE is about to transmit a byte, it turns RTS on and transmits the byte if or when CTS is on. After transmission the RTS lead is turned off again. This mode of working is designed for such systems as 1200bit/s half duplex (V23). With V23, data cannot be sent and received at the same time so the transmit carrier is only switched on when RTS is on. At other times the DCE (modem) is in receive mode.

Clear to Send CTS A signal sent by the DCE in acknowledgement to the RTS indicating that the DCE is able to receive data. As set out above, the RTS/CTS handshake is only fully applicable to half duplex working where the send/receive function is toggled by RTS. A DTE should always halt transmission if CTS goes off for any reason. There are a few occasions when modems turn off CTS (other than in response to RTS going off). Its unlikely you'll come across these very often, but examples are found with modems with buffers to cope with error correction and speed conversion.

Data Set Ready DSR A signal sent by the DCE indicating that it (the modem) is "on-line". A DTE will not transmit unless both DSR and CTS are "on".

Data Carrier Detect DCD A signal sent by the DCE indicating that a data carrier is being received from the far end. Loss of this signal would indicate that the remote computer had disconnected the call. A DTE will only be capable of reception if both DSR and DCD are on.

Data Terminal Ready DTR A signal sent by the DTE to the DCE indicating that it is ready to transmit data. A modem may not be able to go "on-line" unless DTR is on. Often a modem will force clear the telephone connection if DTR goes off during a call.

By the way, control leads are "on" when they are 3 volts or more positive, and "off" when -3 volts or less negative. The TI99/4A transmits on/off as about +8/-8 volts. With the TI99/4A, a disconnected lead is considered "on" which can simplify the wiring. Other computers however work the other way about and certain leads must be connected to a positive source to allow proper transmission and reception.

Since the TI99/4A is configured as a DCE and printers as DTEs, a little care is needed when sorting out the right connections. What you need to decide is whether some form of flow control or handshaking is required and if so which pins need to be connected. The RTS/CTS handshake has been described above, but as we have seen, its unlikely you'll need this. Another form of flow control is used with a printer. The computer can transmit data to it at full speed until its print buffer becomes nearly full. The printer will then usually turn off the DTR lead so telling the computer to halt transmission. When the print buffer becomes less full it will turn on the DTR lead again and data will restart. Sometimes a separate Printer Buffer Full/Busy lead is used instead of DTR.

By comparison, as we have seen, a modem rarely requires control lead handshaking. Data is usually free to flow through the modem at the speed the computer transmits. What is more, a computer may wish to send data to a modem even when the latter is not receiving a carrier or is not on line at all. This is because many "intelligent" modems use the so-called "Hayes Protocol" to send ASCII data to the modem itself to set or change its transmission parameters. What modem communication does need however is flow control with the remote computer's buffer and this is achieved by using the data stream itself: the X-ON/X-OFF protocol. X-ON is ASCII character 17 (or DC1) corresponding to Control Q in most cases. X-OFF is likewise ASCII 19, DC3 and Control S. The remote computer, when its buffer is nearing full, will send X-OFF and the transmitting computer will halt transmission. When ready to restart an X-ON is sent. Since this all goes on within the data stream itself, we need not worry too much about flow control when wiring our RS232 to a modem.

Lets now look at the TI99/4A pin configuration. We need to remember that we have two ports on the one plug.

Function	Port 1	Port 2
Earth	1	1
RD (into TI99)	2	14
TD (out of TI99)	3	16
Spare output	5	13 (the manual calls this CTS)
DTR output	6	6 (the manual calls this DSR)
Signal earth	7	7
RTS output	8	12 (the manual calls this DCD)
DSR (& CTS) input	20	19 (the manual calls this DTR)

In the above list I've tried to show what the pin function is viewing the computer as a DTE, whereas, as mentioned above the port and manual descriptions use DCE conventions. Its worth noting that the TI99/4A uses "conversation" mode on the RTS lead. That is, it raises RTS when ready to send, transmits the byte when or if pin 20 is "on" and lowers RTS again after the byte is sent. There is only a single handshaking input on pin 20 so there aren't separate inputs for CTS or DCD. Pin 6 is always on. It is fed at 12 volts via a 1.8k ohm resistor and is useful for feeding devices that may need a definite positive signal to hold "on". The spare outputs on pins 5 and 13 aren't used and are set high. They can be switched off via Assembly Language instructions to CRU addresses >130A and >130C. This could provide a switchable DTR if desired or perhaps drive an auto-dialling facility within the modem.

Lets now look at how the TI99/4A should be connected to a modem, a printer, another directly wired TI99/4A and a non-TI computer.

MODEM CONNECTION

```

TI99/4A RD   2 <----- 3 RD   MODEM
          TD   3 -----> 2 TD
          DSR 20 <----- 5 CTS
          Earth 7 ----- 7 Earth
    
```

Assuming the modem's DTR and RTS inputs default to "on" if disconnected, no other leads are needed. If this is not the case add as well:

```

DTR  6 -----> 4 RTS
          [-----> 20 DTR
    
```

If you do need full "conversation" mode for half duplex toggling, use this for the RTS input instead:

```

RTS  8 -----> 4 RTS
    
```

PRINTER CONNECTION

Unfortunately printers tend to vary in their pin designations so you must read your printer manual carefully. However, the following is typical.

```

TI99/4A RD   2 <----- 2 Data out  PRINTER
          TD   3 -----> 3 Data in
          DSR 20 <----- 20 or 11 (Buffer full/Busy)
          Earth 7 ----- 7 Earth
    
```

The TI99/4A's RS232 card can be modified to transfer the DSR input onto pin 11 to aid printer connections (Remove strap W5, insert strap W7). You may wonder why a printer might have a data out port. Well some printers can be configured to use the X-ON/X-OFF protocol instead of the direct wired flow control.

INTERCONNECTED TI99/4As

```

TI99/4A RD   2 <----- 3 TD   TI99/4A
          TD   3 -----> 2 RD
          DTR  6 -----> 20 DSR
          DSR 20 <----- 6 DTR
          Earth 7 ----- 7 Earth
    
```

Note that since DTR is always on, this configuration provides no flow control. Generally none would be required.

ANOTHER COMPUTER

In this example I am assuming that the other computer is correctly wired as a DTE. You will note that despite the TI99/4A's strange connections, such a computer can be connected using a "straight" lead as used for connecting to a modem. This is in line with my statement above that the TI99/4A is wired as a DCE. I've therefore shown a very "complete" lead to underline this point.

TI*MES

TI99/4A	RD	2	<-----	2	TD	Another Computer
	TD	3	----->	3	RD	
		4	<-----	4	RTS	
Spare	O/P	5	----->	5	CTS	
	DTR	6	----->	6	DSR	
	Earth	7	-----	7	Earth	
	RTS	8	----->	8	DCD	
	DSR	20	<-----	20	DTR	

So, despite some of the apparently odd cross connections, the above works since both pin 5 and 6 are positive so both CTS and DSR inputs on the other computer are correctly driven for transmission purposes. Equally, when the TI99/4A is transmitting, pin 8 is positive, so both DSR and DCD are correctly driven for reception. Of course, if you are making up a lead from scratch you won't need all these connections.

I hope the above examples show the general principles to be used. In a nutshell, make sure you get the TD/RD connected in the right sense. If flow control is required, find the pin that does this at the connected device and connect it to pin 20 on the TI99/4A.

Its worth mentioning that not all data ports are 25 pin D plugs. On the BBC computer and the Nightingale modem, 5 pin (widespaced) DIN plugs are used. These are called RS432 and support TD, RD, RTS, CTS and Earth only, but this is generally enough to provide any necessary control. On another computer I have, a 9 pin male D plug is used with again a restricted range of pins (but more than enough to do most jobs).

To those of you who have struggled through this article, I hope it has provided some illumination on the vexed question of wiring RS232 connections and why in particular the TI99/4A provides special difficulties. I don't claim to possess special knowledge in this field but if I can be of assistance, my address can be found at the front of this magazine. For those who wish to telecommunicate, I can be reached on Prestel Mailbox 707873778 or Telecom Gold 74:MIK1900. This assumes of course you've got that modem wired up right!

>LOGOFF PETER WALKER ELAPSED TIME Several Hours!

by Christina

After our little brush with GOTO last time...and I know you've been practising ever since!...you know how to change the order in which program lines are executed. We need now to consider "relational tests" which will allow you to put the GOTO statement to useful work.

Relational tests are tests that compare two items and decide whether the relationship between the two items is true or false.

For example:

A relational test to decide whether 6 is greater than 5 would be written as:

6>5

The answer to the test is true.

The symbols used to specify the relational test to be carried out are called RELATIONAL OPERATORS, and they are:

SYMBOL	MEANING
=	is equal to
>	is greater than
<	is less than
>=	is greater than or equal to
<=	is less than or equal to
<>	is not equal to

The important point about relational expressions is that they are either true or false.



Which option A, B or C is the option that completes the following examples correctly:

Relation	Result	Relation	Result
8>7	True	30>=10*2	True
7>8	False	4*9=72/2	?
30>10*3	False	7<8	?
5=5	True	56/2>29	?

Each relational test is constructed in the same way:

An expression - A relational operator - An expression

You will remember that expressions can consist of a mixture of numbers, variable names, and computations. It is the results of the computations and the contents of the variables that are used in the relational test.

If three variables X, Y, and Z are assigned values as follows:

```
10 LET X=7
20 LET Y=4
30 LET Z=12
```

Write down the results of these relational tests:

Relation	Result	Relation	Result
7=X+Y		X=Z/Y	
Y<Z		(2*X)+Y=Z+X	
Z/3<X		X*Y=Z	

The IF-THEN statement

This statement combines the relational operators and GOTO statements. IF something is true, THEN a particular action is taken. For example IF I am hungry, THEN I will eat. Simple, eh? After the keyword IF there must be a relational test argument.

10 IF 3+2=5 THEN GOTO 70 This relational test is true, the program will jump to Line 70. (Answer to test above: option B)

More on IF-THEN next time. See you,

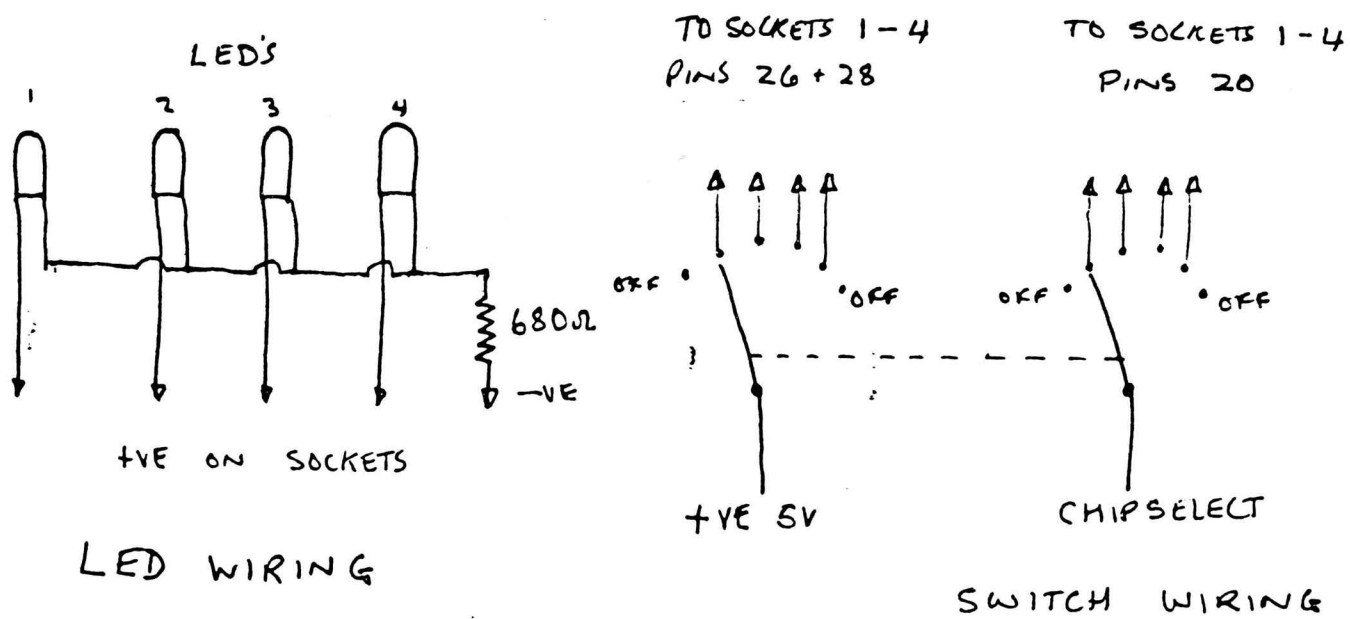
Christina

This is a multi socketed ROMOX module which can be switched to select whichever EPROM is required. Because of the various ways this could be put together I will describe the way I tackled it and leave it to the individual to put their own interpretation into practice.

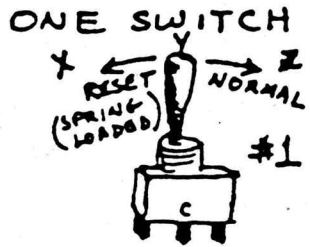
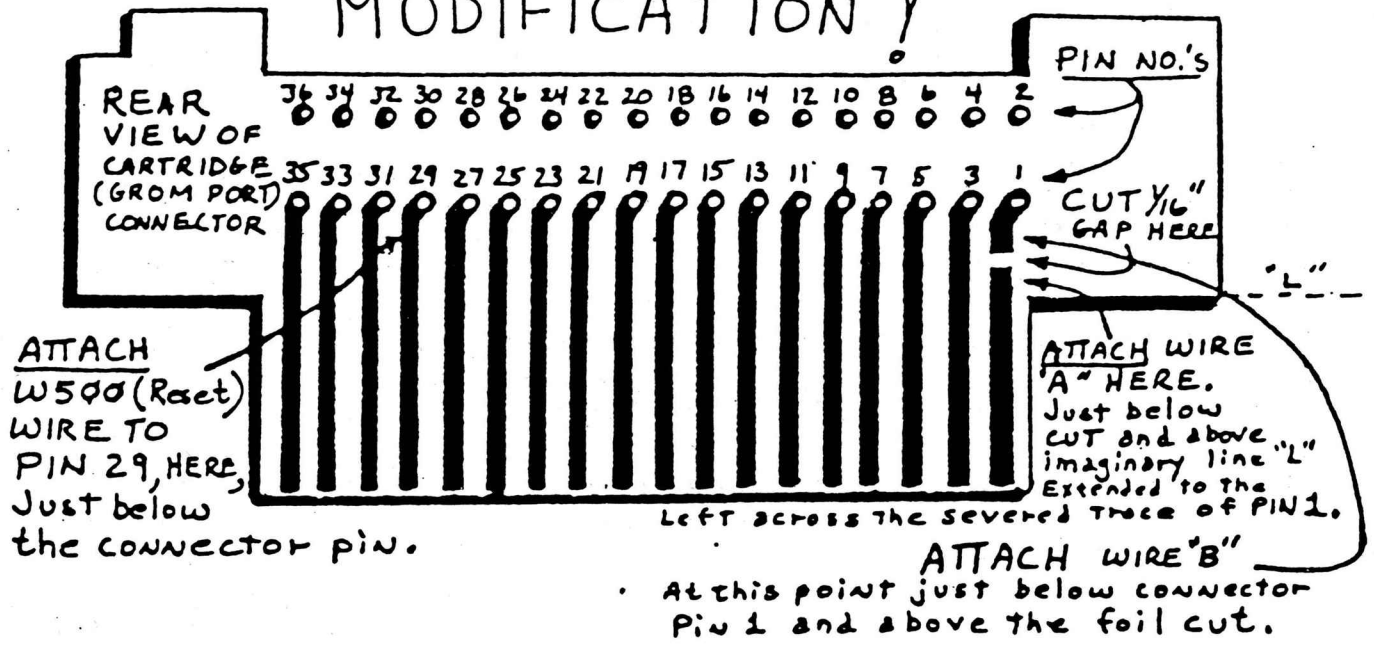
I used a small printed circuit board capable of holding four twenty eight pin DIL sockets which I had to hand. It would be possible to use almost any board capable of supporting the sockets wired in parallel, such as a VQ board etc.

The board is mounted above the ROMOX after removing the EPROM chip and bolted to it with spacers. The only lines which need to be switched are the positive and chip select lines. These are taken to a miniature multipole switch, and chip select and positive are taken to each socket. These lines must be isolated from each other, ie. the connections on socket one must not be connected to socket two etc. All other connections are wired in parallel and connected to the ROMOX board by short lengths of thin insulated wire.

I could only obtain the miniature switch in a six pole version so the two end poles were left unconnected and made "OFF" positions. The other refinement I added was a LED wired to the positive connection on each chip, so you can see at a glance which chip is selected, and a reset switch as per the ZIF modification, the negative end of the LED's is taken to negative via a 680 ohm resistor. Only one LED is on at any one time. I used turned pin DIL sockets which permit fairly easy removal of the chips and aren't as expensive as ZIF sockets. The switch was not glued to the circuit board as were any odd bits of wiring that were lurking about. Of course there is theoretically no limit to the amount of sockets you could employ on a circuit of this type as long as the connections are kept reasonably short. If, however, the board gets too big then the problem of connector wobble in the GROM socket will rear its ugly head, so it could be advisable to use the ROMOX module as a connector only and run a length of multicore cable to a remote socket unit. I would be interested to hear of anybody else's experiences on this type of modification.

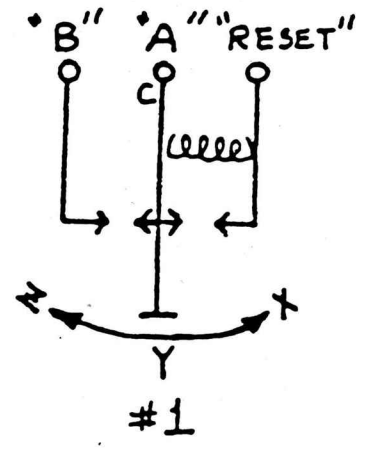


RESET SWITCH, CARTRIDGE INSERT BUT NO RESET MODIFICATION ?

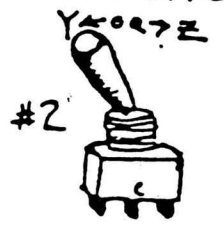


This switch exists but is relatively hard to find. Commonly used in MINI-COMPUTERS DURING THE 60's and 70's. SURPLUS GOOD SOURCE.

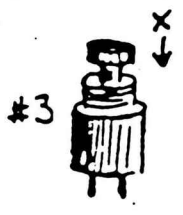
This switch actually has 3 positions:
 Y = Center position, allows for you to insert cartridge into your console with no reset occurring. (useful for cartridge or screen dumps of cartridges)
 X = Push and release allows the console to receive a reset (useful for freeing a locked console, without powering the console off).
 Z = Normal (switch stays in this position when selected) PUTS the cartridge slot back into the mode that T.I. originally designed.



TWO SWITCH OPTION: (IF YOU CAN'T FIND THE SWITCH SHOWN ABOVE)

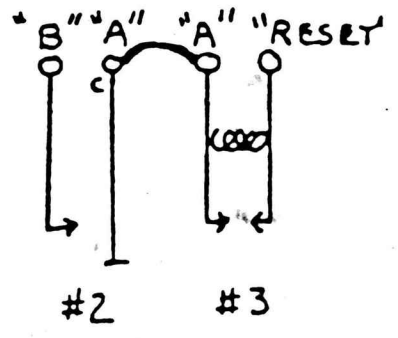


THIS SWITCH NEEDS ONLY TO BE SINGLE THROW, SINGLE POLE. (EITHER AN OPEN OR CLOSED CIRCUIT)



THIS IS A MOMENTARY CONTACT SW. JUST TAP IT TO CLOSE THE CONTACTS BRIEFLY

These 2 switches now can function the same as the single switch above. WIRE "A" WILL BE ATTACHED TO ONE SIDE OF THE TWO SWITCHES.



- FEATURES:
- RESET BUTTON FOR CPU WITHOUT SHUTTING OFF CPU.
 - CARTRIDGE INSERTION WITHOUT RESET (CARTRIDGE DUMPS).
 - NORMAL OPERATION NOT MODIFIED.

This is the easiest hardware modification I've written about yet. TRY IT, YOU'LL LIKE IT! (Your Responsibility) John F. Willforth (112) 527-6656

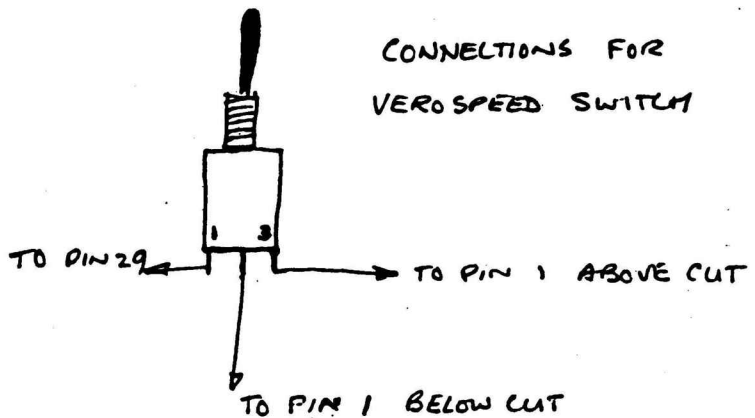
West Penn 99's Club, Jeaneatte, PA

RESET SWITCH NOTES

This reset switch really is the Deluxe version. I've fitted it to two of my consoles and it works perfectly. Debounce components do not appear necessary so I assume the line is debounced within the console.

Contrary to our American cousins the specified switch is easily available in this country ex stock from Verospeed stock no. 63-382296 at £1.61 each plus VAT post free.

Verospeed,
 Stanstead road,
 Bayatt wood,
 Eastleigh,
 Hants, SO5 4ZY,
 phone, 0703 644555



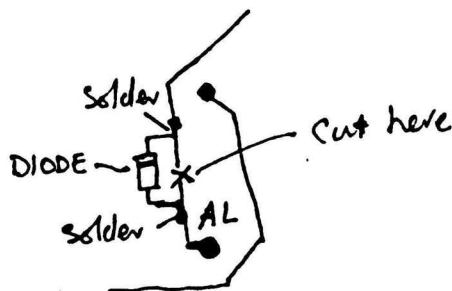
Mike

ALPHA LOCK MODIFICATION UPDATE

Mike Goddard Feb 1988

I recently fitted a diode to one of my keyboards to remove the alpha lock problem as per Geoffrey Coan's article, and it really worked a treat. So, hot soldering iron clutched in my hot sticky little hand, I gleefully attacked my second console, only to find that the keyboard was entirely different on the wiring side. After much shaking of the head I decided to consult Geoffrey who said quite wisely "hmmm a futaba type keyboard, oh, never heard of it! So after some discussion I decided to attack the dreaded futaba and diode its alpha lock whether it liked it or not. I discovered to my astonishment that the track to put the diode into is in fact marked A L on the back of the circuit board. The diodes should be inserted as shown in the drawing with the black band facing away from the A L marking, after making an incision in the track. The diode I used was a 1N4148 which is easily available from most suppliers for the princely sum of about 5P each.

Happy Soldering, Mike.



FUTABA TYPE KEYBOARD DETAIL

MINIMEM Conversion to Rechargeable Battery. Send MINIMEM and Crossed cheque £7.50. To: N.J. PETRY, TENSAL TECHNOLOGY, 15 PENRICE CLOSE, WORLE, W.S.M., AVON BS22 9AH.

SWOPS SWOPS SWOPS SWOPS SWOPS

I WISH TO EXCHANGE 3 COMMAND MODULES: TUNNELS OF DOOM, MIND CHALLENGERS, AND ADVENTURE, FOR ANY SPORT COMMAND MODULES AVAILABLE, LIKE SNOOKER, FOOTBALL AND TENNIS OR ANY OTHERS. I AM INTERESTED IN A STRAIGHT SWOP IF THAT IS POSSIBLE. PLEASE WRITE TO: ROBERT R WILSON, 25 MUIREND STREET, KILBIRNIE, Ayrshire, SCOTLAND KA25 7DG.

FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

A Printer Interface for the TI99/4a computer. Plugs into right-hand side of the console and powered from the 5v supply of the computer. Drives a parallel port printer and includes a BBC type printer lead. Price £15. Apply: Mr. R. BROWN, 27 CHILTERN DRIVE, WESTMOOR, NEWCASTLE ON TYNE NE12 ONU.

FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

TI99/4A £30, EXPANSION BOX £80, 32k RAM CARD £60, SPEECH SYNTHESISER £25, VARIOUS MODULES INCLUDING EXT. BASIC £35, CHESS £12, T:E.2 £12, MANY GAMES, WRITE OR PHONE FOR DETAILS: JOHN DOWNIE, 36 LANCASTER AVENUE, BEITH KA15 1AR, TEL. 05055 2965.

FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

Modules: Video chess, touch typing tutor, Personal Record Keeping. £15 for all three. APPLY: T.J. COOPER, 9 THE ACRES DOWNLEY, HIGH WYCOMBE, BUCKS HP13 5NR.

FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

Hardware: Console with 32 K, Power Supply, Modulator and Cassette Cable £50
Printer interface (Epson) £30 Joystick Adaptor £ 5

Work Station: Purpose built, fan, power supply etc. £20

Modules: Various £5 to £25. Enquire

Tapes: All £2 each.

Books: T1-MES 1-19 £5 T1-Lines Various £5ⁱ
Technical Data Manual (T1) £5.

ROGER MACEY, 9, Cubleigh Close, Moulton, Northampton, NN3 1GF.
TELEPHONE: 0604-46194



USING & PROGRAM
TI-99/4A
INCLUDING READY-TO-R

TI-99/4A
GAME PROGRAMS
BY FREDRICK HOLT

Sprites, A Turtle,
and TI LOGO

MASTERING
THE TI-99/4A
Peter Brooks

THE TEXAS
PROGRAM
BOOK

TANTALIZING
FOR TI-99/4A
HAL REMICK

NEW

JAN

TI PROGRAMS
TI STORE OF LIFE