TI*MES U.K./UG

S.O.M.
6

TI*MES

Autumn 85 ISSUE NUMBER 10

AUTUMN 1985 NUMBER TEN

CONTENTS

INDEX.

Birmingham city centre
54 Museum of Science &
33 Power House Disco
64 University of Aston in Birmingham A3
Birmingham Convention & Visitor Bureau Tourist Information Centre
Car Parks
One Way
Pedestrianised

Nationwide
TI99/4a USERS SHOW
26th October 1985
CIVIC HALL DIGBETH
BIRMINGHAM.
doors open 10.00am
Admission by ticket free to members.

THE THIRD NATIONWIDE
TI99/4a USERS GET TOGETHER

Welcome ______________ to TI99/4a EXCHANGE

TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES
AUTUMN 1985 NUMBER TEN

40,Barrhill,Patcham,BRIGHTON,East Sussex,BN18UF.Tel:0273 503968(evenings)

THE THIRD NATIONWIDE TI99/4a USERS GET TOGETHER

Birmingham

In this issue we give news of the third Show. Yes it is in the heart of Britain. BIRMINGHAM often hosts the best shows and this one will be no exception for TI99/4a enthusiasts. Those of you who have supported previous shows will know that it is good to get together and get to meet other TI users.

You will meet and talk to many of the group writers. See some of the latest things happening to the TI console in the way of hardware/software available from the two 100% TI dealers. Of course many books will be on sale too. You can really get alot of comfort that there is alot going for the TI 99ers.

You are invited to join in the fun of the day, if you wish to help at the show then drop in at the Group stand. Everyone will pick up a bargain for Christmas, or sell that unwanted cartridge etc in the grand TI99 auction. There is a chance to win a valuable prize in the TI*MES draw. The TI OXON club will be there with PETER BROOKS who will show you a trick or two as you get to master the TI99. At the last show we were sorry not to have STEPHEN SHAW, this time he will be there to get you started.

The Civic Hall, Digbeth is not far from the main rail station, the National coach station is on the doorstep. If you come by car you will be amazed at the ease of the drive to the city. Your support will ensure that these shows continue, so come along and really make it your day.

In this issue ......

we feature articles of a very high standard in editorial content. We hope that it covers something for everyone, if you disagree then write and say so. In this edition we give tribute to our very close friends in a distant land of AUSTRALIA, as can be seen from this issue they have contributed some outstanding work which will benefit all TI 99ers around the world.

The happy band of the TI99/4a world extends to many many TI users groups throughout the USA and Canada, all working night and day to ensure that your TI is here to stay.

P.C.W. Show ....

I went along to the computer show of the year, nothing could be seen of Texas Instruments, but that did not deter a number of 99ers from going. Why, well the TI can always be given a modem, a monitor what about a new printer or a drive, it was interesting also to see what was new in the computer world. I went away very satisfied that our TI will live on a few more years yet.

See you in Brum........

Clive

Clive Scally

TI994a Exchange TI*MES newsletter is supported only by its subscribers. This TI users Group is INDEPENDANT of Texas Instruments and is completely non profit making. TI*MES is published quarterly, JANUARY, APRIL, JULY, and OCTOBER months.The annual subscription is £6 and includes 4 newsletters. Editorial etc is provided by group members, other user-groups and other related sources. Views expressed are those of the writer and not necessarily those of TI994a Exchange. Whilst efforts are made to ensure accuracy no responsibility can be accepted by TI 994a Exchange as a result of the applying of such information found within the pages of TI*MES . You are invited to contribute copy for publication in TI*MES. If you would like to make a contribution please submit copy on A4 only this MUST be typed with a disk or tape if a program is included (better still use the word processing program in this issue). Unaccepted material will be returned ONLY if accompanied by a S.A.E. The editors reserve the right to refuse advertising.

RANDOM EYES

Graham Baldwin.

In the last issue I mentioned Enhanced BASIC in passing, and the fact that it can be obtained from the Personal Record Keeping and Statistics modules. Looking back through previous issues of TI*MES, this somewhat obscure subject doesn't seem to have been mentioned often, so, prompted by one or two enquiries about it, here goes.

When a PRK or Stats module is in the cartridge slot the normal menu appears after the title screen, ie. 'Press 1 for TI BASIC, 2 For Personal Records'. On pressing '1' we obviously get TI BASIC, BUT with some additional sub-programs, prefixed with CALL, as usual, that give us Enhanced BASIC. These sub-programs enable us to set up PRK type-files from TI BASIC, giving somewhat more flexibility (and a great many more headaches) in programming and use than 'pure' PRK files allow, albeit with some restrictions.

The first sub-program I'm going to look at, and probably the easiest to get to grips with, is CALL D, the 'D' standing for DISPLAY. Yes, like Extended BASIC, we can actually display information anywhere on the screen (well, almost anywhere) without using the TI BASIC 'print at' simulation!

The syntax of CALL D is as follows:-

```
CALL D(ROW,COL,WIDTH,STRING/NUMERIC)
```

ROW is obviously the screen row, with a value of 1 to 24, top to bottom.

COL is the column value, from 1 to 28.

WIDTH (or size) is the field width of the information to be displayed, to a maximum of 28 characters. A positive value clears the row before the information is displayed; a negative value does not. (This should sound familiar to Ex BAS users, with their SIZE clause.) Although the width is limited to 28 characters (don't bother specifying anything greater; you won't get it), we can still get more than one line on the screen, again like Ex BAS, by putting more information into one CALL, following on with another ROW, COL etc. after the STRING/NUMERIC item.

STRING/NUMERIC is the information to be displayed, and can take the form of a string or numeric constant, variable or expression.

The next CALL to look at is CALL A, or ACCEPT. We are no longer tied to an INPUT at the bottom of the screen, where we don't always want it; again, like Ex BAS ACCEPT AT, we can take information from anywhere on the printable part of the screen. This is the syntax:-

```
CALL A(ROW,COL,WIDTH,RETCODE,RETV)
```

ROW, COL and WIDTH are as for CALL D
RETCODE is a variable that is given a value according to which key has been used to enter information. (No, we don't have to use ENTER.) The codes returned are as follows:-

1. ENTER was used.
2. A null was entered
3. FCTN 7 was used.
4. FCTN 8 was used.

5. FCTN 6 was used.

6. FCTN 5 was used.

7. FCTN 9 was used.

FCTNs 5 to 9 are the BEGIN, AID etc. keys, the use of which can make a program more user-friendly, and when used with CALL A, save us a CALL KEY routine to find which key the user pressed.

RETV is the return variable that will contain the string or numeric data from the keyboard.

There are some optional extras that will fit into the brackets, the first of which is a Field Number, used when setting up PRK-type records in Enhanced BASIC, a subject I propose to tip-toe away from for now. The second is a maximum and minimum value range check of numeric information entered, taking the form ...LO,HI) after RETV.
By the way, you can't break (FCTN 4) out of CALL A...

There are several more CALLs, the details of which are increasingly obscure and I don't have the time or space to do them justice for the moment. TI used to publish an information sheet about them, and I believe Stephen Shaw can supply a booklet that gives some more gen, for a small fee...

****

Here's a short routine to extract a month number, 1 to 12, when the first three letters of any month are entered. It uses POS, a dazzlingly fast device for this sort of job. Try writing the routine in a different way, without using POS, and ponder the speed difference.

```
100 INPUT "MONTH?":M$
110 M=(POS("JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC",M$,1)+2)/3
120 IF M<>INT(M) THEN 100
130 PRINT "MONTH NUMBER IS ";M
```

You could use a similar routine for word recognition in an adventure program...

I've gained some more information on the Adventure module. Apparently it does not contain a lower-case character set and the other goodies I thought it might; the information to set these up is loaded from tape, along with the adventure itself. Also, there appears to be no way of accessing the module from TI BASIC. Shame...

About the most obscure Ex BAS facility must be the pre-scan selectors, !@P+ and !@P-. These don't even find their way into the Ex BAS manual but appear in a seperate leaflet, entitled "Important Product Information for TI Extended BASIC", usually, but apparently not always, supplied with the module.
When a program is RUN it is scanned to reserve space for variables, arrays, DATA etc., which all takes time, hence the delay before a program begins to execute. By including the pre-scan 'on' and 'off' statements at judicious points in the program we can prevent the computer wasting its time (and ours) looking at lines containing variables etc. that already have space reserved, and thus get our program running a little quicker. By the way, pre-scan 'on' is !@P+ and 'off' is !@P-. An oddity of !@P- is its ability to dump us back into TI BASIC if certain types of error are found on pre-scan, which could be useful in some circumstances, otherwise make sure ALL errors are cleared from the program before using it...

Red herring department again. All variables must start with or consist of a letter, right? Of course - er, hang on. Some non-alpha symbols can be used if you so wish ('@' is one of them). No doubt there's some inscrutable TI logic behind it somewhere...

Happy programming,

Graham Baldwin

RAMBLES

RAMBLES by STEPHEN SHAW

Remember, RAMBLES is based on YOUR comments and requests.... if you don't write to say what you want, I shall write what I want to!!! If you don't find Rambles helpful, write and say why' All enquiries welcome, SAE for a direct reply please.... and nb I am not an expert on hardware or machine code!

Write:
10 Alstone Road, STOCKPORT, Cheshire, ENGLAND, SK4 5AH
Overseas readers please send TWO international reply coupons!

OK... gossip first as usual, serious stuff later...

STAINLESS SOFTWARE.... will be closing down at the end of October 1985, due to lack of interest, as losses now reach towards four digits.... a few programs will be placed into the User Club libraries, but most will merely cease to be (legally) available. Note that in such cases copyright will continue to apply!
If you need anything in the Stainless catalogue, please do not delay!

I shall however continue to maintain a library of public domain/freeware programs available to club members at low low prices. SAE minimum for details please! Contributions of both funds and programs are most welcome: but nothing which is copyright please!

(Overseas groups: we don't have many original programs, but swaps are welcome!!!)

PUBLIC DOMAIN: I understand that there is some misunderstanding about this term. In U.K. law, EVERY artistic creation is AUTOMATICALLY copyright : everything created in the U.K. IS COPYRIGHT.
In the USA however, works by US authors must be registered to obtain copyright protection. If a work is published before registration it is said to be PUBLIC DOMAIN which means that the Public own it and may freely copy it.... something which applies to one of the Star Trek tv series, as Paramount failed to register before transmission! (BUT the series is still subject to TRADE MARK protection!).
In the UK, the closest we can come is OPEN LICENCE where an author, although retaining copyright, authorises anyone to copy his work. He may if he wishes make the consent subject to certain restrictions (eg the work is not sold commercially, or with his name removed).
Similarly, in the USA, the FREEWARE range falls into this category of Open Licence, with the author retaining copyright, and some measure of control.
There are a number of programs circulating which are NOT Public Domain nor Open Licence. Please be aware that copying a copyright program is unlawful even if you do not INTEND to deprive the author of his royalties. If you DO have such an intention.....

It has also come to my notice that some TI owners are merrily copying recent commercial copyright works, such as GRAPHX and INFOCOM adventures. This is not merely unlawful, it is insane: how much longer will authors bring out high quality material if nerds like that copy there work without paying a penny for it? If you should happen to have a hot (or even slightly warm) copy of such programs, please give consideration to purchasing an authentic copy (and thereby supporting the commercial infrastructure which is also still necessary) or at the very least, sending the author direct an anonymous donation for his work.

Is there anyone out there who still does not know what copyright is?

In the Summer issue of TI*MES (No.9) John Rice wrote of some goodies...
p.24, TI Writer Companion... I duly sent of my US$6, and have duly had my envelope returned, unopened, marked "DELIVERY REFUSED", leaving me two pounds down on the deal. Be warned! Dr William G Browning, if he exists, does not want to hear from anyone!
pp27 28: I rang Scott Foresman immediately on receipt of issue 9. Nothing left whatsoever.

The text from Phillip Marsden, in compressed format, seemed fairly legible despite the small size, so if all goes well I shall try it with some of this issues RAMBLES.

I shall also try a little formatting routine from the July LA Group newsletter... you will notice if it works! Comments welcome. Magnifying glasses from Woolworths...

CHOKE. ??? . Encountered in LOGO 2 and undocumented. In connection with garbage collection, followed by system lock out. Any comments anyone? Anything else undocumented?

I have sent some cash on to Craig Miller, and now have the June, July and August 1984 issues of the SMART PROGRAMMER. September 1984 has not yet been published (is this a record) although I understand the intention is to ultimately publish the 12 issues promised, and then HALT! with an occaisional (say quarterly) newsheet available thereafter.

The June SP has considerable console maintenance information, a map of GROM 0 (the GROMs have been rewritten several times! Craig maps those from an early 1983 console), and Italian program to change machine code to CALL LOAD statements for X8 (public domain and available from me), and a Forth program to copy disks in 3 passes using one drive only.

The July SP announces the ADVANCED DIAGNOSTIC program from Millers Graphics for US$20, which checks all memory, including 32k, and has a useful disk utility including a visual indication of disk drive speed. NB: the program is supplied on a disk which is (almost!) impossible to copy.

Also in the July SP are maps for GROM 1 and GROM 2. The Forth program is a general DSR utility.

And in the August SP news that Craig was still waiting for payment from CorComp..., an excellent ISR driven machine code program to run with Extended Basic, which places a digital clock at screen top right, even while you are programming! (I've used it for a Speaking Clock program, accurate to about 6 min per day, tells the time every 10 seconds!!!), a DSR memory map of the TI RS232 card, some general assembly language print routines which use BLWP, and fifteen screens of Forth program for music (Star Trek theme over 6 screens).

A Forth question which I have received more than once... HOW do you return to the title screen, as QUIT does not work!!!
If you have loaded -SYNONYMS (or either Editor or any Graphics mode or -FLOAT) then you have access to the Forth word MON. To return to the main TI screen just type in MON and press ENTER.

During the past quarter, I have received an excellent TI related magazine, called SUPER 99 MONTHLY. Only 16 pages, but good enough for me to send for all the back issues!!! At the time of writing the last issue I have is Issue 11, dated July 1985. Back issues are only US$1.50 , inc pp, so a brief summary
Issue: Summary:
1. General disk loader.Use of ERASE ALL. Transliterate. RES. Tombstone City.
2. Game. Merge. Forth ACCEPT AT.
3. Days between dates. Forth Phone list! Multiplan RPI. Stats.
4. Sound echo. Loading XB Iowmem with m/c. LOGO music.
5. Solitaire Checkers Game. XB Mouse. Speech Test. TIW XB. LOGO pattern.
6. Adding a keypad. REDO. M/c sector addressing. TIW hint.
7. 28col prog lister. Multiplan example. M/c equates.
8. Joystick adapter. A routine to extract a part of an existing program.
9. Graphics on TIW!! Number base conversion. Multiplan SYLK files.
10.Echoing PRINT with TE2 speech. MULTIPLAN SYLK files.
11.Review of COMPANION. PEB technical detail. Latest from Myarc.

SUPER 99 Monthly continued...

Sub is US$26.50 air mail or US$16 by sea mail. Excellent value...
SUPER 99 MONTHLY, Bytemaster Computer Services,
171 Mustang Street, Sulphur, LA, USA, 70663.

Super 99 is 16 pages thick, has no ads, and prints original material. It is also a commercial, copyright, publication!!!

The July copy of LA99ers Topics tells us about the TI99/9 from dealer TexComp... which turned out to be a genuine TI99/4A inside an IBM PC casing. I understand that TexComp are not everyones friends....

Two more UK machine code programs came in... and neither would work on my all-TI rig!! I could start worrying about this....

SXB Suppliers JKH Software, have now moved
4911 S. 31st Street, ARLINGTON, VA, USA. 22206

I now have a copy of the Forth Manual relevant to Vn.4.5 (the last) with the correct description of MCHAR. The Manual is on disk... or to be exact, FIVE disks.

MEMBER DISK LIBRARY:
I have a huge number of public domain programs available, far too many to list here. There are some truly powerful utilities as well as the usual games, music and so on. Prices most reasonable... see above for how to get details.
Some of the more recent additions include FUNLWRITER Vn2.1 which not only QUICKLY loads TI Writer from XB but adds quite a few improvements over the module loader. Has to be seen. EVERY TIW owner needs it!
Utilities to save Adventure tapes to disk, to permit MiniMemory to save to disk (and then load from disk!), and lots lots more.
New programs welcome. This library is operated as a non-profit service, and any surplus funds are used to obtain new or revised programs (it can be quite expensive!).

FORTH: No requests for more Forth but a request for less, so you're outvoted this time! Five Forth screens are printed elsewhere for you to experiment with.
To enter the screens, select a bunch of unused Forth screens, lets say the first is Screen N. Type N CLEAR N EDIT enter the printed text, use BACK and FLUSH. The two screens of CALL SOUND(3 voices) are quite separate from the one screen CALL SOUND for one voice. Details in next issue if there is any demand for help and assistance!!! Otherwise, no more Forth unless ye asks for it lads!

REQUESTS FOR ITEMS TO APPEAR IN RAMBLES WANTED!

SLIPPED DISKS....

Your most valuable data disk suddenly won't load, and when you try to look at it with Disk Manager, back comes the answer DISK NOT INITIALISED.

You have ignored all the good advice given you to always keep back up copies of valuable material. What do you do?

After wiping the blood off the wall, and retrieving your console from the garden....

The Disk Manager module will tell you a disk has become uninitialised when Sector 0 has been corrupted or damaged... and with a bad sector zero the disk is no longer available to your system. Or ist it?

The disk system is either remarkably silly or very clever, depending on which way you look at it... as you can very easily recover a disk damaged in this manner.

IF Sector Zero is merely corrupted but physically undamaged, go on to Step B below, but if Sector Zero has been physically damaged and is incapable of being recorded on, start with step A. (How do you find out? Try Step B: if it doesn't work come back to Step A!).

STEP A: Take a blank initiallised disk, and using a sector copy utility (see this copy of Rambles!) copy all the sectors EXCEPT SECTOR 0 from the damaged disk on to the new disk. The new disk should have been initiallised with DISK MANAGER and have a valid disk name! Now go to Step C...

STEP B: Take ANY valid disk initiallised with Disk Manager... it does not matter if it has programs on it or not! Copy Sector 0 from the valid disk onto your corrupted disk. Now go to Step C...

Step C: You now have a disk which enables you to access and catalogue the files on it, but it reports an incorrect total number of sectors used/free.

Let's make the system put this right! Use disk manager module to make a back up disk of the 'incorrect' disk, and the backup will magically have the right details.

Now junk the original disk you had problems with, and make a security back up!

∼∼∼

In response to a news item in a US magazine I sent US$9 to the LA Group for a collection of FORTH articles. These comprise 3 volumes, grand total 43 sides of print, collected from various US user group magazines.

There is a lot of duplication of course, and some of the material has already been passed on to you in TI*MES.

They have kindly given us consent to copy for our members - but at my copying costs this would be 6.75! And some of the originals are not very good copies to start with, being blurred or faint.

Brief items of interest:

Unhappy with default colours for text screen? They are set in line 9 of Screen 51. The value 0F4 is white on blue.

There is an explanation of the ISR Forth Clock by Rick Mirus, on the Forth Screens 1 disk from me. The value 59 on line 2 is the counter limit, and applies to the US 60 Hz system. In the UK 49 is correct (50Hz).

The value of 47 on line 6 is an 'adjustment' to fine tune the clock by adding 470 milliseconds to it every ten minutes!

∼∼

TI BASIC EFFICIENCY...

Some detailed benchmark timings in PCW July 85 led me to take another, closer look at the workings of our machine...

FOR...NEXT loopings are faster on the TI than on the Spectrum....

A(4,5)=B(6,7) is faster on the TI than on the Commodore 64

X=5^2 is faster on the TI than on the Spectrum, the Commodore 64 or the Apple 2. In fact, Extended Basic is ten times faster than the Spectrum!

CALL CLEAR is faster on the TI than on the Spectrum or Commodore 64.

Forgive me for not listing all the items for which the TI is slower...

Comparing TI Basic with Extended Basic, in a short benchtest program it appears that XB is slower, but XB is much faster than TI Basic for such things as SQR,^, ABS, LOG and SIN. ExBas string operations are faster... a simple A$="A" running three times faster in XB.

I also took a look at use of DATA in programs, to try to settle the old chestnut of WHERE in a program DATA should go...

As you might expect the answer is not so simple. It depends on whether or not your program uses RESTORE.

With DATA at the start of your program, RESTORE is faster but READ is slower, while conversely if your DATA is at the end of your program, RESTORE slows down a lot but READ is much faster. NB: this MAY differ on consoles with differing operating systems!!

On average, how many RESTORES do you process, and how many READs?

If the number of RESTOREs TIMES FIVE exceeds the number of READs, place your data at the beginning. If the number of reads is more than five times the number of RESTOREs, place the data at the end. The difference in the timing of RESTORE in a 10k program is unbeleivable!

!!

TIW FORMATTER: Do you wish to prevent the FORM FEED at the END of printing? Make the last line of your letter etc: .PL 1 and the form feed is suppressed! NB: Do not forget to reset PL if you have another document to print!

(Thanks SUPER 99 MONTHLY for that excellent tip)

==

Magazines have now arrived from the UK Forth Interest Group and have been used as packing materials - nothing at all of use. Be warned.

MICROpendium continues, but the sub is now higher than last advised:
AirMail is now US$35.00 and sea mail is US$21.50 still.
MICROpendium, P O Box 1343, Round Rock, TX, USA, 78680

The June issue advised that scrap TI modules were buried in Providence, Rhode Island. Also in the June issue is a DIY article on changing your now excess Editor Assembler module (...now available on disk for ExBas!)... into an 8k battery backed RAM!

FORTH AGAIN... let's see this is issue ten?! Back in issue EIGHT I suggested how you could set up TI FORTH to load quickly using BLOAD. Richard Owen has just been in touch to point out a vital step missing from the top of page 46 (issue 8 that is!).

Having FLUSHed the new screen 3, the new word PAGE is out there sitting in the amended screen three, but it is NOT in memory! So the next bit (' PAGE .) will produce an amusing ?.

Having flushed screen 3, EMPTY-BUFFERS then 3 LOAD. You now have the word PAGE in memory and can use ' PAGE .

(((I can only correct my errors if you tell me you have problems!!!)))

Manorgrove had a sizzling Summer Sale and dropped the price of their dusty TI99/4A to eighty pounds....

REVIEW: TIGERCUB DISK: NUTS AND BOLTS: Wow. 100 utilities!!! (well, one or two tips / demos in there... but still!). You know that CALL CHARSET leaves the lower case letters with their space invader definitions! Jim has provided us with CALL CHARSET2... and several typefaces to choose from as well. There are so many useful utilities on this disk it is impossible to do a full and proper review in less than ten pages (it's OK Clive I'm not going to!). Very well worth the US$20 Jim is asking, and you will be supporting one of the User Groups' most helpful supporters in so doing. If you use XB, and have a disk system, buy this disk.

Earlier I mentioned piracy of a program called GRAPHX. This program is Australian and is sold by Parco (who I notice has a new address:
2 Devonshire Court, Heathpark, Honiton, Devon. Same phone).

Now... GRAPHX is a hi-resolution sketching program. There now seem to be lots of them about... but GRAPHX does stand out above the rest, with a superb manual, and 120% for ease of use. It has helps to assist with the odd VDP restrictions (which every other program ignores!) and of course the obligatory disk and printer saves. Parts of the picture can be saved or moved, and there is even a form of animation capability! The manual includes technical information for machine code programmers to enable them to use GRAPHX files in their own programs. Very highly recommended. If you buy just one sketching program, buy this one.

DRAW A BIT is an older hi res program, and comes in two versions, DAB1 and DAB2. DAB2 is more advanced but unfinished and of course harder to use. The manual supplied is a trifle ropey, and together with the appalling demo screens very nearly put me off progressing with the programs.... which would have been a pity. The two programs are quite different and each has its strong point. Ease of use is not too good, especially with DAB2! but with practice you can quickly come to grips with them.

How to compare these three hi res programs? GRAPHX scores highly on manual, demos, and ease of use. DAB gave me problems with PRINTING- I was able to amend DAB1 to work with my Epson FX80, but still cannot print with DAB2.

DAB allows storage of actual drawing sequences, as well as storage of finished pictures, which can have its uses. DAB2 uses an automatic fill routine, which can be messy if there is a break in the outline, but both DAB1 and GRAPHX go for a semi-automatic FILL so the damage is contained!

DAB2 has many more features than GRAPHX including texture painting and more flexible addition of text to pictures, but lacks the ability to slide parts of the picture around.

DAB is cheaper! and is sold by Arcade Hardware. A serious graphics artist would need all three programs, because of the different effects each offers. And be advised: in hi res mode, the TI is capable of some astonishing pictures. Sample of GRAPHX follows. And do not forget SUPER SKETCH (and printer utility DFX Screen Dump)- Super Sketch may offer fewer utilities, but is handy for the younger sketcher or for a quick free hand sketch.

Dab is written by Dominic Melfi....

Also by Dominic Melfi (and from Arcade) is a THREE GAME DISK containing KIPPY's NIGHTMARE (reviewed on p 18 of TI*MES 3), SPACE STATION 1 (reviewed in TIdings) and BANG BANG SUB. The latter puts you in command of a submarine being attacked by air and by sea. In all three programs the level of difficulty gets harder and harder and harder.... and in this package (twenty quid for three machine code programs!) do offer super value for money. All three games are quite playable - although they will be preferred by speed freaks! And I defy ANYONE to score over 300,000 on any of the games!

Oldtimers may recall CHRISTINE COMPUTING LTD., operated by Ian and Christine Godman down in Watford. Congratulations to the couple on the birth of BARBARA ANN on 1st April 1985 at 4.50am!

A discount mail order price list for TI99/4A software has fallen into my hands, from RAMTOPS, Levenshulme, Manchester. The catalogue has a large number of titles for the TI99/4A... looking very much like a list of titles from an old Stainless Software catalogue...

At no time has Stainless Software sold goods to this establishment, and our spy calling in the shop found no stock available. If anybody has sent money and had nothing back, please contact the Official Receivers Office in Manchester.

A receiving order was made against Mr D Barrett trading as RAMTOPS on 18th July 1985, under reference MANCHESTER 25 of 1985.

After all that chat earlier on about copyright, how come I'm distributing copies of Editor/Assembler to load with MiniMemory and Extended Basic, when the Editor/Assembler files have not been officially released???

It's an excellent question and one which has cost me some very heavy thinking! The TI Writer and Multiplan were freely released in amended versions. I can't help but feel that if TI had had amended versions of Editor/Assembler they would have released them. Certainly, they dropped the price of the package to extremely low levels compared to TI Writer.

I am very well aware that technically I may be in the wrong... but this is the ONLY program known to be copyright I'm sharing!!! I have actually written to TI in the USA twice on the subject, and after over a year, I'm still waiting for a reply.....

Two disks available from the user library...

1. EDITOR ASSEMBLER to load from ExBas with several utilities including an excellent dissassembler (in French!). Disk from Italy.
2. Editor Assembler AND TI Writer to load from Mini Memory.
program from Belgium.

DISK SECTOR ACCESS

FROM BASIC

DEBUGS...

Being corrections to Page 13 of TI*MES Issue Nine (Summer):

The listing on page 12 of issue 9 was, as stated, taken from a working program, and in good order'

However the listing on page 13 wastyped in and had a tiny bug in it if you were using Mini Mem or ED/As, and a huge bug if you were using ExBas...

As shown on page 12, C must have a value of 0 or 255, and line 200 on page 13 does not provide those values' Somehow an = has become a -.

For all versions of this listing please adjust line 200 to read:

```
200 C=-255*(C=1)
```

With this small adjustment the Mini Memory and Editor Assembler versions wil work!

For those of you struggling with the ExBas version, one typo and one omission:

Line 280: the adjustment at the bottom of the page merely repeats the Mini Mem version and is incorrect. For ExBas line 280 should be:

```
280 CALL LOAD(8196,63,248)
```

There is a more serious omission though...

Using Editor Assember or Mini Memory modules, the user has access to Device Service Routines - eg the subroutines in the peripheral roms - via DSRLNKs. Unfortunately, Extended Basic provides no such luxury.

We CAN write a DSRLNK for Extended Basic, but that would involve a bit more code than our simple little program here needs.

Therefore below you will find an adjusted listing, for ExBas, together with the equivalent source code for lines 255 and 260 to the right.

This routine directly accesses the disk controller card, and therefore the routine will work ONLY with the TI Disk Controller Peripheral Card.

GENERAL PURPOSE DISK UTILITY

FOR EXTENDED BASIC.

```
100 CALL CLEAR
110 CALL INIT
120 PRINT "DISK UTILITY"
130 REM
140 INPUT "DISK NO:":D
150 INPUT "SECTOR NO:":S
160 S2=-1*(S>255)
170 S1=S+256*(S>255)
180 PRINT "ENTER":"1.TO READ
":"2.TO WRITE"
190 INPUT C
200 C=-255*(C=1)
210 PRINT "RECOMMENDED BUFFE
R AREA":"14800 & OVER"
220 INPUT "BUFFER ADDRESS:":
B
230 B2=INT(B/256)
240 B1=B-(B2*256)
```

CONTINUED........----/

Source code equivalent to lines 255 & 260:

```
      AORG   >3C32
      LWPI   @>3C12
      LI     R0,>3C00
      MOV    *R0+,R1
      MOV    *R0+,R2
LABEL MOV    *R0+.*R1+
      DEC    R2
      JNE    @LABEL
      LWPI   @>83E0      *END LINE 255 here
      LI     R12.>1100   *CRU address of disk
                         *controller=>1100
      SBO    0
      BL     @>5B38      *Address in card
      NOP                *REQUIRED'
      SBZ    0
      CLR    R0
      MOVB   R0.@>837C
      MDV    @>3C10,R11
      B      *R11        *RETURN
```

This is just ONE screen of graphics!

```
250 CALL LOAD(15360,131,76,0
,6,D,C.B2,B1,S2,S1,0,0,0,0,5
7,108)
255 CALL LOAD(15410,200,11,6
0,16,2,224,60,18,2,0,60,0,19
2,112,192,176,204,112,6,2,22
,253,2,224,131,224)
260 CALL LOAD(15436,2,12,17,
0,29,0,6,160,91,56,16,0,30,0
,4,192,216,0,131,124,194,224
,60,16,4,91)
270 CALL LOAD(16376,84,73,42,
77,69,83,60,50)
280 CALL LOAD(8196,63,248)
290 CALL LINK("TI*MES")
300 STOP
```

....

NB: REF/DEF addresses used all assume that no other machine code routine or program is resident!

I am indebted for this information (especially for the m/c routines) to Richard Blanden.

MY APOLOGIES FOR ANY INCONVENIENCE CAUSED... but as at the time of writing, nobody has written to me to tell me of my errors!!!! or to complain of difficulties!!!

.....

And this is an excellent place to discuss EFFICIENT use of your TI disk system... although the TI does use its disks quite efficiently, knowing what it does can help you to obtain the best from it.

When you place a program (say) onto the disk, the controller looks at the disk index for unused space, and places your program there, marking the index accordingly. Although the disk manager will show you your files in alphabetical order, they actually occur on the disk in the order they are recorded.... assuming a blank disk to start that is.

Suppose you have a long file called XXX which you replace with a shorter file, also called XXX? The new recording will go over the old XXX and the space saved will be MARKED as available (actually the data is still on the disk!). Subsequently you save a very long file YYY. Part of YYY may be placed in the 'saved' space no longer needed for XXX, and another part of the file may be found after file ZZZ... we then refer to file YYY as a FRACTURED FILE.

The disk controller looks at the index, deals with the first part of the file, and then moves the drive head to the next part of the file. This movement can be thought of as wasted time: it is faster and more efficient if files are NOT fractured.

Now take another disk with files A, B and C, recorded in that order. We wish to amend File A, making it longer... the extra will be placed after File C, and File A will again be fractured.

SO.... how to put files together? When you use the DISK MANAGER to copy a disk using BACKUP DISK and copy to a blank disk, the module/controller will record the new disk with entire (unfractured) files, placed on the disk in alphabetical order.

To find out if files ARE fractured you can use one of the commercial disk utility programs (such as Arcade Hardware sells) or you can use the Extended Basic loader for TI Writer (FUNLWRITER Vn1.2) available from me (£4.00).

Efficiency hint number one: Fractured files are best healed!

You have noted that when copying files with BACKUP DISK the files are placed onto the new disk in alphabetical order?
This may not be the best thing for us!!! Suppose our disk has a LOAD file which is used frequently? The best place for a file used often is at the beginning of the disk, while rarely used files are best at the end of the disk.
We can place files just where we want them by taking a blank disk, and using COPY FILE to place them onto the copy disk in the order we want.

As an example, let's look at MULTIPLAN....
When working with Multiplan, the system makes frequent reference to the file called OVERLAY, while files MPCHAR, MPDATA, MPINTR and MPBASE are used only when you start using Multiplan.
Your disk system can use Multiplan more effectively then if the files are copied in the order OVERLAY, MPHLP, MPCHAR, MPDATA, MPINTR and MPBASE.

Efficiency Hint Number Two: Files are stored on a disk in the order recorded: the further the files are from the disk directory the longer the 'head seek' times.

If you use any of the SECTOR COPIERS now available (including TI Forth!) the copy disk will be identical to the master: fractured files will NOT be repaired... but if you have manipulated file order, that order will be retained.

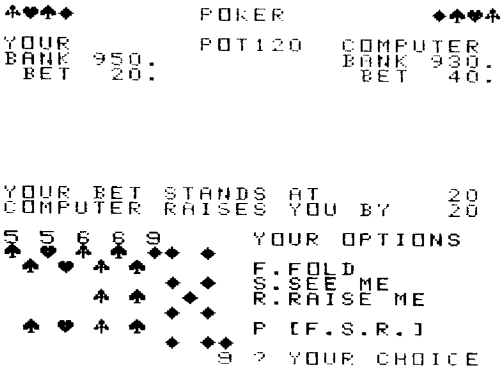
Having mentioned MULTIPLAN.... are you entirely happy with it? There are some odd disk restrictions. The module looks for a disk called TIMP when loading Multiplan, and there is little we can do about that....
BUT the disk program also imposes restrictions, and we CAN change that, using any sector access utility... it helps if a FIND or SEARCH facility is available to help us locate the necessary sector! Fortunately, the amendable parts are all together...
The information is in the following order

DEFAULT DATA DRIVE: Originally set to DSK1.
OVERLAY FILE NAME: Originally set to DSK.TIMP.OVERLAY
MPHLP FILE NAME: Originally set to DSK.TIMP.MPHLP
PRINTER DEFAULT: Originally set to RS232.BA=300

I suggest you FIND "BA=300" to locate the track that needs amending...
on the next page you can find a copy of the disk sector before and after I had amended my copy. A suitable program to do this can be purchased from Arcade Hardware. NB: DO KEEP A COPY OF THE ORIGINAL MULTIPLAN DISK!!!!

Note that the data may be located in a different PART of the sector on your disk. but it will always be in this order. Also NOTE that each of these four entries is preceded by a single byte length indicator: if you change the length of the entry, be sure to amend the length indicator
.

On other machines, ready set-up Multiplan spreads are available for a number of uses: if any TI owner out there is using Multiplan, do you have blank spreads to share? I understand that in the USA Multiplan is being used for some very strange things, including storage of machine code utility blocks!



```
♣♥♠♦          POKER          ♦♠♥♣

YOUR          POT120    COMPUTER
BANK  950.              BANK 930.
 BET   20.               BET  40.


YOUR BET STANDS AT          20
COMPUTER RAISES YOU BY      20

5 5 6 6 9         YOUR OPTIONS

                  F.FOLD
                  S.SEE ME
                  R.RAISE ME

                  P [F.S.R.]
             9 ? YOUR CHOICE
```

Picture at left printed simultaneously with this text using TI WRITER!!!

Does anyone still think that TI Writer is a limited editing program?

It does enable you to do some interesting things, no?!!!

The above screen dump may seem a little strange, perhaps out of place... why is it there? Two reasons!

1. One of the first Extended Basic programs to come my way was a Poker program (check out the October 1982 issue of TIdings!) and since then quite a few versions have passed my way. So it takes a good program to sneak its way into my collection at this late stage! And the program illustrated above IS very nicely done. It is by one of our senior members, Ron Johnson, and can be found on a disk available from me called GAMES-1.

2. This issue of RAMBLES is an innovative one.

a. The double column pages were printed in a single pass.... admittedly there was some preparation required, but the computer did all that!

b. The Forth screens have not been glued in as previously, they also have been printed as one with the text, from files saved directly from the Forth disk.

c. And above, perhaps the ultimate: TI Writer GRAPHICS. Yes, this page has been printed ONLY by the TI Writer formatter, in a SINGLE PASS.

Now... there are some VERY practical points to this: no, it's not a very fast dump of screen graphics, but...

Suppose you ran several businesses... you could design on screen a separate letter head for each one and save it to a TIW file.... and then when writing your letters, you could quite easily use the simple TIW command:

' .IF DSK1.COMPANY1 and the letter heading for Company One will duly appear!

Not bad at all eh???

The screen dump was produced using a program in SUPER 99 MONTHLY, and as such it is copyright! If you would like to have this power, back copies of the magazine are just US$1.50 each including post and packing to the U.K.

The issues to ask for are:
May 1985 (Volume 1 Issue 9)
and June 1985(Volume 1 Issue 10)
Address earlier in this Rambles!

This page was produced on an EPSON FX80 printer. Apart from the text to the right of the actual screen dump, most Epson compatibles should be able to manage it.

Details of changes needed for use with Gemini and Panasonic printers, or any printers compatible with THEM (principal difference is that the Gemini has a left margin command,CHR$(108) which Epson does not use)

==

Enquiries are welcome. Tell me what YOU want to see in RAMBLES! If you are looking for something WEIRD (like how to change the screen colour in ZORK!) or some odd program, drop me a line with an SAE - I may have the answer (I do for Zork!) or the question can be aired here! I may have just the program you are looking for... and tell me what systems you have, what you use them for... the more information I have the better use Rambles can be to YOU!

10 Alstone Road STOCKPORT Cheshire ENGLAND SK4 5AH
(Overseas? Please send two international reply coupons!).

RAMBLES ON FORTH

FORTH RAMBLES....

Referring to page 14 in TI*MES issue 9...
an important reserved part of your Forth disk was missing from the list of Screens not to use...
The Forth loading program, which contains the kernel of TI Forth, is contained on Screens 5 to 21.

Therefore, assuming you have coppied your Forth disk as detailed in issue 8 (with a slight amendment in this issue: see main Rambles), you need to copy only the following screens onto a blank disk, and you will have virtually all of TI Forth on the disk, with the remaining screens available for your own programs or experiments. Copy with SCOPY or SMOVE.

Copy Screens 0 to 21: essential system information!
Copy Screens 51 to 61: Our BSAVEd additional Forth material

Now down to business....

FORTH CALLSOUND

a. ONE VOICE ONLY.
Screen:

```
( CALL SOUND ROUTINE FOR ONE VOICE APRIL 85 SJS )
( USES VDP AREA 14336 TO 14345 )
( duration in ms, volume 0-30, frequency CALLSOUND --- )
: SINIT -31747 C@ 1 OR -31747 C! 3 14336 VSBW ;
: FREQ S->F >F 111860.8 FSWAP F/ F->S ;
: BYTE1 DUP 16 MOD 128 + 14337 VSBW ;
: BYTE2 16 / 14338 VSBW ;
: VOL 2 / 144 + 14339 VSBW ;
: DUR SINIT FREQ BYTE1 BYTE2 VOL 20 / 14340 VSBW ;
: SIL 3 14341 VSBW 159 14342 VSBW 191 14343 VSBW 223 14344 VSBW
0 14345 VSBW ;
: PLAY 56 -31796 C! 0 -31795 C! 1 -31794 C! ;
: CALLSOUND DUR SIL PLAY ;
CR ." ONE VOICE WORDS LOADED & AVAILABLE"
```

To emulate CALL SOUND, I have used the information given in TI*MES issue 6, and have placed data into VDP RAM as a sound table, then told the computer to play it.
The CALLSOUND requires three values on the stack, N1 N2 and N3, where N1= duration in milliseconds, N2=volume 0-30, N3=frequency in cycles per second.
eg: 600 2 220 CALLSOUND
The operation of the CALLSOUND is the same as a Basic Call Sound using a NEGATIVE duration: a second use will cut short any sound still in progress.
If you wish to use an equivalent to a positive call sound, you need to test VDP Location -31794. This becomes zero when the sound has stopped. C@ is used to place the value in this location onto the stack.

SINIT tells the computer the sound information is to be stored in VDP RAM. Note the use of C@. We then tell it that we are passing three values.
FREQ took a few seconds to develop, and together with BYTE1 and BYTE2, translates the frequency in cps to the two values the computer wants (see issue 6). We switch the top stack value (frequency) to a floating point number, then input a floating point number. Then we swap these two floating point numbers on

the stack, so that the frequency input is again on the top. Then we divide,and change the result to a single precision number.
SIL is placed at the end of the VDP data to switch the sound off after the input duration.

CALL SOUND with ONE or THREE voices:
 Screens follow...

These screens are a development from the above, and permit you to use either one or three voices. When loaded you may immediately use either one or three:
 ONE VOICE: same as CALLSOUND above: N1 N2 N3 CALLSOUND
 THREE VOICES:
    REQUIRES 7 items on stack, placed in a slightly different order to the standard Call Sound: Time, Volume then Frequency:

```
    T  V1  F1  V2  F2  V3  F3  SOUNDX CALLSOUND
```

Having used three voices, if you then wish to use just one again, you must switch off voices 2 and 3 with V2V3OFF.

To repeat any sound just use PLAY

To alter voice one but leave voices 2 and 3 as they were, just use CALLSOUND
To amend voices 2 and 3 but not one, use SOUNDX on its own: it requires four values on the stack. Then use PLAY to produce the sound.

Here are the screens:

The screen line numbers are omitted from now onwards... but copy to your Forth screen in the same format! Forth screens have lines numbered from 0 to 15, with line 0 usually used for an index comment in brackets!

```
 ( CALL SOUND 3 VOICES SJS APRIL 1985. USES VDP 14336 TO 14351 )
 ( use as Tms V1db F1hz V2db F2hz V3db F3hz SOUNDX CALLSOUND - )
 ( or Tms V1db F1hz CALLSOUND --- )
 : V2V3OFF 161 14340 VSBW 129 14341 VSBW 191 14342 VSBW
   193 14343 VSBW 129 14344 VSBW 223 14345 VSBW ; V2V3OFF
 : SINIT -31747 C@ 1 OR -31747 C! 9 14336 VSBW ;
 : FREQ S->F >F 111860.B FSWAP F/ F->S ;
 : B11 DUP 16 MOD 128 + 14337 VSBW 16 / 14338 VSBW ;
 : VOL1 2 / 144 + 14339 VSBW ;
 : DUR SINIT FREQ B11 VOL1 20 / 14346 VSBW :
 : B13 FREQ DUP 16 MOD 192 + 14343 VSBW 16 / 14344 VSBW :
 : VOL3 2 / 208 + 14345 VSBW :
 : B12 FREQ DUP 16 MOD 160 + 14340 VSBW 16 / 14341 VSBW ;
 : VOL2 2 / 176 + 14342 VSBW ;
 : SOUNDX B13 VOL3 B12 VOL2 ;
        -->
```

that final line (-->) is important: it tells Forth to load the next screen as well.....   see next page for the next screen, which MUST follow on, consecutively, from the above screen ......

```
( SOUND 3 VOICES CONTINUED SCREEN 2OF 2    SJS APRIL 1985)
 : SIL 3 14347 VSBW 159 1434B VSBW 191 14349 VSBW 223 14350 VSBW
  0 14351 VSBW  ;
 : PLAY 56 -31796 C! 0 -31795 C! 1 -31794 C! ;
 : CALLSOUND DUR SIL PLAY ;
CR ." 3 VOICE WORDS LOADED & AVAILALE"
 ( YOUR CHOICE: ENTER 7 ITEMS DN STACK AND THEN USE: )
 ( SOUNDX then CALLSOUND or USE THREE ITEMS ON STACK AND )
 ( CALLSOUND ON ITS DWN. )
 ( IF YOU WISH VOICE 3 TO BE SILENT ENTER ANY VALID )
 ( FREQUENCY AND A VOLUME DF 30 )
 ( USE V2V3OFF TO REVERT TO SINGLE VOICE AFTER USING 3 )
 ( STEPHEN SHAW )
 ( 10 ALSTONE ROAD  STOCKPORT  CHESIRE ENGLAND  SK4 5AH )
```

Here are the words we have made available:

```
SOUNDX      ( n n n n ---       )
CALLSOUND   (   n n n ---       )
PLAY        (         ---       )
V2V3OFF     (         ---       )
```

This is the most flexible way of handling a CALL SOUND routine I could think of, the only thing lacking being a test of -31794, to enable the sounds to be 'stacked' instead of having just the equivalent of a negative duration CALL SOUND. That little extra I leave to you dear reader!

Note that in Forth, CALLSOUND does NOT have a space in the middle of it!

These Sound screens do require that you have the FLOATING POINT routines loaded as part of your Forth vocabulary.

VSBW is the same as CALL POKEV.   C! = CALL LOAD   C@= CALL PEEK

TI BASIC FASTER THAN GPL ROM....

Programmers who are trying to obtain hi res graphics on the TI often make very heavy use of SIN and COS. Programmers in Forth and Machine Code have been disturbed to find that their programs are still pretty slow... here's why...

The GPL routines for SIN and COS are APPALLING!

In testing out various language segments in a benchtest I found that SIN in TI Basic takes more than four times as long as on the BBC or Commodore, and seven times longer than an Apple... and in a graphics program with 500 or so uses of SIN, that makes a big difference!

SIN and COS have been improved in Extended Basic, taking about half as long as TI BASIC, but still far too slow.

Now look at these two TI BASIC programs:

```
100 REM                            100 REM
110 PRINT "*********"              110 PRINT "********"
120 RAD=1.50                       120 RAD=1.50
130 FOR T=1 TO 1000                130 FOR T=1 TO 1000
138 R=RAD*1000
140 P=(((R*R*R/1E6)*(R*R*76/       140 P=SIN(RAD)
1E7-166))+1000*R)/1000000          150 NEXT T
150 NEXT T                         160 PRINT P
160 PRINT P
```

WHICH program ran fastest? And how did the results compare?

That awful TI BASIC program on the left gave an answer of 0.9974625 and took sixty six seconds to run.
The shorter program on the right gave a more accurate answer of 0.9974949 but it took one hundred and twenty five seconds!!!

Running the program on the right in Extended Basic took 75 seconds, but by changing line 140 to:
140 R=RAD*1000 :: P=(((R*R*R/1E6)*(R*R*76/1E7-166))+1000*R)/1000000
the time was reduced to 56 seconds.

NOW... is the GPL ROM version of SIN faster than my TI Basic version?

The long calculation in line 140 is only accurate from 0 to 1.7 radians, when the error is less than 0.7%, perfectly adequate for screen graphics.

If you think the limitation to about 97 degrees is a problem, please remember the shape of a sine wave!! It is possible to use this equation for all angles by splitting them up into the four quadrants and doing a little manipulation with the answer. eg the SIN of 80 degrees is the SAME as the Sine of 100 degrees. In one case (90-10) and in the other (90+10).

Similarly, the sin of (90+180+10) and (90+180-10) have an identical value but with a negative sign...

FORTH SPEECH

The following two screens are based upon the theory detailed in TI*MES Issue 6.
The C! is equivalent to CALL LOAD.
Speech synthesiser required.
Not the best laid out screens but space is at a premium in TI*MES!!!
You only need to LOAD the first screen, the --> automatically LOADs the next screen, after a pause.

When LOADing is complete the computer will say READY TO START.
You may then use any of the defined words in these two screens to generate speech.
For instance, if you key in (or use in a definition):

```
   _I AM _A TEXAS_INSTRUMENTS NINETY #9 #4 _A COMPUTER
```

that is just what the computer will say!

Nothing difficult about these screens. It would be possible to search through the vocabulary for a particular word, and thus open up the full range, but that would inevitably be slower. I leave that program to someone else.!!
Here we are: two screens for speech:

```
( SPEECH WORDS   9TH APRIL 1985  SJS)  CR ." LOADING WORDS...."
: M -27648 C! ;  : SE 64 M 80 M ;
: THAT_IS_INCORRECT 70 M 65 M 72 M 70 M SE ;
: THAT_IS_RIGHT 78 M 79 M 72 M 70 M SE ;
: READY_TO_START 67 M 75 M 70 M 69 M SE ;
: TEXAS_INSTRUMENTS 70 M 73 M 70 M 70 M SE ;
: WHAT_WAS_THAT 73 M 78 M 71 M 71 M SE ;
: NEGATIVE 76 M 77 M 72 M 68 M SE ;
: POINT 76 M 78 M 64 M 69 M SE ; : #1 73 M 64 M 68 M 65 M SE ;
: #3 74 M 73 M 68 M 65 M SE ;  : #5 65 M 67 M 69 M 65 M SE ;
: #7 72 M 78 M 69 M 65 M SE ;  : #9 68 M 70 M 70 M 65 M SE ;
: POSITIVE 67 M 75 M 65 M 69 M SE ; : #0 67 M 76 M 67 M 65 M SE
; : #2 76 M 69 M 68 M 65 M SE ; : #4 71 M 78 M 68 M 65 M SE ;
: #6 72 M 74 M 69 M 65 M SE ; : #8 71 M 67 M 70 M 65 M SE ;
: ANSWER 67 M 65 M 73 M 65 M SE ;
-->
==================================================================

( SPEECH WORDS PART TWO  APRIL 85 SJS )
: ABOUT 68 M 65 M 71 M 65 M SE ; : FOURTH 73 M 65 M 77 M 66 M SE
 ; : AM 64 M 67 M 72 M 65 M SE ; : _A 68 M 78 M 70 M 65 M SE ;
 : WORKING 76 M 75 M 75 M 71 M SE ; : PLEASE 67 M 73 M 64 M 69 M
 SE ; : PRESS 65 M 67 M 66 M 69 M SE ; : PRINTER 74 M 74 M 66 M
69 M SE ; : NICE_TRY 69 M 74 M 73 M 68 M SE ; : NINETY 78 M 68 M
 74 M 68 M SE ; : DISK 77 M 66 M 68 M 66 M SE ; : _ERROR 79 M 78
 M 72 M 66 M SE ; : HELLO 74 M 65 M 69 M 67 M SE ; : MEMORY 69 M
 64 M 68 M 68 M SE ; : UHOH 68 M 79 M 65 M 71 M SE ; : SORRY 70
M 76 M 65 M 70 M SE ; : TRY_AGAIN 79 M 64 M 64 M 71 M SE ; : LEF
T 72 M 71 M 78 M 67 M SE ; : GOOD 70 M 77 M 64 M 67 M SE ; : SHO
ULD 68 M 66 M 79 M 69 M SE ; : YOU 78 M 75 M 65 M 71 M SE ;
: COMPUTER 68 M 67 M 64 M 66 M SE ; : IS 66 M 67 M 74 M 67 M SE
; : IT 74 M 71 M 74 M 67 M SE ; : PROBLEM 73 M 79 M 66 M 69 M SE
 ; : SAID 65 M 74 M 74 M 69 M SE ; : _I 67 M 73 M 71 M 67 M SE ;
 : HELP 65 M 71 M 69 M 67 M SE ; READY_TO_START
====================================================================
```

HACKERS: ATTENTION: GPL NEWS:

How would you like to have a book which listed in full: A machine code dissassembly of the console ROM, a hex dump of GROM 0, and a GPL dissassembly of GROMs 0, 1, and 2? And also gave you information about each GPL word? One snag...example: GPL Op-Code >0B means EXIT and uses Format type 3. It means: Software-Reset, Return "zum Titelbildschirm bzw. Power-up Routine". It is in GERMAN. Still lots of useful data. The price? TEN POUNDS including post and packing from:

THE MICRO WORKSHOP, Hofkampstraat 15, 7607 NA ALMELO, NETHERLANDS. (Sterling payment is OK). ASK FOR "TI99/4A INTERN"

↑ GPL!

FORTH TRIG...from Sweden...

These two screens give you fast trig for your Forth graphics.
The defined words use ONLY single precision Forth numbers and therefore some adjustment to input and output is required.
In all words, DEGREES are x10, eg to put 90 degrees on the stack, put 900 and if 900 degrees comes off the stack it really means 90 degrees.
RADIANS are times 1000, so 1000= 1 Radian.
This may take a little getting used to, but it makes for greater speed.

You also need to define 2DUP and 2OVER. These can be found on page 1 of Appendix C of your Forth Manual.

```
 ( Fast trig LES )                                 BASE->R  DECIMAL
 : D-R ( deg--rad) 17453 10000 */ ;
 : R-D ( rad--deg) 10000 17453 */ ;
 : (SIN) ( rad--n) DUP DUP 1000 */ DUP 76
   10000 */ 166 - SWAP 1000 */ 1000 + SWAP 1000 */ ;
 : SIN ( deg--n) DUP ABS 900 /MOD 3 AND
   CASE 0 OF ENDOF
        1 OF 900 SWAP - ENDOF
        2 OF MINUS ENDOF
        3 OF 900 - ENOOF ENDCASE D-R (SIN) SWAP +- ;
 : COS ( deg--n) 900 + SIN ;
 : (TAN) ( rad--n) DUP DUP 1000 */ DUP 2033 10000 */
   318 + SWAP 1000 */ 1000 + SWAP 1000 */ ;
 : SQR ( ud--u) 2DUP OR IF -1 0
   BEGIN - DUP 1 SRL 2OVER 2OVER DROP U/ SWAP DROP
    1 SRL - DUP 0 > 0=    -->
```

```
 ( Trig cont. )
   UNTIL DROP ROT ROT 2DROP ELSE DROP THEN ;
 : TAN ( deg--m n) DUP ABS 450 /MOD 3 AND
   CASE 0 OF D-R (TAN) 1000 ENDOF
        1 OF MINUS 450 + D-R (TAN) 1000 SWAP ENOOF
        2 OF D-R (TAN) MINUS 1000 SWAP ENDOF
        3 OF MINUS 450 + D-R (TAN) MINUS 1000 ENDOF
   ENDCASE ROT +- ;
 : (ATAN) ( n--rad) DUP DUP 1000 */ 280 1000 */
   1000 + 1000 SWAP */ ;
 : ATAN ( n m--rad) 2DUP ABS SWAP ABS >
     IF 1000 SWAP */ DUP ABS (ATAN) SWAP +-
     ELSE 1000 ROT */ DUP ABS (ATAN) 1571 SWAP - SWAP +-
     THEN :
 : ASIN^ ( m--n) DUP 2DUP 1000 */ DUP 1000 */ 2000 */ + ;
 R->BASE  ;S
```

Very little comment on that one... its for experimentation! Have fun!

!!

CorComp CONTROLLER BUG: I understand some owners have removed their SPEECH SYNTHSIGERS due to system lockouts when the CorComp Controller is connected. This is due to a BUG in the Controller, and can be avoided.

ON POWER UP, the CorComp Disk Controller cards puts up a special menu selection. If you select from the card, the card sets the scratchpad ram to the effect that NO speech synth is connected. If you then (for instance) try to OPEN a SPEECH file with TE2, your console hangs up.

Avoiding Action: If you wish to use speech, instead of selecting from the CorComp menu, press the SPACE bar, and the normal TI colour bar screen will appear. Press SPACE again for normal menu and speech should operate OK.

The MYARC 128k card is not compatible with CorComps PIO outlet on their RS232 card.

RAMBLES STOP PRESS

FORTH NEWS:

Tom Freeman has passed on details of further changes required to TI Forth screen 58 (Sprites):

Screen 58:

Line 9 should be on line ten, and line ten should be on line nine... after this, amend the new line nine to now read:

```
VDPMDE @ 4 < IF SMTN 80 0 VFILL 300 ' SATR ! ENDIF
```

If your version has: INIT ALL SPRITES you may delete these words.

You should amend the heading to: 01NOV84 TSF

Tom disagrees with the previous fix to Screen 59 and suggests that line 9 should read (in part)....

```
   ...>R 8 SLA SWAP 00FE AND OR SP@ ....
```

From an anonymous enquirer who did not send an SAE either...

" I do not have disk manager ... can I use FORTH to format my disks?"

Answer = yes with reservations. TI FORTH on its own can initiallise a single sided single density drive as follows: Load FORTH and menu choices -COPY and -SYNONYMS.

Place your disk to be formatted in disk drive number one.

Type in: 0 FORMAT-DISK DISK-HEAD

When FORTH has finished, the tracks have been laid down on the disk, and a header has been put on the disk for the Console to use: BUT the disk is full of a file called SCREENS, so, in Basic (or Ex Bas), type in: DELETE "DSK1.SCREENS" and you end up with a blank initiallised disk.

One of the Forth Screens disks mentioned above has routines to initiallise disks with a BLANK header: SSSD with one or two drives, and DSSD with two drives (although it could be amended for one). No double density initiallisation programs to hand as yet. Consider it a challenge!

==

Earlier I asked if THE SMART PROGRAMMER held a record for late publication.... the answer is already to hand... no it isn't. Those of you who have been with me a while may recall a mention of THE CUBIC CIRCULAR way back in TIdings... the Postman has just delivered the SPRING 1983 issue!!!!!

The problem is very similar to that with SP: an enthusiastic guru, writing fairly technical material for a small group, tied up with actually making a living. David Singmaster (now Dr. David Singmaster!) is now a series editor with Oxford University Press... and guess what one of the series books is to be... yep, THE RUBIK CUBE!

> > > > > > > > > > > > > > - - - - - - - - - - < < < < < <

If you have FORTHSCREENS1 from me earlier, you may be interested to know that the excellent Music program originated from TI, who used it to demonstrate the console's capabilities! (They didn't have to tell people it was in Forth and that Forth had not been released....!!!). So, it looks like no more from that source! If your copy of BATTLESTAR is incomplete on SCREENS 1 (there are a total of EIGHT screens!) return your disk with return postage for a free rerecording: I now have the missing screens!

~ ~

Some words are RESIDENT and thus we cannot see what makes them work... or can we? Using a decompiler from the Forth Screens Disks, I obtained:

```
:  FLUSH LIMIT FIRST - B/BUF 4 + / 1+ 0 DO 32767 BUFFER
DRDP LODP ;
: QUIT 0 BLK ! [ RP! CR QUERY INTERPRET STATE @  0=
IF ." ok" ENDIF REPEAT ( RP! ) ;
: KEY kEY 127 AND ;
```

(You will not find kEY in the vocab: using it in command mode causes a lock out!).

```
:  ABORT  SP!  DECIMAL 0 ECOUNT ! CR ." TI FORTH" FORTH
DEFINITIONS QUIT ;
```

and so on.... very few of even the RESIDENT words are defined in machine code!

FORTH LIBRARY: From Milwaukee I now have the TI FORTH International Information Centre library: comprising of SCREENS 1 to SCREENS 6 at normal prices plus FREEWARE FORTH DOODLES. Screens 1 is now complete.

SAE for details: mention your interest in the FORTH items or ask for full library list: an additional contribution for paper and printer wear and tear is helpful. Library list also available on disk (for TI Writer) send blank initiallised disk and return postage!

FRI SEPT 13th....

The brothers and company which we reported in issue 9 as having been found guilty at Wigan Magistrates Court of copying Stainless Software programs, appealed against sentence and conviction. The appeal was to be heard on Friday September 13th, but was dropped from the list and a new date is therefore now awaited. News in next issue maybe.

PINBALL!!! One games program I have been looking for for many years was a Pinball program. Just before TI pulled out they announced that 'Davids Midnight Magic' (a pinball program) was to be converted for the TI. (It never made it!). A couple of XB "pinball" programs have been my way but....

I was therefore DELIGHTED to see MICRO PINBALL from Software Specialitles Inc, a super program which I have have been playing for hours and hours recently.... by the way, it is good enough to make you hammer the keys and try nudging the console! (Nudging doesn't work!).

^^continued......>

Pinball (continued...)

Not perfect (no highscore!) but an excellent presentation of a difficult program. Where can you buy one???

Software Specialities Inc, P O Box 3304, Evergreen, CO, USA, 80439.

This super program, which would have sold for about US$50 in a module I am sure, is just US$10, as are their other programs, T I Toad, Burger Builder, and Midnite Mason. Supplied ON DISK ONLY. Please add US$5 to your order to allow for overseas post. My disks arrived in a fairly flimsy mailer, but somehow made it through the international mails in once piece!! (All programs load with XB, and some also with TIW or EdAs). Micro Pinball is awarded the coveted zillion star award.

SEE YOU IN
BIRMINGHAM
Saturday OCTOBER 26th!!

FROM NEWSLETTERS RECEIVED RECENTLY....

LA AUG 85: Does anyone run a TI BB? If so please write to:

Michael P Galeon,
5430 Baltimore Drive #80
La Mesa, CA, USA, 92041

---he is maintaining an international file of TI BBs.

He needs to know:

Title of board, Phone Number, Address (or just town), name of operator, hours of operation. Notes on contents/ specialities etc.

Write or connect with:

Compuserve# 70317,2636 or SCCG-TIBBS (USA)(619)282-3525

CORCOMP CURSOR AUG 85: New products include a stand alone clock peripheral giving time/date availability to Basic, US$81, and a Box Card which includes the clock, a place to plug the speech synth PCB, and a 64k printer buffer, all for US$110 (from TexComp). Also a new small stand alone 32k which does not need a power supply, for US$119. Other products announced include a stand alone Load Interrupt Switch and an adaptor to allow Vn2.2 consoles to use 3rd party modules. NO mention of any further financial problems or any difficulty with the CorComp disk system.

THE SMART PROGRAMMER. SEPT 84 (!). Quote:"CorComp are not out of bankruptcy yet.... from what we are hearing about cards going bad it doesn't look too good. I also understand they are now charging $50 to repair their defective units." "Most of the CorComp cards lose DATA on a double density Read Track (their new design is a little better)." "You can tell when your [CorComp] disk controller is starting to go out when it starts to return NO DISKETTE or NO DRIVE errors or I/O ERROR 06. With their old design this was an annoyance. However, with their new design when this occurs some of our

continued.........)

our/... (cont)

friends have told us that it has erased some of their files off the diskette".

Mr Miller mentions that CorComp have still not paid him for his work on the disk system software, and they will find it difficult to correct some software bugs as he has the source code AND still owns the Copyright. Unless they pay him he sees no reason to do more work for them!

Mr Miller also announces new products: GRAM CRACKER to plug into the module port, allows you to dump module contents to disk OR TAPE and reload, also store in battery backed GRAM (yes, GRAM!). In a typical use the Kracker would be loaded with the module contents of Editor Assembler, Extended Basic, TI Writer. (The LA newsletter above also suggests you can put the disk portions of TI Writer/Editor Assembler in their!). ALL instantly available - AND MODIFIABLE.

You can amend the contents of GROM 0- the console operating system - or amend Exended Basic. Or add your own CALLS and so on. From about $200. Also news of programs THE EXPLORER (on screen display of everything that is happening!) and ADVANCED DIAGNOSTICS which throughly tests RAM and DISK and includes a disk fixer. Technical contents include How To Use BLWP, how to execute CALL FILES(1) in Assembly Language, an overview of the GERMAN ExBas module, a note of how Forth stores its screens (buffer area), some modifications to the Forth 40 col Editor, and a DSRLNK source code.

Speaking of Forth again.... I have seen in a number of Newsletters the following note: When you have typed EDIT 67 and then want to look at screen 68, just press FCTN 4. You do not need to press FCTN 9 and then type EDIT 68!!!

What I HAVE NOT seen is the converse: with 67 on the screen, you want to go back to 66.... answer: just press FCTN 6.

This information is on SCREEN #38, where the codes in VED refer to BASIC FCTN key codes. eg Code 2 (FCTN 4) calls +SCR while OC (=12)(=FCTN 6) calls -SCR.

I have ordered a couple of interesting utilities to report on in the next newsletter, internatienal mails prevailing.... DATABASE 1 from SPC SOFTWARE CO, which allows up to 1000 records per SSSD disk (small records!), interfaces with TI Writer, and has some intriguing utilities! and SPELL 1.0 from SSI (of Micro Pin Ball fame!) a spelling checker for TI Writer. Watch out for the January issue.

Happy Computing!

Stephen Shaw

CASSETTE WORD PRO

..--. (C) By Brian Rutherford.

How you create TI-WRITER files without the expense of expansion...

This small word processor is designed for people who only have the console, (Ext Basic) and a cassette recorder. This is the word processor part,(the Formatter programme is also included for those who are lucky to have expansion and wish to convert the files). The main idea is for you to type in your article for the magazine and save it on tape, you then give a copy to the editor, who has TI WRITER. He then uses the formatter programme to convert your cassette files into files the TI WRITER can read, and print out to.

After you have typed in the programme and run it, the main menu is displayed with all the options plus the words "free 8000". That is the number of characters you can input before the memory is full. That figure is constantly updated as you input text or edit text. Any time you need to know how much more you can type in, you only need to go back to the main menu by typing // at the begining of a new line. You may type in five lines at a time before you need to press ENTER. If you are part way through a word at the finish of the fifth line do not hyphenate the word just carry on typing it when you start the next line. Also if you finish a word or leave a space at the end of the fifth line, leave a blank at the start of the next line, otherwise the formatter will join the two words together. To leave a blank line just press enter to input a null string. It is a good idea to always leave a blank line between paragraphs so the formatter does not get itself mixed up. To change a line with the change line option, the ACCEPT with out AT is used, along the lines of my article in the June magazine. That is you have to move the cursor over the parts of the line you want using the FCTN D key, to enable the computer to read what is on the screen. The other options lead you through as you use them. The options change line, delete line and insert line allow you to opt out of that mode if you realise you have pressed a wrong key. By specifying a text line number that is invalid. i.e. line number zero or a number higher than you are up to. Also change line and delete line show you the line first and ask yes/no, which is another failsafe for you. The change word option allows you an out by typing a null when the prompt OLD WORD is shown. When you save your files the computer tells you how many lines there are to save, I advise you to write that number down on the cassette that you save your file on, as that is the first thing you will be prompted for when you reload the file for any reason. The screen will also show you what line it is saving or reading, so you know where it is up to. Nothing is more irritating with cassette files is not knowing where they are up to. Use tapes with about ten minutes a side if you have long files. If you find ANY BUGS please let me know, as it is not always easy to think of all the possible situations that can go wrong with a programme of this type.

Mini Formatter. for those with expansion...

This is the formatter programme for Mini Word Processor. Not only does it convert the above cassette files into files that TI Writer can read, it will output directly to a printer also. (So those of you who have expansion can use).

I hope you wrote the number of lines that were saved on the cassette, as that is the first thing you will be prompted for when you run the programme. Then after the data is loaded, the screen shows you two choices, 1 Output 2 Exit. Press 1 for output, and the screen changes to look like this:-

Output to
1 Printer
2 Disk
3 Both

If you press 1, PIO is displayed opposite the word Printer as the default. After accepting the default or typing in your printers device file name, the screen clears again. You are then prompted for a line length 10 to 80 characters long. If you input a value outside those parameters the cursor just goes back for you to try again. Next you are prompted for a left margin, if the line length plus the left margin exceed 80, a warning message is shown on the screen, just press any key and the programme goes back to let you have another go at inputing a line length and left margin. The next promp is for page size A4 or quarto, again you only need to press 1 or 2. The programme still prints the same number of lines to a page, only A4 has a larger margin a top and bottom of the page. Then the last prompt is Right justified Y/N. Press Y or N, upper or lower case is does not matter, the screen clears again and the message "Working on it" is displayed. When the printout is finished the programme goes back to the first screen "1 Output 2 Exit" .

When you select Disk output, the default DSK1.FILE is displayed opposite the disk option, press enter to accept the default or change as necessary, again you will be prompted for a line length the same as for the printer output. But that is all, the file is then printed straight to disk in DISPLAY VARIABLE 80 formatt. When that is finished, you are taken back to the output or exit screen again.

The third option of course takes you through both sets of prompts and does both jobs together, But dont try that if you have done a CALL FILES(1) earlier, as it is printing to 2 files at once.

When you type your text in with the word processor, the following commands can now be used.
To tab a line over a number of places type //number of places/ and then the line. ie.
//49/J. Blow
//50/35 Whatsit St.
//51/Kings Cross.
Pressing enter after each line. In that way you can tab your address over to the right side of the page with out having to count the spaces. When using that command in that type of way, due regard for the line length you are going to specify must be taken into account. If the number of spaces you tab over and the length of the text exceed the line length you specify in the formatter programme, the line will be wrapped around into the next line along with many other sorts of funnies. If you were only using the command to indend a line, then the full 5 lines can be typed in. Lastly to centre a line or a heading type /c/ or /C/ and then the line or heading, will centre it. Again take care with the line lengths.

I think those two little commands will make life a lot easier, especialy letter writting, also the tabs are added to the lines befor they are printed to disk as well as the printer. And dont forget the editor is waiting for your letters and articles to start rolling in dont let him down.
Brian R.

(C) Brian Rutherford, Hunter Valley 99ers, AUSTRALIA.

```
(100 REM !~~~~~~~~~~~~~~!
110 REM !     MINI      !
120 REM !WORD PROCESSOR!
130 REM !      BY       !
140 REM !   BRIAN R.    !
150 REM !_____________!
160 OPTION BASE 0 :: DIM L$(
66):: DISPLAY ERASE ALL :: F
OR L=0 TO 12 :: CALL COLOR(L
,16,1):: NEXT L :: L=0 :: L$
(0)="8000"
170 CALL SCREEN(5):: CALL ME
N(L$()):: CALL CH(9,K)
180 ON K-48 GOSUB 200,210,22
0,230,240,250,260,270,280
190 GOTO 170
200 CALL IN(L$(),L):: RETURN
210 CALL CL(L$(),L):: RETURN
220 CALL DL(L$(),L):: RETURN
230 CALL INS(L$(),L):: RETUR
N
240 CALL CW(L$(),L):: RETURN
250 CALL SCREEN(14):: CALL S
F(L$(),L):: RETURN
260 CALL SCREEN(14):: CALL R
F(L$(),L):: RETURN
270 CALL PF(L$(),L):: RETURN
280 CALL SCREEN(7):: DISPLAY
 AT(10,8)ERASE ALL BEEF:"1 P
urge file": :"       2 Exit"
 :: CALL CH(2,K):: IF K=49 T
HEN L=0 :: L$(0)="8000" :: R
ETURN
290 DISPLAY AT(10,1)ERASE AL
L:"HAVE YOU SAVED YOUR FILE
Y/N"
300 CALL KEY(3,K,S):: CALL S
OUND(-10,110,5):: IF K=78 TH
EN RETURN ELSE IF K<>89 THEN
 300
310 DISPLAY ERASE ALL :: STO
P
320 SUB MEN(L$()):: DISPLAY
AT(1,12)ERASE ALL:"MENU" ::
DISPLAY AT(2,11):"______" ::
 DISPLAY AT(4,8):"1 Add text
": :"       2 Change line"
330 DISPLAY AT(8,8):"3 Delet
e line": :"       4 Insert l
ine": :"       5 Change word
": :"       6 Save file": :"
       7 Load file"
340 DISPLAY AT(18,8):"8 Prin
t file": :"       9 Purge fi
le & exit": :"
  free ":L$(0):: SUBEND
350 SUB CH(X,K):: DISPLAY AT
(23,2)BEEP:"Your choice"
360 CALL KEY(3,K,S):: IF S<>1
 THEN 360 ELSE IF K>48 AND K
<(X+49)THEN SUBEXIT
370 DISPLAY AT(24,1)BEEP:"A
NUMBER BETWEEN 1 AND";X :: G
OTO 360
380 SUBEND
```

```
390 SUB LI(A,B,A$,D,L):: DIS
PLAY AT(A,B):A$
400 ACCEPT AT(A,19)VALIDATE(
DIGIT)BEEP:A$ :: IF A$="" TH
EN 400 ELSE D=VAL(A$)
410 IF D<1 OR D>L THEN DISPL
AY AT(23,1):"<<< No such lin
e >>>" :: CALL KC :: D=Ø42Ø
SUBEND
430 SUB KC :: DISPLAY AT(24,
1)BEEP:"PRESS ANY KEY TO CON
TINUE"
44Ø CALL KEY(3,K,S):: IF S<1
 THEN 44Ø
45Ø SUBEND
46Ø SUB IN(L$(),L):: CALL KE
Y(5,K,S):: DISPLAY AT(15,1)E
RASE ALL BEEP:"Type // to le
ave input mode" :: PRINT L$(
L)
47Ø L=L+1 :: LINPUT "":L$(L)
:: IF L$(L)="//" THEN L$(L)=
"" :: L=L-1 :: SUBEXIT
48Ø IF L=66 OR VAL(L$(Ø))<Ø
THEN PRINT "<<< MEMORY OR AR
RAY FULL >>>" :: CALL SOUND(
5ØØ,11Ø,1Ø):: SUBEXIT
49Ø L$(Ø)=STR$(VAL(L$(Ø))-LE
N(L$(L))):: GOTO 47Ø
5ØØ SUBEND
51Ø SUB CL(L$(),L):: DISPLAY
 ERASE ALL :: CALL LI(23,8,"
Which line",D,L):: IF D=Ø TH
EN SUBEXIT
52Ø CALL LC(L$(),D,K):: IF K
=78 THEN SUBEXIT
53Ø LE=LEN(L$(D))/28 :: IF L
E >INT(LE)THEN LE=INT(LE)+1
54Ø CALL KEY(5,I,K):: DISPLA
Y AT(16,1)ERASE ALL BEEP:STR
$(D);"!-":L$(D):: P=1 :: A$=
""
55Ø FOR I=1 TO LE :: P$=SEG$
(L$(D),P,28):: DISPLAY AT(24
,1):P$ :: ACCEPT P$ :: A$=A$
&P$ :: P=P+28 :: NEXT I
56Ø IF LEN(A$)>191 THEN DISP
LAY AT(16,1)ERASE ALL BEEP:"
LINE TOO LONG TRY AGAIN"57Ø
IF LEN(A$)>191 THEN DISPLAY
AT(16,1)ERASE ALL BEEP:"LINE
 TO LONG TRY AGAIN" :: CALL
KC :: GOTO 54Ø
58Ø DISPLAY AT(16,1)ERASE AL
L BEEP:A$ :: DISPLAY AT(24,1
):"Is change OK? Y/N"
59Ø CALL KEY(3,K,I):: IF K=8
9 THEN L$(Ø)=STR$(VAL(L$(Ø))
+LEN(L$(D))):: L$(D)=A$ :: L
$(Ø)=STR$(VAL(L$(Ø))-LEN(L$(
D))):: P$,A$="" :: SUBEXIT
6ØØ IF K<>78 THEN 59Ø ELSE 5
4Ø
61Ø SUBEND
62Ø SUB DL(L$(),L):: DISPLAY
 ERASE ALL :: CALL LI(23,8,"
Which line",D,L):: IF D=Ø TH
EN SUBEXIT
63Ø CALL LC(L$(),D,K):: IF K
=78 THEN SUBEXIT
64Ø L$(Ø)=STR$(VAL(L$(Ø))+LE
N(L$(D))):: FOR I=D TO L-1 :
: L$(I)=L$(I+1):: NEXT I ::
L$(L)="" :: L=L-1 :: SUBEND
65Ø SUB INS(L$(),L):: DISPLA
Y ERASE ALL :: CALL LI(23,1,
"Before which line",D,L):: I
F D=Ø THEN SUBEXIT
66Ø L=L+1 ::`FOR I=L TO D+1
STEP -1 :: L$(I)=L$(I-1):: N
EXT I :: CALL KEY(5,I,S):: L
INPUT "":L$(D):: L$(Ø)=STR$(
VAL(L$(Ø))-LEN(L$(D))):: SUB
END
67Ø SUB LC(L$(),D,K):: DISPL
AY AT(1,1)ERASE ALL BEEP:STR
$(D);"!-":L$(D): :"This line
......Y/N?"
68Ø CALL KEY(3,K,S):: IF K<>
78 AND K<>89 THEN 68Ø
69Ø SUBEND
7ØØ SUB CW(L$(),L):: CALL KE
Y(5,I,J)
71Ø DISPLAY AT(16,1)ERASE AL
L:"Old word": :"New word" ::
 ACCEPT AT(16,1Ø)BEEP:OW$ ::
 IF OW$="" THEN SUBEXIT ELSE
 ACCEPT AT(18,1Ø)BEEP:NW$
72Ø CALL LI(2Ø,8,"From line"
,SL,L):: IF SL=Ø THEN 72Ø
73Ø CALL LI(22,8," To  line"
,EL,L):: IF EL=Ø THEN 73Ø
74Ø LE=LEN(OW$):: FOR I=SL T
O EL :: IF LEN(L$(I))<LE THE
N 81Ø ELSE A=1
75Ø P=POS(L$(I),OW$,A):: IF
P=Ø THEN 81Ø ELSE IF P=1 THE
N 76Ø ELSE IF SEG$(L$(I),P-1
,1)<>" " THEN 77Ø
76Ø CH$=SEG$(L$(I),P+LE,1)::
 IF CH$="" OR CH$=" " OR CH$
="." OR CH$="," OR CH$=";" T
HEN 78Ø
77Ø A=1+P :: IF A>LEN(L$(I))
THEN 81Ø ELSE 75Ø
78Ø CH$=SEG$(L$(I),1,P-1)::
C$=SEG$(L$(I),P+LE,LEN(L$(I)
)):: L$(Ø)=STR$(VAL(L$(Ø))+L
EN(L$(I))):: L$(I)=CH$&NW$&C
$
79Ø L$(Ø)=STR$(VAL(L$(Ø))-LE
N(L$(I))):: GOTO 75Ø
81Ø NEXT I :: CH$,C$,OW$,NW$
="" :: SUBEND
82Ø SUB SF(L$(),L):: DISPLAY
 AT(9,11)ERASE ALL BEEP:"Cas
sette" :: DISPLAY AT(1Ø,11):
"________" :: DISPLAY AT(11,
5):"Press!-"
83Ø DISPLAY AT(13,8):"1 For
CS1" :: DISPLAY AT(15,8):"2
For CS2" :: CALL CH(2,K):: I
F K=49 THEN C$="CS1" ELSE C$
="CS2"
84Ø DISPLAY AT(15,1)ERASE AL
L:"There are";L;"lines to sa
ve" :: OPEN #1:C$,INTERNAL,O
UTPUT,FIXED 192
85Ø FOR I=1 TO L :: DISPLAY
AT(12,4)ERASE ALL:"Saving li
ne number";I :: PRINT #1:L$(
I):: NEXT I :: CLOSE #1 :: S
UBEND
86Ø SUB RF(L$(),L):: DISPLAY
 ERASE ALL :: ON WARNING NEX
T :: INPUT "How many lines t
o read ":L :: GOTO 88Ø
87Ø GOTO 96Ø
88Ø OPEN #1:"CS1",INTERNAL,I
NPUT ,FIXED 192 :: FOR I=1 T
O L :: DISPLAY AT(12,4)ERASE
 ALL:"Reading line number";I
 :: INPUT #1:L$(I)
89Ø L$(Ø)=STR$(VAL(L$(Ø))-LE
N(L$(I))):: NEXT I :: CLOSE
#1 :: SUBEND
9ØØ SUB PF(L$(),L):: DISPLAY
 ERASE ALL :: FOR I=1 TO L :
: PRINT STR$(I);"!-":L$(I)::
 IF I/4=INT(I/4)THEN CALL KC
91Ø NEXT I :: CALL KC :: SUB
END
```

WORD PROCESSING

WORD PROCESSING

```
100 REM |~~~~~~~~~~| @#$&
110 REM |  MINI    |
120 REM |FORMATTER|
130 REM |   BY     |
140 REM |BRIAN  R.|
150 REM |_________|
160 OPTION BASE 1 :: DIM L$(
66):: DISPLAY ERASE ALL :: F
OR L=0 TO 12 :: CALL COLOR(L
,16,1):: NEXT L :: CALL SCRE
EN(5):: L=0
170 CALL RF(L$(),L)
180 DISPLAY AT(10,8)ERASE AL
L BEEP:"1 Output": :"
2 Exit" :: CALL CH(2,K):: IF
 K=50 THEN DISPLAY ERASE ALL
 :: STOP
190 CALL CUTUP(L$(),L,K):: I
F K=49 OR K=51 THEN CLOSE #1
 :: IF K=49 THEN 180
200 CLOSE #2 :: GOTO 180
210 SUB KC :: DISPLAY AT(24,
1)BEEP:"Press any key to con
tinue"
220 CALL KEY(3,K,S):: IF S<1
 THEN 220
230 SUBEND
240 SUB CH(X,K):: DISPLAY AT
(23,4)BEEP:"Your choice"
250 CALL KEY(3,K,S):: IF S<1
 THEN 250 ELSE IF K>48 AND K
<(X+49)THEN SUBEXIT
260 DISPLAY AT(24,1)BEEP:"A
number between 1 and";X :: G
OTO 250
270 SUBEND
280 SUB RF(L$(),L)
290 DISPLAY AT(24,1)ERASE AL
L BEEP:"How many lines to re
ad" :: ACCEPT AT(24,24)VALID
ATE(DIGIT):A$ :: IF A$="" TH
EN 290 ELSE L=VAL(A$)
300 OPEN #1:"CS1",INTERNAL,I
NPUT ,FIXED 192 :: FOR I=1 T
O L :: DISPLAY AT(12,4)ERASE
 ALL:"Reading line number";I
 :: INPUT #1:L$(I):: NEXT I
310 CLOSE #1 :: SUBEND
320 SUB CUTUP(L$(),L,K):: CA
LL LO(K):: P$="" :: I=0 :: I
F K=50 THEN CALL PLINE(P$,I,
1)ELSE CALL PLINE(P$,I,0)
330 PL=I :: P$="" :: FOR I=1
 TO L :: LL$=P$&L$(I):: P$="
" :: IF LL$="" THEN CALL PLI
NE(LL$,K,0):: GOTO 430
340 LL=LEN(LL$):: IF SEG$(LL
$,1,3)="/C/" OR SEG$(LL$,1,3
)="/c/" THEN LL$=RPT$(" ",IN
T(PL-LL)/2)&SEG$(LL$,4,LL)::
 CALL PLINE(LL$,K,0):: GOTO
430
350 IF SEG$(LL$,1,2)="//" TH
EN P=POS(LL$,"/",3):: LL$=RP
T$(" ",VAL(SEG$(LL$,3,P-3)))
&SEG$(LL$,P+1,LL):: LL=LEN(L
L$)
360 IF LL<=PL THEN CALL PLIN
E(LL$,K,0):: GOTO 430
370 FOR J=PL+1 TO 1 STEP -1
:: P$=SEG$(LL$,J,1):: IF P$<
>"" AND P$<>" " THEN 420 ELS
E P$=SEG$(LL$,1,J-1):: LL$=S
EG$(LL$,J+1,LL):: LL=LEN(LL$
)
380 IF LL$<>"" THEN CALL PLI
NE(P$,K,1):: GOTO 370
390 IF I=L THEN CALL PLINE(P
$,K,0):: SUBEXIT
400 IF SEG$(L$(I+1),1,3)<>"/
C/" AND SEG$(L$(I+1),1,3)<>"
/c/" AND SEG$(L$(I+1),1,2)<>
"//" AND SEG$(L$(I+1),1,4)<>
"    " AND L$(I+1)<>"" THEN
430
410 CALL PLINE(P$,K,0):: P$=
"" :: GOTO 430
420 NEXT J
430 NEXT I :: SUBEND
440 SUB LO(K):: DISPLAY AT(6
,1)ERASE ALL BEEP:"Output to
" :: DISPLAY AT(8,6):"1 Prin
ter": :"      2 Disk": :"
 3 Both" :: CALL CH(3,K)
450 IF K=50 THEN 470
460 DISPLAY AT(8,18)BEEP:"PI
O" :: ACCEPT AT(8,18)SIZE(-3
):P$ :: IF P$="" THEN 460 EL
SE OPEN #1:P$ :: IF K=49 THE
N SUBEXIT
470 DISPLAY AT(10,18)BEEP:"D
SK1.FILE" :: ACCEPT AT(10,18
)SIZE(-9):P$ :: IF P$="" THE
N 470 ELSE OPEN #2:P$,OUTPUT
,DISPLAY ,VARIABLE 80
480 SUBEND
490 SUB PLINE(P$,K,S):: IF K
 THEN 570
500 DISPLAY AT(4,4)ERASE ALL
 BEEP:"Line length 10-80": :
"   Left margin"
510 ACCEPT AT(4,24)VALIDATE(
DIGIT):P$ :: IF P$="" THEN 5
10 ELSE LL=VAL(P$):: IF LL<1
0 OR LL>80 THEN 510 ELSE IF
S THEN 560
520 ACCEPT AT(6,24)VALIDATE(
DIGIT):P$ :: IF P$="" THEN 5
20 ELSE LM=VAL(P$)
530 IF LL+LM>80 THEN DISPLAY
 AT(11,1)ERASE ALL BEEP:"LIN
E LENGTH PLUS LEFT MARGIN IS
 LONGER THAN EIGHTY..." :: C
ALL KC :: GOTO 500
540 DISPLAY AT(8,8):"Paper s
ize" :: DISPLAY AT(10,4):"1
A4": :"   2 Quarto" :: CALL
CH(2,X):: IF X=49 THEN X=11
ELSE X=9
550 DISPLAY AT(23,2):"Right
justified Y/N" :: CALL KEY(3
,J,S):: IF J<>78 AND J<>89 T
HEN 550
560 DISPLAY AT(12,10)ERASE A
LL:"Working on it" :: L=0 ::
 K=LL :: SUBEXIT
570 IF K<>49 THEN PRINT #2:P
$ :: IF K=50 THEN SUBEXIT
580 IF J=89 AND S=1 THEN CAL
L RJ(P$,LL)
590 IF L=0 THEN PRINT #1:""
:: PRINT #1:""
600 PRINT #1:TAB(LM);P$ :: L
=L+1 :: IF L<56 THEN SUBEXIT
610 FOR I=1 TO X :: PRINT #1
:"" :: NEXT I :: L=0 :: SUBE
ND
620 SUB RJ(P$,LL):: P=0 :: I
F LEN(P$)=LL THEN SUBEXIT
630 A=2 :: FOR J=1 TO LL-LEN
(P$)
640 P=POS(P$," ",P+A):: IF P
=0 THEN A=A+1 :: GOTO 640
650 P$=SEG$(P$,1,P)&CHR$(32)
&SEG$(P$,P+1,LEN(P$)):: NEXT
 J :: SUBEND
```

ED: If you have any comments we will pass them on to Brian for you if you wish.

TIPS FROM TIGERCUB

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

GROCERY SHOPPING LIST

Are you desperate for some way to convince your wife that your computer and PEB and printer and all are not just a too-expensive plaything? Maybe this will do the job.

The first thing to do is to prepare a file of the grocery items she might want to buy. It will be especially useful if you can list the items in the sequence in which she will come to them in the aisles of her favorite store. This little program will set up the file. Type END when you are finished.

```
100 OPEN #1:"DSK1.BUYLIST",O
UTPUT
110 INPUT A$
120 IF A$="END" THEN 150
130 PRINT #1:A$
140 GOTO 110
150 CLOSE #1
```

If you have TI-Writer, you can also use that to create the file, edit it and add to it - but BE SURE to delete all the carriage return symbols and any blank lines at the end. Save it under the filename BUYLIST.

Next, this program will hopefully get your wife t actually sit down at the keyboard and try out your computer. It will (through the list and ask if she wants to buy. If she types in any quantity other than 0, it will output the item name and quantity to the printer. At the end, she will be given the opportunity to add any other items.

```
100 CALL CLEAR
110 OPEN #1:"DSK1.BUYLIST",I
NPUT
120 OPEN #2:"PIO"
130 LINPUT #1:A$
140 IF EOF(1)THEN 210
150 DISPLAY AT(12,1):A$
160 DISPLAY AT(12,LEN(A$)+2)
:"0"
170 ACCEPT AT(12,LEN(A$)+2)S
IZE(-4):Q
180 IF Q=0 THEN 130
190 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
200 GOTO 130
210 DISPLAY AT(12,1):"ADDITI
ONAL? Y"
220 ACCEPT AT(12,13)VALIDATE
("YN")SIZE(-1):Q$
230 IF Q$="N" THEN 300
240 DISPLAY AT(12,1):"ITEM?"
250 ACCEPT AT(12,7):A$
260 DISPLAY AT(14,1):"QUANTI
TY?"
270 ACCEPT AT(14,11):Q
280 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
290 GOTO 210
300 CLOSE #1
310 CLOSE #2
320 END
```

The list will be in enlarged print, so that no one in the store will see her putting on her reading spectacles. And after each item and quantity is a blank square to be checked off when she picks up the item.

You might also point out that she could use the checkoff blocks to mark the items she has coupons for, and she could jot down prices on it to be sure she isn't cheated at the checkout counter, or to shop for better bargains elsewhere.

The program is set up for the Gemini printer. You may need to change the "PIO" to the name of your printer, and other printers may not have the open block character CHR$(175) available.

Of course, you can also use this program for more important things, such as shopping for computer software....!

If you type the period key while holding down the CTRL key, the printer interprets the resulting blank space as CHR$(27), even though the computer knows it is really CHR$(155). Since CHR$(27) is the ESC or "escape code" which tells the printer to interpret the following characters as function command codes, you can for instance set up the printer for emphasized double-struck double-width underlined italics by OPEN #1:"PIO" :: PRINT #1:" E G W"&CHR$(1)&" -"&CHR$(1)&" 4 ", using CTRL . in the blanks. I have been overlooking another very useful feature, the skip-over perforation. PRINT #1:" N"&CHR$(6), again with CTRL . in the blank, causes the paper to advance to the top of the next page when there are only 6 lines left at the bottom of the page (providing that you started at the top, of course). This makes it possible to LIST "PIO" a program, or PF PIO from TI-Writer Editor, without printing right across the perforations.

Ghosts! Did you ever read data from a file, and find that you were getting data from a file that was no longer on the disk? It can happen, at least if you are reading from a RELATIVE file in the UPDATE mode. When you delete a file, only its address is actually deleted - the data remains on the disk until it is overwritten by a new file. If the new file is shorter than the old one, and you try to read beyond the end of the file, you may awaken the ghost!

Are you making use of those special characters that are available on your Gemini printer? You didn't know about them? Try this.

100 OPEN #1:"PIO" :: 110 PRINT #1:" (hold down the CTRL key and type 1234567/ and then hold down the FCTN key and type <>/Ø;BHJKLMNOY) ". RUN . Surprised? Some of those can be very useful, such as the true division sign that you get with FCTN H. There are many more of these that you can access by CHR$. For a complete list of them and their CHR$ codes, run this -

100 OPEN #1:"PIO" :: FOR CH=160 TO 254 :: PRINT #1:CH;CHR$(CH);:: NEXT CH :: CLOSE #1. Unfortunately, these can't be used out of TI-Writer.

Here's a handy little routine to practice up on your typing.

```
100 CALL CLEAR
110 CALL CHAR(94,"3C4299A1A1
99423C")
120 CALL SCREEN(5)
130 CALL VCHAR(1,31,1,96)
140 CALL COLOR(1,8,16)
150 FOR SET=2 TO 12
160 CALL COLOR(SET,2,16)
170 NEXT SET
180 PRINT TAB(10);"TIGERCUB"
: :TAB(8);"TOUCH-TYPING": :T
AB(11);"TUTOR": :TAB(9):"  T
igercub Software": :
190 REM by Jim Peterson
200 PRINT "  Watch the scree
n, not the":" keyboard!": :"
  Letters and numbers will"
210 PRINT " appear on the sc
reen grid":" in position cor
responding":" to their keybo
ard position.": :"  Type the
m and they will"
220 PRINT " disappear.": : :
" Press any key"
230 CALL KEY(0,K,ST)
240 IF ST=Ø THEN 230
250 CALL CLEAR
260 CALL CHAR(32,"FFBØ8Ø8Ø8Ø
8Ø8Ø8")
270 CALL VCHAR(1,3Ø,1,192)
280 CALL HCHAR(14,1,1,384)
290 CALL VCHAR(1,4.1,14):: C
ALL VCHAR(5,6.1,11):: CALL V
CHAR(8,7,1,6):: CALL VCHAR(1
1,8,1,3):: CALL VCHAR(8,29,1
,6)
300 CALL VCHAR(11,28,1,3)
310 CALL CHAR(48,"ØØ3A444C54
6444B8")
320 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
330 RANDOMIZE
340 K=ASC(SEG$(KEY$,INT(42*R
ND+1),1))
350 GOSUB 370
360 GOTO 420
370 X=POS(KEY$,CHR$(K),1)
380 Y=ABS(X>11)+ABS(X>22)+AB
S(X>33)+1
390 R=Y*3
400 C=((X-ABS(Y>1)*(Y-1)*11)
*2)+4+Y
410 RETURN
420 CALL HCHAR(R,C,K)
430 CALL KEY(3,K,ST)
440 IF ST=Ø THEN 430
450 GOSUB 370
460 CALL GCHAR(R,C,G)
470 IF G<>32 THEN 500
480 CALL SOUND(-1ØØ,11Ø,Ø,-4
,Ø)
490 GOTO 340
500 CALL HCHAR(R,C,32)
510 CALL SOUND(-1ØØ,1ØØØ,Ø,1
ØØ5,Ø)
520 GOTO 340
```

Here's one for the kids to have fun with. I'm sorry I lost track of who published it.

```
100 CALL INIT :: FOR J=1 TO
100 :: PRINT J :: FOR P=1000
 TO 1 STEP -J :: CALL LOAD(-
31456,P):: NEXT P :: NEXT J
```

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just $15.ØØ postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just $19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only $3.ØØ each!(plus $1.5Ø per order for casette, packing and postage, or $3.ØØ for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Folks, I just can't afford to keep mailing out these Tips if you don't BUY something once in awhile! When you send me an order or ask for my catalog, mention your users group so I'll know there is someone still alive out there!

If you know of any schools in your area, especially elementary schools, that have TI-99/4As in the classroom, won't you please give me their address? I'll send them a free catalog.

Danny Michael has improved his graphics screen dump to include rotate and double size! It is in assembly, very fast, and runs out of XBasic, E/A module or Mini Memory. He has also written an assembly Neatlist program which lists an XBasic program to a printer in single line statements, indented, expanded, etc., very useful for debugging, setting up pre-scan, etc.

These are freeware, pay if you want and whatever you want. Just send an initialized disk for either one, or two disks (or SSDD or flippy) for both, in a returnable mailer with ENOUGH RETURN POSTAGE, to

Danny Michael,
Rt 9 Box 460
Florence, AL 35630.

MEMORY FULL,

Jim Peterson

32k Ram DIY Project

by Graham Wolstenholme.

This article is not intended to be of a technical nature. The Australian extract, which describes the manufacture of a 32K Ram Expansion covers this area in depth. It is provided purely as a document to outline the experiences of making your own 32K Ram.

Hopefully it may encourage others, who like me are complete idiots when dealing with all things electronic and who also, due to other commitments - her in doors and the kids - have to watch the capital expenditure!

Perhaps like other users I often considered the purchase of a 32K Ram Expansion, either Standalone or the more expensive, but versatile, P.E. Box (P.E. stands for Phenomally Expensive) but dismissed it as a luxury item.

However, I recently acquired a LOGO II module and the desire for a 32K Expansion was rekindled. Unfortunately, after a number of phone calls I established that the only available expansion, at that time, was a new P.E. Box.

Then along came dear Uncle Clive and Auntie Audrey to the rescue. Through their contacts they had obtained details, from Australia, of a DIY 32K Ram.

The Australian article was comprehensive and even with my limited knowledge I could understand most of it. The design was simple, just four Ram chips mounted on a suitable veroboard strip and connected to the computer via an additional socket, mounted inside the computer. The socket being connected via flexible wires taken from the rear of the GROM port socket.

Working strictly to the instructions I soon had all the necessary connections made and a veroboard circuit produced, ready to take the chips.

Next, before connecting the circuit, I was advised to first test the computer operated correctly. With all those extra wires on board this was good practice and would reduce possible damage.

Having reassembled the computer I eagerly switched on to see if everything was OK. My eagerness was soon shattered as I viewed a colourful display of various characters for approximately 10 seconds followed by a blank screen.

Don't panic, I thought, as I reached for the Panic Button, maybe its static generated by her in doors. There had been quite an amount of static generated during the week whilst I slaved over a hot soldering iron.

The problem, on investigation, was found to be misalignment of the GROM PCB in its socket. All those wires soldered to the rear of the GROM were causing it to twist and hence create bad contacts. I assume this was the case since all the lines were first checked with a Multimeter to ensure no shorts or dry joints.

Since this method of construction might possibly cause problems in the future, when I hopefully had the unit operating, I reconsidered the connection method. Subsequently, looking back through some old issues of the TIMES (The New TIstament) I established that the majority of connections were already available on the computer side port.

Using this method would, of course, mean that instead of a neat on-board expansion the unit would be an external standalone unit. However, on reflection this perhaps was a more attractive proposition, since faults in the future (hopefully none) could be rectified without desmantling the console.

(NOTE:- This method will limit the facilities available at the side port unless a duplicate edge connector is incorporated in the design. My sole purpose for constructing the unit was to run LOGO II, consequently future hardware purchases requiring side port connections were not considered.)

With the above modified design only six internal connections are required. These lines were taken to a six pin DIN socket, mounted at the rear of the console above the cassette port. The remainder of the lines being available on the side port edge connector.

The veroboard circuit was mounted in an attractive plastic box, purpose made by Tandy for use with their TRS80. A suitable card edge connector was mounted on the side of the box to suit either the Speach Synthesizer or console side port. Connection to the six internal wires of the console being made via a six wire cable terminating in a matching six pin DIN plug.

The unit was initially connected and tested without the Ram chips in place. This time I was not treated to a colourful display - everything was normal.

In the Australian article they suggest testing the unit one chip at a time, using Call Init, Call Load, Call Peek. However, I found that with Ext Basic it would not respond with only one chip inserted. (I suggest that with the MINMEM module this would not be the case.) It did, however, recognise the presence of the LOGO II module, ie. selecting LOGO II from the title screen resulted in standard blank screen with cursor at top left.

Consequently I saw no alternative but to install all the chips. Not, I might add, a recommended method but without the MINMEM module I saw no alternative. (The more knowledgable might be able to advise differently?)

So, fingers crossed on the panic button, I switched on - no blue flashes, no signs of expensive chips frying. With a nonchalant press of the key I selected LOGO II. The screen changed to the standard background colour and, after what appeared to be an eternity (5 seconds), the display printed 'WELCOME TO LOGO II'. The response was followed instantaneously by a performance of quadruple reverse double flips and a complete rendition of Yorkshire Morris Dancing. Yes, I know its usually a group activity but what response would you have done?

The project was a success. Checks with both LOGO II and Ext Basic confirmed, to the best of my ability, that everything was functioning correctly.

The cost of the entire project came to approximately £36. This includes all wiring, connectors, veroboard, four Ram Chips, plastic box, etc. A considerable saving over the present alternatives.

Finally, I would like to thank the following people who, without their help and advice I could not have completed the project:-

Viv Comley - who took time to reply to my letter and for the words of encouragement.

Phil Marsden - for his technical help and patience whilst dealing with a complete idiot at the other end. of the phone.

Berni Elsher and Phil West - of TI USERS PERTH , for without their comprehensive article I could never have commenced.

Clive and Audrey - for their enthusiasm for everything TI and TI 99/4a EXCHANGE, the USER GROUP.

Stephen Shaw - for his advice on operating the system with the 32K and Ext Basic.

GRAHAM WOLSTENHOLME

ED: If you require photostats of the project we shall be happy to supply, please send large S.A.E. plus 50p to TI EXCHANGE.

TI MONITOR CONNECTION

BY VIV. COMLEY

Console video output signals.

Many ordinary tv sets do not give full justice to the picture quality that can be obtained from most home computers. Television programme pictures are not as demanding on the tv set's performance as static pictures generated by home computers. A screen full of text can show up such problems as poor focus, poor convergence (colour shading at edges), mis-tuning, and vision-on-sound. The last problem can be particularly annoying, causing a buzz on the sound that varies with the amount of text on the screen. Vision-on-sound may be reduced or eliminated by careful tuning.

Some computers supply extra video output sockets to which a tv monitor can be connected. A monitor is a 'cut-down' version of an ordinary tv set, the remaining circuits however are often of a higher standard. The monitor does not have a tuner or any circuits capable of handling signals received from an aerial, and frequently an audio amplifier and loudspeaker is not included. Because of this, it is not suitable as a second tv set, and may only be used as a dedicated VDU for the home computer.

The 99/4A does not supply suitable signals for direct connection to an RGB monitor, although it is possible to connect it directly to a black and white monitor and a suitable audio amplifier without too much of a problem (see issue 9 of TI*MES for signal identification on the video output socket).

Issue 4 of TI*MES showed how it was possible to obtain a composite signal from the external modulator box, which could be used to supply a tv set or video cassette recorder with a composite video input socket. The pictures will be better than using the aerial input socket, but will not appear as crisp as that from a true RGB monitor.

The signals supplied by the 99/4A console are the luminance signal, (which is the brightness information in the picture, and is given the symbol 'Y'), the two colour difference signals (given the symbols (R-Y) and (B-Y)), and the audio signal. The luminance signal may be used to drive a black and white monitor if the colour difference signals are ignored. For the full colour picture, the separate red, green and blue signals are formed by suitably combining the Y, (R-Y), and (B-Y) signals. This is normally carrid out in the tv set to give the red, green, blue, voltages to drive the display tube. A colour monitor does not have the circuits required for combining, so this must be done by a separate interface (see OXON TI users group newsletter TI-LINES, volume 1, issue 13).

If an RGB monitor is used, supplied from the 99/4A console through an interface, there is a further limitation on available picture quality. Monitors are available with three different resolution capabilities, standard, medium and high. The resolution capability chiefly refers to the size of the phosphor dots or stripes on the face of the screen. The smaller the size, the higher is the definition available. The dots or stripes on a standard tv set (standard resolution) are normally clearly visible on close inspection of the screen. These large dots will mean that on close viewing, as in programing, the display will have a granular appearance, similar to looking through a net curtain. The phosphor dots of a high resolution monitor, on the other hand, should be very difficult to see.

It is unfortunate that the 99/4A was not supplied with different choices of video output signals, as is the BBC micro. The 99/4A can make full use of improved picture quality, but people may be understandably reluctant to spend around £50 to construct an RGB interface on top of the £200 or over for the monitor, when the basic price of the console is less than £100. However, as the price of the 99/4A plus add-ons can easily stretch to around £1000, giving a very useful system, it seems odd to have to accept a display quality that is compromised by cost-cutting in the basic console.

Drop me a letter if you have any queries on this, and I'll try to answer it. Please do not ask about specific tv sets though.

V. Comley

Mr V Comley,
Dormer Cottage,
7, St. Vincents Hill,
Redland,
Bristol BS6 6UP.

TI WRITER TIP

TI WRITER TRICK! by Ed Kennedy

Here is a program suggestion for routine users of TI-Writer. The next time you want to include a program listing in a letter or document perform the following operation:

1. Load the program into memory using TI-Writer. Place a carriage return character at the end of each program line
2. Save the program onto your diskette using the following:
   LIST "DSKn.(filename)"
   where n=1,2,or 3 for the disk drive to be activated
3. Include in your letter or document's main file the following:
   IF DSKn.(filename)

The key to this procedure is the second step. By LISTing to the disk you have saved the program in DIS/80 format which can be utilized by TI-Writer. Using this process you avoid needless typing and errors for inclusion of program listings in a letter or document.

```
100 CALL CLEAR
110 !FOR B=1 TO 24::PRINT::NEXT B
120 ACCEPT AT(5,1)SIZE(1)BEEP VALIDATE(UALPHA):A$(0+A)
122 !ACCEPT AT(5,1)BEEP:A$(0+A)
130 PRINT A$(A)
140 GOTO 120
```

CIN-DAY "99" USER

The above program was inserted in this article using this procedure. For those fellow 99'ers with Extended BASIC try RUNning this program in order to get an interesting 32-column effect while using X-BASIC (28-column is normal).

CIN-DAY USER GROUP
HAPPY FIFTH ANNIVERSARY

ARCADE HARDWARE

211 Horton Road, Fallowfield, Manchester M14 7QE. Tel: 061-225 2248

SOFTWARE

MICRO-SURGEON	£ 17.95
DEMON ATTACK	£ 17.95
MOONSWEEPER	£ 17.95
FATHOM	£ 17.95
BUCK ROGERS	£ 17.95
BIGFOOT	£ 17.95
SUPERFLY	£ 17.95
SPACE BANDITS	£ 17.95
HONEY HUNT	£ 17.95
SEWERMANIA	£ 17.95
SOUNDTRACK TROLLEY	£ 17.95
METEOR BELT	£ 17.95
ADVENTURE/PIRATE	£ 17.95
RETURN TO PIRATE ISLE	£ 17.95
TUNNELS OF DOOM (very few)	17.95
FROGGER	£ 24.95
Q*BERT	£ 24.95
POPEYE	£ 24.95
MIDNITE MASON	£ 24.95
PARSEC	£ 9.95
ALPINER	£ 9.95
T.I. INVADERS	£ 7.95
MUNCHMAN	£ 7.95
INDOOR SOCCER	£ 7.95
CAR WARS	£ 9.95
HOPPER	£ 14.95
CONGO BONGO	£ 14.95
ARCTURUS	£ 38.00
KILLER CATERPILLAR	£ 38.00

BOOKS

COMPUTE! PUBLICATIONS

Guide to ExBas applications	£10.95
T.I. Collection Volume 1	£10.95
33 Programs for the TI99/4A	£10.95
Guide to Sound & Graphics	£10.95
Beginners guide to Assembly	£11.95

BASIC BASIC ENGLISH DICTIONARY £15.95
Translates all Basic dialects

PROGRAMS FOR TI HOME COMPUTER £14.95

MILLERS SMART PROGRAMMERS GUIDE
FOR SPRITES £ 6.95

PROGRAMS

EXPLORER	£ 24.95
DIAGNOSTICS	£ 19.95
DISC REPAIR KIT	£ 19.95
NAVARONE DATABASE	£ 65.00
DFX SCREEN DUMP	£ 24.95
MULTIPLAN	£ 69.95
NAVARONE CONSOLE WRITER	£ 39.95
EXTENDED BASIC	£ 69.95
MINI-MEMORY	£ 65.00
T.I. LOGO II	£ 69.95

FULL RANGE OF INFOCOM GAMES IN STOCK.

I'VE RUN OUT OF ROOM! PHONE ME FOR THE NEW CATALOGUE.

SEE YOU IN BIRMINGHAM

ARCADE HARDWARE

211 Horton Road, Fallowfield, Manchester M14 7QE. Tel· 061-225 2248

PERIPHERALS

MYARC MINI-BOX	£450.00
MYARC RS232 CARD	£125.00
MYARC DISC CONTROL CARD	£185.00
MYARC 128k CARD	£249.95

DISC DRIVES

SINGLE SIDED INTERNAL	£125.00
DOUBLE SIDED INTERNAL	£155.00
SINGLE SIDED INTERNAL	£155.00
DOUBLE SIDED EXTERNAL	£185.00

PRINTERS

ALPHACOM 42 thermal £145.00

SHINWA CPA 80 (C) £230.00
I rate this very good value!

CANON PW1080A (C) £402.00
Full spec matrix printer

CANON PJ1080A (C) £575.00
7 colour ink jet printer

JUKI 2200 £345.00
An excellent daisywheel printer
cum typewriter. (True Centronics)

JUKI 6100 £460.00
Full spec daisywheel printer
(True Centronics only)

CPP 40 (C) £ 95.00
4" wide 4 colour printer/plotter

JOYSTICKS

PERSONAL PERIPHERALS	£24.95
COMPETITION PRO	£T.B.A.

STAND ALONE

BOXCAR 32k R.A.M.	£125.00
BOXCAR RS232/PARALLEL	£119.95

OTHER ITEMS

RS232 CABLE	£ 18.00
CENTRONICS CABLE	£ 18.00
CASSETTE CABLE	£ 7.50
TI*MES BINDERS	£ 3.95
CASSETTE'N'GAME FILE	£ 22.50
FLIP'N'FILE 50	£ 29.95

EDUCATIONAL MODULES

FROG JUMP	£ 14.95
PYRAMID PUZZLER	£ 14.95
PICTURE PARTS	£ 14.95
STAR MAZE	£ 14.95
SPACE JOURNEY	£ 14.95
DRAGON MIX	£ 14.95
NUMBER MAGIC	£ 14.95

G
TIGEBRUIKERS HOLLAND
Sorting in super Assembler

by Evert Smies

SHELL-METZNER SORTING ROUTINE IN ASSEMBLER

The Shell-Metzner sorting method is a fast sorting process. Quicksort is slightly faster, but that method is recursive, which means that it is rather difficult to implement in Basic and that it may require a lot of memory. In the worst case Quicksort needs memory for twice the number of variables to be sorted. The Shell-Metzner in its simplest form hardly requires any extra memory.

The Shell-Metzner method may be explained by means of the following example.

Suppose the following array of strings is to be sorted:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
D	C	B	A	F	D	A	E

The array is divided in groups of two elements at a distance of the half of the length of the array. In this example the distance becomes 4 and we compare (1) with (5), (2) with (6), (3) with (7) and (4) with (8). Each pair of elements will be put in the desired sequence (here ascending). The exchange of (3) and (7) yields the following intermediate result:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
D	C	A	A	F	D	B	E

Now the distance between the elements to be compared is halved and we compare (1) with (3), (2) with (4) etc. Exchanging (1) and (3) leads to:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A	C	D	A	F	D	B	E

(2) and (4) are exchanged next:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A	A	D	C	F	D	B	E

Then (5) and (7) are exchanged:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A	A	D	C	B	D	F	E

Every exchange is followed by a back-tracking procedure which checks if the last exchange requires exchanges between elements which were already dealt with. After the last exchange this is required indeed and (3) and (5) are exchanged:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A	A	B	C	D	D	F	E

Distance 2 does not require any more exchanges, so the distance is halved again. Now (1) is compared with (2), (2) with (3) etc. The exchange of (7) and (8) yields the sorted array as final result:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A	A	B	C	D	D	E	F

The following Basic subroutine is based on this method and will sort a string-array in ascending order. Change "<=" in line 1180 to ">=" for descending order.

```
100 OPTION BASE 1
110 DIM A$(100)
    <your program lines>
1000 N=100
```

```
1010 M=N
1020 M=INT(M/2)
1030 IF M=0 THEN 1160
1040 K=N-M
1050 J=1
1060 I=J
1070 L=I+M
1080 IF A$(I)<=A$(L) THEN 1140
1090 B$=A$(I)
1100 A$(I)=A$(L)
1110 A$(L)=B$
1120 I=I-M
1130 IF I>0 THEN 1070
1140 J=J+1
1150 IF J<=K THEN 1060 ELSE 1020
1160 RETURN
```

In this subroutine the elements of the string-array are exchanged. This usually takes some time, because every exchange requires a new memory allocation for the strings, which triggers a "garbage collection". A better solution at the price of a greater memory requirement is to use an index-array and to merely exchange the indices. Again in Basic this gives the following program.

```
100 OPTION BASE 1
110 DIM A$(100),A(100)
    <your program lines>
1000 N=100
1001 FOR I=1 TO N
1002 A(I)=I
1003 NEXT I
1010 M=N
1020 M=INT(M/2)
1030 IF M=0 THEN 1160
1040 K=N-M
1050 J=1
1060 I=J
1070 L=I+M
1080 IF A$(A(I))<=A$(A(L)) THEN 1140
1090 B=A(I)
1100 A(I)=A(L)
1110 A(L)=B
1120 I=I-M
1130 IF I>0 THEN 1070
1140 J=J+1
1150 IF J<=K THEN 1060 ELSE 1020
1160 RETURN
```

The sorted array must now be retrieved by A$(A(I)) instead of by A$(I) as in the first program. In the second program A$(I) gives the original sequence both before and after sorting.

Sorting in assembly language is much faster. I converted the second Basic prorgam into an assembler routine, which can be called from your Basic program as a subprogram.

The assembler program also uses an index-array. This array is initialized by the subprogram itself and contains 16 bit integer numbers. The index array is accessed from Basic by CALL PEEK. This in itself is not an example of beauty, because it makes the Basic program dependent on the load address of the assembler program, but it gives a very efficient performance. Of course it is possible to make the index-array an argument of the subprogram, but this will require much more memory. Any number in Basic requires 8 bytes, whereas integer numbers in assembler only require 2 bytes. Besides the assembler program uses a part of memory which is not available to Basic programs. For TI Basic either the 32K RAM Expansion or the Mini Memory is used. In Extended Basic the 8K part of the 32K Expansion is used.

```
* SYNTAX FOR SUBPROGRAM CALLS FROM BASIC
*
* OPTION BASE 1
*
* CALL LINK("ALFSRT".(N),A$())
* N = NUMBER OF ELEMENTS TO BE SORTED <= LENGTH OF A$(..)
* A$ = ARRAY TO BE SORTED
*
* A$() IS SORTED IN ASCENDING ORDER OF ASCII-VALUE
*
*
* ":=" IN COMMENT LINES MEANS "BECOMES"
*
*
FAC    EQU >834A Floating point ACcumulator
*
* REFS FOR EDITOR/ASSEMBLER AND MINI-MEMORY
*
       REF XMLLNK,NUMREF,STRREF  E/A AND MM ONLY
*
* EQUATES FOR EXTENDED BASIC
*
*
GPLWS  EQU >83E0 GPL Workspace  EXT BASIC ONLY
XMLLNK EQU >2018 Console ROM-routines  EXT BASIC ONLY
NUMREF EQU >200C NUMeric REFerence routine  EXT BASIC ONLY
STRREF EQU >2014 STRing REFerence routine  EXT BASIC ONLY
*
*
STATUS EQU >837C GPL STATUS byte
STRBF1 BSS >100  STRING BUFFER 1 [A$(A(I))]
STRBF2 BSS >100  STRING BUFFER 2 [A$(A(L))]
RETURN BSS 2     BASIC RETURN ADDRESS
NUMBER BSS 2     NUMBER OF ELEMENTS TO BE SORTED (=N)
*
MYWSP  BSS >20   OWN WORKSPACE EXT BASIC ONLY
*
       DEF ALFSRT
ALFSRT MOV R11,@RETURN    SAVE BASIC RETURN ADDRESS
*
       LWPI MYWSP         LOAD OWN WORKSPACE EXT BASIC ONLY
*
       CLR R0          R0:=0
       LI R1,1         R1:=1
       BLWP @NUMREF    FETCH 1ST ARGUMENT (=NUMBER OF ELEMENTS)
       BLWP @XMLLNK    CONVERT TO INTEGER
*
       DATA >1200      E/A AND M-M ONLY
*
       DATA >12B8      EXTENDED BASIC ONLY
*
       MOV @FAC,R3     R3:=FAC [M=N]
       MOV R3,@NUMBER  NUMBER:=R3
       INC R1          R1:=R1+1 FOR 2ND ARGUMENT
       LI R4,INDEX     R4:=STARTING ADDRESS OF INDEX ARRAY
VULIND INC R0          R0:=R0+1
       MOV R0,*R4+     INITIALIZE INDEX ARRAY WITH 1,2,3,...
       C R0,R3         COMPARE R0 WITH NUMBER
       JGT SORT        IF GREATER THEN START SORTING
       JMP VULIND      ELSE GO ON INITIALIZING
SORT   MOV R3,R4       R4:=R3
       CLR R3          R3:=0
       DIV R1,R3       DIVIDE R3&R4 BY R1 [M=INT(N/2)]
       MOV R3,R3       COMPARE R3 WITH ZERO
       JNE LOOP1       IF UNEQUAL CARRY ON
*
```

```
* PREPARE RETURN TO BASIC
*
       CLR R0          IF EQUAL R0:=0
       MOVB R0,@STATUS CLEAR GPL STATUS BYTE
*
       LWPI GPLWS      LOAD GPLWS EXTENDED BASIC ONLY
*
       MOV @RETURN,R11 PUT RETURN ADDRESS BACK INTO R11
       RT              RETURN TO BASIC
*
*
LOOP1  MOV @NUMBER,R5  R5:=NUMBER
       S R3,R5         R5:=R5-R3 [K=N-M]
       LI R6,1         R6:=1 [J=1]
LOOP2  MOV R6,R7       R7:=R6 [I=J]
LOOP3  MOV R3,R8       R8:=R3
       A R7,R8         R8:=R8+R7 [L=I+M]
       LI R2,STRBF1    PUT ADDRESS OF 1ST STRING INTO R2
       MOV R2,R9       R9:=R2
       SLA R7,1        R7:=2*R7 (WORD INDEX AND NOT BYTE INDEX)
       MOV @INDEXM(R7),R0  FETCH INDEX AND PUT INTO R0
       SETO *R2        MAKE SPACE FOR 255 BYTES
       SRL R7,1        R7:=R7/2 (RESTORE R7)
       BLWP @STRREF    FETCH FIRST STRING [A$(A(I))]
       LI R2,STRBF2    PUT ADDRESS OF 2ND STRING INTO R2
       MOV R2,R10      R10:=R2
       SLA R8,1        R8=2*R8 (WORD INDEX AND NOT BYTE INDEX)
       MOV @INDEXM(R8),R0  FETCH INDEX AND PUT INTO R0
       SRL R8,1        R8:=R8/2 (RESTORE R8)
       SETO *R2        MAKE SPACE FOR 255 BYTES
       BLWP @STRREF    FETCH SECOND STRING [A$(A(L))]
       MOVB *R9+,R11   R11:=LENGTH OF 1ST STRING
       MOVB *R10+,R12  R12:=LENGTH OF 2ND STRING
       SRL R11,8       SHIFT TO MAKE A NUMBER OF IT
       SRL R12,8       DITTO
VOLGND DEC R11         R11:=R11-1
       JLT INORDE      IF <0 THEN DONE WITH THIS PAIR OF STRINGS
       DEC R12         R12:=R12-1
       JLT WISSEL      LENGTH 2ND STRING LESS THAN LENGTH 1ST HENCE EXCHANGE
*
*      ALL LETTERS WERE EQUAL SO FAR
*
       CB *R9+,*R10+   COMPARE STRINGS BYTE BY BYTE (I.E. LETTER BY LETTER)
       JL INORDE       IF LETTER 1 < LETTER 2 THEN DONE WITH THIS PAIR OF STRINS
       JEQ VOLGND      IF EQUAL THEN NEXT LETTERS
WISSEL SLA R7,1        R7:=2*R7
       SLA R8,1        R8:=2*R8
       MOV @INDEXM(R7),R0 EXCHANGE INDICES [B=A(I)]
       MOV @INDEXM(R8),@INDEXM(R7)             [A(L)=A(I)]
       MOV R0,@INDEXM(R8)                      [A(L)=B]
       SRL R7,1        R7:=R7/2 (RESTORE R7)
       SRL R8,1        R8:=R8/2 (RETSORE R8)
       S R3,R7         R7:=R7-R3 [I=I-M]
       CI R7,1         COMPARE R7 WITH ONE [IF I<1]
       JLT INORDE      IF LESS THEN NEXT PAIR OF CURRENT DISTANCE
       JMP LOOP3       ELSE PREVIOUS PAIR OF CURRENT DISTANCE
INORDE INC R6          R6:=R6+1 [J=J+1]
       C R6,R5         COMPARE R6 WITH R5  [IF J<=K]
       JGT SORT        IF GREATER HALVE DISTANCE
INDEXM JMP LOOP2       IF NOT CARRY ON WITH SAME DISTANCE
INDEX  BSS >1000       INDEX ARRAY OF 2048 ELEMENTS (2048 WORDS= 4096 BYTES)
       END
```

In fact the assembler source listing contains two different programs for use with TI Basic with Editor/Assembler or Mini Memory module and for use with Extended Basic. The difference between the programs is due to differences between the Linking Loader of Editor/Assembler and Mini Memory and the Loader of Extended Basic. Please refer to section 24.2 of the Editor/Assembler Manual.

The assembler program can be tested with the following Basic programs, provided that no other assembler programs (BSCSUP excepted, see later) are loaded. As the program sorts strings, 11 will precede 2, just as AA will precede B. For the comparison of Basic floating-point numbers a different assembler routine is required.

TI Basic with Editor/Assembler or Mini Memory and 32K RAM:

```
100 OPTION BASE 1
110 DIM A$(100)
120 FOR I=1 TO 100
130 A$(I)=STR$(INT(RND*10000))
140 NEXT I
150 CALL LINK("ALFSRT",(100),A$())
160 FOR I=1 TO 100
170 CALL PEEK(PA+2*I,V1,V2)
180 PRINT A$(V1*256+V2)
190 NEXT I
200 STOP
```

PA in line 170 must be set at -22878 for Mini Memory. For the Editor/Assembler you must also load the BSCSUP program before the sorting routine and PA must be set at -22938.

Extended Basic with 32K RAM:

```
100 OPTION BASE 1:: DIM A$(100):: FOR I=1 TO 100 :: A$(I)=STR$(INT(RND*10000)):: NEXT I
110 CALL LINK("ALFSRT",(100),A$())
120 FOR I=1 TO 100 :: CALL PEEK(10198+2*I,V1,V2):: PRINT A$(V1*256+V2):: NEXT I
130 STOP
```

One thousand variables are sorted in approximately one minute.

Although the program was written to be assembled with the Editor/Assembler, it can also be entered by means of the Line-by-Line Assembler, supplied with the Mini Memory. You will not need the 32K RAM for this version.
In this case use the Editor/Assembler version with the following modifications:

-All symbols must be replaced by two letter symbols.

-DEF ALFSRT must not be used. Instead update the REF/DEF table according to the instructions of page 20 of the Line-by-Line Assembler booklet.

-The external REFerences XMLLNK, NUMREF and STRREF should be EQUated to >601C, >6044 and >604C respectively.

-The symbols INDEX and INDEXM must NOT be placed in the program, but must be replaced by the following EQUates:
IM EQU >7116 (IM in lieu of INDEXM)
IX EQU >7118 (IX in lieu of INDEX)

Instead of an index-array of 2048 elements an index-array of 1524 elements is then available. PA in line 170 of the TI Basic test program must be set at 28950.

When using the sorting routine, the Line-by-Line Assembler will be overwritten by the index-array.

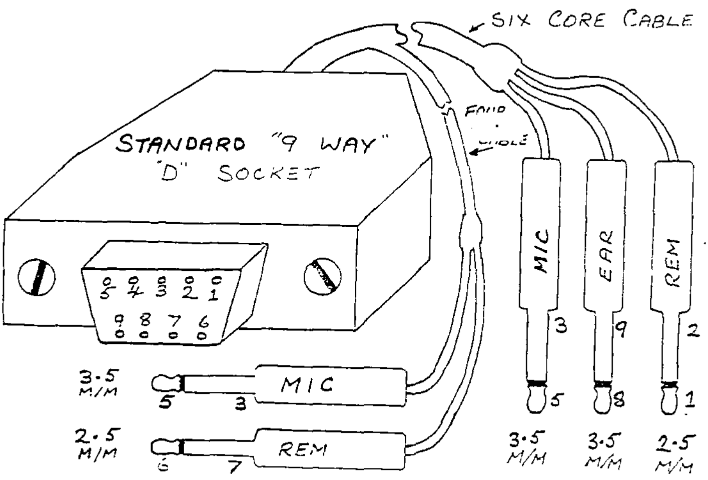
Evert J. Smies
Meiendel 13
2036 HN HAARLEM

Netherlands
Telephone +31 23 354307

D.I.Y. HARDWARE

by DEREK FORD

CASSETTE INTERFACE CABLE for C.S.1. & C.S.2 USE



The plug is a standard '9.WAY.D. SOCKET.'. The contacts ,numbered one to nine, are 'HOLES' into which the pins of the plug can be inserted. It costs about £3.

The jackplugs are : 3 off 3.5mm for EAR (1) and MIC(2)
: 2 off 2.5mm for REM(otes).

You need two lengths of cable the same length,about 1/2 metre or so, one cable should be six-core ,the other four-core (or suitable substitutes) . Use multi-strand wire for flexibility, your local electrical shop will advise you. Each wire must be insulated from the others . The wires will make up entirely separate circuits, I.E. you cannot have one 'common' return for several circuits , as feed-back and short circuiting occur.

Connect one end of one wire to number one terminal on the '9.WAY.D.SOCKET' and the other end of the same wire to number one marked on the diagram (THE TIP OF THE 2.5 mm JACK PLUG)

Connect the other nine wires to their respective numbers on the diagram,noting that terminals 3 & 5 on the '9 WAY D SOCKET' have two wires each connected to them,these go to the two MIC jackplugs respectively.Also that terminals 6 & 7 are connected to the second REM jackplug (for C.S.2.)

You will note that terminal 4 on the '9.WAY.D. SOCKET.' is not used on this interface lead for C.S.1-C.S.2. use.

A very small minority of cassette recorders have their REM sockets wired in reverse , if you find your T.I. does not control your cassette motor , reverse the wires in the REM jack plug, (or leave the REM plug out).

You must identify the four 3.5mm jackplugs either by the colour of the cables or by marking the body of each jackplug as on the diagram.

Remember C.S.1 will 'OLD' and 'SAVE' but C.S.2. is only to 'SAVE'.

Derek Ford

DEREK FORD.

Derek Ford.
37 Stotfold Road.
Maypole.
Birmingham.
B14 5JD.

EXTENDED BASIC TUTORIAL

by TONY MCGOVERN
HUNTER VALLEY TI99ERS AUSTRALIA
(Formally Newcastle of TISHUG)

HUNTER VALLEY SPECIAL

XB STYLE WITH SUBPROGRAMS

Let's now stand back a bit and look at the best way to construct XB edifices. Assume at this stage that we are in the process of developing a program, but not yet to the point where scrunching program length has become important. The first thing to note is that by giving the subprograms good descriptive names you have already gone a long way to making your program self-explanatory.

How big should individual subprograms be allowed to get ? After all, one of the reasons for using them is to break up big programs into manageable hunks. We will use the term 'line' to refer to a multi-statement XB line identified by a line number. My own prejudice is that, except in special circumstances, subprograms should be no more than about 10 lines long, and mostly rather less than that.
What makes an exceptional circumstance? An obvious one is in title blocks, like that in SIMPLIST which was left as an almost bare stub. A full version would provide graphics and advice screens, which can be tediously long to write, but contain very little in the way of branching decisions or variable assignments.
Another example is where a familiar routine, that already works, is used with little variation as in COLIST where the disk directory routine from the Disk Manual is incorporated as a subprogram with only minor changes. In any such situation where long subprograms are justified, the lists of parameters passed will be short or non-existent.

The other extreme is short one or two liners which are frequently CALLed for small special tasks, more or less your own customized extension of the built-in set of subprograms. In the middle there are middle-length subprograms with extensive parameter lists and the logic which carries the burden of program flow.

Some subprograms may be CALLed only once from within another subprogram but are of value in making your code easier to read and modify. These are associated with the branching of program flow by means of IF..THEN..ELSE statements. In either TI BASIC or XB, FOR-NEXT loops may extend indefinitely with NEXT acting as delimiter. Unfortunately in extending BASIC to XB, TI did not provide an "ENDIF" statement as in TI-FORTH, but only the 'endif' implied by the end of a XB line.
This means that any alternative actions determined by the IF.. condition have to fit within that XB line or involve a GOTO somewhere else unless the usual simple drop-through to the next line is enough. The XB manual already explicitly forbids inclusion of FOR..NEXT loops within IF..THEN..ELSE statements.
No doubt you are already used to getting around this little deficiency by placing the looping code in a subroutine and using a GOSUB. Subprograms can be used instead, following THEN and ELSE to give more complex alternative possibilities, but still staying within the confines of a single line with a minimum of leaping about with GOTOs.

This brings us to the subject of the 'dreaded GOTO'. A great deal of heat, and not necessarily much light, has been expended on this subject. It is after all just another statement available in many languages, and has perfectly predictable immediate consequences. At the machine code level, jumps enable the computer to do more than just chomp along a single track of instructions.
The question is whether it is help or hindrance in high level languages, and whether other ways of controlling program flow can replace its explicit use to advantage. TI-FORTH does without it, but that most procedural of languages, TI-LOGO, still finds it useful. Pascal tries to do without it. What we do have is XB, and XB can't do without GOTOs. If anything should be considered as reprehensible in a high-level language, it is any need to provide PEEK and POKE.

The great weakness of GOTO as a language element is that it is so readily abused, because undisciplined use makes the program code inefficient and hard for people to follow. The genuine message from 'structured programming' ideas is not that BASIC is bad for having GOTOs, but that most BASICs (TI console Basic is typical) make it necessary for the programmer to exercise real restraint if terrible tangles of GOTOs are to be avoided.

Once you use XB subprograms to chop up a program into small hunks, then you have automatically eliminated great leaps around with GOTOs. All you need then is to remember the comments on using subprogram CALLs as statements in IF..THEN..ELSE and take a little care in laying out the logic flow, you will find it very much easier to debug or develop programs.
Backwards GOTOs over more than one or two lines of code, or any forward GOTOs at all, should only occur under the most regular of logical layouts, as in SUB BASICLINE in the SIMPLIST example. Single recursive lines such as in line 620 of SIMPLIST are very effective. It's a pity that the designers of XB didn't add the "MYSELF" function as in TI-FORTH to enhance such constructions.

One last little matter before we go on to other topics. Many languages with local procedures also allow specification of global variables, accessible from any part of the program. XB does not allow for separate global variables, and it can be quite tiresome when a parameter defined at the end of one subprogram chain is only needed at the end of another chain, and has to be passed all the way up and down in parameter lists. A way around this is to use the static value feature of XB subprograms.

```
        3000 SUB PAGELENGTH(A,B):: I
        F A THEN C=B ELSE B=C
        3010 SUBEND
```

If the write flag is set as CALL PAGELENGTH(1,66) the value 66 is stored in the subprogram local variable C, while CALL PAGELENGTH(0,PL) will retrieve that value into PL. This is clumsier than having global variables, but is also more protected from unwanted interference. XB does not enforce any hierarchy of subprogram levels, so PAGELENGTH can be written to, or read from, at any level in the program. The example is for one parameter only, but is easily extended.

VI. PRE-SCAN SWITCH COMMANDS

The little supplementary booklet that comes with the current Version 110 of Extended Basic introduces a new pair of reserved words,!@P+ and !@P-. These have the form of a tail remark (XB manual p38) and so are ignored entirely by the earlier V.100 of XB. If the XB interpreter finds an exclamation mark ! outside any DATA string or string enclosed by quotes, it treats the rest of that line as though it were a REM statement. The V.110 interpreter has the added ability to recognize this pair of words beginning with ! as being distinct from normal tail remarks when used as a single word statement. Their use is allowed only at the end of a line so that V.100 just ignores them, not creating any incompatibility problems between versions, something that TI was always conscientious about. TI then couldn't let these commands actually do anything! So why are they there ?

The XB manual addendum, p7, tells the story. These switch commands allow you to control the operation of the pre-scan through the program by the interpreter -- that agonizing time interval after RUN is entered before the program starts executing. The interpreter is grinding its way through your program, byte by byte, ignoring only the messages in DATA, REMs and tail remarks.
Other than these there is nothing that it can afford to ignore until it has actually looked at it. The pre-scan sets up the storage areas and lookup procedures for variables, arrays, data, sub-programs and DEFs used by the interpreter as the program runs. Of course once it has set aside space for a variable and its lookup linkages, then it doesn't need to do it again or even to have to decide it has already fixed it up earlier. The pre-scan switch commands allow the programmer, from a superior vantage point, to turn the pre-scan off and on throughout the program so that it only looks at what it really needs to look at to do its job.

What does the programmer gain by going to all this extra trouble? The most obvious result is a reduction of pre-scan time. This can be significant in long programs. The 6 to 7 seconds for TXB, a 12K program, may still seem long but beats 4 times that. In a later chapter we will see how it can be used to fine tune run time behaviour as well. What price does the programmer pay for these benefits? The necessary penalty is the memory space taken by the extra statements. The hidden penalty, incurred while writing programs, is the inscrutable bugs that may be introduced into the code and the loss of some program checking during pre-scan such as FOR-NEXT nesting.

Let's work our way through the XB manual's prescriptions. Some of these help give insight into the way XB conducts its affairs. My experience is that some of the restrictions need not be followed strictly as laid down, as long as the essential spirit is observed, while some are absolute, and others are in between.
These last are the ones where it is possible to imagine another version of XB doing things differently while still being according to the book. This is always the danger in using unspecified properties or "undocumented features". It is not such a problem with XB since TI pulled the plug on the 99/4a and made XB a language as dead as Latin.

(1) DATA statements :-

The pre-scan locates the first DATA statement and sets XB's data pointer for the first READ operation to use. If the first DATA is skipped in the pre-scan, then RESTORE must be invoked before the first READ to set the data pointer correctly. If this is done, the XB manual's advice can be ignored.

(2) Variables :-

Each variable must be scanned once, otherwise XB won't have it in its linked list of pointers to names and storage locations. This can be the source of some truly evil program bugs, where a syntax error message results from a line of code which looks perfectly correct. The reason can be that injudicious positioning of pre-scan switch commands has left the interpreter with something that should be a variable, but can't be located as such. Being a non-variable is a much worse fate than merely being set to zero.
OPTION BASE 1 affects how storage is allocated and normally precedes any array references. If hidden from the pre-scan by !@P- then the default Ø will apply.

The manual says that the first occurrence of any variable or array must be included in the pre-scan. This would seem to be necessary for arrays, in the DIM statement, unless you are using the default (no DIM) dimensioning. Simple variables can be pre-scanned anywhere as long as it's at least once. Try the little sample program

```
         1ØØ CALL CLEAR :: !@P-
         2ØØ I=1 :: PRINT I
         3ØØ !@P+
         4ØØ I=2
```

Run this program and there will be no problems. Delete line 4ØØ and see what happens. Now you will have a syntax error in a line that by itself is perfectly correct.

(3) Sub-programs :-

The XB manual recommends that the first CALL to any sub-program be included in the pre-scan. It would appear that if the first CALL to a user defined sub-program occurs after its own SUB (from within a later sub-program) then the necessary inclusion of the SUB and SUBEND markers suffices.
(ED Note: we know there are at least nine versions of GROM Ø in the consoles, and presumably several variations on ExBas Vn 11Ø. Tony's comments MAY not apply to every system. If they do not apply to yours, please write in and tell us all about it!...ss)

Built-in sub-programs of course do not have associated SUB statements, so a CALL must be included in the pre-scan if the program is to run normally. Try this example.

```
         1ØØ FOR I=1 TO 1ØØØ :: !@P-
         2ØØ CALL SCREEN(12)
         3ØØ !@P+
         4ØØ NEXT I
         5ØØ SUB ANYTHING :: CALL SCR
         EEN(3):: SUBEND
```

This will run even though SCREEN is pre-scanned only in a subprogram. Delete line #5ØØ and it will crash if you are running XB with the 32K memory expansion. In VDP RAM (console only) it still executes but only at about 1/3 the speed.

What happens if an array is referenced in the parameter list of a sub-program, but not dimensioned until a later sub-program? If you recall the discussion on passing arrays by reference, you won't be surprised to find that XB is smart enough to hold over assigning space for the array until it comes across a genuine program reference. Try this little example:

```
        100 CALL SECOND
        200 SUB FIRST(A())::PRINT A(20):: SUBEND
        300 SUB SECOND :: !@P-
        400 DIM A(20):: CALL FIRST(A())
        500 !@P+
        600 SUBEND
```

This program crashes with a syntax error in 400 in SECOND. Now delete the pre-scan commands and the program will run. If you further delete DIM A(20):: in line 400 the program will crash in 200 with a subscript error.

(4) DEF,SUB and SUBEND :-

Do as the book says. XB needs these in the pre-scan to set things up correctly.
The pre-scan switch doesn't have much effect unless the program is of substantial size, so it isn't worth worrying about too much in the early stages of a program's development beyond being prepared for the possibility.
The XB manual supplement (p10) shows how all variable and sub-program declarations may be gathered together to minimize the range of the pre-scan, by using a GOTO to jump over the list to the first executable statement.
This can be ignored since XB does not do a complete check for correct syntax until it comes to execute the line. This is the only virtue one can ascribe to XB's failure to reject all invalid lines at entry time. The same technique can be used within a sub-program, and I have found it very convenient for this same GOTO to reserve a hiding place in which to tuck away the subroutines accessed by GOSUBs within the sub-program.

E.B.Tutorial continued Page 43->>>

HUNTER VALLEY 99'ERS NEWS

TI 99/4A

NEWSLETTER OF HUNTER VALLEY
TI99/4A USERS GROUP

TEXAS INSTRUMENTS

Extended BASIC for TI99/4A

TUTORIAL

BY Tony McGovern

continued from p42

VII. ACCEPT AT and other RAMBLINGS

TI Extended Basic is a very substantial language. The XB cartridge contains 12K of ROM and 3 and a bit (the 4th one isn't full) GROMs at 6K apiece. This is on top of the 8K of console ROM and whatever parts of the 3 console GROMS are still used in XB. The tragedy of the TI-99 is that GROMS and GPL were ever invented. I guess it was TI's way of trying to keep the software market sewn up. The end result as we all know is that they shot themselves in both feet with uncanny accuracy. Instead the TMS9900 CRU addressing to bank switch plain ordinary ROMs or even just used GROMs only as sources of code to load into RAM, they could have had a machine that did justice to its CPU, a real home minicomputer ..... that's all past history now.

I have been pondering on what TI should have done way back when the 99/4 was first designed, that could have been easily done at the time (or even when it was updated to the 99/4a). My conclusion is that the machine should have been given 4K of fast 16-bit CPU RAM instead of a measly 256 bytes. There would have been plenty of room with a little rearrangement and/or better decoding of memory-mapped devices (VDP, sound, speech, GROMs). This would have meant that Basic and XB system areas, sprite tables, full screen buffers, string buffers, value stack, and so on could have been in fast RAM, and even console Basic could have had full scope for character and sprite definitions (as in TI-LOGO for instance). Their cartridges could then have easily been a lot better, and let's face it, many of the earlier ones were pretty hopeless, and the later ones are all limited by lack of honest CPU RAM. Really good programs have only just started to appear (TENNIS for example), a year after TI laid the 99/4a to moulder in its grave. TI would then have never been dragged into that marketing war to the death (TI's that was) with that vastly inferior machine, the VIC-20. I have a suspicion that the 256 bytes happened because part of TI management wanted to protect their existing evaluation board and smaller minicomputer business.

Proposed specs for the CorComp 99/64 are floating around the User Groups now. Perhaps this is the 99/8 rising from the ashes. It reads well, but I have a few reservations. The VDP sounds just like the 99/4 VDP rather than the more advanced model that TI is working on. And only 64K RAM as standard in a new machine in this day and age ? They may be thinking of it as a way to sell high profit sole-source memory cards, and if so they haven't learned anything from the TI-99/4a's demise at the hands of Commodore. I hope that 99/4 compatibility does not mean that GPL and GROMs form any part of the normal operation of the new machine. The biggest reservation I have concerns the Enhanced, Extended Basic. This looks very powerful but has one disastrous omission, as it does not support honest procedures, sub-programs with local variables as in XB. That's not enhanced, that's just caught the Microsoft / Apple / IBM PC Basic disease (Commodore Basic is beneath contempt). Genuine procedures are a major requirement for a good Basic. TI Extended Basic (showing its engineering/instrument company heritage) has had this feature for years now. I suspect TI's programmers who obviously put enormous effort into doing the user subprograms properly were disappointed that so little good use was made of their finest feature in magazine articles and books. What was needed in XB was for user defined functions to be upgraded to the same level of performance as subprograms, and editing improved.

If CorComp has really given up this feature (enhanced ??) then I may very well pass their machine by, despite its TI-99 compatibility, when the time comes to upgrade. Other Microsoftish clutter seems to have crept in too. As it is I am waiting until the Intel 8088 has faded to an unpleasant memory. The immediate improvement really needed in XB sub-programs is a means of examining variable values in any sub-program when program execution is halted by BREAK or errors. TI should have done it in XB by retaining the EDIT command of console Basic, allowing it to access user subprograms by name. Anyone listening out there? If so add single command array operations, full syntax checking on entry, 80 column display capability with formatting power to match, bit-map screen functions, fast program execution and anything else will then be gravy. Then TI-99/4a owners will be most pleased to join in. The bad news is that TI is starting to cut back on support for the 9900 family despite its excellent qualities, and so it is becoming less attractive for new designs.

Enough ramblings and back to the tutorials ! What then is the most powerful feature in XB after SUB and CALL? A good candidate is the file system, but as this is already built into the console I will stick with commands specific to XB. The prime candidate is ACCEPT AT and its qualifying clauses. This was emphasized by the recent appearance (mid-84) in a computer magazine of a long article on machine code for adding this function to IBM PC Basic (which doesn'nt have sub-programs either). ACCEPT AT is very useful and powerful, but has some undocumented features as well as some subtle and treacherous bugs, and is well worth talking about in this series.

The simplest level of ACCEPT AT combines the INPUT routine with its access to editing features, with cursor positioning on the display screen by the AT clause. So far this is just the input version of DISPLAY AT. The difference from INPUT is that there is no provision for prompt strings, but a DISPLAY AT soon fixes that. It also accepts input to a single variable only, and not to a whole variable list. As ACCEPT AT and DISPLAY AT do not scroll the screen, their repeated use can give a much better effect than INPUT when graphics elegance is important. Construct your own examples here or work the XB manual examples. Remember that the cursor is in XB color group 0 if you are trying to dress up the graphics.

BEEP allows an audible prompt with only one program byte (we'll talk about program length later on if it keeps going long enough). Of course constant repetition of beeps can get a little wearing. The ERASE ALL clause provides an alternative to CALL·CLEAR for clearing the screen. As compared with CALL CLEAR, ERASE ALL is slower to execute, (it seems to be line at a time) but takes less program space. Its effect is slightly different also. This little program which uses ERASE ALL with DISPLAY will make both speed and screen effects easy to see.

```
100 CALL CLEAR :: CALL COLOR(0,3,3)
110 FOR I=1 TO 100 :: CALL CLEAR :: NEXT I
120 FOF I=1 TO 100 :: DISPLAY ERASE ALL :: NEXT I
130 CALL SCREEN(11):: FOR I=1 TO 1000 :: NEXT I·
```

That's the simple pieces of ACCEPT AT -- now it starts to get interesting. VALIDATE allows the programmer to decide what characters are acceptable in a response. The computer honks (that's the word in TI-FORTH) at unacceptable inputs. Three predefined types are available. UALPHA accepts only upper-case alphabetic characters -- very useful for filenames and suchlike. This is not quite the same as depressing the alpha-lock key as it it only accepts letters, and so is incompatible with input to a numeric variable. If you are in the habit of verifying wet paint signs by touch, try that for a change. The DIGIT type does just what its name implies, and NUMERIC allows the input of any floating point number as well as plain positive integers. As with INPUT, all numbers are acceptable to a string variable, but numeric variables are fussier.

Now what if these predefined types aren't right for what you want ? Suppose only digits 1 to 4 are acceptable, as in a menu choice of 4 items labelled 1 to 4. In console Basic extra lines of code would be needed to check the input, but ACCEPT AT handles this with the clause VALIDATE("1234") or VALIDATE (I_LIKE_IT$) where the string variable has previously been set to "1234". To put it more formally, only the characters in the string argument of VALIDATE can be entered at the keyboard to be ACCEPTed.

The SIZE clause allows ACCEPT AT to be used with almost no interference to screen displays. It blanks out the specified number of characters, providing an input window of finite length, and if the length specified is negative, the characrers already in the window are not erased, and form an immediate input for ACCEPTance. This is very handy for making default choices obvious to the user. Let's enter a little program to get at the essentials.

```
100 CALL CLEAR :: DISPLAY AT(12,1):RPT$("_",28)
200 ACCEPT AT(12,2)SIZE(3):A$
300 DISPLAY AT(15,2):A$;LEN(A$)
400 CALL KEY(Ø,K,S):: IF S>Ø THEN 100 ELSE 400
```

You most likely have the Alpha-lock depressed. If so let it off, and RUN our little program. Just press ENTER the first time round, next time hit <space> first, and finally <space> first before hitting another key. This shows that <space>s after the last honest character entered are ignored. Try some VALIDATEs here too, if you wish. Now with the program as given, alter SIZE(3) to SIZE(-3). It now ACCEPTs whatever is in the was or is placed in that 3 character input window.

Now that's all very simple, but it brings us to the edge of the undocumented wilderness. Alter the CALL KEY(Ø,K,S) in the last line to CALL KEY(3,K,S) and RUN the program again, this time entering letters. Observe what happens the second time around. This answers the question of what keyboard mapping ACCEPT AT uses -- like CALL KEY(Ø,K,S) it uses the last one, whatever that was. Try split keyboard units in the last line. At the machine code level, a particular byte in the CPU scratchpad RAM has to be set to the key unit before calling the SCAN routine. I interpret the behaviour as showing that in the XB modules of my experience that ACCEPT AT does not alter this byte. The XB manual however does not document this behaviour at all. If XB weren't a dead language that would be a caution signal. It does need to be watched in your programs, if your last CALL KEY wasn't the key unit you want for ACCEPT AT. On the positive side you can control ACCEPT AT with a prior dummy CALL KEY to ease input for the user. An example is when a program requests input of a filename, setting the key unit to 3 makes letters come out as upper-case while still allowing other characters. Brian Rutherford first brought the anomalous behavior to my attention.

Now that's not too bad, but there is worse to come. Insert a VALIDATE("123") clause in the ACCEPT AT and RUN the program. No problems there with SIZE(3), but SIZE(-3) is trickier. You can't enter invalid characters from the keyboard but unaltered "_"'s slip through. The VALIDATE appears to be exercised as characters are entered from the keyboard, and not as the edit buffer contents are transferred into the target variable. The decision to ignore trailing blanks in the input window is taken then however. Presumably a negative SIZE pre-loads the edit buffer with the screen window contents without doing a VALIDATE check. Ultimately this is not a real problem since the programmer can control what is on the screen before ACCEPT AT is invoked. Once again, the XB manual does not bind ACCEPT AT to work this way.

This behaviour does leave a weak spot in ACCEPT AT which can only be considered as a bug, but not an intractable one. Suppose you have a menu choice of items, say 1-4 by number, with default 1 pre-loaded in the SIZE(-1) window, and a VALIDATE("1234") clause to ensure proper entry for a numeric variable . What can possibly go wrong ? An evil-minded program tester would immediately delete the default using FCTN-1. An attempt to enter the blank will then cause the screen to scroll with a WARNING message. This is not a fatal error, but might as well be if your background is a carefully composed graphics screen. The workaround for this problem is not difficult, but the best one also resolves an even worse bug, so I will leave it for a little while. I do consider suppression of error trapping or warning messages by global ON ERROR or ON WARNING to be poor programming practice. The best safety net is one that is never used, only tested.

Now go back to the original sample program and change every every A$ to an array element A$(2). Default dimensioning will do. Nothing changes. Next alter your A$(2) in the ACCEPT AT to A$(1+1). Now it works only if there is also a VALIDATE clause, but the SIZE window is disabled and input can even spill over into the next line. No, it's not useful as a multiline ACCEPT ! The solutions to this and the previous problem are the same --- always ACCEPT into a temporary simple string variable, and then process the return, and do not ACCEPT a numeric directly or ACCEPT into an array element with computed index. Both of these problems were turned up by my testing crew during the writing of TEX-BOUNCE, and served as a reminder that program testing should never be left to the author of a program. The same holds true for writers of languages!

Might as well keep on going with the entomology lesson. The sub-program CALL ERR fails to clear errors when the DSR routine cannot find the external device, as in attempts to access an empty disk drive. The work-around this problem is to have a second bash at CALL ERR after further trying for a file on the device which failed to OPEN. The OPEN cannot be CLOSEd without crashing the program or invoking this extra step to flush out the Peripheral Access Block. The Feb/85 Newsdigest carries a letter giving that essential extra step. Also Ross Mudie called up earlier with a similar approach. I had put it aside as something to be dealt with from Assembly language, which is the way I will do it in the next update of COLIST.

The instruction ON BREAK NEXT is useful, particularly in games, for disabling the FCTN-4 (BREAK) key action. However a CALL SOUND with duration greater than 33 over-rides that. Just why is not so far obvious to this outside observer. [I suspect Tony is referring to the situation when two CALL SOUNDS are contiguous. The actual CALL takes about 33 milliseconds. If the first CALL is for a longer sound, the computer waits for the first sound to finish before passing to the other. Presumably there is a "hole" here which permits BREAK to function...ss]

CONSOLE ONLY USERS....

So far the series has had a detailed look at user SUBprograms and the ACCEPT AT statement, the two most powerful features of TI's Extended Basic, and also at the prescan switch commands lurking in the V110 manual addendum. For the next few sessions we will continue with topics which are of immediate relevance to console-only users, namely squeezing programs to fit in memory, and extracting maximum speed from XB. Please let me know of areas you would like covered. My policy so far has been to concentrate on those parts of XB which are especially powerful, not already included in console Basic, not well documented, and not as widely appreciated as they should be. The next few tutorials will be on getting the most into and out of the machine while using XB. On the other hand I can see no point, and have even less interest in writing about, say, SOUND or SPRITEs from the very beginning, as these are fairly well documented in the manuals and the subject of many books and articles. That isn't to say that subtleties in using them won't come up from time to time.

There has been a gap of a few months in appearance of Tutorials, mainly due to pressure of work. The time spent on the TI-99 has been almost entirely devoted to Assembly language programming, much of it in association with XB, and this will provide some real substance for future HV99 News articles, either in this series or separately. One of these projects has been to get TI-Writer running from XB. Why do that ? Well... TI have always been good guys in that most serious disk software can be backed up on disk as often as needed, but they were of course relying on the infamous cartridge GROMs for protection and exclusion of others. Now cartridges are a lot less fragile than disks, but they can die too. So I don't want ever to have to suffer Imagic Australia's less than impressive service and/or rapacious pricing policy if our TI-Writer module ever claps out. May save some module swapping on occasion too. Yes, we do have a spare XB module!

Of late I have been working with Microsoft Basic and Turbo Pascal on CP/M machines with Z-80 processor in science laboratory applications. I must say that the more I see of Microsoft Basic the more I regard it as a cancer that should have been eradicated years ago when computers grew up to have more than 8K of memory. It is only now with their Apple Macintosh version, judging that from reading magazine hype, that they have at last surpassed the level of expressiveness that XB had years ago. TI did a lot of things to screw up this machine, some of historical origin, some quite intentional, but it takes coming from the engineered TI-99/4a file and device handling system to CP/M to make you realize how weak and primitive CP/M is in this area. On the other hand Turbo Pascal almost makes that Swiss straight-jacket feel comfortable, and even CP/M doesn't seem so bad with Turbo. An excellent product at a realistic price that makes one realize that pirates are only the minor league of brigands in the software business. We can only dream that someone will bring out a native code Pascal anywhere near as good as Turbo for our machine. The TI P-code version is most unattractive at Imagic's past and present exorbitant price (eight times that of Turbo). Now that more compact consumer type products are replacing the massive PE box, P-code cards will fade into oblivion.

It is also another example of how TI had this death-wish to hobble the most powerful micro-processor in any home micro available here, with interpreted languages. Shed a tear for TI-99 Basics with their two layers of interpretation (Basic and GPL), on top of working indirectly from the byte oriented VDP memory and GROMs.

Enough raving on for now and on to the real business. Let's now look at how to face up to that 'MEMORY FULL' message. This even comes up when you have memory expansion with a total of 48K of RAM in various guises. Programs always seem to end up needing more memory than is available! I do feel some unease in discussing this topic as many of the things that are done in compacting programs can only be regarded as poor program practice otherwise, as they make the code obscure and difficult to modify or develop further. The other great trade off that must be considered when scrunching programs is speed of execution. Given an equal level of skill in program writing, coding for speed usually results in a longer program than would otherwise be written. Perhaps the easiest example to see is unrolling of a short loop which is repeated a fixed number of times. A FOR-NEXT loop gives compact code but carries a penalty of the loop overhead which could be avoided by writing out the contents of the loop the appropriate number of times. The subject of coding for speed will be taken up in detail in later Tutorials, and speed sacrifices with compacted code will only be noted in passing. The richer the language the more opportunities there are to optimize code one way or the other. Console Basic offers many fewer ways to do this than does XB and is much less fun.

At what stage should you bother trying to make your code compact ? Remember that XB can only OLD or RUN one program at a time, so apart from loading time from cassette, or disk space, there is no reason at all to scrunch a program that runs in the smallest memory it is intended to run on. Most users with disks now have the 32K memory expansion, so this means the bare console. Minimem Basic programs to store in the module's RAM are the only ones you have real incentive to make smaller still. Unless you know from the start that you are going to run short of space because of large arrays of numbers, or a need for maximum string storage room, be expansive -- document your program thoroughly with REMs, use lots of SUBprograms, use obvious explanatory names for variables, avoid reusing variables for unrelated uses ..... and then you run out of room.

Now first of all a program has to be short enough to load. This is purely a function of program length. Next it has to be able to complete prescan when RUN. For prescan to succeed there must be enough room left over after the prescan for variable pointer and subprogram tables to be set up, and room set aside for numeric values, at 8 bytes per number. String variables are not assigned space until it is actually required, so it is possible for a program to crash later because it can't find enough room for strings. The well known hiccupping of long Basic programs occurs while Basic scratches around to reclaim string space when it has run out of new space. XB does it too, but it is a lot faster at 'garbage collection'. Now let's look at how to squeeze programs in, starting with things that affect the program length only.

The most obvious thing to do is to remove REMs from your program. I would suggest that this be left till later in the development process as you put them there in the first place to help. At the least keep some for now. If you have been following earlier Tutorial advice to use lots of clearly named subprograms then you don't need many REMs. For the same reasons you should not abbreviate subprogram names beyond recognition at this stage. Basic as an interpreted language, where the source code is also the run-time code, has this problem that commentary and explanation are not eliminated by a compiler or assembler and compete for memory space with the executing program. One way round the problem is to restore REMs to a file copy after intensive development is over, even if it does make it too long to RUN. The REMS can always be removed later.

Now it's time to look at what makes an XB program as long as it is. To get started let's look at two very short programs to clear the screen.

```
100 CALL CLEAR
```

Before entering this clean up the machine with NEW and SIZE it. Then enter this program and SIZE it again. The difference will be the length of the program 13928-13914 = 14 . I will mostly quote SIZEs on the basis of a console only machine for simplicity, but there are some interesting differences. With memory expansion XB lists high memory and stack separately, and ignores low memory altogether. XB stores the program and numeric variables in high memory (24K), while the stack - 12K of VDP memory - contains variable descriptions, subprogram details, PABs, and the string storage space. This ALL has to fit in 14K of VDP RAM with XB/console only. Console Basic doesn't use memory expansion for Basic at all. Now try a second program which does almost the same thing

```
100 DISPLAY ERASE ALL
```

and SIZE it. Only 9 bytes now ! Although the LIST of this second program is longer, the computer thinks it is shorter. Consult your XB manual p40 where you will find all three words DISPLAY, ERASE, ALL are listed as reserved words, as is CALL but not CLEAR. Reserved words are treated differently -- when you enter the line they are recognised and "tokenized" as one byte symbols with ASCII values >127. 'CLEAR' takes 7 bytes, one the token for a string without quotes, one for a length byte, and 5 for the string itself. Why use tokens ? For one thing it shortens the program length, and also makes it easier for the interpreter to recognize them when the program is running. XB's range of tokens is very limited and built-in subprograms are the way XB gets around this.

Now you don't have to take my word for this. If you have an expanded system you can write programs using CALL PEEK to explore stored programs, or better still use the E/A DEBUG (reassembled as uncompressed object code so the XB loader can handle it) for a quicker look. With console XB you can at best get an indirect insight by entering <CTRL+various keys> in a REM statement and LISTing that. Be careful, you can crash the computer in ways wondrous to behold that way. Someone forgot to tell the computer not to try to turn token values back into reserved words when LISTing REMs. Ever notice when writing file specifications that keywords that do extra duty elsewhere LIST with the extra space, but the others do not. EASY-BUG in Minimem also allows you to look directly into VDP RAM or cartridge RAM to see Basic programs in their internal state.

In TI Basics, unlike those which store programs as ASCII files, the line number is always stored as a 2 byte integer, and it makes no difference to program length to use line #1 or line #10000. Try various line numbers in one of the examples above. If you are peeking around in the program, don't expect to find the line number at the start of its program line. It is in a separate table below the program, and each 4 byte entry has the line number followed by the location of the line itself. The line # table is sorted into order, but new or edited lines are always added to the lower address end of the program block. The program lines themselves are preceded by a length byte and terminated by a null (>00) byte. I won't go into it here but you can use this general information to interpret the various time delays when you edit a line or enter a new line.

From this you can see that there is a 6 byte overhead associated with every new line number. Now enter the program lines above as lines #100 and #200 and SIZE. Next combine them as a single line

```
100 CALL CLEAR :: DISPLAY ER
ASE ALL
```

and SIZE again. There is a saving of 5 bytes. The reserved word "::" has cost 1 byte, but 6 bytes have been saved by having one line fewer. Now if you scrunch a 500 line console Basic style of program into 200 XB multi-statement lines you have gained 1500 bytes. Of course you can't do this to every line because line numbers, as well as being line editor markers, are also where GOTOs and GOSUBs go, so you will usually end up with a few short lines you can't condense. FOR-NEXT loops work perfectly well within or across multi-statement lines. The use of prescan switch commands is costly because you end up with !P+ and SUBEND on separate lines at the end of each subprogram so treated. Still, it's usually worth doing even though a long program may have several hundred bytes tied up in prescan switching. In desperation at the end you can always remove prescan switches starting with the shortest subprograms.

How much room does a variable take up ? Take a simple numeric variable. There are 8 bytes for the radix-100 floating point form that both TI Basics use for all numbers (they even do 1+1 to 14 significant figures every time - another reason they are slow). Next the interpreter has to be able find where this value is stored so there's 2 bytes for a pointer to the value, and 2 more to point to the name associated with this value. Further it has to record the nature of the variable, whether it is numeric or string, simple or array, DEFed or normal. Also in a Basic language which allows long variable names a length record is also likely, though not absolutely necessary. All told there is a practical minimum of 14 bytes of overhead for every simple numeric variable.

As I have noted in other connections in this series, TI in its self-defeating secretive way, never explicitly specified the details. TI Basic is most likely highly consistent in this from model to model, because any console can be called on to work with separate E/A or Minimem Basic support utilities such as NUMREF. On the other hand each XB module contains its own set of support utilities, and only has to be internally self consistent. There is information in TI's published data (XB, E/A, Technical manuals), giving details of VDP stack entries built by the E/A CALL LINK with some hints as to changes for the XB version. So to use XB LINKs at this level of detail you have to work by implication. Now it is done from time to time, but TI does not seem to have guaranteed explicitly to programmers that such procedures would work with all XB modules, or that the LINK stack entries are similar to internal table entries. Most likely they do and are. Only TI knows for sure. Then again TI lost big while Apple and IBM make lots of money being more open about their machines, though Apple seems to be developing more secretive ways as it gets older and more arrogant.

Time for some little program experiments again. Enter the miniscule program

```
100 A=0
```

Before you do anything else work out how many bytes this uses. The answer is 11. In accepting the line the editor has already figured 'A' for a variable (because it starts with an allowable character) and not a reserved word and it is represented exactly as it occurs, no token involved. On the other hand it doesn't yet care that '0' is meant to be a number and treats it as an unquoted string. If it isn't an honest number, say 2N, it will only find out later when it RUNs and tries to convert it to a floating point number.

SIZE the program, then RUN it and SIZE again. XB does not reset everything until you have made an editing change, as you know from debugging efforts after BREAKing (fctn-4) program execution. At this stage you get more information from an expanded system, which will show 8 bytes of memory used and 9 bytes of stack. Now repeat the process with a longer variable name. The length is reflected both in the original program length and in the stack used. The stack usage is 2 bytes plus the variable's name length more than the minimum we figured out before. Most likely the 2 bytes are for a linked list structure to help table searching, and there is a symbol table entry of the variable name. Now turn off your expansion system and be like everybody else with console only, and repeat the above. Now you will find the increase over the program length is always the 14 bytes we figured earlier no matter how long the variable name is. Now try

```
100 A23456789012345,A2345678
9012345=0
```

Still 14 bytes RUNtime overhead ! Change the second A to a B to make a distinct variable name, 15 bytes long. Only another 14 bytes overhead ! So how come ? Maybe it's doing without the full word list link and squeezing things up a bit, but what about the symbol table ? The only consistent conclusion is that it doesn't have one as such, but points to the first location of the variable name in the program as located by the prescan. Read the Tutorial on prescans again. XB always searches in VDP RAM for variable names even if the program itself and the numeric values pointed to are stored in expansion memory.

If you wanted to make a faster interpreted Basic, you would, in the prescan, replace all variable names by some token plus a storage pointer to eliminate table searches. Which is just what TI claim in their Software Development Handbook to have done with the Basic for their 990 minicomputers. Unfortunately they failed to make an honest machine of the 99/4.

That should be plenty to chew on for this inaugural issue of the HV99 News. The next Tutorial will continue with the principles of program scrunching, getting more into the program writing end of things.

Funnelweb Farm
215 Grinsell St.
Kotara, NSW 2288
Tel (049)52-3162

(C) by TONY MCGOVERN

END OF EX BAS ARTICLE~~~~~~

World TI Users Spotlight

Calling all TI groups around the World TI*MES puts you in the Spotlight.

TI ENTHUSIASTS OF SINGAPORE

P O BOX 698   TIONG BAHRU   SINGAPORE 9116   SINGAPORE

GREETING FROM TEXAS INSTRUMENTS ENTHUSISATS OF SINGAPORE.

Fellow computer users,we are a group of Singaporeans,who own and operate the Texas Instruments 99/4A Computer and we would like to acquaint you of the many benefits membership of T.I.E.S. can offer you.

SOFTWARE LIBRARY

We are in the process of building up our software library. Some of these programs are from members of our group and from similar groups in U.S.A.,U.K. and Australia. Free copies of these programs are available free to members during each meeting.

TECHNICAL LIBRARY

We have a small library of technical information relating to the TI-99/4A and other peripherals. This information are available to members upon request.

REGULAR MEETINGS

Regular meeting are held whenever possible and a computer will be available to demonstrate new software and/or equipment. Other interesting literature or news received will also be shared during the meeting.

EXCHANGE OF IDEAS

Members are encouraged to discuss any problems or ideas relating to the use of the computer or other relevant subjects. Topics discussed in the past meetings were:-

TI-WRITER--Formatter Commands.
Printing with daisy wheel printer.
Installation of enhancements.
XBasic --File processing
Arrays handling
Sprites routine
Error codes and their meanings
Others --Turn a SS disk into a Flippie
Joystick routine
Cleaning of module contacts
The on-going tutorials on FROTH.

These meetings of minds certainly resulted in more enjoyable and rewarding computing for all our members. And we look forward to the 128K RAM Card, DSDD Controller, 'cheap' 1200 baud modem and other such goodies. Any UK TI Users who wish to contact us may write to me, Colin Lee c/o TIES P.O.Box 698, Tiong Bahru, Singapore 9116.

****
****
****

HOWARD WRITES AGAIN

by Howard Greenberg .....yawn!

Panic stations! A 'phone call from our editor....."Howard, are you putting anything in TI*MES this edition." "Yes, when do you need it?". So this is being written at the sort of hour when vampires are usually around, with Wilson Pickett singing In the midnight hour.

I could pretend that I'd waited until the last minute in order to give you the very latest news, (which I have), but I just didn't realise how late I was with my article.

Unfortunately, it's necessary to start this article with a warning. Just after the Brighton show, when I quite literally received the Infocom adventures the day before the show, I learned of a pirate copy of Hitch Hikers being passed around. Now Infocom put protection on their discs, but it's not foolproof, and within a week of my selling the first copies of this game, I discover it's being handed around. I still don't know who the originator of this is, but I've promised the secondary pirate a visit to a courtroom if I learn of this happening again. The same applies to the primary pirate. Legal proceedings are so expensive that most people who threaten them can't carry out their threats. I can't afford to waste money on solicitors either, but if someones threatening my livelihood, then I'll have no choice. STOP IT!

More bad news. Despite my advertising that Thorn-EMIs Computer War would be "coming soon" (To be the epitaph of every computer manufacturer) it now seems unlikely that a finished version will see the light of day. I have seen the program, it is one of the best games I've seen for the TI99/4A and completely original, but heels are now being dragged and if it can't be made before this Xmas, then there's little point in doing it. A great shame.

Some very late good news. A phone call from Sweden indicates that someone has cracked the problem of getting into Prestel. As you may know, the TI99/4A and RS232 can only use single speed baud rates (e.g. 300/300 or 1200/1200) whereas Prestel and other bulletin boards use 1200/75 baud, which our machine cannot handle. Our Swedish freind seems to have overcome this, and has put his Terminal Emulator program in cartridge form. If I can, I'll get a copy to show off at Birmingham.

A couple of new products which are well worth raving over. I'm awaiting delivery of two discs from Millers Graphics which I should certainly have in time for the show. One is Diagnostics (£19.95) which checks out mostly the disc system, but other areas too. The other is quite simply the most staggering program I've seen. Explorer is incredibly sophisticated and the Manual alone contains £24.95s worth of information. If you're into assembly programming, then this has got to be the best thing yet. There is virtually nowhere in the console that's forbidden for you to go. You can single step a program (even a cartridge) and modify it if you want. Mr. Miller has gone to extraordinary lengths to protect his program too, including developing a new disc format which can't be overcome with any of the disc copy programs available. But more than that, the only item Explorer won't look at, is itself. Even by using a load interrupt switch, (the favourite means of doing things you're not supposed to). On pressing the l.i.s. the program checks out what's happened and realising you're trying to go somewhere you're not supposed to, returns you to the master title screen. A truly brilliant program and I'd recommend it even if I weren't selling it.

My last piece in TI*MES contained an incorrect item. Although Myarc were trying to make a 128k version of their card that could work in conjunction with the TI 32k card, it didn't quite happen that way. I was under the impression that it was going ahead, and by a coincidence, I received the new cards from

Howard Greenberg .....yawn!

Myarc at the moment Myarc 'phoned me to tell me they's actually sent 32+96k versions. I didn't mind that they'd been unable to make the 128k only and had sent the original versions. What I did mind was the fait accomplis they'd done. The card still be brought up to 512k, which is now becomming more economically feasible. Last time I wrote, the cost of replacing the 64k R.A.M.s with 256k equivalents was more than £500.00. The cost of the 256k R.A.M.s is now nearer £12.00 each and dropping. Myarc have now also developed a Super Extended Basic for the 128k card, which promises a fourfold increase in speed along with a whole set of new commands. (DRAW, PLOT, CIRCLE and others.) I have neither price nor details yet, but hope to have at least the information for Birmingham. It will only work with the 128k card though.

Black Mark number 2 for Myarc. After very considerable delay, they finally sent copies of their disc manager to me. If you have a Myarc Disc Control card and have not yet received a Myarc Disc Manager from me yet, please let me know and I'll run off a copy. It's very good indeed, much better than the TI Disc Manager II. But...It won't work with the TI disc control card. It makes certain CALLs to the control card, and if it doesn't find a Myarc operating system there, you're presented with the master title screen.

Following John Rice's review of the COMPUTE! series of books for the TI99/4A, I've decided to stock them. Titles are listed in the adverts (elsewhere) and all I'll say in addition is that I back up his opinion of "Beginners guide to Assembly Language". It is the best book on the subject, if only because Peter Lottrup is a teacher and therefore not only does he know his subject (as do all writers on the assembly set) but he can also teach. Very important.

Thanks Graham Baldwin for his comments on my salesmanship. Personally, I'd alway felt I'd have had trouble selling Liferafts on the Titanic. Incidentally, I've still got a stand alone disc Controller!

The news is about to break anyway, so I may as well spill the beans. There is a new computer under development. It will be made by Myarc and will be about 90% compatible with all TI software. It will have a very fast and capable BASIC and will be able to run TI modules too. All expansion will be external, so you'll be able to use your peripherals as they are. (This is the reason why Myarc didn't make a 128k only memory card). Price is hoped to be in the BBC region (£300.00-£400.00) but availability is uncertain. I will be sending a mail-shot when I'm certain of its release and of course WATCH THIS SPACE. It will also have a plug in option to enable it to run IBM software, (This is pretty vague at the moment) as an extra.

I'm running out of space and news. Look forward to meeting you in Birmingham, but for now. BYE

HOWARD GREENBERG. Howard

MANCHESTER MUSINGS

by John Rice

CLUBS AND PUBLICATIONS

The TI Home Computer Users Club News (Summer 1985 edition) eventually arrived, together with the news that Philip Thomson of Home and Business Microsystems in Edinburgh will be getting in touch with me when my membership expires, to see if I want to enjoy the benefits of his Users Club. I'll let you know what tempting offers I receive!

National Ninety-Niner went a bit quiet over the summer, but a quick (20-minute!) phone call to Luci Veith in Bakersfield, California gave me the news that my June and July/August copies of the mag must have gotten lost in the mail, so I'm now awaiting a reconsignment. Luci also told me that they'd had to change their printer, which has upset both printing and mailing a bit, and accounts for the fact that the collected edition of past issues of National Ninety-Niner, which has been due for reprinting for ages, is unlikely to be available for a couple more months - apparently their new printer hasn't had his vacation yet.

MICROpendium continues to arrive regularly each month with all the latest US hardware and software news. Recent issues have contained a series on how to make your own 8K battery-backed RAM cartridge (butchering an Editor/Assembler module and games module in the process). Also there's been news of Myarc's new memory card for the PEB and Corcomp's new multi-function card.

BOOK REVIEWS

I picked up some super bargains at the Personal Computer World Show a couple of weeks ago - getting in early on a trade day and having a root through the sale books on one of the stands.

1. "The Best of TI99/4A Cartridges" by Thomas Blackadar; ISBN 0-89588-137-3; 133 pages; Sybex, 1984.

It reviews about 20 TI cartridges in depth under the general chapter headings of Home Management, Education and Games. I'd never come across this book before anywhere. In these days when most purchases are by mail order (sometimes across the pond) - except, of course, at TI User Shows - it's handy to be able to read a review before you buy. I picked the book up for £2.00 - there was no indication of its normal price. Quite useful.

2. "TI in Wonderland" by Fred D'Ignazio; ISBN 0-8104-6415-2; 121 pages; Hayden, [illegible].

3. "The TI Playground" by Fred D'Ignazio; ISBN 0-8104-6414-4; 116 pages; Hayden, 1984.

These two books contain 21 and 23 "programs for learning and fun". The actual age of the "youngster" for which they are intended isn't stated, but the programs cover everything from recognizing capital letters to a quiz with questions like "Who invented the telephone?". Each program is introduced with a section For Parents and Teachers, one For Kids, then the Program, and then Typing Hints, Highlights, Variables and a Do-It-Yourself section which suggests extensions to the program. The latter section enables the programs to be made useful to an extended age range. I paid £1.00 and £2.00 for these books, but I've no idea what the usual price is - though I've seen them at around £6.00 elsewhere. Well worth buying if you have kids (or are a kid!) in the 4 to 9 age group.

COMPUTE! BOOKS

Good news concerning COMPUTE! publications. Your bookseller or dealer can order them from:

HOLT SAUNDERS LTD, Customer Services Department,
1 St.Anne's Road, EASTBOURNE, East Sussex BN21 3UN.

Individuals can order them from the Accounts Department. Payment by credit card is welcome. Telephone orders to FREEPHONE 2568. Enquiries for more information (e.g. price) to the Marketing Department (A and P Division) or phone 0323-638221.

SOFTWARE

I'm sorry that Scott Foresman had sold out by the time my last article appeared. Perhaps someone has news of where all their stock went to?

At the risk of incurring your collective wrath again, I'll stick my neck out and mention some "bargains" that are available from:

TexComp,P.O.Box 33084, Granada Hills, CA 91344, USA.

They accept VISA cards, and you can order direct on 0101-818-366-6631. I placed an order on their answerphone back in mid-July and received the correct goods in mid-September, which is reasonable going for surface mail. It does come as a bit of a shock to have a postman at the door demanding £22.69 "postage due" on behalf of the taxman (import duty and VAT)! (Incidentally, if anyone out there in TI land wants £22.69 worth of TO PAY stamps - let me know - they're beautifully lightly cancelled). And what was in this bumper parcel I'd ordered? One of each of the following.

"Story Machine" (PHM 3178), the Spinnaker module that was issued at the same time as "Facemaker" (PHM 3177), one of my nieces' favourite modules (particulary when the face sticks out its tongue!). Story Machine helps children write simple stories with a limited vocabulary, and illustrates the story as it's written - only allowing gramatically correct sentences - like "The boy eats the dog" (good one that!) - well illustrated! This is a super module, as is Facemaker, and both are available at $9.95 each.

The other modules which I thought were reasonably prices were Terminal Emulator II (PHM 3035) at $9.95 - necessary for some of the TI speech cassettes; Mini Memory (PHM 3058) at $38.95 - including free Mini Writer word processor program, which I'll try and review next time; and a TI-produced "original" - but version 110 - Extended BASIC at $49.95 (which seems to be the US price for most of the licensed versions as well).

Then for the fun! Now that Channel 4's started it's new winter season, settle down with (American) Football (PHM 3009) at $9.95. Not exactly a lightningly fast game, but it does come with a book of rules and explanations of jargon, so it might increase your TV enjoyment. I also obtained Physical Fitness (PHM 3010) at $9.95. You can decide whether that module's any use when you see me at the next TI User Show!

SOFTWARE REVIEWS

1."Advanced Diagnostics" ; MILLERS GRAPHICS, 1475 West Cypress Avenue, San Dimas, CA 91773, USA; Disk plus 34-page book; $19.95 (+ $3.50 Foreign Shipping and Handling per order).

Requires 32k Memory Expansion, Disk Drive (Double Density requires Corcomp Controller) and one of Extended BASIC, Editor/Assembler or Mini Memory.

This is without doubt the most useful program anyone with a disk system could wish for. It performs the functions of a disk manager, including sophisticated copy facilities, and also provides extensive disk editing features. If you've ever wanted to see into the no-man's land between disk sectors, you can do that too. You can also alter the head step time to optimize head movement when using the program, and you can view a graphic display of the average RPM of your disk drives. A CPU, VDP and 32k Extension RAM test is included, and on-screen HELP is available at all times. Screens can be dumped to any valid output device and the whole program has a very friendly user interface. However, its real power lies in its ability to perform pre-stored sequences of commands held on disk files. Just to get an idea of how powerful this feature is, a number of sample command files are supplied, such as disk test utilities, a utility to initialize a new disk and copy a disk to it, and one to format a box of new disks. The manual tells you all you need to know about disks but didn't know where to look. Altogether an essential package for any disk owner who wants to get into the guts of the disk system.

2."EXPLORER"; MILLERS GRAPHICS; Disk plus 105-page book; $24.95
Requires 32k Memory Expansion, Single or Double Density Disk Controller (currently not compatible with Myarc's Mini Peripheral System) and one of Extended BASIC, Editor/Assembler or Mini Memory.
This software does for the console and modules what Advanced Diagnostics does for the disk hardware. However, since there's rather more available on the software front, this program will keep you engrossed for hours. In its simplest terms, EXPLORER is a machine language interpreter - a program loaded into the 99/4A which looks at 9900 machine code instructions, decodes them like the hardware does, and executes them - at about 1/300th of normal speed (1/1000th if interrupts are enabled). So what? Surely BASIC works slowly enough as it is! Well, while it's running it can keep track of up to 3 dynamic memory windows for any area of CPU memory, VDP memory or GROM/GRAM memory (except where EXPLORER is located). It can also check for reads (or writes, if appropriate) to any specific VDP, GROM, GRAM, ROM or RAM addresses, and stop when found. Whilst this is happening, it can also display the current instruction in assembler source code and the current register values on the screen. You can scan memory for particular HEX or ASCII strings and, when a program is running, you can flip back and forth between the program's display and EXPLORER's. The sample explorations in the book take you through the power-up routine, executing CALL SCREEN in BASIC (takes 71,265 machine instructions) and in Extended BASIC (only 32,651 instructions).
The real power of the program comes if you have a Navarone Widget (available from Arcade Hardware) or a Myarc or Corcomp Disk Controller. Then you can execute other modules under your control. I successfully traced through the start-up of the Disk Manager 2 module, although because the interpreter is so slow, it can't actually handle disk, RS232 or cassette calls. Instructions are given on tracing through different types of cartridges. The comprehensive manual ends with 25 pages of useful information about the contents of the Console GROMs and VDP and CPU RAM usage in BASIC and Extended BASIC.
Miller's Graphics service was good. My copy of EXPLORER arrived defective, but it was replaced by return of (transatlantic airmail) post. For anyone developing machine code programs, this software is almost indispensible. My only complaint is that you can't dump a screen to the printer, but I understand this feature was omitted to reduce the size of EXPLORER so that you can actually load a reasonably sized assembler program for it to trace.

DAISYWHEEL PRINTER

I've been meaning to get a daisywheel printer for some time, but I'd promised myself that I wouldn't until I could find one that printed at about 20 charactes per second and cost under £200. Imagine my delight to find a special offer at the Personal Computer World Show on Quest's stand of a Quen-Data DWP1120 printer at £199. It was a bargain too good to miss (though I opted out off lugging it home on the tube and train). It arrived last week. I plugged it into the PIO interface on my Corcomp card - nothing! Dismay! I plugged it into a Toshiba HX-10 MSX computer - it worked! Panic! A quick phone call to Howard Greenberg at Arcade Hardware resulted in me collecting interface, cable and printer and whizzing round for him to fix it. He checked the cable - OK. Howard tried his TI card - no success. Desperation! He tried it with a Myarc card - SUCCESS! Now the problem seemed to be the timing of the signals on the "Centronics" interface from the TI and Corcomp cards. There's apparently a new EPROM available for the TI card which cures the problem. Presumably there's a similar one (same?) for Corcomp. In the meantime I've joined Howard's happy Myarc customer list - otherwise this would never have got printed!
Before I start work on the usual screen dump programs and working out how to exploit the Quen-Data's graphics features, does anyone else own one? If you do, please write and let me know what software you've developed for it.

See you at Birmingham in October, I hope.

John Rice, 7 Lincoln Road, SWINTON, Manchester M27 3WR

YOUR LETTERS

Mr G.EGAN of Staleybridge, Cheshire writes:- Here is a helpful hint for those with Minimem, 32k exp and "Widget". When writing a program instead of "SAVE CS1" type "SAVE EXPMEM2" every ten lines or so. If you get a lockup a press of the reset button and "OLD EXPMEM2" has the program so far back in the memory in seconds! You can use "SAVE EXPMEM1" as well as save minimem IE three instant save and old files without the labourious use of CS1. You could finally save to CS1(or CS2) before switching off the console. As long as the console is turned on any info in 32K expansion is retained.

Richard SPEED of West Sussex writes:- Why don't you organise some sort of coach trip for TI Users in the Southern region to BIRMINGHAM SHOW. The only time I've ever been able to speak to another TI user was in Brighton, and two years with no contact is a long time.

ED: Sorry Richard but we just haven't the time to organise this as we are kept pretty busy with the newsletter, library, letters and the arrangements for the show itself. It is a very good idea for people to get together to make the tip to the show and save on costs. If you are going to the show, want a lift or can provide a lift please call one of the AREA CONTACTS below and perhaps something can be worked out.

George HUNT of Skipton, Yorks writes:- A special thank you, for all your time, work and effort in running TI*MES. Please keep up the technical side to TI*MES and gladly we shall pay bigger subscriptions for a larger magazine. Running a TI club from my home I am willing to help anyone should they live in or around this area.

ED: Many of our members have asked for larger or more frequent newsletters but we are faced with the dilema that increased costs may cause a decline in membership. One solution may be a once a year special magazine which can be purchased separately. We'll let you know when/if this is available.

Syd MICHEL of Faslane writes:- I have found a bug in the LINE by LINE assembler supplied with the MINIMEMORY.I have received many enquiries regarding the difference between OLD and NEW when entering the Line by line assembler.(See page 5 of the L byL leaflet paras B + C) This is what the diference SHOULD be but good ol TI cocked it up again and the L by L ass treats OLD and NEW as... NEW! ... Due to popular request I have had a look at the code and eventually I came up with the bug(s). Use EASY-BUG to alter the following location contents from the list below.

```
ADDRESS.......OLD CONTENTS........CHANGE TO.
>71A9 .......>3E .................  >8E
>71AF .......>3E .................  >8E
>7229 .......>3E .................  >8E
```

Having made these changes don't forget to save the changed program, the L by L ass should no behave itself (fingers crossed)

PLEASE REMEMBER IF YOU ARE WRITING TO ANY OTHER MEMBER OR A CONTRIBUTER TO THE MAGAZINE TO INCLUDE A STAMPED ADDRESSED ENVELOPE FOR YOUR REPLY.

AREA CONTACTS.

These are members who have the wish to have contact with other users in their area. Let us know if you wish to be included.

Henry Clark,60 St Pauls Road, New England,PETERBOROUGH, Cambs. 0733 42642
Harry Pridmore, 17 Jerrards Close, HONITON,Devon. 0404 41856
John Carter, 16 Sherwood Ave, NORTHAMPTON. 0604 842760.
Simon Pryce, 48 Mount Street, SHREWSBURY,Salop. 0743 67799. Interested in amateur radio
John Bingham, Rygghagen 7B, 4070 Randaberg, Stavanger, NORWAY. 04-599228
Stephen Meadows, Stonebrook, Old Forge Lane, Nutley,E.SUSSEX.TN22 3EL.
Richard SPEED, 18 The Spring, Burgess Hill, W.SUSSEX.
George Hunt, 44 Sharphaw Ave, Skipton, N.YORKS. Skipton 5840.
FORTH INTEREST. Stanley Dixon, 28 Grange Park Road, LEEDS, LS8 3BB.
PASCAL INTEREST. Stanley Dixon, 28 Grange Park Road, LEEDS, LS8 3BB.
Graham Hilton, 8 Sandwich Close, Saint Ives, CAMBRIDGE. 0480 65228.
Richard Owen, 17 Highfield Ave, Litchard, Bridgend, Mid-Glam, SOUTH WALES.
Alan Davey, 88 Halcombe Estate,Chard, SOMERSET. BULLETIN BOARD SUNDAYS 10am-10pm. Tel 04606 4511.
David Moerel, 6 Chyrose Road, St Day, Redruth, CORNWALL.
MR.B. Sholanke, WELWYN GARDEN CITY, HERTS. Welwyn 27232.
Andy Hopkinson, 16 Linden Wlak, Nth Baddesley, SOUTHAMPTON. 0703 732801.
G.Broomfield, 42 Layton Rd,Parkstone,Poole, DORSET. 0202 722542.(Interested in DIY Hardware)

BACK TO BASICS

TI*NIES PAGE

PROGRAM TO CREATE YOUR OWN SECRET CODED CRYPTIC MESSAGE

```
100 REM TI*MES LIBRARY
110 REM TI99/4 BASIC
120 REM AUTHOR UNKNOWN
130 REM CODE H090/1985
140 REM not to be sold
150 REM =============
160 @=1
170 _=2
180 CALL CLEAR
190 CALL CHAR(60,"0000000000
10102")
200 CALL CHAR(62,"00282828")
210 CALL CHAR(91,"0004081020
100804")
220 CALL CHAR(93,"0020100804
08102")
230 CALL CHAR(152,"FF8181818
18181FF")
240 FOR A=@ TO 8
250 CALL COLOR(A,4,4)
260 NEXT A
270 PRINT TAB(8);"KZACDXBZOC
TA":" ":" "
280 PRINT "DTEH CZXBZOO NEMM
 SGIGMXC O"
290 PRINT "SEZGKD HPRHDEDPDE
XW KACTOZ"
300 PRINT "QWS GWKXSG Q OGHH
QBG XU PC"
310 PRINT "DX WGOZMA HGIGW M
EWGH, XZ"
320 PRINT "ED NEMM OKKGCD QW
 GWKZAC-"
330 PRINT "DGS OGHHQBG."
340 PRINT "DTGW ED NEMM TGMC
 AXP DX"
350 PRINT "HXMIG DTG KZACDXB
ZOO."
360 FOR A=8 TO @ STEP -@
370 CALL COLOR(A,_,4)
380 NEXT A
390 FOR A=@ TO 400
400 NEXT A
410 A$="CRYPTOGRAPHY"
420 B=13
430 C=9
440 GOSUB 460
450 GOTO 510
460 FOR A=@ TO LEN(A$)
470 D=ASC(SEG$(A$,A,@))
480 CALL HCHAR(B,C+A,D)
490 NEXT A
500 RETURN
510 A$="THIS PROGRAM WILL DE
VELOP A"
520 B=16
530 C=_
540 GOSUB 460
550 A$="DIRECT SUBSTITUTION
CIPHER"
560 B=17
570 GOSUB 460
580 A$="AND ENCODE A MESSAGE
 OF UP"
590 B=18
600 GOSUB 460
610 A$="TO NEARLY SEVEN LINE
S< OR"
620 B=19
630 GOSUB 460
640 A$="IT WILL ACCEPT AN EN
CRYP-"
650 B=20
660 GOSUB 460
670 A$="TED MESSAGE."
680 B=21
690 GOSUB 460
700 A$="THEN IT WILL HELP YO
U TO"
710 B=22
720 GOSUB 460
730 A$="SOLVE THE CRYPTOGRAM
."
740 B=23
750 GOSUB 460
760 PRINT : :
770 PRINT "DO YOU WANT TO":"
(1)TYPE IN A MESSAGE TO BE
ENCODED":"(2)TYPE IN A CODED
 MESSAGE":" ":"TYPE 1 OR 2"
780 CALL KEY([,E,F)
790 IF F<@ THEN 780
800 IF E=50 THEN 1060
810 IF E<>49 THEN 780
820 CALL CLEAR
830 PRINT "CODE NOW BEING DE
VELOPED. BEPATIENT. IT TAKES
 TIME.": :
840 DIM G(26),B$(90),C$(90)
850 FOR H=@ TO 26
860 B$(H)=CHR$(H+64)
870 NEXT H
880 FOR H=@ TO 26
890 RANDOMIZE
900 G(H)=@
910 NEXT H
920 FOR H=@ TO 26
930 C=INT(26*RND)+@
940 IF G(C)=@ THEN 970
950 C=C+@
960 IF C>26 THEN 930 ELSE 94
0
970 C$(H)=CHR$(C+64)
980 IF C$(H)<>B$(H)THEN 1000
990 IF C$(26)=B$(26)THEN 850
 ELSE 930
1000 G(C)=[
1010 NEXT H
1020 FOR H=@ TO 26
1030 B$(H)=C$(H)
1040 NEXT H
1050 CALL CHAR(44,"FFFFFFFFF
FFFFFFF")
1060 PRINT "READY. TYPE UP T
O 6 AND 1/2 LINES. DO NOT HY
PHENATE. USESHIFT [ FOR COMM
A AND SHIFT"
1070 PRINT "] FOR QUOTATION
MARKS."
1080 PRINT "HIT ENTER AT END
 OF 4TH LINEAND WHEN FINISHE
D."
1090 INPUT A$
1100 IF (LEN(A$)=109)+(LEN(A
$)=110)THEN 1110 ELSE 1130
1110 INPUT D$
1120 A$=A$&D$
1130 IF E<>50 THEN 1160
1140 E$=A$
1150 GOTO 1280
1160 A$=A$&"X"
1170 PRINT : :"WAIT< PLEASE.
.."
1180 I=LEN(A$)
1190 J=ASC(A$)-64
1200 IF J>[ THEN 1250
1210 F$=SEG$(A$,@,@)
1220 E$=E$&F$
1230 A$=SEG$(A$,_,I)
1240 IF LEN(A$)=@ THEN 1280
ELSE 1190
1250 A$=SEG$(A$,_,I)
1260 E$=E$&B$(J)
1270 IF LEN(A$)<>@ THEN 1190
1280 K=LEN(E$)
1290 DEF L(M)=ASC(SEG$(E$,M,
@))=32
1300 IF K<29 THEN 1500
1310 K=LEN(E$)
1320 N=N+@
1330 O=28
1340 O=O*N
1350 IF K<=O THEN 1460
1360 IF L(O)+L(O+@)THEN 1320
1370 IF L(O-_)=[ THEN 1400
1380 E$=SEG$(E$,@,O-_)&" "&
SEG$(E$,O-@,K)
1390 GOTO 1440
1400 IF L(O-@)=[ THEN 1430
1410 E$=SEG$(E$,@,O-@)&" "&S
EG$(E$,O,K)
1420 GOTO 1440
1430 E$=SEG$(E$,@,O-@)&"-"&S
EG$(E$,O,K)
1440 K=LEN(E$)
1450 GOTO 1320
1460 IF K<197 THEN 1500
1470 PRINT "MESSAGE TOO LONG
"
1480 E$=""
1490 GOTO 1060
1500 CALL CLEAR
1510 PRINT "TO TRY SUBSTITUT
ING A LETTERAS FOR EXAMPLE T
O SUBSTITUTEA FOR Y< TYPE Y=
A. THEN< IF"
```

```
1520 PRINT "YOU DECIDE IT'S
WRONG< TYPE X=X.": :"  WAIT
FOR THE TONE BEFORE":"TYPING
 ANOTHER CHANGE.": :
1530 PRINT TAB(5);"PRESS ANY
 KEY"
1540 CALL KEY([,E,F)
1550 IF F<@ THEN 1540
1560 CALL CLEAR
1570 FOR P=9 TO 12
1580 CALL COLOR(P,_,10)
1590 NEXT P
1600 CALL COLOR(13,16,16)
1610 CALL COLOR(14,8,8)
1620 CALL COLOR(15,12,12)
1630 CALL COLOR(16,_,16)
1640 FOR A=@ TO 7
1650 CALL HCHAR(A,@,128,32)
1660 NEXT A
1670 FOR A=8 TO 14
1680 CALL HCHAR(A,@,136,32)
1690 NEXT A
1700 FOR A=15 TO 17
1710 CALL HCHAR(A,@,144,32)
1720 NEXT A
1730 IF K>140 THEN 800
1740 B=_
1750 Q=9
1760 R=19
1770 GOTO 1810
1780 B=@
1790 Q=8
1800 R=18
1810 C=3
1820 S=Q
1830 FOR A=@ TO K
1840 T=ASC(SEG$(E$,A,@))
1850 IF T>=65 THEN 1880
1860 M=T
1870 GOTO 1890
1880 M=152
1890 CALL HCHAR(B,C,T)
1900 CALL HCHAR(Q,C,T)
1910 CALL HCHAR(R,C,M)
1920 C=C+@
1930 IF C<31 THEN 1980
1940 C=3
1950 B=B+@
```

```
1960 Q=Q+@
1970 R=R+@
1980 NEXT A
1990 B=16
2000 C=3
2010 FOR U=65 TO 90
2020 CALL HCHAR(B,C,U)
2030 C=C+@
2040 NEXT U
2050 CALL SOUND(300,500,[)
2060 CALL KEY([,E,V)
2070 IF V<@ THEN 2060
2080 CALL SOUND(50,800,5)
2090 IF (E<65)+(E>90)THEN 21
00 ELSE 2130
2100 IF E=61 THEN 2060
2110 CALL SOUND(200,110,[,-@
,[)
2120 GOTO 2060
2130 W=W+@
2140 ON W GOTO 2150,2170
2150 X=E
2160 GOTO 2060
2170 Y=E
2180 W=[
2190 FOR Z=@ TO K
2200 AA=ASC(SEG$(E$,Z,@))
2210 IF AA=X THEN 2240
2220 NEXT Z
2230 GOTO 2050
2240 AB=S+INT(Z/28)
2250 C=Z+_-(INT(Z/28)*28)
2260 IF C<>_ THEN 2290
2270 C=30
2280 AB=AB-@
2290 AC=AB+10
2300 IF X<>Y THEN 2340
2310 AD=Y
2320 AE=152
2330 GOTO 2360
2340 AD=Y+32
2350 AE=AD
2360 CALL HCHAR(AB,C,AD)
2370 CALL HCHAR(AC,C,AE)
2380 CALL HCHAR(17,X-62,AD)
2390 GOTO 2220
2400 END
.FI;AD;
```

TI*NIES PAGE !!

LIBRARY REVIEW

by Graham Hilton

REVIEW: E25 FLAGS OF COUNTRIES EX.BASIC, SPEECH SYNTH. REQUIRED.
This is a program which could be of great interest to students and anyone with an interest in geography and general knowledge.
the program generates pictures of about 25 flags belonging to various countries of the world and you are invited to type in which country the flag belongs to. If you do not know, you can press ENTER and the answer will be printed with the Capitol City also. The various flags are drawn quite quickly and accurately, there are also spoken comments from the speech synth, which makes it that bit more interesting. Extensive use is made of SUBPROGRAMS in Extended basic. This could provide an interesting way to brush up on general knowledge of countries.

REVIEW: E25+ FLAGS OF COUNTRIES VOLUME II EX.BASIC, SPEECH SYNTH. REQUIRED. The same program as E25 but with another 25 or so flags of other countries. Well recomended for general knowledge.

REVIEW: G37 FLIGHT SIMULATOR EX.BASIC.
This program attempts to simulate flying an airplane by instruments only. There is no view of anything except the control panel. Included with this prog is a another short prog which when LISTED enables you to find out which keyboard controls work the various controls on the aircraft. eg. throttle elevators,rudder etc.
I found this program extremely difficult to play. I am not your average Biggles, and I must admit that I have yet to get the plane off the ground! I always manage to either run out of runway or crash... It should provide a challenge to anyone with a fair degree of patience.

REVIEW: TIWRTLOAD EX.BASIC and full system required.
This exellant utility provided by Funnelweb Farm of Australia is an extended basic loader for TI WRITER! If this prog is put on the same Disc as your TI WRITER Disc, it will allow you to load TI WRITER from EXTENDED BASIC!!!
It also has a Disc cataloger as well. It will no doubt save a lot of wear and tear on the cartridge port as you can just leave EX.BASIC in most of the time, instead of having to bung in the TI WRITER module. (ED: This program is FREE to all TI99/4a Exchange members who at the same time request disk based programs from the TI*MES library, it comes with complete documentation, program is written by TONY McGoven who requests that charges are not made on this.)

REVIEW: UTILITY option EX.BASIC and full system and TIWRTLOAD (part of above)
An extremely useful utility again from Funnelweb Farm. This can be loaded from the UTILITY option 3 of TIWRITER E/B LOADER. (ED: Will load E/A based programs either TI-Writer or Graphics can be selected). UTIL1 is a Disc formatter, again saves wear and tear on the old cartridge port. (ED: This program has been replaced with a EX Basic DISK MANAGER program which we have received from THE OTTAWA TI99/4a Users CANADA. It has all the functins of Disk Manager 2 module but much improved with extremely FAST operations are available. This is distributed as FREEWARE, again no charge but would ask that donations are sent to the writer of both programs.) I found this to be really useful for me, because I have a double sided Discdrive which has been connected in a way which the computer thinks I have 2 separate drives connected ie. Disc 1 and Disc 2. The program will format these two drives but it will also format it as a DOUBLE SIDED drive! This really expands my options as in effect I can choose to have two single drives (side 1, side 2) or one Double sided drive. fantastic!!!

CURSOR

By Dot Matrix

(AGONY PAGE FOR BOTH READER AND WRITER)

I've been crowded out of the last few issues by more venerable and certainly more masculine writers but I am happy to relate that I have not been wasting my time. I have just completed a short MSC sponsored course called "Access to Information Technology". Wow I hear you say - heavey. Well it can certainly be a conversation stopper at the Playgroup Coffee morning.

The course had everything going for it:- on when the kids were safely locked up in school, cheap and not far from Sainsburys so I could do my shopping afterwards (or even during if it all got too much!). The pamphlet in the library showed a picture of a woman sitting in front of a computer monitor so I reckoned it should come somewhere within capabilities. After all one does have to enlarge one's mind hasn't one, so my pea-like appendage should become at least the size of a sprout, or should I say small cabbage?

I learned that Information Technology is "the devices and techniques used to store, process, manage, transmit and communicate information. I encompasses various technologies such as computing, microelectrics and telecommunications" Great but what does it MEAN. During the next five weeks we were given "hands on" experience with a BEEB each to play around with. Not of course as good as the TI fans, you can't turn the sound down for a kick off. We played with a word processor, a beebuggy, a graphics tablet, modems, Prestel and a £54,000 engineering machine which could be pogrammed to produce any shape in a few seconds - they wouldn't let us get our "hands on" that one! We toyed with robotic arms, one built from a kit and the other which looked as if it could either produce or devour a car in one hour flat.

Each session was accompanied by a Ian Mcthingy Hyphen Davis video to show the layman and woman what the BBC computer was capable of, providing of course that you have a second processor, disc drives, electronic etcs.... It left me with feelings of wonderment, not the least of which was how much Mc was getting in Royalties.

Encouraged by the course and hungry of more of this new technical revolution I gate crashed a CAD/CAM exhibition (got in by pretending to be someone's sekketary). CAD stands for computer assisted design. This can range from architects plans to complex circuit diagrams; reproductions of Old Masters to advertising artwork.Bet you didn't know that Pampers nappies logo was designed on a computer. It was fascinating to watch the great machines at work.

To complete my education this summer I went to the other extreme and visited the PCW show in London. My memories of this are as follows. Noise. Heat. Crowded staircases. Crowded aisles. Carrying huge bag containing sandwiches, drinks, badges, stickers, paper hats and numerous leaflets. Seeing Rambos on lots of stands. This must have been everyone's original show stopper this year. Remembered thinking that the screen Rambo was surely taller than the Rambo lookalike on one stand who was only 5'3" or was that just his chest measurement. Thinking where are the kids going to put all these badges and stickers. Winning a tee shirt on the PCW stand answering a highly intellegent question,being too embarrassed to ask for a large one for self and ended up with a small one for the kids. Hoping they'll take it in turn to wear it. And the most memorable part - this is the only function I have attended where there has been no queue in the Ladies toilets!

Dot.

BULLETIN BOARD

ADVERTS

SALE OF MODULES.
PARSEC £10.95.
TI.INVADERS £6.95.
MUNCHMAN £3.95.
ADDITION/SUBTRACTION 1. £5.95.
AMAZE-ING (FEW LEFT) £5.95.

****

TI99/4A DUST COVERS. Best selling standard cover in semi-clear durable PVC with bound and stitched edges £3.25+35 P+P. **De luxe cover** in lined leathercloth with clear keyboard panel. (RRP £4.99) £4.00 +35 P+P.

TV AERIAL SPLITTER (SWITCHED) will prevent wear on TV socket and cable. Simply flick a switch to £2.25 +25 P+P

****

T.I.BOOKS.

KIDS AND THE TI99/4a - Datamost.
Learning Basic and the magic tricks of the TI can be great fun so why learn any other way. The drawings help you understand and remember the material so you'll be writing your own programs by Lesson 5 and forever after. Learn to write computer music
create computer graphics and drawings
make your computer talk + ask questions
create games and quizzes.
edit and debug programs fast
organize your mind and think creatively
and best of all **it's fun.**
NOT JUST FOR KIDS!
236 pages and spiral bound.
£6.00 including post and packing.

GETTING STARTED WITH THE TI99/4A by Stephen Shaw....£4.50

MASTERING THE TI99/4A by Peter Brooks.£5.50 or **both** these books **for only £9.00** These prices include postage and packing.
Send Order to COMPUTER HOME SERVICE, 40 BARRHILL PATCHAM BRIGHTON BN1 8UF.
PLEASE ALLOW 28 DAYS TO PROCESS

```
1. IF YOU PRESS(ENTER)OR CTRL(M)WITHOUT
   A PAGE(LETTER) OR RESELECT(R) PREFIX
   YOU WILL RETURN TO THIS WELCOME PAGE
=======================================
2. YOU CAN USE THE DATA CTRL SEQUENCES
   AS LISTED IN THE TE2 MANUAL. QUICK-
   REF-GUIDE PAGE.1 DETAILED PAGE.16-19
=======================================
3. EG.IF YOU WISH TO DUMP THE NEWS TO A
   PRINTER THEN LOGOFF DO THE FOLLOWING
  >A.PRESS RESELECT(R)AND ENTER. GOTO C
  >B.SELECT PAGE(LETTER)AND PRESS ENTER
  >C.OUTPUT(CTRL 2)PAGE TO THE PRINTER
  >D.REPEAT B AND C ABOVE TILL FINISHED
=======================================
4. EXIT.(LOGOFF)THE 4/ABC NEWS ANYWHERE
   SIMPLY TURN YOUR MODEM TO  OFF-LINE!
   THEN CHECK PHONE FOR A DIALLING TONE
   **THE REVIEW KEYS(PAGE 25)WILL STILL
   BE ACTIVE AFTER YOU HAVE LOGGED OFF!
=======================================
   PRESS R AND ENTER FOR SELECTION PAGE
```

```
ENTER     4/ABC NEWS SELECTION PAGE
=======================================
  A.       TEST PAGE ONE
  B.       TEST PAGE TWO
  C.       TEST PAGE THREE
  D.       TEST PAGE FOUR
  E.       TEST PAGE FIVE
  F.       TEST PAGE SIX
  G.       TEST PAGE SEVEN
  H.       TEST PAGE EIGHT
  I.       TEST PAGE NINE
  J.       TEST PAGE TEN
  K.       TEST PAGE ELEVEN
  L.       TEST PAGE TWELVE
  M.       TEST PAGE THIRTEEN
  N.       TEST PAGE FOURTEEN
  O.       TEST PAGE FIFTEEN
  P.       TEST PAGE SIXTEEN
  Q.       TEST PAGE SEVENTEEN
  R.       RESELECT THIS PAGE
=======================================
    PRESS R OR PAGE LETTER AND ENTER
```

```
   WOULD YOU LIKE TO FILL THIS PAGE
   --------------------------------
WITH NEWS VIEWS OR INFORMATION ON THE 4A

DO YOU HAVE ANY HINTS OR TIPS TO PASS ON

MAYBE A QUESTION OR A PROBLEM TO PUT UP

ARE YOU LOOKING FOR ANY 4A ITEMS TO BUY
DO YOU HAVE ANY ITEMS YOU WANT TO SELL

OR DO YOU JUST WANT YOUR NAME(INTERESTS)
AND PHONE NUMBER PUT UP SO THAT OTHER 4A
USERS CAN CONTACT YOU IF THEY WISH TO

THEN CALL 4/ABC NEWS ANYTIME(NOT SUNDAY)
========================================
PRESS R OR PAGE LETTER AND ENTER
```

DIAL 04606-4511

OFFICIAL TI99/4A MODEM EXPERT **WANTS TO HEAR FROM YOU PLEASE SUBMIT NEWS AND VIEWS FOR TI 99/4A B C NEWS.** MORE DETAILS **IN THE NEXT** ISSUE OF TI*MES
DIAL 04606 4511 FOR DETAILS

TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES
TI*MES small adverts **small adverts** small adverts TI*MES
TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES

>Swap Standalone CORCOMP RS232 for PE BOX +RS232. >TOM O'SULLIVAN,LIMMERICK PHONE 061-393163

>FOR SALE ALPHACOM 42 PRINTER. C/W DIRECT CONNECT I/F. for TI99/4a only 12 months old. £95. >(05436) 3551. Ask for DOUG WARD.

FAMILY TREE program in T.I.Basic. Research your forefathers. Will hold 200 entries. Facilities to search,modify, add to, multi + single entry display. Compatible with Genealogical Research Society's layout. Runs on standard TI99/4a. Program tape, listing and information sheet(includes addresses) £2.50.
Derek Ford. Tel 021 430 5484.

FOR SALE -MINI-MEMORY: Boxed as new. Complete with manuals and assembler. Plus Texas Assembly Language publication. £36 or swap. Tel. 0532 642619.

FOR~SALE- TI99/4a computer, extended basic, joysticks, Supersketch, various games/modules plus many TI books. A real bargain £198.00. Tel. 061-902-0430.

FOR~SALE- MINIMEMORY plus Assembly language manual £30. 12 consecutive copies of 99er magazine £14 contain many programs and articles. Bargain.
Telephone Northwich(0606) 782276.

FOR~SALE - TI99/4a computer, Ext. basic, joysticks, games and books. (Will separate) £100. Tel. Swindon 0793 751446

FOR~SALE- TI 99/4a computer together with instruction books and tapes, some user group magazines, all connecting leads and dust cover. Tel Bexhill 220443

FOR~SALE- TI 99/4a, 32K Expansion, RS232, disk controller, Disk drive (internal), Ex.bas and TI Writer. £400 the lot (cannot split) Ring 021 773 1009

MODULES:- Extended basic, TE 11, Touch Typing tutor. CASSETTES:- Pentathlon, Trainer Plane, Lionel and the Ladders, Monbase 5, Floor planner. All reasonable offers accepted. Tel (0308)22865 evenings.Ask for Nigel

WANTED~UCSD~PASCAL/P-CODE~SYSTEM. Contact Peter Calcraft, 13 Royal Mews, Princes St, Dorchester,Dorset DT1 1RL. 0305-67658.

FOR~SALE £525? Expansion box, FS232/32k memory/disk controller cards. Internal disc-drive, disk manager 11. Also TI Forth, Display enhancement package, minimemory, Ed.Ass, Terminal Emulator 11, Personal record keeping and misc. games(modules, tape and disk).
Tel 0296 31509 evenings/weekends.

FOR SALE >Standalones RS232 interface + 1 disk controller + Tachyon 32k memory. >£200. the lot may split phone 01 643 4643.

WANTED ~ to buy or hire for photocopying, PASCAL assembler/linker section of the Texas Instruments USDC PASCAL manual. ~ V.Comley, Dormer Cottage, 7 St Vincents Hill, Redland, BRISTOL BS6 6UP.

COMPLETE SYSTEM+TI99/4a FOR SALE. Expansion Box, 32k, RS232, Disk drive, Shinwa printer, EX Basic, Minimem, TIwriter, E/A module, TE2 + 30 modules +30 Disks. £500.00 ONO. >Telephone Clevelys 852096

TI*MES back issues still available are ONLY issues 4 ,5,8 + 9. They cost only £1 each to members. TI99/4A EXCHANGE, BRIGHTON.

MINIMEM Conversion to Rechargeable Battery. Send Minimem and Crossed cheque £7.50, To- N.J.Petry, Tensal Technology, 3, Lester Drive, WORLE,W.S.M., Avon, BS22 0NG.

FOR~SALE Parsec -£10. Home budget management -£5. Brand new, unused in original boxes with guarantee and intructions. Tl 0604-46194.

ADVERTISE ON THIS PAGE FOR 5P A WORD. (MEMBERS ONLY) SWAP SHOP ADS FREE!Trade Advertisements Rates on application.

Nationwide
TI 99/4a USERS SHOW
26th October 1985
CIVIC HALL DIGBETH
BIRMINGHAM.

doors open 10.00am to 5.00p

Admission by ticket free to members. Guests £1.00

Come and see what is new, always a good get together. There will be Auctions(a chance to sell your goodies to other TI99/4a Users),displays,a competition, lots of bargains, and of course major support by ---->>>>

ARCADE
HARDWARE

Stainless

PARCO
Electrics

Oxon TI Users.

COMPUTER HOME SERVICE
******** **** *******

TI99/4a Exchange

Local Groups

[illegible]

Bring your Christmas shopping list
lots of TI99/4a goodies from Francis Farrish
and Howard Greenberg (Arcade). >>> Your support will
of course ensure that the show continues next year <<