



NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 9, Number 3

April, 1990

Registered by Australia Post - Publication No. NBH5933

RENEW NOW!

& FOR ONLY \$30

YOU CAN HAVE ACCESS TO FIXES LIKE THIS!



Subscriptions now due

P.O. Box 214, Redfern, New South Wales, Australia, 2016

\$3

TIsHUG News Digest

April 1990

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board**Co-ordinator**

Dick Warburton (02) 918 8132

Secretary

Terry Phillips (02) 797 6313

Treasurer

Geoff Trott (042) 29 6629

Directors

Rolf Schreiber (042) 84 2980

Russell Welham (043) 92 4000

Sub-committees**News Digest Editor**

Bob Relyea (046) 57 1253

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

Merchandising

Percy Harrison (02) 808 3181

Publications Library

Warren Welham (043) 92 4000

Software library

Rolf Schreiber (042) 84 2980

Technical co-ordinator

Lou Amadio (042) 28 4906

Regional Group Contacts**Carlingford**

Chris Buttner (02) 871 7753

Central Coast

Russell Welham (043) 92 4000

Coffs Harbour

Kevin Cox (066) 53 2649

Glebe

Mike Slattery (02) 692 0559

Illawarra

Lou Amadio (042) 28 4906

Liverpool

Larry Saunders (02) 644 7377

Northern Suburbs

Dennis Norman (02) 452 3920

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$30.00
Overseas Airmail Dues A\$60.00

TIsHUG Sydney Meeting

The next meeting will start at 2.00 pm
on 7th of April at Ryde Infant School,
Tucker Street, Ryde.

Printed by
The University of Wollongong
Printery

Index

Title	Description	Author	Page No.
Apple peeks	Software hints	Callaghan, Mike	17
c99 function library	Software hints	Agee, Warren	27
Co-ordinators report	General news	Warburton, Dick	2
Disk peripheral	Specifications		21
Editor's comment	General interest	Relyea, Bob	1
Extended BASIC Tutorial	Software hints	McGovern, Tony	7
Forth to you too!	Software hints		24
Four-A/talk	Software hints	Gaskill, Bill	9
From the bulletin board	Mail to all		22
From the service bench	RAMdisk	Trott, Geoff	29
Hi resolution graphics	Software hints	Shaw, Stephen	26
How does it work?	General interest	Trott, Geoff	29
Newsletter update	General interest	Relyea, Bob	31
Program to type in	Grue stew		15
Programs from TI*MES library	Software review	Shaw, Stephen	25
RAMdisk folly	General interest	Forthwrite	28
Regional group reports	General interest		31
Reviews	General interest	Shaw, Stephen	23
Secretary's notebook	Club news	Phillips, Terry	2
Small list	Software hints	Rebel, E.P.	27
Sorcerer part 2	Adventure hints	Scorpio	10
Techo time	Multifunction card	Amadio, Lou	5
Tetris, a game	Software review		30
TI Artist plus	Software review	Shaw, Stephen	14
TI bits #13 and 15	Software hints	Swedlow, Jim	18
Tips from the tigercub #40	Software hints	Peterson, Jim	19
TIsHUG software column	Club software	Schreiber, Rolf	3
Wit and Wit Out	Software review	Lang, Chris	11
Younger set	Program, adventure	Maker, Vincent	17

TIsHUG Fairware Author of the Month

The Fairware Author for this month is Barry Boone for his Program Archiver. The latest version available from the shop is version 3.03. All Donations collected at the meeting and sent in will be mailed to him this month.

Editor's Comment**Articles for TND ?**

by Bob Relyea

I was reading an article by Garry Christensen of the Brisbane Users Group last night about user groups in general. The article concerned may be in this month's issue. At any rate, it prompted me to write about something that was on my mind regarding the submission of articles for the club magazine. In the article mentioned above, Garry tries to encourage everyone who is a club member to write up something for inclusion in one of the issues of the group's magazine. He even goes to the extreme of saying that it does not matter if it is 'gibberish', just as long as you write something. Although I would not encourage the inclusion of gibberish in our fine magazine, the point that he was making was clear enough. Many of us may have the impression that only the ones with a lot of technical knowledge can get articles into the magazine. I would like to assure you that this is not true. The object of this article is to explore ways of submitting articles for the TND.

One of the main points that I want to make, especially for the sake of the 'younger set' and for members of any age who do not have an expanded system, is that you do not need an expanded system or word processing facilities to prepare an article. That is right! All you really need is a bare console and a cassette recorder. Let me repeat, YOU ONLY NEED A CONSOLE AND A CASSETTE RECORDER to prepare an article. In fact, one person who regularly submits articles for the TND is a cassette user only. So the next point is to explain how you would go about it all with a cassette recorder.

continued on page 4

Co-ordinator's Report

by Dick Warburton

I was pleasantly surprised at the number of club members who turned up at the March monthly meeting. I was delighted at the spirit which pervaded the meeting. Members seem to be getting some of the things they need from the meetings. Though the start was shaky, because I could not get my system to work properly, (I left most of the important parts sitting on a table at home), and my ramdisks crashed, and we were unable to demonstrate the eprom ramdisk as we had hoped, etc. we still seemed to have a really busy enjoyable day. My problems did not end there. I am still trying to put the monitor interface circuit together so that it actually works. At the meeting I saw what it is supposed to do. I have had one faint line on the screen so far plus some squawks from the speaker. First I read the Dick Smith catalog wrongly, and put all 14 transistors in back to front. I was sure it would now go, but alas, nothing. I checked it all again to find I had the PNP transistors in wrongly. Back to the soldering iron, and another quick surgical transplant. Certain of success, I plugged in and switched on. Nothing! Back to the drawing board. Well, careful inspection uncovered a short. I knew I had it right this time. A little more surgery and I was on the right track. I hope brain surgeons do not have weeks like this. Just think of the possibilities in among the neurones. Another failure! I reckon I have spent at least one full week of my present life trying to get this interface to work. Now when we get our project group operative, we can help check each other's work. I am sure that it is easy to make mistakes which are hard to recognise, because we get used to them. One day you may hear the cheer from our household. Your coordinator has his monitor going well.

This fixing and making things is addictive. I suppose there are other computer groups which do things for themselves, but I would be very surprised if there are more than a few in any group but ours which actually do the modifications and work themselves. TI members seem unique. We take it for granted that we can do many things for ourselves. I wonder how many Tishug members get printer or modem cables made up commercially, or throw disks away because files are corrupted. It is quite amazing to think about the range of skills we have developed in our club. Some have learned to fix consoles, many have successfully built 32K expansions, or successfully completed ramdisks. Our members have designed the circuits, done the artwork, organised the kits, and helped each other to complete projects successfully. I do not know any other club like ours. I hear about computer club meetings where members spend a lot of time sharing software, playing games, or simply watching commercial demonstrations of software. We have something unique and something to be proud of. Hopefully, the spirit evident at the March meeting will grow as more people gain more satisfaction from their own efforts in the club. The more we put in, the more we personally get out of membership. I am finding my usage of the TI becoming more satisfying, because I am gaining more skills, and more understanding than I thought possible. I do not think that I will ever be a really competent programmer, but I now know which end of a soldering iron to use, and hopefully can perform simple miracles like resurrecting dead consoles. We need some more new members. If we can increase our membership we can pay our way better and provide better services for our members. Cost per head rise as our numbers drop. We have approximately 200 members. Imagine if each member brought one new person into the club per year. Our numbers would grow steadily. If you know someone with a TI, or an ex member, tell them what we are doing, and encourage them to come to a meeting. Our next monthly meeting should be a beauty. There should be bargains galore. Bring your money with you. We will advertise this meeting, and hope to attract some new blood. If you are new to a meeting, make sure you come and say hello to myself or one of the committee members. We want everyone who comes to feel welcome. Make sure of course that you renew your membership fees \$30.00 this month.

On a different topic altogether, I am trying to get together some systems and software to be used in kindergartens and with young slow learning children. I am interested in buying for the school or kindergarten appropriate software modules, consoles with power supply and PAL encoder, as well as speech synthesisers. Any donations will be gratefully received also. I also wish to build up a library of appropriate educational software on disk, for school use. I see a really useful role for the TI at school. The early learning programs are really good. The ability of the TI to use a TV screen is appreciated at school. The TI's graphics on a TV are far superior to the 2GS we use. I was appalled using a 2GS to see how blurred and indistinct the text became on a large TV. It was almost impossible to read. The TI leaves it for dead. If we can get some more TIs being used in the school, we might arouse some local interest among parents in the TI. If you have some practical ideas for publicising our club and its activities, let me know, and we may be able to implement them. If you want to join the growing numbers of members who are taking an active interest in helping out, tell me that you want to help. We will find you a useful and rewarding task. By the way, if you hear of any real bargains for our members, let the committee know so we can hunt them up for the club.

With membership renewal coming up, let me quote the words of the immortal bard:

"When you are on a good thing, stick to it".

See you at the next meeting.

Dick Warburton.

Secretary's Notebook

by Terry Phillips

Well it certainly looks like we are on the right track moving the meeting venue to the Ryde Infants School. The March meeting was one of the best attended for years with 65 to 70 members and visitors being present. By the look of it most seemed to be enjoying themselves and certainly the shop did a roaring trade. The next meeting will probably be even bigger as members come along to the BUY SWAP SELL day in an attempt to offload their unwanted items and perhaps even pick up something they have been after at a bargain price.

It was nice to see Katherine Rebikov, one of our few lady members at the meeting. Hope you can get along to future meets as well, Katherine.

I had a long chat with life member and former Secretary, John Robinson on the telephone a few weeks back. John was all set to come along to the AGM when illness in the form of a ruptured appendix sidelined him. John was calling from his hospital bed where he was recovering from surgery after removal of the offending appendix. Hope by now you are well on the way to recovery, John, and that now the meeting venue is much closer to your home, you can get along to some of the meetings and renew old acquaintances.

We have two new members to give a big welcome to this month -

Stephen Marsden - Condell Park
Garry Keats - Bowral

Hope you can both make it along to the meetings and join in the groups activities.

Rolf Schreiber has now completed his task of evaluating local and foreign newsdigest exchanges with the result that in some cases exchanges will be discontinued, while in other cases this group will seek exchange on a membership basis. Chief reasons for this are, of course, rising costs associated with production and mailing of our own newsdigest and the poor quality of some we receive in exchange. Rolf is contacting each of the affected groups explaining the situation to them.

continued on page 20

TiSHUG Software

Column

by Rolf Schreiber

At the March meeting I formally took possession of the TiSHUG Software library. I am presently engaged in becoming familiar with the library's contents as well as putting together disks suitable for this month's distribution. A revised software library catalogue should be available shortly.

The library consists of two series of software, comprising around 500 disks.

Series 1 consists of 230 volumes (SSSD disks), with some volumes combined on DSSD disks. This series contains most of the early software that was obtained from other User Groups around Australia and overseas. It is mainly Public Domain software, some of which has not seen the light of day in years. However, for members new to the group, this software is worth a second look. Also, most of this software is suitable for transfer to cassette for Users without 32K memory expansion.

Series 2 consists of over 360 volumes and is increasing in number every month. This series contains mainly disk-only software, but also has software suitable for downloading to cassette for those Users without a disk drive but with a 32K memory expansion. Most of the monthly software releases for sale in the shop are drawn from this library. As well there is commercial software which is not available for distribution, except for sale under licence. Such items are clearly marked in the software catalogue.

It is my intention to make the public domain software readily available to all the regional groups. With this in mind, I would like to hear from a member from each regional group who is disk based (preferably someone with a double density controller card and dual drives) who would be interested in taking on the job of Regional Group Public Domain Librarian. I would also like to hear from isolated TiSHUG members in outback NSW, or interstate with regard to a similar proposal. The job would entail duplicating public domain software (as well as FAIRWARE) for local TiSHUG members (and encouraging any non members or past members to join TiSHUG) as well as transferring programs to cassette for those TiSHUG members without disk drives. I would also like to hear from cassette only users (who cannot make it to our main meeting or any regional group meetings) about their software needs.

Software releases for March 1990

DISK A358 is the complete FUNNELWEB V4.21 upgrade (Fairware from Tony McGovern), now available as one disk in DSSD format and as 2 disks for those without a double density controller card. Disk A358(A) is DSSD and contains all the programs, while A358(B) is SSSD and contains all the documentation. Included on this release is version 4.2 of John Birdwell's Disk Utilities. Since this did not originate from Tony McGovern, it was not present on the disks sold at the March meeting - if you bought the disk then should see Percy Harrison about getting an exchange disk if you want the latest version of Disk Utilities as well. Please remember that both Funnelweb and DSKU are FAIRWARE, and if you use these programs, you should make a donation (through TiSHUG, or direct to the authors) if you have not already done so. If you have never made a substantial contribution to the author of either of these excellent programs in the past, any amount in the region of \$5 - \$20 would be considered adequate.

DISK A361 is called TEACH YOURSELF BASIC, an original TI software release comprising ten lessons. It comes on a DSSD disk and photocopied documentation is available on request for an additional \$1.00. This disk is also available on cassette by prior arrangement. Anyone using these programs should be

aware that references are made to the TI99/4 computer, which used different keys for some of the functions.

DISK A362 is BEGINNER'S BASIC TUTOR, also originally from Texas Instruments. It comes as a SSSD disk and contains 8 tutorial style lessons. The material covered is similar, but not identical to that found on disk A361. This disk is also available on cassette by prior arrangement, while photocopied documentation is also available for an additional \$1.00.

DISK A370 is Volume 4 Number 1 of the HOME COMPUTER MAGAZINE 'ON DISK' series, and features all the TI99/4A programs found in that issue of the magazine itself. Originally the disks were only available to subscribers to HCM, and were never widely circulated. These programs are all of excellent quality and well worth a look. Included on this SSSD disk is APOLLO (TI LOGO), LARRY'S TEN FIDDLE TUNES (music in BASIC), MELTDOWN (Extended BASIC game), MUSIC ASSEMBLER (music writing utility requiring a Mini Memory module), MUSIC MAGIC (music in Extended BASIC), SLOTS (BASIC game) and TOWER OF HANOI (a brain teaser in BASIC). This disk is also available on cassette.

DISK A375 is Volume 5 Number 1 of the HOME COMPUTER MAGAZINE 'ON DISK' series, and features all the TI99/4A programs found in that issue of the magazine itself. Included on this SSSD disk is ORBITAL DEFENDER (a game in BASIC, also an Extended BASIC version), ELECTRONIC BACKGAMMON (Extended BASIC gambling game), KORS-ELF (typing skills program in Extended BASIC), PERSONAL LOAN CALCULATOR, ORGANIZER (a menu driven series of Extended BASIC programs designed to organize your thoughts on any subject), QUIZ-PRINT, RAZZLE DAZZLE (computer graphics in BASIC or Extended BASIC), and LOGO SAILING. This disk is also available on cassette (with the exception of the 'ORGANIZER' programs, because of size restrictions)

Software releases for April 1990

DISK A1 contains a rapid loader (in Extended BASIC) for the INFOCOM series of text adventure games, as well as a utility to print out the vocabularies of the various INFOCOM adventures. Also included for your enjoyment is a sample game/tutorial on four of the early INFOCOM adventures. The disk is DSSD and is not available on cassette.

DISK A78 is the EXTENDED BUSINESS GRAPHS package, originally available from GREAT LAKES SOFTWARE in USA. This company is no longer in business and the package is no longer commercially available. The disk is SSSD and automatically loads from Extended BASIC. The program produces graphs in various formats from entered data eg bar and pie graphs, and also allows these graphs to be printed out. Photocopied documentation is available on request for \$2.00. This professionally produced package is highly recommended for anyone needing to display any information in a graphical format.

DISK A145 contains all the 12 adventures released by TI for the ADVENTURE module. The format is DSSD and all the programs on the disk require the adventure module or adventure loader program, available on request. A photocopied hints booklet is available on request for \$2.00 extra. This disk is also available on cassette.

DISK A325 is the TEACH YOURSELF EXTENDED BASIC tutorial package originally released by TI. The disk is SSSD and includes a great demonstration program, as well as 7 tutorial style lessons. A photocopied manual is available on request for \$1.00 extra, and all these programs are also available on cassette.

DISK A363 is the PROGRAMMING AIDS I package from TI. The disk is SSSD and is available on cassette as well. A photocopied manual is available for \$2.00

continued on page 4

continued from page 1

All you need to do is to boot up the computer and go to Basic or Extended Basic if you have it. Then you type in 'NUM', which is the automatic line numbering, and after the line 100 appears on the screen, type in REM or ! if you have Extended Basic. They both mean the same thing and they mean that a REMark is going to follow. Whatever statement you write in is ignored by the computer as far as its role in a program execution is concerned. After you type in REM, space once and start writing your article. If you are using Basic then you will get around 120 characters in (including line numbers, REM, and spaces) and with XB you get around 140. With Basic this is about three and one-half lines and with XB about another line more. When you have run out of space press 'enter' and line 110 will automatically come up and you go through it all again typing the next portion of your article. You continue this process until your article is finished. When it is finished you SAVE it to cassette and note the line numbers. In fact, save it twice on the tape in two different locations just in case one lot gets corrupted. Note the line numbers in both cases.

You can give this to me at a club meeting and I will try to get it in the next issue. In case you are interested, here is what happens to your article after I receive the cassette. I load it up on my computer and then I type in 'LIST "DSK2.filename"' and press enter. This loads your article onto a disk which is in a form that the TI-WRITER word processor can call up (a DV 80 file). I then call it up on the word processor, chop off the line numbers, remove the REMS and change it around to normal paragraph style. When I am finished editing and printing it there is no difference to its appearance in the magazine than if it had been typed up on a word processor to begin with.

What about some more of you cassette users getting a few things ready for the magazine? If you have made up a program then just save it twice like I described above. There must be members out there who have done some experimenting with LOGO. It does not matter how simple the game or article is, there is a place for it in the magazine. If you are unable to attend the meetings then post the cassette (or disk) to me at:

Bob Relyea
38 Vanderville Street
THE OAKS, 2570 NSW
(046) 571 253

I will see that the disk or cassette is returned for future use.

If you have an expanded system and have time to put carriage return symbols at the end of each paragraph and at the beginning of each blank line, it would be a big help. While I am on the topic of carriage return symbols, I find that it is possible to greatly speed up the process of putting them at the end of program lines by making use of the Replace String command (RS). For instance, if I am editing a program with a lot of CALL SOUND statements a Replace String comes in handy. All the CALL SOUND statements end with an end parenthesis ')'. To get a carriage return symbol at the end of these, press Fnct(9) and type RS. After the prompt comes up type in //) / and then ENTER. This will replace each parenthesis with another parenthesis and a carriage return symbol at the end of it. When you are in the command mode, to get a carriage return symbol you press Ctrl(U), Ctrl(M), Ctrl(U). If you are in the edit mode then the Ctrl(M) is replaced by Shift(M). Having the carriage return symbols in place is an enormous help to the editor.

How Geoff found the time to do the job he did with the editing is a mystery! Beyond the editing of the disks and cassettes that come to hand is the review of the other Users Groups magazines around the world. This is a big task especially if you like a program and want to include it in the TND. Of course, it all has to be typed up, and some of those programs and articles are very long and it is time consuming. So here is the last request - if anybody in the group has time I will have

ready at each meeting an article or two (or three?) which need to be typed up. If you have time on a particular month get one off me and give it to me at the next meeting, and I will take it from there. Acknowledgement will be given for your efforts.

Living a ways away from a major population centre means that things have to be coordinated well for everything to be ready in time to ensure that we get our magazines before the meeting. This means that I have to have everything in Rolf's hands by the second Tuesday or Wednesday of the month. This means that if you want to give me an article or notice via the BBS then it has to be uploaded by no later than the first Saturday after the meeting or earlier, if possible.

After having looked over the magazines of the other users groups in the U.S. I am convinced that we have the best one. I hope that we are able to keep it that way, and your help will ensure that we do. o

continued from page 3

extra. The programs on this disk are all TI BASIC utilities and include display at, accept at and screen print routines, a lowercase routine, a second ASCII character set to display highlighted text, a character definition utility, and the original disk catalogue program.

DISK A366 to DISK A369 contain a four disk tutorial on TI BASE from Martin Smoley. Each disk is SSSD and is jam-packed with examples and demonstrations on the use of TI BASE, as well as the complete text of all the TI BASE articles which have appeared in the TND over the last six months. If you bought TI BASE and have not used it yet because you found it too difficult to get started (Yes, Lou this means YOU) then this is the answer to your prayers. The four disks come as a package and are not available separately.

DISK A371 is Volume 4 Number 2 of the HOME COMPUTER MAGAZINE 'ON DISK' series, and features all the TI99/4A programs found in that issue of the magazine itself. Included on this SSSD disk is FROGO, a game in TI LOGO, CANNIBALS, a BASIC game, THE ELECTRONIC HOME SECRETARY, which produces the tones necessary to dial a touch-tone telephone from the phone numbers stored in a computer file, SEA OF STATES, an entertaining educational game in Extended BASIC, TABLUT, a 14th century board game in Extended BASIC, and using files with TI LOGO. This disk is also available on cassette.

DISK A376 is Volume 5 Number 2 of the HOME COMPUTER MAGAZINE 'ON DISK' series, and features all the TI99/4A programs found in that issue of the magazine itself. Included on this SSSD disk is EVACU-POD, a rescue game in Extended BASIC, IT FIGURES, a useful mathematics utility in Extended BASIC, LASERITHMETIC, an educational game, MUSIC KEY, a music program, SWITCH 'N' SPELL, a spelling aid, and a variety of updates to programs that appeared in previous issues of HCM. Some of the programs on this disk are suitable for cassette use. o

For Sale

For sale, for sale!! 1 fully expanded TI system including RS232, CLOCKCARD with print buffer, 300 baud (with phone attached) modem, Corcomp DS/DD drive controller, 32K expansion, 3 DS/DD 5.25 drives (2 attached and 1 spare), 3 keyboards, heaps of software including Legends II and also the SCI-FI BBS program (3 different versions all working from Techie to TI-NET) which also includes GAMEROOM plus E/A, MINI -MEM and HEAPS of other progs on disk. Disk progs also include a cracked version of 4A/TALK, the premier term emulator prog for the TI. No reasonable offer will be refused. Please either contact me via TEXPAC BBS, SCI-FI BBS (646-4865) or by writing to Greg Hope, 90 Harrow Rd., Auburn, 2144. o

Techo Time

with Lou Amadio

Errata - Direct I/O Interface

Somehow the PCBs of the Direct I/O Interface were incorrectly manufactured with the tracks on the front and back of the board reversed. Fortunately, as this board has plated through holes, this makes no difference to the electrical operation. However, some of my instructions in the original article (Oct '89 TND) may not make sense.

The diagram at the top of page 6 of the July 1989 TND shows how the interface fits in relation to the console. Also, as mentioned in the article, ensure that the console edge connector is mounted at least 8 to 10 mm proud of the PCB to allow proper seating in the recess of the console I/O port.

New Hardware Projects

In the February TND Dick Warburton reported on a number of projects which were either underway or were being considered. Two projects in particular worthy of mention are the RAMdisk with EPROM file storage and the hard disk controller.

EPROM RAMdisk

This project was conceived by Craig Sheehan, who as you may know, will not be in Sydney this year due to other commitments.

Although I have yet to see the new version of the RAMdisk, Dick reports that it works very well.

Hard Disk Controller

This would probably be a PEB card and would not replace your current floppy controller. According to Geoff Trott, however, this project would require a great deal of effort to get off the ground. One of the driving forces is, of course, the unrealistic price of the Myarc hard and floppy disk controller card - currently about \$390 (IBM HDCC are about \$100).

On the positive side, 10Mb hard disk drives are selling for about \$100 second hand.

We need some positive feedback on this one to determine if it is worth progressing.

While on the subject of disk controllers, another idea is to provide limited hard disk access through your current floppy disk system. Without knowing if this is feasible, the idea is to be able to read from and write to a previously formatted hard disk using a modified DSR and disk manager. A data interface would be required between the floppy controller and the hard disk drive.

Project Parts

Printed circuit boards are available through the club shop for the 32K Memory Expansion (both 8K and 32K chip versions), Direct I/O Interface system, the RGB colour monitor interface as well as the Multi-Function Card. In some cases, kits will be made available where possible.

The club has purchased a large number of components from Peter Schubert, mainly associated with the projects that Peter has designed. It is hoped that most of these parts will find their way into kits, but there should be no reason why you cannot purchase individual items as required. A parts list will be published soon in the TND.

My thanks to Rolf Schreiber and George Meldrum for their assistance in producing the MFC kits. More kits are available for those wanting to build a double density disk controller.

Hardware From The USA

Rolf has informed me that he has ordered a Zeno board and a RAMBO kit from the USA. More on these when they arrive.

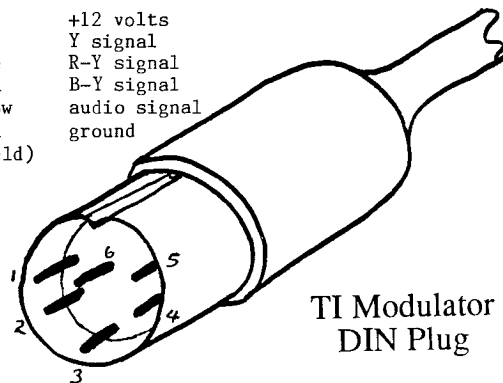
RGB Colour Monitors

We have ordered an additional 30 colour monitors through Bob Keast. Once again we do not know how long this deal will last, so consider the offer seriously. At \$122 (includes interface PCB) it is a cheap way to release a colour TV for other uses. Ring Rolf on 042-842980 to place your order.

Errata - RGB Monitors

Some feedback that I have received from members building the RGB interface may help others avoid similar problems. Some did not like the idea of using their modulator cable for the interconnection between console and interface and wanted to build their own. If you have a plastic boxed modulator, then you only have to look inside the case to see what coloured wire goes to which function at the DIN plug (console end). However, this is not easily determined if you have a metal boxed modulator, so I am reproducing the TI modulator DIN plug connections and wiring colours so that you can make your own cable.

Pin	Colour	Function
1	red	+12 volts
2	blue	Y signal
3	white	R-Y signal
4	green	B-Y signal
5	yellow	audio signal
6	black	ground
7	(shield)	



TI Modulator
DIN Plug

In making the connections between the console and the interface PCB, Eric Ockenden discovered that the pin numbering on circuit diagrams for DIN plugs is not always consistent and consequently incorrectly wired his interface. After a great deal of trouble shooting, he gave up and built another interface based on a TI design. It was not until he had similar problems with the new interface that he discovered his error.

On the positive side, we now have an alternative RGB interface design to experiment with.

Well, which is the best interface? Eric's view is that the TI version has good colour saturation but Geoff's design has better resolution.

Anyway, my thanks to Eric for his help in providing the negatives for the PCB production. This has saved the club (and ultimately its members) a lot of money.

If you own a swivel base Wang RGB colour monitor, you would have been advised to change the resistor associated with the vertical oscillator pot to assist in locking the picture. I have since found that 12K will work instead of the 15K mentioned previously.

Werner Kanitz and Tony Bell have reported a similar problem with their interface - a light horizontal shadow slowly moving down the screen. This is particularly noticeable in Basic. Geoff has suggested that could be caused by inadequate filtering in the power supply, and suggested that a 1 to 10 uF capacitor be placed between the base of the PNP transistor (G-Y amp, and the junction of 1K and 2.7K) and earth.

I also recommend powering the audio amp from the console regulated +12 volt. This reduces load on the power supply as well as lowering the noise in the audio section.

In some areas, we have found that the interface makes a good radio receiver! Try earthing the metal case of the pot and/or reducing the impedance of the input to the IC. This can be done with a 5K or a 1K pot (instead of 10K) together with a similar input resistor.

MFCs - Adjusting The Disk Controller

The disk controller portion of the MFC project requires special adjustment with a CRO. Club members should advise beforehand if this service is required at one of the meetings.

Dip Switch Settings on The MFC

The 8 way DIP switch on the MFC is used to control the disk drive motor stepping speed. The ultimate setting depends on the type of drives that you have. Modern fast drives will operate at 6 mS while the original Shugart single sided drives need a setting of 15 mS. Faster step times means faster loading and saving times for your files.

Drive #	DIP S/W
1	1 and 5
2	2 and 6
3	3 and 7
4	4 and 8

Two switches are used to set the head step times for each drive.

1st S/W	2nd S/W	Step Time
0	0	3 mS
0	1	6 mS
1	0	10 mS
1	1	15 mS

0 = on, 1 = off

PIO, RS232 and 32K on the MFC

by Lou Amadio and Geoff Trott

The following article describes how to add 32K memory expansion and/or a PIO printer interface and/or an RS232 serial interface to the Multi-Function Card (MFC). The article presumes that you have already built and tested the disk controller portion of the card as described last month.

Parts List (Common Items Only)

(Additional parts for specific options are listed separately with the assembly instructions).

2 x 0.1 uF monolithic	1 x 560 ohm resistor
1 x 4.7 uF 16V TAG	1 x 1.2K resistor
1 x 10 uF 16V TAG	3 x 4.7K resistor
2 x BC549 transistors	2 x 1N914 diodes
1 x 3 mm yellow LED	1 x 74LS86
1 x 3 mm green LED	1 x 74LS125
1 x 3 mm red LED	1 x 74LS00
2 x 33 ohm resistor	1 x 27256 EPROM (see shop)
3 x 100 ohm resistor	

The following steps outline the parts required for expansion beyond the disk controller stage described last month. Part A is used in conjunction with parts B, C or D.

See component layout on page 16

Part A - All Options

1) Add all of the additional resistors, capacitors and diodes which are identified on the overlay diagram in this article.

Take care that you do not solder any parts in the wrong place. Carefully locate the parts using the overlay diagram. The 3 mm LEDs should be mounted side by side so that they are visible through the PEB lens.

(Unidentified components on the diagram were added previously when the disk controller was built).

2) Socket all remaining chip sites. Before inserting any ICs, bend the legs in a little by holding the chip on its edge over a flat surface. The chip numbers on the overlay diagram (Uxx) refer to the identity on the original circuit diagram (available separately).

3) Insert chip U7 (74LS86)

4) Remove the link from pins 2 to 3 on U21 and insert the chip (74LS125)

5) Insert chip U22 (74LS00)

6) Change the DSR chip (U18), if necessary, to accommodate the new options (contact the club shop).

Important:

When testing this (or any other PEB) card ensure that the power has been off for about 2 minutes prior to inserting or removing the card otherwise you risk damaging the card.

Part B - Memory Expansion

The following are required to add 32K memory expansion to the MFC.

All steps in part A above plus:

1) Remove link 3 to 7 on U4 and reinsert chip (74LS02, including all pins) in the socket.

2) Insert U2 CMOS static RAM (62256). This device is sensitive to static electricity - avoid handling the pins.

Test the memory expansion by inserting the card into your PEB (or other expansion system) and typing the command "SIZE" from Extended Basic. You should be greeted with a message that indicates an additional 24K of program space.

Part C - PIO Printer Interface

The following parts are required to add a facility to interface a centronics parallel printer.

All of the steps in Part A above plus:

1) Insert chip U8 (74LS138)

2) Insert chip U9 (74LS32)

3) Insert chip U20 (74LS259)

4) Insert chip U23 (74LS74)

5) Add the printer cable IDC PCB socket (16 way, ie 8x2, male).

Test the interface with a parallel printer by listing (LIST "PIO") from Basic or with a word processor. (You will need a special cable from the printer to the 16 pin IDC socket. Note that the TI PIO interface is non standard.)

Part D - RS232 Serial Interface

This option is required to connect a telephone modem, serial printer or any "real world" device to the console. Two serial ports are available.

continued on page 18

Extended BASIC Tutorial

by Tony McGovern, Funnelweb Farm

We now continue the Tutorials with detailed ways and means of scrunching program length. As I remarked before, it is a subject I am not completely comfortable talking about because, while this series has been devoted to better XBasic programming, most things you can do to scrunch Basic programs make them less readable by ordinary mortals, given reasonable programming skill in the first place. The other reason for my reluctance is that this kind of discussion tends to degenerate into a collection of unrelated items, to yet another set of "tips", when I really want this series to be a gentle but systematic look at the workings of the machine and its language(s).

Anyway let us start at the small end of things and work up to the larger scale. Last time we looked at the space taken by simple variables. The most obvious thing is to keep variable names short. I do not recommend this until late in the piece because it is such a cheap and obvious way of gaining bytes that you might as well have the help of descriptive variable names until you are absolutely desperate for bytes. Absolute desperation has not occurred until you have had several rounds of byte saving already. The shortest variable name has only one letter character, but TI Basics also officially allow "@" (shift-2) and "-" (fctn-U) as variable names. It has to be a fairly long SUBprogram before you need more than 26 simple numeric variables but it can happen. On this console there are 3 other single characters which can be used as variable names. Experiment to find if they exist on your machine. The nagging problem is that they are not documented.

There is another way to use variable names to shorten a program. Remember from last time that a one digit numeric constant is treated as a string and takes 3 bytes, while a single letter variable takes only 1 byte. If a particular numeric value occurs frequently in a SUBprogram, 0 or 1 being common examples, then it may be worth the overhead, 14 bytes plus the defining statement, for a new variable of that value if you can then save 2 bytes on numerous occasions. A frequently used longer numeric constant, as might occur in CHAR or SPRITE manipulations, yields more bytes each time. It is a matter of doing careful book-keeping and byte counting in each SUBprogram. Once you start down this track be alert for further gains -- if you have defined S=7 and F=5 then it saves a byte to write S*F instead of 35. If you can reuse an already defined variable name then the investment is paid back faster, but this requires keeping very careful track of program flow. Go back to the example of a Key/Joystick routine in an earlier Tutorial and see if you can shorten it by reducing the number of variables used.

Replacement of numbers by variables has precedents in other languages. In TI-Forth the numbers 0,1,2,3 are not treated directly as numbers but are defined words in the language.

There is another little way that cunning entry of characters can shorten programs. This is in the entry of graphics characters with ASCII values above 127 in the upper colour groups of XB by writing strings with DISPLAY AT instead of H VCHAR CALLs. Characters in this range can be entered in strings in program statements by use of the CTRL key, rather than by using the CHR\$ function. It does tend to make the program incomprehensible as these echo as blanks to the screen. They will appear with their defined shapes if the line is called up for editing after RUNNING the program. These codes are also used as XB tokens and can only be used within strings. I should add in passing that I am in total agreement with the TI designers' choice not to allow abbreviated (direct token) entry of Basic keywords. If you want that sort of thing you should be back on your Sinclair or Commodore, and you probably do not believe in relocatable object files either.

The use of arrays to represent small collections of numbers needs detailed working out. The gains from less variable table overhead and simplified parameter passing to SUBprograms have to be balanced against the extra bytes needed for each program reference. Let the program logic be your initial guide.

This idea of using fewer bytes to represent quantities leads on to the larger subject of data compaction. One byte can carry 256 different values, and one third to one half of those can be conveniently entered from the keyboard. It is sheer overkill to use an 8 byte floating point number to represent just a few values, or even just a logical (Boolean) variable which really needs only one bit. Some languages compact Boolean variables as bits in a word or words. The CRU single bit bus of the TMS-9900 provides an ideal mechanism for bit storage and testing, but as in so many other areas the 99/4A hardware does not do justice to its CPU. The later TMS-9995 in fact has a little on-board CRU memory for just this purpose.

Opportunities for data compaction are limited in XB both because of the structure of the language (it has only character strings, floating point numerics and arrays of these as data types) and the convoluted, slow way it is implemented via GROMs and VDP memory. Any scheme for coding or compacting needs computation to pack and unpack the data. At the machine code level the tradeoffs between memory use and speed are different from those in Basic, especially TI-99 Basics, because Basic is so much slower. In my experience the use of string variables to compact data in active parts of a program is almost always doomed to failure because of slow string handling by XB and pauses for garbage collection. Data compaction can be useful though in setting up initial graphics designs or for music data. There are only so many different notes, in pitch length and volume used in any given short musical piece, and since each note takes time to play and is handled by the machine on an interrupt driven basis, this time can be used to do the computations needed to unravel the data for the next note.

Let's have a look at the graphics screen example. Suppose that in setting up a game screen, either one of two characters, maybe the same pattern in two different color groups, has to be written to 20 locations in various parts of the screen. The simplest way is a whole succession of CALL HCHARs - assuming the display is not suited to generation with DISPLAY ATs - and that is the way you will find it done in many programs (just like long lists of CALL SOUNDS). What is totally unforgivable is to find incompetent magazine or commercial programs with inefficient coding that force inconveniences like CALL FILES(1) or (2) on the user when it could have been avoided.

```
1000 CALL HCHAR(23,12,105)
1010 CALL HCHAR etc etc
```

This takes over 600 bytes. How can it be shortened? One way, a bit of a dead end in this example, is to use multi-statement lines. This would be shorter by 30 bytes or so, and marginally faster. The real improvement is to eliminate the repetition of CALL HCHAR - remember CALL is cheap but HCHAR is expensive - by using a loop and DATA statements.

```
1000 FOR I=1 TO 20 :: READ A ,B,C :: CALL HCHAR(A,B,C)
:: NEXT I
1010 DATA 23,12,105, etc etc
```

Now all but one of those HCHARs have gone. The price paid is loop and DATA execution overhead and the increased possibilities for clerical errors since the DATA items have been divorced from their proper context. At this stage you may be feeling very pleased with yourself, but then you find that to add another feature to your program you need more space. Now is the time to reflect seriously on data compression. A column index for HCHAR can only have the values 1 to 32 and rows 1 to 24. One of these values can be expressed by 1 byte with possibilities to burn. Say you use 1 byte for each row or column value then. Expressing the bytes efficiently as DATA is the next problem - there are a few bytes of overhead for each item in a DATA list, and DATA lists of a lot of short items are notorious for causing a "line too long" error. So let us pack them in a single string and use SEG\$ to unpack them, with ASC to turn a ASCII character back to a value for HCHAR. A minor problem is that characters 1 to 32 can not be entered directly in XB, so just use characters starting with "A" and

subtract 64. The opposite problem may occur with the string for the character values if upper graphics sets are being used. Then just use lower values and add a correction. So now the code might look like

```
1000 READ A$,B$,C$ :: FOR I=1 TO 20 :: CALL
HCHAR(ASC(SEG$(A$,I,1))-64,ASC(SEG$(B$,I,1))
-64,ASC(SEG$(C$,I,1)+32):: NEXT I
1010 DATA "W... ", "L... ", "I..."
```

You could further pack the data into a single string and modify the SEG\$ statements accordingly, but it might not be worth it. Remember now that the problem posed involved writing only two different characters and work out how you could compact things still further for this limited case. This example is based on one of the methods that was used to squeeze TXB into console memory. An extreme example of data compression comes when the data is regular enough that it can be generated by a formula or procedure. This is something that has to be worked out in each case.

The use of loops as in the examples above applies in other situations, particularly in CHAR definitions. XB allows the use of multiple arguments in CHAR, COLOR, SPRITE and suchlike SUBprograms. This is better and faster than using individual SUBprogram CALLs for each item in the list. The real dilemma comes when you try to use a loop to compact the program further. Critical parts of the program may be slowed down unacceptably so that you may find yourself using compact slow code in some parts of a program and longer but faster forms elsewhere. Just in passing I should remind you to null out on exit from a SUBprogram, any string variables not required to keep their value till the next CALL. This particularly applies to string variables used for READ, INPUT, PRINT etc operations involving long strings. Remember that it is the length of a program while RUNNING that really counts.

More Bugs in XB

Now funnelweb spiders are not exactly the nicest critters to be found around Funnelweb Farm, but they do have the virtue that if you do not bother them they leave you alone too. Unfortunately the bugs that infest Extended Basic are not nearly so accommodating in keeping out of the road in the first place, and insist on making their presence felt.

I have looked at a few in previous XB tutorials where they came up naturally in the subject matter, mostly in ACCEPT AT. This time we have two beautiful specimens. The first of these was exposed in the Spring 85 issue of TI*MES from the UK by John Bingham. To see it at work, enter the following little program

```
100 I=1 :: IF I=1 THEN J=1 :: GOSUB 200 ELSE K=1 :: J=2
110 PRINT K;J :: STOP
200 RETURN
```

Before you run this, predict what it is going to print out ! Then run it and see what you get. Next reverse the order of the statements between THEN and ELSE and run it again. Now it should work the way you expected, K=0 and J=1. If your XB gets in right the first time it is different from the one I have. The presence of the GOSUB just before ELSE seems to have upset XB's mechanism for keeping track of ELSE, and the program has gone ahead and ignored the ELSE and the statement after it and executed the following statement which it should have ignored entirely while proceeding to the next line. Try substituting a dummy SUBprogram CALL for the empty GOSUB. XB then works just as expected. Yet another reason for using SUBprograms instead of GOSUBs.

The XB manual lays down a few prohibitions on what can go into IF does not mention this little beauty. It does seem, despite the warnings, that FOR..NEXT loops can follow the final ELSE without problems, but this usage is not to be recommended as it may not hold good for all XB modules.

I must admit that reading this news had me a little worried, as I have written long and thoroughly debugged XB programs with some tricky IF..THEN..ELSE footwork, and had never picked up this problem. How come ? The first saving grace is that the Tutorial advice I gave is

for real, and I use very few GOSUBs and very many SUBprograms unless I am absolutely desperate for more bytes. This was frequently the case in the writing of TXB, and the central SUBprogram, one of 12 in the program, itself contains 12 subroutines written in to save bytes. Careful study of the code for TXB showed that none of the GOSUBs was written in a way that would let this evil bug loose. When you look back at something like this, you wonder whether you had scrapped particular pieces of code that never quite worked properly, for entirely wrong reasons.

The second bug mentioned has not yet been fully explored. It showed up in a CALL LINK from XB to an assembler routine which also passed in a string variable to be examined by the routine. It happened in the COLIST program, the all-singing all-dancing version of the SIMPLIST program which appeared in an earlier XB Tutorial. The symptoms were that the machine crashed utterly when it was printing out a line of its own listing, after it had already printed several hundred lines, many quite similar to the one that caused to the trouble. Not just an error caught and reported by XB, but a full blown paralytic seizure. It turned out after some TRACE work to be in the very line that was being processed for printing, and to be associated with the LINK name being at a particular position in the line of text being passed in with STRREF. The problem seems to be CALL LINK extending its link name search over places it should not, but more research is needed. Further reports in Entomology Corner in a future issue.

One disappointing discovery came up in the bug hunt. I disassembled the XB machine code utilities loaded by CALL INIT to see if the problem was in STRREF. The good news is that that code looks OK, but the bad news is that STRREF reads strings out of VDP in the same slow way that the console does, 1 byte at a time, resetting the VDP addresses each time. This is the tortuously slow way GPL does it because the console has hardly any CPU RAM, but it scarcely seems necessary in STRREF which can only be used when there is expansion RAM present.

Another bug in CALL LINK has been reported in the US of A in some older XB modules, I suspect prior to the models of V110 sold in Australia. This comes when the link name is supplied as a string variable, as in CALL LINK("A\$"). If a garbage collection is performed in VDP RAM by the XBasic interpreter in between assigning A\$ and using it in CALL LINK, this routine would lose track of where A\$. I have not encountered this bug myself, but I will try to stir it up in the XB modules I have. This reported bug brings to mind the strange state of affairs in XB where it is possible to DELETE a file by string variable reference but not to RUN a program file by similar reference, leading to a small cottage industry of ways around this deficiency.

A Few Last Little Items

As you are now no doubt aware, not all Basic programs will run correctly in XB, and some fail altogether, usually because XB supports fewer color groups than console Basic. Suppose you want a program that can run under either Basic or XB, and determine from either without crashing which interpreter is in control. If it can run under either, then you should be able to edit it in either! Some suggestions have been made to use PI which is a reserved word in XB giving the usual math value. This runs afoul of edit processing though. A better way is to look at the first value returned by RND. The initial seeds have quite distinct values. Just do not call RANDOMIZE first.

If you are writing such a program it will also be necessary to take care in using colons in PKLNT statements to cause multiple line scrolling. XB will enter two colons in a row in a statement as a statement separator token. It is necessary to use semi-colons in between to stop then being run together when a line is being called up for editing as in

```
200 PRINT "A";:;:"B"
```

continued on page 24

Subscriptions now due

April 1990

TiSHUG NEWS DIGEST

Page 8

Four-A/Talk

by Bill Gaskill, USA

DISCOVERIES:

- Art Green, the Canadian assembly language wizard who wrote the Macro Assembler programs, has released V4.3 of his TI-Writer upgrade. To be real honest with you, I did not even know that there was a V1.0! None the less, Art has done as neat a job with TIW as he did with Macro Assembler, except that the TI-Writer upgrade is better documented. Or maybe it is that I understand TI-Writer better than I do assembly language. Whatever, Art's TIWV4.3 upgrade is a FairWare offering that is available at yesterday's price of \$10 (suggested donation). A copy may be ordered from your user group disk librarian.

But wait! I am going to tell you WHY you should send the \$10 to Art and get a copy.

1. Like the Mike Ballman, Ed Jones and John Johnson TIW rewrite for the Horizon Ram Disk, Paolo Bagnaresi's BA-Writer, Tom Knight's TK-Writer and the McGovern's FunnelWeb system, the RAG version of TI-Writer has its own loader, so you do not need the TI-Writer module. The RAG loader lists A-Editor, B-Formatter or C-Utility.

2. Unlike the Ballman, Jones and Johnson collaboration, but similar to the McGovern's FunnelWeb system, the RAG version allows you to "install" the program to your system. That means that you can configure the program with your defaults for the;

- printer name.
- screen. -colours.
- tab settings.
- word wrap on/off.
- line number display on/off.
- defined character set.

3. The Show Directory function shows a catalog display much like the original one in the TIW module, which is a vast improvement over the one used in the BJ and J version.

4. The RAG version has several new commands such as QQ for an immediate quit without further prompting, Ctrl comma to go to the top of the file, Ctrl period to go to the bottom of the file, a PC (printer control) command has been added that allows control codes to be sent directly to the printer without changing the line count. It also allows you to setup a printer without having to use transliterates.

5. You can also define your own underline character, boldface character, required space character and mailing list control character. Art has also added a Conditional Page break option that tells the formatter program to do a page eject if less than a certain number of lines are left on the current page.

6. Perhaps one of the neatest features is the Chain File option that allows multiple floppy disks to be used in the formatter. This means that you can have a file that is hundreds of pages long, on several floppies, and still have the file printed as a single document. The CF option causes TIW to prompt you to insert the next disk and then press <ENTER>. Once that is done, processing resumes. Wow!

7. The formatter program is faster and more compact. In fact, it is only one file in the RAG version instead of two as it is in the original TI-Writer, the BJ and J version, BA-Writer, FunnelWeb etc.

8. Loaders for Extended Basic, Mini-Memory, the original TI-Writer cartridge and the SuperCart are included.

Many other "little" nicities have been added to make the RAG SOFTWARE version of TI-Writer my word processor of choice. It loads and works marvelously from my Horizon Ram Disk, though like all other TIW clones I have, it will not catalog my hard drive. It resists the loss of characters on word wrap better than anything I have seen to date, and it also scrolls text or windows the screen faster than anything I have seen to date, regardless of the size of the file.

I have not mentioned much about the formatter improvements, but I will say that there are several. I also have not mentioned the speed with which the program operates in general. Cursor speed is NOTICEABLY quicker as are block operations such as COPY and MOVE. Overall, the program is "smooth". Do yourself a favour and look into the Art Green TI-Writer V4.3. I know that you think a word processor is a word processor and all TIW clones are the same. NOT SO in this case.

NEWS:

- Andi Wise, editor for the newsletter at the Eugene, Oregon 99/4A Users Group, Box 11313 Eugene, Or. 97440, has compiled the most complete list of 99/4A Users Groups I have seen to date. There are over 500 listings in the data base, both past and present, U.S. and foreign. She really has gone to a lot of effort to provide us with this much needed resource. SouthWest 99er BJ Mathis, who also has an excellent UG data base, contributed as an information source to the project. Andi compiled the data base in Mark Beck's Creative Filing System.

Not intending to compete with Andi, I have adapted her data to TI-Base so that TI-Base owners could also have access to the information, in a command file programmed environment. The TI-Base version is available in your club's library. Andi is asking a paltry \$5.00 ShareWare fee for her file in CFS format. If you procure the TI-Base version, I ask that you also send her \$5.00. Please do not send any money to me for the TI-Base version. Andi did the work and deserves the financial support for her Users Group.

- In case you have not heard, Asgard's Chris Bobbitt is stirring things up with an offer to support the development of a prototype of the "Next Generation Computer" for the TI Community. From what I have read, it appears that he has lost faith in the Geneve or in Myarc, or has gotten tired of waiting for the Geneve to be a complete machine. It will be interesting to see what comes of it.

- Texaments have released MICROdex for TI-Base. All you TI-Base owners now have a chance to see the first third-party application available for your TI-Base data manager. The MICROdex libraries are available for \$14.95 for MICROdex I and \$9.95 for MICROdex II or \$22.95 for both. Please include \$8.00 S/H (UK). MICROdex is available from;

Texaments
53 Center Street
Patchogue, New York 11772
516-475-3480

- Back to Asgard again. Pro Page 99 by Ed Johnson. According to Chris Bobbitt, Asgard president, Pro Page will let you compose a full 8 1/2" X 11" page at once, with up to 28 pictures of any size on the page, and they may be placed anywhere you want. Also, both large and small type fonts [two FIXED sizes] for text will be supported as well as line drawing. Utilities to convert TI-Artist fonts and instances into Pro Page format will be included, with other conversion utilities planned for Picasso to Pro Page 99 format also. Wow! The newsletter editors out there ought to have a field day with this product [maximum 66 characters across the page!!!] Of course it does column layout of text files too, as well as importing and exporting of TI-Writer DV/80 files.

Renew now for only \$30

TI-99/4A REFERENCE LISTS:

I discovered Mike Wright's TI-99/4A Reference Lists while at the Fest-West in San Diego. Because I had to leave Sunday morning, and did not get to spend the whole weekend there, I just barely noticed Mike's product in the Genial ComputerWare booth in a last minute sweep of the Fest before leaving. I bring the topic up again, after having read and re-read the lists, because I am convinced that they are the most complete, and the most professionally presented resource of their kind available to the 99/4A community. At \$5.00 plus a couple of dollars to cover the cost of shipping and handling, they are a bargain. There must be hundreds of hours of research invested in them to come with the 40 plus pages of laser printed information, all of which has been verified by actual product. The list also contains the most complete and accurate description of books for the 99/4A I have seen since Barry Traver's list. If you are a 99/4A enthusiast, you WILL WANT the TI-99/4A Reference Lists by Mike Wright, 45 Centerville Drive, Salem, New Hampshire 03079. Honest!

TRIVIA:

Did you know that...

-The Peripheral Expansion Box project ordered by Don Bynum, designed to do away with the cumbersome chained peripherals setup of the 99/4, was officially completed in January 1982?

-In December 1983 Louisville, Colorado 99er Jim Robinson tried to start the International 99/4A Users Group complete with a bi-monthly newsletter named 4A Forum? I never heard of it again, but that does not mean it did not succeed. Has anyone else heard of it or been a member?

-AtariSoft once listed Robotron: 2084, Stargate and Super Storm in their advertisements as being available for the 99/4A? They certainly showed us a lot of support after the "bailout" by TI, but I have yet to run across those titles in a 99/4A retailer's catalog or anywhere else. Mike Wright's TI-99/4A Reference Lists do not show them either.

-Charles LaFara's International 99/4 Users Group published only seven issues of Enthusiast 99, despite being in existence from September 1980 until May 1985? The magazine issues were May, July, September and November 1983, and January, March and a May/June 1984 issue.

-While most of us are familiar with four of the product designators used with the 99/4A line of computers, TI actually had seven of them? PHA-Accessories, PHD-Diskettes, PHL-Libraries, PHM-Modules, PHP-Peripherals, PHT-Cassette tapes and PHV-Value packs.

-Bill Bies, author of the Zaxxon clone "Arcturus", also wrote a Centipede clone named "Arthropod"? Wonder what Bill is doing these days? Sure would like to see him back amongst the active 99ers.

-The hottest 99/4A joystick today is the Epyx 500XJ. [UK NAME is KONIX SP EDKING- now updated to KONIX NAVIGATOR]. TexComp's Jerry Price advises that it has the most advanced design of any joystick available. It ought to really make MunchMan turn those corners. I bought one but do not have it yet.

-Cities named DANVILLE out number all other community names in the number that have 99/4A Users Groups. Yep! There is a joke out here in the west that no matter what state you go to west of the Mississippi River, you will find a body of water named Beaver Creek. It must be the same kind of thing for towns named Danville. There are four Danvilles, the Danville 99ers in Kentucky, the East Central Illinois Users Group in Illinois, the Southside 99/4A Computer UG in Virginia and the Susquehanna Valley 99ers in Pennsylvania. The next most common community name seems to be Springfield, with groups in Illinois, Missouri and Virginia.

-Aside from the Danvilles and Springfields, there are some other interesting community names where 99/4A User Groups can be found. How about Red Deer, Whitefish, Horseheads or Papatoetoe for starters?

Until next time...

Sorcerer, part 2

A Walk Through, by Scorpia, USA

Ok, so now you are out of the Hall, but look where the spell took you: to the same forest you were dreaming of at the start of the game. And there is the Hellhound, too! And this time it is no dream! You better not wait around this time, so immediately head Northeast to the Forest Edge.

Here you will find a Snake Hole, as well as paths North and East. The North path is mined with magical mines. This is a red herring in the game, as there is no way to go safely along the path. You can ignore it without fear of missing something important. The Snake Hole is another matter, so climb down into it, and then down again to the Slimy Room, and South from there into the Crater.

Go West to the Chasm's Edge. You can not jump across the chasm, but flying over is no problem. Learn Izyuk twice, cast it on yourself once. Now just go West twice and you will be on the other side. Go North, and you will be in a room with what appears to be a tree of coins. However, that is an illusion, and you will only be able to get one coin. Take it, return to the Chasm Edge, then Izyuk and fly back across.

Now go back the way you came, all the way to the Forest Edge. This time, go East to the Meadow (do not stop to admire scenery; those are MEAN locusts on the horizon!), then Northeast to the Riverbank. Learn the Pulver spell, then Pulver the river. It will dry up, and you can move East into the river, where you will see a small cave to the Northeast. Go there.

Inside, you will find several items. Get the scroll with the Fweep spell, and Gnusto it. Then get the bat guano, but leave the vial; it has no use in the game. Now go down the hole, and you will be at the Pit of Bones. If you go South, you will find the Torture Chamber, which has another useless potion, so go Southwest into the Dungeon instead. From here, go up into the Ruins.

Learn Izyuk again, then go West across the drawbridge (careful, do not fall in!), then West again to the Meadow, where you now cast Izyuk on yourself. You have time to do that and move before the locusts arrive. Once you are flying, go Northeast to the Riverbank, and this time, Southeast to the Fort Entrance. You need to use Izyuk because the river bank has a distressing tendency to crumble after the first visit.

Around about now, you are probably feeling sleepy. Do not worry about it, just lie down and sack out for awhile. You may or may not have a strange dream. Ignore any dreams, as they are just "for show", and have no important clues to the game. When you waken again, go East into the Parade Ground. There is a flag at the top of a tall flagpole; lower the flag and search it. You will find an Aqua Vial. Take that, it will come in handy later.

Now, go East again, and you will be at the cannon. If you look inside, you will see what appears to be a pile of scrolls. Actually, they are not scrolls at all, but a group of Yipples, peacefully sleeping in the barrel. However, there IS one real scroll in there, and you will need it later. So, drop the bat guano into the barrel, and the Yipples will take off, leaving the real scroll, with the Yonk spell for you to take.

You are now just about finished above ground. Return to the entrance, learn Izyuk twice, and fly Northwest to the River Bank, and Southwest to the meadow. Here you should Izyuk again (the drawbridge is like the river bank) and go East twice to the Ruins.

"Sorcerer" is copyrighted 1984 by Infocom

This walkthru is copyrighted 1984 by Scorpia

Wit or Wit-Out

by Chris Lang, USA

The WIT series contains 5 Educational word games which are Fairware. The author is:

Chris Lang,
1906 Jackson Rd,
Baltimore, MD. 21222 USA

The author's requested donation is \$10 for User Group members and \$15 for non-user group members.

INSTRUCTIONS

OBJECT OF THE GAME:

To be the player with the highest score after a predetermined number of turns (or rounds) has been played. From 2 to 6 players may compete.

EQUIPMENT REQUIRED

- 1 floppy disk or cassette tape containing the Wit Or Wit Out game program (included)
- 1 instruction booklet (included)
- 1 TI 99/4A computer console (not included)
- 1 colour monitor (or l.r.f. modulator and a colour TV set)(not included)
- 1 Exceltec (or T.I.) Extended BASIC command module (not included)
- 1 disk drive (for disk version only) (not included)
- 1 disk controller card (for disk version only) (not included)
- 1 32K memory expansion card (for cassette version with disk system attached and turned on; also for disk version) (not included)
- 1 peripheral expansion box with peripheral expansion card (for disk version only) (not included)
- 1 cassette recorder with interface cables (cassette version only) (not included)

NOTE: For disk version only, separate units can be used in lieu of the peripheral expansion box and all cards listed above.

PREPARATION:

Connect all equipment (not included with this package) as shown in each equipment's respective instruction manuals and insure that the equipment is working properly. Read and study this entire instruction booklet carefully before proceeding to play the game.

GAME DESCRIPTION:

Each player, in turn, spells out a word using all of the letters that appear at the top of the screen and receives a score for that turn based on the amount of letters in the word and the colours of the letter blocks, each of which contains one letter. After a predetermined number of rounds is completed, the player with the highest score wins the game.

PRELIMINARY STATEMENTS

After loading the program and entering the RUN command, the title screen will appear along with a brief musical interlude. Next, will appear a series of statements; each statement requiring information given by the computer operator (yourself). A list of these statements, along with a description of the data needed by each, is as follows.....

PLEASE ENTER THE NUMBER OF PLAYERS FOR THIS GAME (2, 3, 4, 5, OR 6).

When this statement appears, type in the amount of players that will be playing this game; then press enter.

NOW ENTER THE TURNS EACH PLAYER WILL TAKE BEFORE ENDING THE GAME (1 TO 99).

After each player, in succession, has completed a turn, one round of play is over. When the statement above appears, enter the amount of rounds you wish to complete. Then, after completion of that number of rounds, the game will end.

PLAYER #: ENTER YOUR NAME.

When this statement appears, enter the name of a player. This statement will continue to appear, incrementing the player # by one, for each player's name, who will be playing, to be entered. After entering all players' names, the game will start.

PLAYING THE GAME

Just before the play of the game starts, the computer will randomly select 8 letters, for each player, and put each of the letters in one of 8 letter blocks. Each letter block will also contain a design that will surround the letter and will be coloured either yellow, green, orange, red, blue, or violet. This colour is randomly chosen by the computer. Hereinafter throughout this instruction booklet, and for simplicity, a letter block with a coloured design that surrounds a letter (always coloured black) will be referred to as a "blue letter" or a "red letter" or a "coloured letter", etc., depending upon the colour of the letter block design.

Also, before game play commences, the computer will select two consonants and 1 vowel, 3 letters altogether, and assign each of them a random colour. These three coloured letters will appear on the screen at a later time and will be used by each player in forming a word.

When play begins for each player's turn, the screen will be blank momentarily. As soon as the screen turns yellow, the playing area will be displayed showing the player's name at the upper right-hand corner, with the player's total score to date and turns previously taken at the lower left-hand corner. At this time, a large white square at the center left-hand side of the screen will appear with a black triangular arrow above it.

Almost immediately, you will see plus (+) and minus (-) signs flashing in the center of the white square. Eventually, either the plus or the minus sign will remain in the square and the word "add" or "take" will be displayed in the message area, located below the square. One of these plus or minus signs will be randomly selected by the computer. If the selection is "plus", then the word "add" will appear. If the selection is "minus", then the word "take" will appear.

As mentioned previously, the computer automatically selects 3 coloured letters to start the game. These 3 letters will have either other letters added to them, or letters taken away from these original three, depending upon the plus or minus sign. This add or take selection is the first of 3 selections that the computer will make before any input is required by you.

For the second computer selection, the screen turns violet and a white wheel replaces the add/take square. Then, numbers (from 0 to 3) appear in the wheel and begin to spin around, eventually coming to a stop. The number at the top of the wheel, and directly under the black arrow, when the wheel stops, is the computer's second selection; and represents the amount of coloured letters that you will have to either add to, or take from, the previous 3 letters (hereinafter referred to as "the previous word").

After the computer makes this number selection, that amount appears in the message area, along with the previous message. If this number wheel stops at the number zero (0), then alongside the two-word message will appear the word "letters", and the third and final computer selection becomes unnecessary, as explained in the next paragraph.

This third and last of the computer selections will be the colour. First, the screen will turn light green and the wheel will remain on the screen; but this time, colour segments will spin around in the wheel, eventually coming to a stop with the arrow pointing to

one of six colour segments. This colour will represent the colour of the letters the player is either adding or taking. The phrase in the message area is now complete and will tell you whether to add or take how many of what colour letters. As explained before, the colour of the letter refers to the colour of the design in a letter block. Of course, if the number wheel comes up zero (0), then the colour wheel is not needed and does not appear on the screen.

After all three computer selections are made, the wheel will disappear and the screen will turn grey. Starting at the top left of the screen, in a diagonal fashion, "the previous word" is displayed. If the phrase in the message area says to add one or more letters, the 8 coloured letters previously selected by the computer for that player is also displayed at the lower right of the screen. Next, a series of statements will appear in the message area.

SERIES OF STATEMENTS

Each statement in this series will require input by you via the keyboard.

YOU NOW HAVE __ SECONDS TO TYPE IN THE LETTER(S)
YOU (add letters) WISH TO ADD FROM YOUR STOCK

YOU NOW HAVE __ SECONDS TO TYPE IN THE LETTER(S)
YOU (take letters) WISH TO REMOVE FROM THE WORD

After the player is told to add or take so many coloured letters, one of the two statements, above, will appear in the message area (unless the number wheel stopped at "0"). At this point, a timer is activated, which will allow the player 15 seconds for each letter in "the previous word", plus, if the player is to add letters, 15 seconds for each letter that is to be added to "the previous word".

If the player is "taking" letters, he must now type in a letter matching the specific colour selected by the colour wheel and then, if he is taking more than one letter, he must type in one letter at a time, each letter matching the selected colour, until the amount of letters selected by the number wheel is reached. Each time a correctly coloured letter is typed, it will be removed from "the previous word". The remaining letters in "the previous word" will be used to form a new word during the rest of that player's turn.

If the player is "adding" letters, he must now type in a letter matching the specific colour selected by the colour wheel and then, if he is adding more than one letter, he must type in one letter at a time, each letter matching the selected colour, until the amount of letters selected by the number wheel is reached. Each time a correctly coloured letter is typed, it will be removed from the player's stock of 8 letters at the lower right on the screen and then, that same letter will be added to the end of "the previous word". All of the letters in "the previous word", including the newly added letters, will be used to form a new word during the rest of that turn.

PRESS "P" TO PASS OR "ENTER" TO PLAY YOU HAVE __ SECONDS

If the number wheel selected "0" letters to either add or take, the player's turn immediately proceeds with the above statement. In fact, no matter what combinations of selections previously randomized by the computer, all players' turns will, at some point, be confronted with the above statement, which requires a decision by the player.

Many strategies are involved in determining whether to pass during that turn, or to play out the turn until its completion. Some of those strategy tactics are listed on the next two paragraphs. None of the following strategies are concrete "dos" and "don'ts", however.

A player may elect to play out the remainder of his turn if one of the following conditions exists:

1. A definite and legal new word can be formed by arranging all of the visible letters in the word on the screen.

2. There is enough time remaining to complete a new word.
3. The player's score is either close behind or far enough ahead of the other players that he is willing to gamble on the legality or spelling of a new word, and is willing to risk adding 5 adding 5 points to each opponent's score if he wrong.
4. By playing, instead of passing, the player wishes to add on a turn taken to all previous players' turns, who have passed in succession.

A player may elect to pass the remainder of his turn if one of the following conditions exists:

1. No legal word known by the player can be formed using all of the visible letters in the word on the screen.
2. There may not be enough time remaining to arrange the letters and spell out a new word.
3. The player would like another set of 3 letters, selected by the computer, to work on during his next turn and does not want his present turn to count for no score.

SORRY YOU HAD TO PASS! BETTER LUCK NEXT TIME!!!

When a player elects to pass, the above statement will appear briefly and his turn is completed. No score will be given to that player and play will resume with the next player up. If, at any time, all players in succession elect to use their pass option, regardless of which player started passing, the current word on the screen will be replaced by 3 new coloured letters, selected at random by the computer; and no player will be charged with taking that last turn. In other words, a new 3-letter word will be selected and the entire round of play will be started over again, beginning with the player who started the round of passing.

NOW REARRANGE THE LETTERS TO FORM A NEW WORD! YOU HAVE __ SECONDS LEFT!!!

When a player elects to play, after adding to, or taking letters from, "the previous word", he must now decide on spelling a new word using every visible letter currently in that "previous word". All of these letters must be used to receive any score for that turn. Simply spell out the new word, one letter at a time, each letter being typed in its correct order of spelling and, at once, each letter currently being typed in will disappear from "the previous word", finally leaving no letters showing in the upper portion of the screen.

If, when you are rearranging letters in "the previous word" to form a new one, the number wheel had selected "0" letters, before-hand, to add or take, you will not be allowed to re-spell "the previous word" as it appears currently on the screen, unless "the previous word" is a newly randomized 3-letter word selected by the computer; because, if you should attempt this sneaky little feat, a "cheat" statement will appear in the message area, the so-called new word, by your standards, will not be allowed, and 5 points will be deducted from your score; and your turn is over.

When adding or taking one or more letters, however, you will be allowed to receive a score by spelling out the letters, in order, in "the previous word", providing that those letters, typed in that same order, spell a legal word.

When no letters are showing in "the previous word", and after spelling out a new word, then the new word will appear in "the previous word's" place and that player's turn is over; also, this new word will become "the previous word" for the next player up. The timer, activated immediately after the add/take square, the number wheel, and the colour wheel selections, is reset at this point; and will start its count down again after those three selections are made for the next player up.

After a player has completed his new word, and his turn has ended, that new word will be subject to verification of legality as explained in the next three series of statements.

IF THERE IS A CHALLENGE TO THE ABOVE NEW WORD PRESS "C" OTHERWISE PRESS "ENTER"

Any words found in a standard dictionary are permitted except those capitalized, those designated as foreign words, abbreviations, and words requiring apostrophes or hyphens. Consult a dictionary only when a new word is challenged to check for spelling and/or usage.

If a new word is assumed to be a legal or permitted word by all other players, then press enter when the above statement appears. The current player up will receive a score for that word and play will continue to the next player. However, if the word is challenged by another player, press "C" and the following statement will appear.

CHECK THE DICTIONARY THEN PRESS "A" FOR ACCEPTABLE OR "N" FOR NOT ACCEPTABLE

Simply follow the instructions given by the above statement when it appears in the message area.

If the new word is found to be acceptable, and the "A" key is pressed, the player up will receive a score for his new word and play will continue with the next player up. However, if the new word is not acceptable, and the "N" key is pressed, the following statement will appear.

WORD NOT VALID!! EACH OPPONENT WILL NOW RECEIVE 5 EXTRA POINTS

Any new word found to be unacceptable will add 5 points to all other players' scores and no points for the current player. Play will then proceed with the next player up.

TOO MUCH TIME!!! EACH OPPONENT WILL NOW RECEIVE 5 EXTRA POINTS

Whenever this statement appears briefly in the message area, it means that the amount of time allotted to the current player, to complete his turn, has run out before he has accomplished that task. All other players, at this time, will gain 5 points each on their scores and play will then proceed with the next player up.

If time runs out while a player is in the process of adding or taking letters, the added letters completed will remain as part of "the previous word"; but any letters taken from "the previous word" will remain missing when the next player up receives his turn.

If time runs out while a player is spelling out a new word, all of the letters in "the previous word" that has disappeared from view will be returned.

NOTE: FOR ANY MISSING LETTERS FROM A PLAYER'S 8-LETTER STOCK, WHEN HIS TURN HAS ENDED, THE COMPUTER WILL GENERATE NEW ONES BEFORE THE NEXT PLAYER'S TURN.

SCORING

A player will receive a score for spelling out a new word when the word is not challenged by any other player, and when the word is challenged and is verified (via the dictionary) to be acceptable. This score will be calculated as follows:

1. For each yellow letter in the word, the player receives 1 point.
2. For each green letter in the word, the player receives 2 points.
3. For each orange letter in the word, the player receives 3 points.
4. For each red letter in the word, the player receives 4 points.
5. For each blue letter in the word, the player receives 5 points.
6. For each violet letter in the word, the player receives 6 points.

GETTING READY TO PLAY

FOR DISK VERSION ONLY: Make sure that the Extended BASIC command module is inserted into the command module port of your console. Turn on all peripherals, then turn on the console. When the master title screen appears, press any key to display the module's main

menu. Now select Extended BASIC from the menu and wait until the word READY appears on your screen. Insert the Wit Or Wit Out program disk into disk drive #1. Type in the following statement: RUN "DSK1.WIT/WITOUT"

FOR CASSETTE VERSION ONLY: Make sure that the Extended BASIC command module is inserted into the command module port of your console. Turn on your monitor or TV set, then turn on your console. When the master title screen appears, press any key to display the module's main menu. Now select Extended BASIC from the menu and wait until the word READY appears on your screen. Insert the Wit Or Wit Out program cassette into your cassette recorder. Type in the following statement: RUN "CS1" and follow the loading instructions on your screen.

continued from page 14

CGP220, GP100, GP250, GP550, NX1000 Rainbow, OKI92 and PROWRITER. Although configured for EPSON, if you select to print pictures horizontally, some of the edges will be cut off, a retrograde step from TIA. If then you select to rotate the pictures, you may choose full height or half height- compare in the illustration picture B at far right, top with picture B in the picture below. Apart from that adjustment the only way you can amend picture size is to select to print 1, 2, or 3 pictures at the same time- next to each other. And having made this selection you MUST print 1, 2 or 3 pictures, even if you do not want more than 1! If you select 1 picture, rotated, full length, it will fill a single page. You will readily appreciate how much of the picture is lost if you do not rotate it! You may not now amend the size of line spacing. As I greatly enjoyed the ability with TIA of printing in nine sizes, PLUS the ability to choose line spacing, I am going to keep TIA and use that for my print outs. I find this new print section a bit of a disaster area. The more so as on my black and white tv the selection markers are totally invisible!!!! Although a separate segment, you are still allowed to print the picture in memory, as well as reading and printing disk files.

A new option is BANNER PRINT, with a fixed size enlargement. Approximately the middle third of the screen (note that accuracy! it is unhelpful!) can be printed sideways on about three sheets of paper. The ability to put an outline around the picture is neither here nor there to me, but it is an available option.

One really good option is the ability to print "colour" on your monochrome printer, and I am sending in the Graphx butterfly for illustration. Note that with the very much reduced number of print sizes available it is not possible to obtain even a close approximation to screen aspect ratio- we have either super panavision style, or reduce the tv screen to a square!

The PRINT routine appears to have been written for a Geneve- with any option it takes VERY VERY much longer to print out than TIA, and is possibly the slowest screen dump now available.

As I say, I shall continue using TIA for my printouts!

DSRs are supplied to enable you to use a Geneve Mouse or a Mechatronic Mouse.

The original conversion routines are supplied- this portion is unchanged, and there has been no move to accommodate the new graphics formats introduced since 1985, which is a pity. You may convert Draw N Plot, Draw a Bit, and Graphx only.

Overall impression ?

Well I am glad I have a ram disk. I think I shall be using this only when I need to use the odd new effect, and sticking to TIA for most of my graphics work, as too much useful has been lost. This IS only Version 1.0, and I shall be keeping an eye out for further versions... Not a bad review, but considering the huge amount of work that has gone into this product, I would have hoped for a greater gain, and a smaller loss!

Stephen Shaw. October 1989.

TI-Artist Plus! Version 1.0

Review by Stephen Shaw, England

This is a thorough rewrite of THE TI99/4A Graphics program, TI Artist, and involves some entirely new things, some improvements, and inevitably some losses.

The manual is 38 pages long- and there is an important update file on disk which you must print out!-, and the program is supplied on THREE disks- yes, if you only have a single disk system you are going to be doing a little disk swapping. There are more program segments, which means more waiting for segments to load - use of a ram-disk is even more useful now!

Let us look at each segment in turn, in the order they are presented in the manual, with comparisons to the original TI Artist (Version 2).

TI ARTIST- the drawing mode. We now have nine cursor speeds selected by pressing a number key, and also a very useful single-step key, which moves the cursor one step without auto-repeat, very handy for detail work. We have lost the ability to write on the picture in this mode, but instead we can select background/foreground colours from the drawing screen, instead of having to go back to the menu. We have lost the capability of printing from this mode (PRINT is now a separate mode-see below) but instead have an ARC function, which draws a quarter ellipse between two points. All you do is select the points- if they are vertically or horizontally lined up, they are connected by a straight line. If you move them out of that alignment, you get a preset arc between them. I have not been able to work out the advantages of a preset quarter ellipse- the approach taken by Graphx remains superior! There is a "spray paint" option which some may find of use, and if you wish you may amend the screen (background) colour, although this has no effect on storing or printing the picture. As PRINT is now a separate section, it is no longer possible to directly print a ZOOMed portion of screen, you must first SAVE it when in ZOOM mode and then reload it.

ENHANCEMENT section has changed only a little- we still have copy, move, slides, and instances, but FONTS have been moved to a separate section of program.

FONTS is a new section of program. By placing it in a separate section, it is possible to use longer font files, so in some cases we no longer need to keep lower case and upper case in separate files! and with very large fonts, we can now fit all of A to Z in one file! The entry screen is multi-line, the use of which is going to be limited to placing very short messages on screen, as you may now enter messages which will NOT fit! There is no length check on entering, only when you Test it and see you have most of your message missing. Also the input line is shorter, which is bad news if you use fairly small fonts, and already found the input line on the short side! Each single line of entry may be left or right justified or centred - again, lacking WYSIWYG or any other indication of what text will fit, apart perhaps from CENTRE this is going to be of limited use. There are three columns for each entry line, which contain the letters NNC. Unusually unfriendly for TI Artist users- there is no indication of the meaning of this cryptic display! What these do is to allow you to choose to OUTLINE the letters (as per TIA), SHADOW the letters, (or outline AND shadow the letters!), and the final figure selects Right/Left/Centre. The new modifications to the fonts greatly extend the already huge choice of fonts available, and shadow-outline is very effective with some fonts.

VECTORS is an entirely new section, and is best illustrated with some pictures! The choices are threefold:

- i. Frame an area of picture and then enlarge or reduce that section, without moving it.
- ii. Take a part of the picture out and save it

into another file (using _V tail). It is then available for various manipulations.

iii. Outline a bit of the picture and then distort it, using some preset patterns, but you have control over the amount of distortion.

The manual does not really try to describe this, and I remain unsure of the function of a couple of items, but I am sending in some sample prints! The main illustration is at the bottom of the page and each picture is marked A to Z or a to t. Details as follows:

A...Vector option ii above, selecting SPIN 150. As our picture is only 2-dimensional and spin is a 3-D operation, the effect is to narrow the picture, causing some distortion. Although we have 3D operators, there is NO 3D function available!

B...Vector option ii, ROTATE 90 degrees.

C...Vector option ii, ROTATE 180 degrees.

D...Vector option ii, ROTATE 210 degrees. Note distortion!

E...Vector option ii, SCALE Vert and Horiz, 71%. Note loss of pattern.

F...Vector option ii, TIP 210 degrees. Again a 3D operator, merely causing some fore-shortening.

G...Vector option ii, SCALE Vert and Horiz 114%. Note distortion to pattern!

H...Vector option i, SCALE enlargement, note break up of pattern!

I...Vector option i, SCALE reduction.

** Note reduction and enlargement is available as in two ways! One way changes what is on screen, the other changes a saved image and then places it in a selected location. **

J...Vector option ii, "HORIZONTAL 150 degrees". No idea!

K...Vector option ii, ROTATE 45 degrees.

L...Vector option ii- unchanged original image.

M...Vector option ii, "VERTICAL 145 degrees". No idea!

N...Vector option iii, Special Effect C, no change horizontally but producing two vertical squeezes- see stress lines in pattern!

O...Vector option iii, Special Effect A, reducing height of right hand side only

P...Vector option iii, Special effect F, no change to vertical scale but with a central horizontal squeeze- see stress lines in pattern.

Q...Vector option iii, Special effect H, tilting whole picture, here moving right side upwards.

- a. Original font, letters TS.
- b. Same letters outlined (Using FONT section)
- c. as a but shadowed. (Using FONT section)
- d. as a but outlined AND shadowed. (Using FONT section).
- e. Rotate 180 degrees.
- f. rotate 20 degrees.
- g. rotate 90 degrees.
- h. rotate 30 degrees.
- i. rotate 45 degrees.
- j. spin 20 degrees.
- k. tip 24 degrees.
- l. "horizontal 150 degrees"
- m. opt ii scale 114%
- n. opt ii scale 82%
- o. "horizontal 200 degrees"
- p. opt ii scale- vertical 95%, horizontal 120%.
- q. Special effect E, making bottom part fatter.
- r. Special effect D, sliding top to right, instant italics!!!!
- s. Special effect H, tilting down.
- t. Special effect A, making right hand side taller.

There are a total of eight special effect, A to H.

MOVIES... another brand new section, and with a new animated title screen to demo it too. Could not get the hang of this one. I think my approach is too much coloured by COMIC SHOW 4.0, which strikes me as being more flexible and easier to use, but then again I met Comic Show 4.0 first... I really cannot comment on this section except to say I will not be using it! Movies prepared by this section are not stand alone but require a PLAY program, which is supplied, and which is PUBLIC DOMAIN.

PRINT is now a separate section, and if you are used to TIA I think you may find this a retrograde step. Supplied for EPSON printers, but you can use a utility program supplied to enable you to use: continued on p 13

```

10 REM *****
20 REM ***
30 REM *** GRUE STEW ***
40 REM ***
50 REM *****
60 GOSUB 110
70 GOSUB 670
80 GOSUB 1240
90 GOSUB 2160
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 630
150 PRINT "IN THIS GAME. YOU
ARE A BRAVE HUNTER. YOU
ARE ALSO VERY HUNGRY. SO Y
OU'VE"
160 PRINT "DECIDED TO GO GRU
E HUNTING."
170 PRINT
180 PRINT "A GRUE, AS EVERYO
NE KNOWS, IS THE KEY INGRED
IENT IN GRUE STEW. YOU AR
E GOING TO"
190 PRINT "ENTER A SERIES OF
UNDER- GROUND CAVES IN S
EARCH OF THE STEW BASE, TH
E GRUE."
200 PRINT
210 PRINT "IF YOU CAN BAG TH
E GRUE, ANDGET OUT OF THE CA
VES, THEN YOU WILL GET YOUR
STEW AND WIN THE GAME!!"
220 FOR I=1 TO 4
230 PRINT
240 NEXT I
250 G= 610
260 G= 630
270 PRINT "ONCE IN THE MAZE,
YOU CAN EITHER MOVE TO A
DIFFERENT CAVERN OR SHOOT A
N ARROW"
280 PRINT "INTO AN ADJOINING
CAVE IN HOPES OF HITTING
A FEROCIOUSGRUE."
290 PRINT
300 PRINT "I WILL ASK: MOVE
OR SHOOT?, AND YOU MUST REPL
Y 'M' FOR MOVE OR 'S' FOR S
HOOT."
310 PRINT
320 PRINT "IF YOU DECIDE TO
MOVE, YOU CAN DO SO IN ANY
OF THE FOURCOMPASS DIRECTION
S. WHEN"
330 PRINT "ASKED WHICH WAY,
ENTER 'N' FOR NORTH, 'S' FO
R SOUTH, 'E' FOR EAST, OR
'W' FOR WEST."
340 FOR I=1 TO 2
350 PRINT
360 NEXT I
370 GOSUB 610
380 GOSUB 630
390 PRINT "IF YOU DECIDE TO
SHOOT, YOU WILL BE ASKED WHI
CH WAY, AND YOU MUST ENTE
R 'N' FOR"
400 PRINT "NORTH, 'S' FOR SO
UTH, 'E' FOR EAST, OR 'W'
FOR WEST."
410 PRINT
420 PRINT "IF YOU HIT THE GR
UE, YOU WILL BE TOLD AND
YOU MUST TRY TO EXIT THE C
AVES."
430 PRINT
440 PRINT "BUT...THERE ARE O
THER THINGSIN THE CAVES. THE
RE ARE GIANT BATS THAT W
ILL PICK"
450 PRINT "YOU UP AND DROP Y
OU ELSE WHERE."
460 FOR I=1 TO 5
470 PRINT
480 NEXT I
490 GOSUB 610
500 GOSUB 630
510 PRINT "THERE ARE BOTTOML
ESS PITS. IF YOU FALL INTO
ONE OF THESE, YOU'LL NEV
ER GET OUT!"
520 PRINT
530 PRINT "OF COURSE THERE I
S THE GRUE HIMSELF. THOUGH N
OT AN AGGRESSIVE CREATU
RE, HE WILL"
540 PRINT "EAT YOU IF YOU CO
ME TOO CLOSE."
550 PRINT
560 PRINT "THERE ARE ALSO EA
RTHQUAKES THAT MOVE THINGS
AROUND IN THE CAVES (BATS,
PITS, THE"
570 PRINT "GRUE, AND THE EXI
T)."
580 FOR I=1 TO 6
590 PRINT
600 NEXT I
610 INPUT "PRESS ENTER WHEN
READY TO CONTINUE: ":ANS$
620 RETURN
630 CALL CLEAR
640 PRINT TAB(7); "***GRUE ST
EW***"
650 PRINT
660 RETURN
670 REM
680 REM ***SETUP***
690 REM
700 CALL CLEAR
710 RANDOMIZE
720 DIM RO$(20),TR(20,4)
730 PRINT "PLEASE WAIT...";
740 FOR I=1 TO 20
750 READ RO$(I)
760 NEXT I
770 FOR I=11 TO 15
780 RO$(I)=RO$(10)
790 NEXT I
800 DATA YOU ARE IN A SMALL
ROOM WITHROCKS AND DEBRIS SC
ATTERED EVERYWHERE.
810 DATA DUCK YOUR HEAD IN H
ERE; AS LARGE ROCK STALACTI
TES HANG FROM THE CEILING.
820 DATA THE ROOM HERE SLOPE
S DOWN- WARD.
830 DATA THE ROOM IS VERY SM
ALL; BUT I THINK WE CAN MAKE
IT THROUGH OK.
840 DATA THIS IS A VERY LARG
E ROOM WITH A LARGE BOULDE
R IN THE CENTER OF IT.
850 DATA THIS IS THE CENTER
OF A NARROW PASSAGE THAT
CONNECTSOTHER ROOMS.
860 DATA THIS PASSAGE IS VER
Y LOW; BUT IF WE CRAWL; WE
CAN MAKEIT.
870 DATA THIS IS A VERY DIRT
Y ROOM; IT HAS BEEN PARTIAL
LY FILLEDIN BY THE LAST EART
HQUAKE THAT HIT.
880 DATA THIS ROOM IS ABOUT
AVERAGE SIZE; BUT IS FILLED
WITH A PUNGENT AROMA THAT
IS VERY NAUSEATING.
890 DATA YOU ARE IN A SMALL
PASSAGE- WAY,,,,,
900 DATA A SMALL HOLE IN THE
CEILING LETS LIGHT FROM OUT
SIDE THROUGH,BUT YOU WOU
LD NEVERFIT THROUGH IT.
910 DATA S"" :E HAS LEFT A
LIGHTED T ON THE WALL A
ND IT ILLUMINATES YOUR PA
SSAGE.
920 DATA A RIVULET OF WATER
SLOWLY TRICKLES FROM A HOL
E IN THE WALL.
930 DATA A SMALL HOLE TO YOU
R LEFT ATTRACTS YOUR ATTEN
TION; BUTIT IS TOO SMALL TO
BE OF ANYCONCERN.
940 DATA YOU ARE IN A LOW DE
PRESSION IN THE CENTER OF A
MEDIUM- SIZED ROOM.
950 FNR(X)=INT(RND*X)+1
960 I=1 TO 20
970 PRINT ".";
980 F=0
990 FOR J=1 TO 4
1000 GOSUB 1150
1010 F=F+TR(I,J)
1020 NEXT J
1030 IF F=0 THEN 990
1040 NEXT I
1050 YO=FNR(20)
1060 GU=FNR(20)
1070 IF GF=0 THEN 1090
1080 GU=-1
1090 EX=FNR(20)
1100 B1=FNR(20)
1110 B2=FNR(20)
1120 P1=FNR(20)
1130 P2=FNR(20)
1140 RETURN
1150 IF (FNR(3)=2)+(TR(I,J))
THEN 1160 ELSE 1170
1160 RETURN
1170 RO=FNR(20)
1180 IF RO=I THEN 1150
1190 DI=FNR(4)
1200 IF TR(RO,DI)THEN 1150
1210 TR(I,J)=RO
1220 TR(RO,DI)=I
1230 RETURN
1240 REM
1250 REM ***PLAY***
1260 REM
1270 CALL SCREEN(8)
1280 CALL CLEAR
1290 GOSUB 630
1300 FOR I=1 TO 15
1310 PRINT
1320 NEXT I
1330 PRINT RO$(YO)
1340 FOR I=1 TO 5
1350 CALL SOUND(100,262,5)
1360 FOR J=1 TO 5
1370 NEXT J
1380 NEXT I
1390 FOR I=1 TO 4
1400 CO=TR(YO,I)
1410 IF CO<>EX THEN 1440
1420 GOSUB 1610
1430 PRINT "EXIT NEARBY..."
1440 IF CO<>GU THEN 1470
1450 GOSUB 1610
1460 PRINT "I SMELL THE GRUE
!!!"
1470 IF (CO=B1)+(CO=B2)THEN
1480 ELSE 1500
1480 GOSUB 1610
1490 PRINT "FLAP...FLAP...FL
AP..."
1500 IF (CO=P1)+(CO=P2)THEN
1510 ELSE 1530
1510 GOSUB 1610
1520 PRINT "I FEEL A DRAFT!!
!"
1530 NEXT I
1540 IF FNR(15)<>4 THEN 1630
1550 GOSUB 1610
1560 PRINT

```

```

1570 PRINT "<<<<EARTHQUAKE>>>>"
1580 PRINT
1590 GOSUB 1060
1600 GOTO 1330
1610 CALL SOUND(100,1760,2)
1620 RETURN
1630 PRINT
1640 INPUT "MOVE OR SHOOT:":
MS$
1650 IF (MS$="M")+(MS$="S")T
HEN 1680
1660 PRINT "TYPE IN 'M' OR '
S'..."
1670 GOTO 1640
1680 INPUT "WHICH WAY:":ANS$
1690 PRINT
1700 FOR I=1 TO 4
1710 IF ANS$=SEG$("NSEW",I,1
)THEN 1750
1720 NEXT I
1730 PRINT "ENTER 'N','S','E
','W'"
1740 GOTO 1640
1750 IF MS$="S" THEN 2000
1760 IF TR(YO,I)THEN 1810
1770 GOSUB 1610
1780 PRINT "YOU CAN'T GO THA
T WAY..."
1790 GOTO 1330
1800 PRINT
1810 PRINT "OK..."
1820 PRINT
1830 YO=TR(YO,I)
1840 IF YO<>EX THEN 1870

```

```

1850 WL=0
1860 RETURN
1870 IF YO<>GU THEN 1900
1880 WL=1
1890 RETURN
1900 IF (YO<>P1)+(YO<>P2)THE
N 1930
1910 WL=2
1920 RETURN
1930 IF (YO<>B1)+(YO<>B2)THE
N 1330
1940 PRINT "BATS HAVE YOU!!!"
"
1950 PRINT "THEY'RE LIFTING
YOU UP!!!"
1960 PRINT "OOOOOH, WHERE AR
E WE NOW???"
1970 PRINT
1980 YO=FNR(20)
1990 GOTO 1330
2000 IF TR(YO,I)THEN 2050
2010 GOSUB 1610
2020 PRINT "CLUNK!!!"
2030 PRINT "THE ARROW BOUNCE
D OFF THE WALL"
2040 GOTO 1330
2050 IF TR(YO,I)<>GU THEN 21
30
2060 GOSUB 1610
2070 PRINT "OUCH!!!"
2080 PRINT "YOU BAGGED A GRU
E!!!!"
2090 PRINT "NOW TO FIND THE
WAY OUT!!!"
2100 GF=1

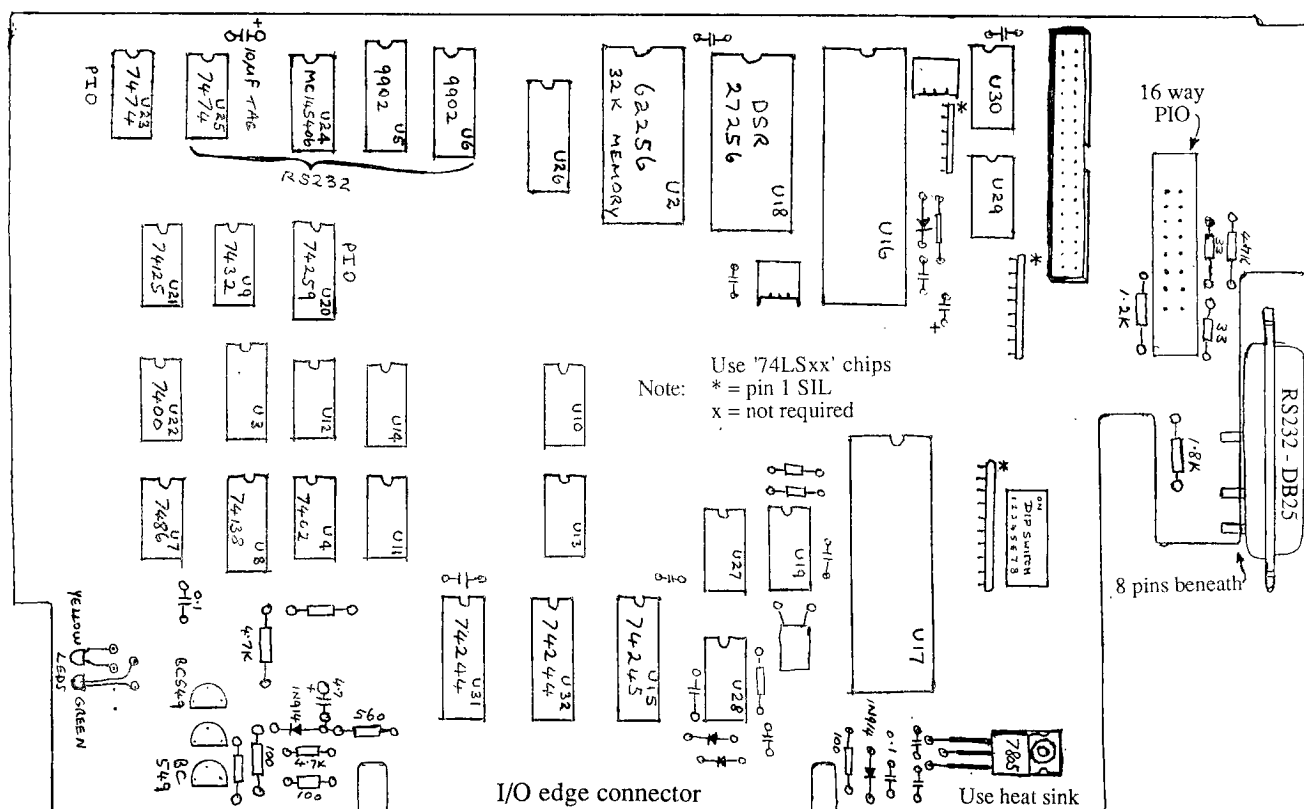
```

```

2110 GU=-1
2120 GOTO 1330
2130 COSUB 1610
2140 PRINT "THE ARROW MISSED
      THE GRUE!!!"
2150 GOTO 1330
2160 REM
2170 REM ***END***
2180 REM
2190 IF (WL=0)*(GF)THEN 2200
      ELSE 2230
2200 PRINT "YOU HAVE REACHED
      THE EXIT WITH YOUR GRUE!!
      ! YOU WILL HAVE A FILLING S
      UPPER"
2210 PRINT "TONIGHT FOR SURE
      !!!!!"
2220 RETURN
2230 IF WL<0 THEN 2260
2240 PRINT "YOU HAVE REACHED
      THE EXIT WITHOUT ANY GRUE
      !!!! YOU ARESURE TO STARVE!!
      !!"
2250 RETURN
2260 IF WL<1 THEN 2290
2270 PRINT "YOU BUMPED INTO
      THE GRUE!!!!HE ATE YOU BEFOR
      E YOU COULD MOVE!!!!"
2280 RETURN
2290 IF WL=2 THEN 2300
2300 PRINT "YOU FELL INTO A
      PIT!!!!"
2310 PRINT "YOU FELL A LOOOO
      OONG WAY..."
2320 RETURN

```

④



32K Memory, RS232, PIO on Multi-function Card

Jenny's Younger Set

DEAR JENNY,

This is a sample quiz of a new technique in writing quiz programs. If anyone has any suggestions/improvements for it, I would like to hear it. Maybe those who want to comment on it can write into the YOUNGER SET.

VINCENT MAKER

```

100 CALL CLEAR
110 RIGHT=0
120 WRONG=0
130 GOTO 190
140 RIGHT=RIGHT+1
150 RETURN
160 WRONG=WRONG+1
170 RETURN
180 CALL CLEAR
190 GOTO 300
200 PRINT "1)SOUTH SYDNEY"
210 PRINT
220 PRINT "2)NORTH SYDNEY"
230 PRINT
240 PRINT "3)PARRAMATTA"
250 PRINT
260 PRINT "4)ST GEORGE"
270 PRINT
280 PRINT "5)CANBERRA"
290 RETURN
300 PRINT "THIS IS AN EXPERIMENTAL METHOD OF PROGRAMMING
    QUIZ PROGRAMS."
310 PRINT
320 PRINT "WHO WON THE FIRST RUGBY LEAGUE(R.L.)
    PREMIERSHIP?"
330 GOSUB 200
340 PRINT
350 PRINT "PRESS THE RIGHT NUMBER:"
360 CALL KEY(O,Y,U)
370 IF U=0 THEN 360
380 IF Y=49 THEN 410
390 GOSUB 160
400 GOTO 420
410 GOSUB 140
420 PRINT
430 PRINT "WHO WON THE RUGBY LEAGUE PREMIERSHIP IN
    1922?"
440 PRINT
450 GOSUB 200
460 PRINT
470 PRINT "PRESS THE RIGHT NUMBER."
480 CALL KEY(O,G,H)
490 IF H=0 THEN 480
500 IF G=50 THEN 530
510 GOSUB 160
520 GOTO 540
530 GOSUB 140
540 PRINT
550 PRINT "WHICH TEAM'S GOALKICKER KICKED UP HIS 1000th
    POINT IN THE 1986 GRAND FINAL?"
560 PRINT
570 GOSUB 200
580 PRINT
590 PRINT "PRESS THE RIGHT KEY"
600 CALL KEY(O,U,I)
610 IF I=0 THEN 600
620 IF U=51 THEN 650
630 GOSUB 160
640 GOTO 660
650 GOSUB 140
660 PRINT
670 PRINT
680 PRINT "WHO HOLDS THE RECORD FOR SUCCESSIVE
    PREMIERSHIPS?"
690 PRINT
700 GOSUB 200
710 PRINT
720 PRINT "PRESS YOUR CHOICE"
730 CALL KEY(O,T,Y)
740 IF Y=0 THEN 730

```

```

750 IF T=52 THEN 780
760 GOSUB 160
770 GOTO 790
780 GOSUB 140
790 PRINT
800 PRINT "WHICH CLUB HAS ONLY ONE PREMIERSHIP TO IT'S
    NAME?"
810 PRINT
820 GOSUB 200
830 PRINT
840 PRINT "PRESS THE RIGHT ONE."
850 PRINT
860 CALL KEY(O,R,T)
870 IF T=0 THEN 860
880 IF R=53 THEN 910
890 GOSUB 160
900 GOTO 920
910 GOSUB 140
920 IF RIGHT>0 THEN 950
930 A$="TERRIBLE.0/5"
940 GOTO 1090
950 IF RIGHT>1 THEN 980
960 A$="NOT THAT GOOD AT ALL.1/5"
970 GOTO 1090
980 IF RIGHT>2 THEN 1010
990 A$="NOT VERY GOOD.2/5"
1000 GOTO 1090
1010 IF RIGHT>3 THEN 1040
1020 A$="NOT BAD.3/5"
1030 GOTO 1090
1040 IF RIGHT>4 THEN 1070
1050 A$="PRETTY GOOD.4/5"
1060 GOTO 1090
1070 A$="VERY GOOD.5/5"
1080 GOTO 1090
1090 PRINT "YOUR SCORE IS ";RIGHT;"CORRECT AND ";WRONG;"
    INCORRECT."
1100 PRINT A$
1110 END

```

o

Apple Peeks

by Mike Callaghan, USA

The information presented here might help if you are planning to convert a program from Apple BASIC to the TI99/4A.

HEXADD PEEK	EXPLANATION
AA68 -21912	GIVES DEFAULT DRIVE NUMBER
AA6A -21910	GIVES DEFAULT SLOT NUMBER
C000 -16384	READ KEYBOARD
C010 -16368	CLEAR KEYBOARD STROBE
C020 -16352	TOGGLE CASSETTE OUTPUT
C030 -16336	TOGGLE SPEAKER, ONE CLICK
C050 -16304	GRAPHICS MODE
C051 -16303	TEXT MODE
C052 -16302	ALL TEXT OR GRAPHICS
C053 -16301	MIX TEXT "A" GR MODE
C054 -16300	PRIMARY PAGE 1
C055 -16299	SECONDARY PAGE 2
C056 -16298	LORES GRAPHICS
C057 -16297	HIRES GRAPHICS
C060 -16288	CASSETTE OUTPUT
C061 -16287	READ PUSHBUTTON 0
C062 -16286	READ PUSHBUTTON 1
C063 -16285	READ PUSHBUTTON 2
C064 -16284	READ GAME PADDLE 0
C066 -16282	READ GAME PADDLE 2
C065 -16281	READ GAME PADDLE 1
C067 -16281	READ GAME PADDLE 3
C057 -16227	HIRES GRAPHICS
C100 -16128	READ SLOT 1
C200 -15872	READ SLOT 2
C300 -15616	READ SLOT 3
C400 -15360	READ SLOT 4
C500 -15104	READ SLOT 5
C600 -14848	READ SLOT 6
C700 -14592	READ SLOT 7
CC02 32	WINDOW LEFT
0021 33	WINDOW WIDTH
0022 34	WINDOW TOP
0023 35	WINDOW BOTTOM

o

TI bits, Numbers 13 and 15

by Jim Swedlow, CA USA
edited by Stephen Shaw, England

TI WRITER TIP

Find String (FS) is a powerful tool for finding something in a document. Just hit FCTN 8 and then enter FS. Your TI Writer gives you this prompt:

FIND enter /string/ :

You enter your string and use the slash as limiters. If you want to find the word "John", you would enter /John/. If you wanted to find John only when it is used as the last word in a sentence, you would enter /John./.

Should the "John" you find not be the one you wanted, you would go back to command mode and enter FS again. You will find /John./ still there. You just press enter and the search resumes.

Lets say, however, that now you want to find the word "Mo". But /John./ is on your screen. You could delete /John./. You could type in Mo but then you would have this:

/Mo/n./

Need you worry about the text after the second slash? No. Your TI Writer only searches for the information between the first and second slash. It ignores everything to the right of the second slash.

You will have a problem with that if you use Replace String, but that is another story.

TRICK QUESTION

Answer to the last trick question: How many birthdays does the average man have? One -- you celebrate it many times but you are born once.

MAKING A DEGREE MARK IN TI WRITER

An owner in Huntsville Texas wrote me and asked if I know how to make TI Writer type a degree sign on a TI Impact Printer (it is a Epson MX80). A degree mark is not one of the standard ASCII characters. Although many newer printers can print it, the MX80 can not.

The only way I could figure to do it was to combine TI Writer's transliterate command and the MX80's graphics ability. After a bit of experimentation, I hit on this:

This redefines the left bracket ([). The first two characters (27,76) tell the MX80 to invoke graphics. The next two (7,0) tell it that there will be seven graphics characters. The last seven characters define the degree mark.

This is not a perfect solution as, if you right justify, the right margin will be a bit uneven. It should work, however, on most Epson and compatible printers.

SOME THOUGHTS ON WORD PROCESSORS

Of late I have occasion to use a number of word processors on other machines. I learned word processing on TI Writer and I wanted to see how the 4A stood up.

TI Writer is limited by the 4A's design. Eighty columns and a full keyboard make text management (warning: buzz phrase alert!!) much easier. Otherwise, TI Writer fares well.

Just about anything you can do with the big name packages, you can do with TI Writer. Sometimes it is a bit harder, but it can be done. TI Writer is a powerful

and flexible tool. It has some abilities, like transliterate, that are superior to other word processors.

The others are slicker because they have much more memory available. They can do things with one or two key strokes that take five or ten with TI Writer - but they can be done on the 4A.

If you are writing a book, it might be worth the cost to move up. [I wrote MY book ENTIRELY with TI Writer and had no difficulty or trouble AT ALL, shifting paragraphs and sentences around all over the place. Stephen Shaw.] But for correspondence, writing this column and similar jobs, TI Writer can do anything you need it to do. And that is a fact!

BLACK FRIDAY PLUS FIVE

It was five years ago that TI announced that they were dropping the 4A. October 28, 1983. A date that changed everything for 4A owners.

We moved from the main stream of computing to a cul-de-sac. Software and hardware became scarcer and scarcer. Retailers dropped from many to only a few.

And yet a cul-de-sac is not a bad place to live. In fact, they are preferred. Ours turned out to be pretty good. Five years later, exciting software continues to appear. This has been the year of graphics applications, with many innovative programs coming out.

There are some signs of strain. User groups report declining membership and money problems. TI owners are slowly moving to other machines (often with three letters).

The end, however, is not upon us. Our 4A still has strong support from retailers, developers, publishers, user groups and owners. I expect to be writing on the sixth anniversary of black Friday.

TI SERVICE

I called TI (USA) on September 24th and got the current costs for exchanging TI equipment. These prices do not include your cost of sending the item to Lubbock or the amount TI adds for return postage. Call TI before sending them anything.

(Compare THESE with UK prices, and again ask TI-UK why UK owners have to pay a premium!):

4A Computer.....	\$30.50
PE Box.....	\$55.00
XB Module.....	\$33.00
Disk Controller..	\$44.00
32K Card.....	\$44.00
RS232 Card.....	\$33.00
Speech Syn.....	\$30.00
TI Joysticks.....	\$9.75

o

continued from page 6

All steps in Part A above plus:

- 1) Insert U8 (74LS138)
- 2) Insert U9 (74LS32)
- 3) Insert U24 (MC145406) - RS232
- 4) Insert U25 (74LS74)
- 5) Insert U6 (9902) for RS232/1
- 6) Insert U5 (9902) for RS232/2

7) Add the DB25 RS232 female connector to the back of the MFC. Align the pins accurately and solder where required.

Test the interface(s) with an appropriate serial device.

o

Tips from the Tigercub #40

by Jim Peterson, Tigercub Software, USA

Copyright 1986 TIGERCUB SOFTWARE
156 Collingwood Ave. Columbus, OH 43213

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

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.00 each plus \$1.50 per order for PPM. Entertainment, education, programmer's utilities.

Descriptive catalog \$1.00, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid. Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid.

```
*****
*
*   Tips from the Tigercub volume 3 is now ready.
*   Another 62 programs, routines, tips, tricks,
*   numbers 25 through 32. Also $15 pp. Any two
*   Tips disks $27 or all 3 for $35 postpaid.
*
*****
```

Nuts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Nuts & Bolts No. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST	PROGRAM-TUTOR
PROGRAMMER'S UTILITIES	KID'S GAMES
BRAIN GAMES	BRAIN TEASERS
BRAIN BUSTERS!	MORE GAMES
ACTION REFLEX AND CONCENTRATION	MANEUVERING GAMES
TWO-PLAYER GAMES	WORD GAMES
ELEMENTARY MATH	MIDDLE/HIGH SCHOOL MATH
MUSICAL EDUCATION	VOCABULARY AND READING
KALEIDOSCOPIES AND DISPLAYS	

For descriptions of these send a dollar for my catalog!

The READFILE subprogram on my Nuts & Bolts #2 disk has a backward parentheses in line 21161. This is the corrected line -

```
21161 DISPLAY AT(17,1):"OPEN PRINTER #":"NAME? " ::
      ACCEPT AT(17,15)VALIDATE(DIGIT)SIZE(-3):P ::
      ACCEPT AT(18,7):P$ :: OPEN #P:P$ :: GOTO 21163
```

When Texas Instruments developed Extended Basic, they took away the ability of Basic to redefine or colour the characters in sets 15 and 16, ASCII 144 to 159, in order to make room in memory for sprites (they did let us have colour set 0 instead. That is why Basic programs which use sets 15 and 16 will crash if you try to run them in Xbasic.

Finally, John Behnke published in the Chicago Times newsletter an amazing routine which gave us back those missing sets. His routine was 13 sectors long. Recently, Richard Heath published in the L.A. newsletter a shortened version. And, without having any idea how it works, I have managed to scrunch it down to only 4 sectors -

```
1 CALL BXB
29999 !BXB by Jim Peterson, adapted from VDPUTIL2 by
      John Behnke/Richard Heath
30000 SUB BXB :: CALL INIT :: CALL LOAD(8194,37,194,63,
      2,40)
30001 CALL LOAD(16368,80,79, 67,72,65,82,37,58,80,79,
      75,69,86,32,37,168)
30002 !
30003 FOR J=1 TO 136 :: CALL LOAD(9529+J,ASC(SEG$([ \[ \] $
      ,J,1))):: NEXT J :: SUBEND
30004 SUB CHAR(A,A$):: CALL LOAD(9500,A):: CALL LINK("PO
      CHAR",A$):: SUBEND
30005 SUB COLOR(A,B,C):: CALL LOAD(9492,8,15+A,(B-1)+
      C-1)
30006 CALL LINK("POKEV"):: SUBEND
```

Note than line 30002 is missing. That is because there is no way to key it in. Once again we need a program that writes a program -

```
100 FOR J=1 TO 136 :: READ A :: M$=M$&CHR$(A):: NEXT J
110 OPEN #1:"DSK1.BXBADATA",VARIABLE 163,OUTPUT :: PRINT
      #1:CHR$(117)&CHR$(50)&"[ \[ \] $"&CHR$(190)&CHR$(199)&
      CHR$(136)&M$&CHR$(0)
120 PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
130 DATA 2,224,37,20,3,0,0,0,2,5,48,48,2,6,37,2,205,133,
      2,134,37,17
140 DATA 17,252,4,192,2,1,0,1,2,2,37,1,2,3,18,0,212,131,
      4,32,32,20
150 DATA 208,4,9,80,2,32,3,0,2,1,37,2,2,0,8,2,7,11,0,2
      ,8,7,0,193
160 DATA 1,192,193,193,180,97,133,145,135,21,1,113,136,6
      ,198,145
170 DATA 135,21,1,113,136,210,70,10,198,177,137,220,198,
      2,131,37,10
180 DATA 17,240,4,32,32,36,16,6,2,224,37,20,3,0,0,0,4,32
      ,32,32,4
190 DATA 192,216,0,131,124,2,224,131,224,4,96,0,112
```

RUN the above program to create a file BXBADATA on the disk. Then load the BXB program, and enter MERGE DSK1.BXBADATA. The unprintable line will pop into place. SAVE this completed BXB routine in MERGE format, and merge it into any Basic-only program. If you want, the result can be run through a Compactor program and turned into multi-statement program lines for more speed.

Or, you can write an Extended Basic program using all 16 character sets for graphics and colour - actually 17, because set 0 is also available. Even the characters 24 through 31 can be redefined! Craig Miller has warned against fooling around in that area of memory, but there seems to be no problem with redefining the cursor (30) or the edge character (31). Sprites can only use characters between 32 and 143 and their colour cannot be changed with CALL COLOR(#,). I have not found any other bugs, but have not had time for much experimenting.

Here is an easy Tigercub challenge - run this one in Basic, not Extended Basic.

```
>LIST
100 DISPLAY AT(1,1):0
>RUN
0
0
```

Why did it print the zero twice?

I wrote this next one primarily for blind users. It converts each PRINT or DISPLAY directly to speech output and also provides a speech prompt for INPUTs.

```
100 !PRINT SPEAKER by Jim Peterson - to add OPEN #1:
      "SPEECH",OUTPUT and convert PRINT and DISPLAY
      statements to PRINT #1
110 !Also writes a PRINT #1 for INPUT prompts
120 !Program to be converted must first be SAVEd in
      MERGE format. Recommend it be RESequenced before
      SAVing, to make room for INPUT lines
130 PSS=CHR$(156)&CHR$(253)&CHR$(200)&CHR$(1)&"1"&CHR$(
      181)
140 DISPLAY AT(3,1)ERASE ALL : "INPUT FILENAME?": "DSK" ::
      ACCLPT AT(4,4):IF$ :: OPEN #1:"DSK"&IF$,INPUT ,
      VARIABLE 163
150 DISPLAY AT(5,1): "OUTPUT FILENAME?": "DSK" :: ACCEPT
      AT(6,4):OF$ :: OPEN #2:"DSK"&OF$,OUTPUT,VARIABLE
      163
```



```

160 PRINT #2:CHR$(0)&CHR$(1)&CHR$(159)&CHR$(253)&CHR$(
    200)&CHR$(1)&"1"&CHR$(181)&CHR$(199)&CHR$(6)&
    "SPEECH"&CHR$(179)&CHR$(247)&CHR$(0)
170 LINPUT #1:M$ :: P=POS(M$,CHR$(156),3):: A=POS(M$,
    CHR$(162),3):: Z=POS(M$,CHR$(181),3)
180 I=POS(M$,CHR$(146),1)::IF I=0 THEN 210 :: IF Z=0 OR
    Z<I THEN PRINT #2:M$ :: GOTO 240
190 M2$=SEG$(M$,1,1)&SEG$(M$,2,1)&PS$&SEG$(M$,I+1,Z-I-1)
    &CHR$(0):: PRINT #2:M2$
200 PRINT #2:SEG$(M$,1,1)&CHR$(ASC(SEG$(M$,2,1))+1)&SEG$(
    M$,3,255):: GOTO 240
210 IF P+A=0 THEN PRINT #2:M$ :: GOTO 240
220 M=MAX(P,A)
230 M$=SEG$(M$,1,2)&PS$&SEG$(M$,M+1,255):: PRINT #2:M$
240 IF EOF(1)<>1 THEN 170 ELSE CLOSE #1 :: CLOSE #2
250 DISPLAY AT(12,1)ERASE ALL:"Type NEW and Enter" ::
    DISPLAY AT(15,1):"Type MERGE DSK":OF$ :: END
    *****

```

MOLLY DARLING

```

100 CALL CLEAR :: CALL SCREEN(5):: FOR SE=1 TO 12 ::
    CALL COLOR(SE,16,5):: NEXT SE
110 DISPLAY AT(3,8):"MOLLY DARLING": " Written and
    performed by":TAB(9);"Eddy Arnold" :: DISPLAY AT(
    24,1):"Programmed by Jim Peterson"
120 FOR D=1 TO 200 :: NEXT D :: DISPLAY AT(12,1):"Just a
    moment.....": ".....looking for my music..."
130 DIM N(100),N2(100),A(250),B(250),C(250):: F=110 ::
    FOR J=1 TO 80 :: N(J)=INT(F*1.059463094^(J-1)+.5)::
    NEXT J
140 DATA 16,11,8,16,8,11,16,4,11,18,11,8
150 DATA 20,16,11,23,11,16,25,21,16,28,16,21
160 DATA 23,20,16,23,16,20,23,11,16,23,16,11
170 DATA 20,11,16,20,16,11,20,8,11,20,11,8
180 DATA 20,11,16,25,16,11,23,11,16,20,8,4
190 DATA 18,16,10,18,10,16,18,16,10,18,11,16
200 DATA 18,15,11,18,9,15,18,11,9,18,9,3
210 DATA 28,8,1,28,13,8,28,8,13,28,13,4
220 DATA 27,20,18,27,18,20,20,18,12,20,12,18
230 DATA 25,21,16,25,16,21,25,13,16,25,16,13
240 DATA 27,23,21,27,21,23,27,23,18,27,18,21
250 DATA 28,23,20,28,20,23,28,20,16,27,16,20
260 DATA 30,21,13,28,13,21,27,21,13,25,13,21
270 DATA 23,20,16,23,16,20,20,11,16,20,16,11
280 DATA 30,23,13,28,13,23,23,20,13,20,13,16
290 DATA 25,21,16,25,16,21,25,21,16,27,16,21
300 DATA 28,23,20,20,16,11,18,15,11,20,11,15
310 DATA 16,11,8,16,8,11,16,9,1,16,1,9
320 DATA 16,11,8,16,8,11,16,1,8,16,13,1
330 DATA 25,21,16,25,16,13,25,13,9,25,9,4
340 DATA 23,20,16,23,16,11,23,11,8,23,8,4
350 DATA 21,18,11,21,11,9,21,9,6,20,6,3
360 DATA 21,16,11,20,16,11,20,11,8,20,8,4
370 DATA 18,13,10,18,10,6,18,6,1,20,13,10
380 DATA 22,18,13,28,22,18,27,18,22,25,22,18
390 DATA 23,18,15,23,15,11,23,11,6,23,6,3
400 DATA 23,21,15,23,15,11,23,11,9,23,9,6
410 DATA 16,13,8,16,8,13,16,13,8,18,13,9
420 DATA 20,11,8,21,8,11,20,11,8,18,11,6
430 RESTORE 140 :: T=16 :: GOSUB 480 :: RESTORE 140 :: T
    =4 :: GOSUB 480 :: RESTORE 180 :: T=12 :: GOSUB 480
    :: RESTORE 140 :: T=16 :: GOSUB 480
440 RESTORE 210 :: T=28 :: GOSUB 480 :: RESTORE 170 :: T
    =4 :: GOSUB 480 :: RESTORE 250 :: T=4 :: GOSUB 480
    :: RESTORE 280 :: T=4 :: GOSUB 480 :: RESTORE 190 ::
    T=8
450 GOSUB 480 :: RESTORE 140 :: T=16 :: GOSUB 480 :: R&S
    TORE 290 :: T=48 :: GOSUB 480 :: RESTORE 140 :: T=16
    :: GOSUB 480 :: RESTORE 410 :: T=8 :: GOSUB 480
460 RESTORE 310 :: T=8 :: GOSUB 480 :: GOTO 490
470 GOTO 490
480 FOR J=1 TO T :: X=X+1 :: READ A(X),B(X),C(X):: A(X)=
    A(X)+12 :: B(X)=B(X)+12 :: C(X)=C(X)+12 :: NEXT J ::
    RETURN
490 DISPLAY AT(10,1):"Control volume of 3 voices":"using
    1, 2 and 3 keys for":"louder and Q, W and E for":"
    softer.":"
500 DISPLAY AT(15,1):"Control speed using 'F' for":"fast
    er and 'S' for slower."
510 DISPLAY AT(18,1):"Change key using 'A' for":"higher
    and 'D' for lower."
520 DISPLAY AT(21,1):"Press 'Z' for minor key, 'X'":"for
    major key.": V1,V2,V3=10 :: F,P,Y=0 :: X=200

```

```

530 FOR J=1 TO 192 :: CALL SOUND(-999,N(A(J)-Y),V1,N(B(J)
    -Y),V2,N(C(J)-Y),V3):: FOR T=1 TO X/50 :: P=1^X ::
    NEXT T
540 CALL KEY(0,K,S):: IF S<1 THEN 710 :: ON POS("123QWEF
    SADZX",CHR$(K),1)+1 GOTO 710,550,560,570,580,590,600
    ,610,620,630,650,670,690
550 V1=V1-1-(V1=0):: GOTO 710
560 V2=V2-2-(V2=0)*2 :: GOTO 710
570 V3=V3-2-(V3=0)*2 :: GOTO 710
580 V1=V1+2+(V1=30)*2 :: GOTO 710
590 V2=V2+2+(V2=30)*2 :: GOTO 710
600 V3=V3+2+(V3=30)*2 :: GOTO 710
610 X=X-20-(X<2) :: GOTO 710
620 X=X+20 :: GOTO 710
630 IF F=1 THEN GOSUB 700
640 Y=Y-1-(Y=-20):: GOTO 710
650 IF F=1 THEN GOSUB 700
660 Y=Y+1+(Y=6):: GOTO 710
670 IF F=1 THEN 710 :: GOSUB 680 :: GOTO 710
680 F=1 :: Y=0 :: FOR W=3 TO 27 STEP 12 :: N2(W)=N(W)::
    N(W)=N(W-1):: N2(W+5)=N(W+5) :: N(W+5)=N(W+4):: N2(W
    +10)=N(W+10):: N(W+10)=N(W+9):: NEXT W :: RETURN
690 IF F=0 THEN 710 :: GOSUB 700 :: GOTO 710
700 F=0 :: FOR W=3 TO 27 STEP 12 :: N(W)=N2(W):: N(W+5)=
    N2(W+5):: N(W+10)=N2(W+10):: NEXT W :: RETURN
710 NEXT J :: J=192 :: FOR V=10 TO 30 :: CALL SOUND(-999
    ,N(A(J)-Y),V,N(B(J)-Y),V,N(C(J)-Y),V):: NEXT V ::
    FOR D=1 TO 500 :: NEXT D :: GOTO 530

```

MEMORY FULL Jim Peterson

continued from page 31

CIM 99, Decembre, 1989: A collection of reviews of some nice software such as TI Base and some assembly programs - all in nice French, so do not ask me to tell you too much about what it says!

LEHIGH 99'er, December, 1989: Beyond some programming hints and an article on the Geneve were the nice, new colourful U.S. postage stamps on the back of the newsletter.

LEHIGH 99'er, Summer, 1989: A brief look at some faires and software; A 12.5 volt conversion for Mechatronics Epromer; A look at TICOFF - the (T) (I) (C)omputer (O)wners' (F)un (F)aire (T)he (I)BM (C)lone +(O)wners (F)un (F)aire; a comparison of XBasic and the Microsoft GW Basic; a brief advertisement for Page Pro 99; a brief tutorial on designing characters and fonts; the Kiddie Corner.

continued from page 2

Due to circumstances beyond this groups control it has become necessary to relocate the BBS to a new site which will, unfortunately, mean that some members living in the outer metropolitan area will now be charged at STD rates to access the system(The new number is (02) 456 4606 ED). If calls are kept to off-peak times then costs will not be that significant.

Items discussed at the last Directors meeting which should be of interest to members were:-

1. Russell Welham put forward a proposal to sell through the shop a folder to house newsdigests. It is envisaged that the folder would have the groups logo and title on the front cover. Russell is sourcing out a likely supplier and will report back to the meeting when he has the required information and costing structure.

2. It was agreed to purchase some 3.5 inch disks for sale through the shop. These should be available at the April meeting. Price per 10 is yet to be assessed.

3. The group is to contact a major distributor of printers to assess the possibility of becoming an agent for their sale.

That is about all the news for this month. Hope as many members as possible can get along to the meetings where you will be made feel very welcome.

TI99/4A Disk Peripheral

Functional Specifications, part 1

CONSUMER GROUP LUBBOCK, TEXAS 79414
COPYRIGHT 1980 TEXAS INSTRUMENTS ALL RIGHTS RESERVED.
DATE: MARCH 28, 1983 VERSION 2.0
SUPPORTED FILE MANAGEMENT OPTIONS

The disk peripheral supports most of the options in the File Management Spec. for the 99/4 Home Computer. The supported options include:

- Sequential and Relative record (random access files)
- Fixed and Variable length records
- Internal and Display file types
- Out, Input, Update, and Append access modes
- Program Load and Save functions

The I/O routines supported by the disk peripheral are:

OPEN - Open an existing file for access. This routine must have the drive number or the disk name and the filename to open. CLOSE - Close a file for access. The PAB can be released and the disk peripheral software deal locates some buffer area in VDP memory. Since the number of files that can be open at once is limited it is advised each file is closed as soon as it is no longer needed.

DSKX.FILE-ID

DSK.VOLNAME.FILE-ID

X is the drive ID number (1-3), "volname" is a volume name ID and "file-id" is an individual file ID. Both "volname" and "file-id" can be strings of up to 10 characters long. Legal characters for these strings are all the ASCII characters, except the "." character and the space. The first form of the file name specification shows the direct drive ID option. The user can specify either DSK1, DSK2, or DSK3 as drive numbers. Only the specified drive is searched for the given file-id. The second form of the file name specification is the symbolic form. The disk drive is not explicitly assigned, but assigned through the volume name ("volname"). All drives are searched in sequence for the given volname, i.e. DSK1 first, then DSK2, then DSK3. The first drive with the given volname on its disk will be used for the file-id search. It is allowed to use two or more disks with the same volname in the system, however, if the specified file-id does not exist on the first drive with the given volname, the other disk drive(s) with the same volname will not be searched. Whichever form is used, the file-id has to be unique for the selected drive, i.e. if a new file is created, the file-id used must differ from all other file-ids on that drive, or the existing file will be replaced by the new one, unless it is protected. The file-id in the OPEN statement has to correspond to a data file. If the file was created by a SAVE command, an OPEN for that file will give an error, unless the file is opened for OUTPUT mode, in which case the program file will be replaced by the new data file.

FILE ORGANIZATION Option

The two file organisations the user can specify are:

1. SEQUENTIAL - Access the file in sequential order, comparable to tape-access. The file may be accessed in any of the four I/O modes. Record type may be specified as FIXED or VARIABLE. File type may be specified as INTERNAL or DISPLAY.

2. RELATIVE - Access the file in random order. The open mode can be any of the available four modes, and the record type must be FIXED, and may be either INTERNAL or DISPLAY. Due to BASIC limitations, the combination RELATIVE and APPEND is not supported. This combination is trapped out as an error.

The default file organization is SEQUENTIAL.

Both the SEQUENTIAL and RELATIVE specifications can optionally be followed by an initial record allocation specification. This spec. indicates the number of records to be allocated initially. In case the record length has been specified as VARIABLE, the allocation will be made for maximum length records. The number of records initially allocated has to be less than 32767, in order to stay within the record addressing range of a file management system.

RECORD-TYPE Option

The record-type option is used to specify the size of each record in the file. This size can be either FIXED, all records have the same length, or VARIABLE, with a maximum length optional. If the file organization specified is RELATIVE, the only legal type is FIXED, which is also the default for relative record files.

Both the FIXED and the VARIABLE options can be followed by an expression indicating the actual or maximum record length. Since the length is used to reserve buffer space in the BASIC interpreter, a user is advised to select the length as precisely as possible. Larger record lengths mean fewer variables can be used by BASIC.

The disk peripheral defaults the record lengths for both the FIXED and VARIABLE options to 80 characters. The default record-type for SEQUENTIAL files is VARIABLE; for RELATIVE files it is FIXED. If a file is opened for any I/O mode other than OUTPUT, and it already exists, the record length, has to match the stored length. If no record length is given the DSR will default to the stored length. The maximum record length for FIXED records is 255, and for VARIABLE length records it is 254.

FILE-TYPE Option

The file-type option can be used to specify the format of data to be stored in the file. There are two formats available:

1. DISPLAY - Stores data in readable format, i.e. as it would be printed on a printer. If the data has to be read back by the machine, this format is not recommended.
2. INTERNAL - Stores data in machine readable format. Since most files on the disk will be read by machine, this format is recommended. It relieves the user of storing separate data like quotes and commas in the file in order to make it suitable for an INPUT command. It avoids the overhead of converting the internal machine representation for the numbers and strings into a representation that is readable for humans and vice versa.

Again, if the file exists and the I/O mode is not OUTPUT, the specification has to match the value stored at file creation. BASIC uses DISPLAY as a default, which means that if data is stored in INTERNAL format, the user always has to indicate this in the OPEN command.

FILE-LIFE Option

BASIC only recognized the PERMANENT option as a file-life specification. Since it is also the default it can be eliminated.

CLOSE

The key word DELETE is optional with the CLOSE statement. In case DELETE is specified, the file is not only disconnected from the file number, but the disk space taken up by the file is released, and the file-id is erased from the disk's catalog. This means the file can no longer be accessed, not even with an OPEN statement (see DELETE statement).

A few examples of the CLOSE statements are:

CLOSE #240 - close the file associated with #240

CLOSE #240:DELETE - same as above, but also deletes the file

PRINT STATEMENT

The PRINT statement can be used to write information out to a file that has been previously OPENed. The PRINT statement can only be used for files that have been opened for access in either OUTPUT, UPDATE, or APPEND mode. A PRINT to a VARIABLE record length file will always set a new EOF mark, causing data behind the current record to be lost. The general form of the PRINT statement is:

PRINT #file-number[,REC record-number][:print-list]

For a detailed description of the PRINT statement, refer to the 99/4 BASIC Language User's Reference Guide.

INPUT STATEMENT

The INPUT statement can be used to read information from an existing file. The INPUT statement can only be used for files that have been OPENed for access in either INPUT or UPDATE mode.

The general form of an INPUT statement is:

INPUT #file-number[,REC record-number]: variable-list

RESTORE STATEMENT

The RESTORE statement repositions an open file to its first record, or at a specific record if the file is opened for RELATIVE mode and the RESTORE contains a REC clause.

The general form for the RESTORE statement is:

RESTORE #file-number[,REC record-number]

EOF FUNCTION

The EOF function can be used to test for the end of file during I/O operations. Three conditions are indicated by the EOF routine:

0 Not EOF (End of File)
1 Logical EOF (End of File)
-1 Physical EOM (End of Medium)

Physical EOM can only be detected if the device is at its physical end and the file is at its logical end. The general form for the EOF function is:

EOF(file-number)

The EOF indication only has meaning in the case of sequential access to files, since for random access the next record to be read or written cannot be determined from the current one. Therefore, the EOF subroutine will assume that the next record to be read/written is the sequentially next record. The logical EOF indicates that the next sequential read/write operation will attempt to access a record outside the current file. In general this indication will only be used for read operations.

Because of pending BASIC INPUT conditions, it is possible that the EOF subroutine indicates "EOF", even if the next INPUT statement will yield no EOF error, since it can read data from the current record. Something similar can happen if it indicates "no EOF" and the next INPUT statement reads more than one record. In this case the INPUT might be terminated with an error. To avoid this type of situation, the user is advised to use only non-pending INPUT statements (INPUT statements without a trailing comma), so that each record corresponds to one INPUT statement.

For random access to files, the EOF subroutine can only give meaningful results if the access is converted to "semi-sequential" access, i.e. if the record is

positioned through a RESTORE statement and then sequentially accessed through any I/O statement without REC clause specification. After the RESTORE the EOF subroutine will indicate that the condition for the next record is (EOF, EOM or available), without issuing an I/O error.

Note that there is one EOM condition that cannot be detected by the EOF subroutine. This condition occurs when the datablocks on a disk become so scattered that not enough datablocks can be allocated for a file. In this case a PRINT operation will be aborted with an I/O error, even though there is enough space available on the disk, and the EOF function does not indicate an EOM condition.

NOTE that the software file protection does not offer any protection against complete disk re-initialization. The only way to avoid file loss in that case is to "write protect" the disk itself by placing a tab over the notch on the right side of the disk. This will disallow any write operation to the disk, giving a hard error as soon as the disk is being accessed for write operations. Notice that this type of write protection is only intercepted on the actual write operation. The disk software will not disallow destructive access to the disk until the moment it actually tries to modify part of the disk. o

From the Bulletin Board

BBS NEW LOCATION TEETHING PROBLEMS.

Would all users please advise if you are having problems with the BBS at the new location. Please note type of problem encountered and advise via mail to SYSOP. Please take special note if the telephone line sounds noisy, hum or crackles etc if you are having problems in a usage session, by lifting your phone receiver just before you log off then listen for noise etc when you disconnect your modem. The new BBS phone number is: (02) 456 4606.

Ross Mudie, Sysop, 10/3/90

MAIL TO : EDITOR
MAIL FROM : SARA
SENT ON Tuesday 06/03/90 at 21:36:27
Dear Rob,

Here is some news that I ask you consider for publication which is self explanatory.

Recently I called for Help! on the BBS as I had unwittingly corrupted a new disk purchased from TexCorp USA along with the manual. Have THE INSTRUCTIONS but no program! I thought that I would appeal to the members, after all you only need one response. I thought that they all would not assume that the other guy would answer the call. "No answer was the stern reply" so much to my surprise the call was mentioned in the News Digest.

The good news is that I received a call from 600 kilometres away to say that I could expect a copy in the mail if I would be good enough to supply my address! My faith in mankind was given a Big Boost. My heartfelt thanks go out to a member of Tishug's North Sydney Group to one who cares.

Thank you again,

Chris Devlin
17 Lowanna St.
Belrose NSW 2085 o

A Tip from Stephen Shaw, UK

What is the use of the Editor Assembler module if you have Funnelweb? Well you do get a nice machine code manual with it. Apart from that, you also get the source code for Tombstone City on disk and of course you can bung in 8K of RAM and have yourself a Super Space module! o

Reviews

by Stephen Shaw, England

JOYSTICK: KONIX NAVIGATOR

£14.99 each, including post and packing from: Konix Products, Unit 35, Rassau Industrial Estate, Ebbw Vale, Gwent NP3 5SD. Access and Visa accepted; telephone orders to 0495 350101

A few issues back I reviewed the Konix Speedking joystick (sold in the USA as the Epyx 500XJ). Konix, despite their name, are a little Welsh company who seem to have the joystick market well sown up, with millions of Speedking joysticks out there, their latest addition just had to be obtained!

The Navigator Joystick is 99.9% perfect. The only question in my mind is why were all joysticks not made this way eight years ago? It is a real joy to use, and will quickly help you to double your previous high scores! (That .1% reduction is for the shiny surface of the grip by the fire trigger, the remainder of the unit is textured for a happier hold. It would have been nice to have a textured front face as well.)

Like the Speedking the Navigator uses micro switches and a steel shaft, so a long life seems secure!

The Navigator joystick I have seems just a little more "positive" than the "Speedking" I have (a little more pressure is needed, but not a great deal. You may recall I mentioned that the Speedking was for holding in the left hand only as the fire button was on the right hand side. The Navigator solves this problem by placing the fire button in the centre and by redesigning the shape of the body.

The Navigator joystick will sit easily between the thumb and first finger, with the thumb on one side and the other fingers on the other side, while the fire trigger falls neatly under the end of the first finger. The joystick can be used for prolonged periods without fatigue. It does not use up any desk space, as so many joysticks do, and this is a good thing for me because I do not have any desk space to plonk a giant joystick on! The Navigator is held in one hand while the other hand controls the stick.

This joystick has an "autofire" switch. I have seen these advertised and wondered what they do. It appears that they allow you to hold the fire button down, and send a train of fire pulses, something which we ancient TI99/4A owners really have little use for, as off the top of my head I cannot think of any TI program which requires you to repeatedly press fire. We seem to have autofire built in. Yet again, TI gave us something noone else seems to have thought of! If you do switch the autofire on when using this joystick with the TI99/4A, it has no effect on most games, but on some games with rapid joystick scans, it can get in the way, allowing you only a small number of repeated shots in Parsec for instance (making it impossible to overheat the laser!). In general you would use it with autofire off.

The joystick has two D style 9 pin plugs. Use the black one with your TI99/4A and also use one of the ATARI to TI99/4A joystick converters.

This is an excellent product, very strongly recommended, and it is Welsh.

=====

TI-Base UPDATE.

TI-Base is now into Version 2.00. I have written to everyone who purchased TI-Base through me giving details of the update procedure.

New items/augmentations include:
Sorting can be on up to 8 nested fields
READ can be used with quoted strings and there is a new READSTRING.
There is a CONVERT to more easily amend data base structures.
MEMORY gives details of memory usage.
EJECT is put in from dBase II.
SNAP is a screen dump command.
Printer output can now be directed to disk.
There is a limited file print command to print command files.
TRACE to printer has been added.
There is slightly more memory available for your use.
Global conditional changes can be made.
Printer control codes can be incorporated and used in PRINT directives.
SUM command, which can also be conditional/global.

Thus a product which was already superb is even better. The new rather fatter 66 page manual has more examples of usage too. And the price remains at US\$25,

A fuller review of the revised data base appears in the November 1988 issue of MICROpendium, and back issues are available to current subscribers. The final grade was A-, the minus coming from Ease of Use, as the data base is designed for maximum power and flexibility, options which usually require some sacrifices in user friendliness. The reviewer concluded that "It is difficult if not impossible to measure TI-Base against the competition since it does not have any competition".

As I am now unlikely to obtain the necessary quantity of orders for quantity discount, the price if you wish to order through me is twenty pounds, inclusive.

=====

MACFLIX by J Peter Hoddie.

Published by Genial Computerware for US\$15 plus shipping. PO Box 183, Grafton, MA USA 01519.

Another graphics standard is born for the TI99/4A! How would you like to have a single graphics disk file to print out an A4 size picture in one pass, with a resolution of 700 pixels by 480 pixels (or thereabouts)? MACFLIX is not a drawing program!!! It is a program which allows you to view and print high resolution pictures produced on the Macintosh using MacPaint, and also adds the ability to transform parts of the Macintosh picture to TI-Artist format.

If you were to split a Macintosh picture up into bits for TI-Artist and then print them using Artist options 1.3.8., you can transform an A4 portrait into (with some sticky tape) a fair sized poster some 33 odd inches long; naturally pixels are larger on this!

What use is it? On its own, not a lot, unless you have access to Macintosh pictures! Which brings in a query: would you like the disk library to add Macintosh pictures, in addition to the RLE pictures? Let me know if interested. The Macintosh pictures I have at present are quite impressive.

Correct, you cannot see a full Macintosh picture on a TI99/4A screen in one go! You have to window around the thing until you have the bit you want, then save that screen as a TI-Artist picture, or you can print the whole thing (Epson and Prowriter printer standards only).

The maximum disk file size that this program can cope with is about 90 to 100 sectors, so some Macintosh pictures may not fit; the bottom parts of the pictures are lost on larger files.

Geneve compatible, and can save in both Ny-Art formats too.

A good and inexpensive program.

=====

o

Forth to you too! Session 7

Author unknown

This is in answer to questions regarding the use of sprites and other graphics capabilities of the 99/4A with Forth. As we all know, we got a pretty good deal because TI built a fair amount of graphics into this little machine. Anyone who has seen PARSEC etc. can vouch for that. And all of them can be utilised in TI-Forth, with commands (statements) which are very similar to the ones employed in BASIC and XBASIC.

However, there are a couple of things which must be done in order to use any of the available VDP (graphics) modes. First of all, the appropriate LOAD OPTION must be booted, i.e. -GRAPH and -VDPMODES. If you are using the 64 column editor you only need to boot -SPLIT2, the rest of them are already booted. Another thing you will have to do is to fix a bug in line 10 of screen 58. It should read as shown below:

```
VDPMODE 4 < IF SMTN 80 VFILL 300 ' SATR ! ENDIF
```

(in other words the ! after 300 should be a ' {tick}). And while you are at it, fix line 1 of page 10 in Chapter 6 of the manual to read: HEX 3800 ' SATR ! Also, on screens 53, 54 and 55 the last word of line 1 should be SETVDP2, NOT VDPSET2. For some strange reason it is correctly shown on screens 51 and 52. Finally, on screen 59, line 9, change the OOFF to OOF6.

Now let's take a look at what is available in Forth. I have drawn up the following chart for easy reference:

VDP MODE	ASCII CHARS	CHAR DEF?	FG/ BG	HCHAR VCHAR	SPR1- TES	SPR. MOT.	DISPLA Y SIZE
TEXT	YES	YES	NO	YES	NO	NO	40x24
GRAPH- ICS	YES	YES	YES	YES	YES	YES	32x24
MULTI	NO	NO	NO	NO	YES	YES	64x48
BITMAP	NO	NO	NO	NO	YES	NO	256x192
SPLIT	same as BIT-MAP with 8 lines text on bottom						
SPLIT2	same as BIT-MAP with 4 lines text on top						

There are essentially four display modes, except one of them, BITMAP, comes in three versions. When Forth boots, the display is in TEXT mode: 24 lines of 40 characters each. This is the mode used by the 40-column editor. New characters can be defined but all characters have the same foreground/background colours. There is no control of individual character sets. The easiest way to set text/background color is by using nn 7 VWTR (see tut 1). If -TEXT is booted you can return to text mode by entering TEXT.

The screen display of the GRAPHICS mode is identical to that of BASIC or XBASIC: 24 lines of 32 characters. And, as the chart shows, its features the same also. But beware! If you want to set FG/BG colours, charsets are numbered from 0 and start at ASCII 0. The first set containing displayable characters is number 4 in Forth. It is not stated in the manual but you can go to Appendix A (ASCII KEY CODES) and divide each column into groups of 8 characters, then number them - starting with zero! - and you will know what sets to use. If -VDPMODES is booted you can enter this mode by invoking GRAPHICS.

Then there is the MULTICOLOR mode. I have seen those 'crazy quilt' demos, what else it might be good for, I do not know. A 'character' is a block of 4x4 pixels, thus there are 48 lines of 64 blocks. Each of them can be set to a different colour. Use MULTI to enter this mode.

The BITMAP mode provides the highest resolution. Each of the 256 pixels in the 192 lines can be controlled individually. But there is a drawback. No

automation for sprites. Since standard characters can not be displayed TI has provided a neat feature with split modes where a large part of the display is in bitmap and the remainder can display ASCII characters. This comes in real handy when you want to experiment with the various graphic words since you will be able to see what you are typing. The words for entering the various bitmap modes are GRAPHICS2 (full-screen bitmap), SPLI (text at bottom) and SPLIT2 (text at top). By the way, if you opted for the 64-column editor, it uses SPLIT.

Study chapter 6 of the TI-FORTH manual. You will notice that: a) all you need to know is there, b) you are already familiar with most commands by way of BASIC, and c) if you have absorbed enough Forth by now, it is easy to see that the main difference is that parameters (in typical Forth fashion) go on the stack before the operative word is invoked. If you want to use sprites, pay particular attention to the section on Sprite Initialization (page 8 of Ch.6). Here is a brief example which shows how sprite is set up:

```
0 ( SPRITE SAMPLE )
1
2 BASE->R HEX
3 : SETUP
4     GRAPHICS ( set standard graphics mode )
5     2000 SSDT ( set sprite descriptor table )
6     1028 4482 4428 1000 2A SPCHAR ( char def )
7     3F 2F OF 2A 1 SPRITE ( sprite parameters )
8     15 0 1 MOTION ( motion parameters )
9     2 #MOTION ; ( set sprite in motion )
10
11 DECIMAL R->BASE
13 SETUP
14
15
16
```

This example is for standard GRAPHICS mode. If you wanted to use bitmap line 4 would have to read GRAPHICS2, (or SPLIT or SPLIT2), line 5 would have to include 3800 ' SATR ! but lines 8 and 9 would be superfluous. As this screen is loaded a sprite is displayed and set in motion. You can enter TEXT and return to the text mode. Typing SETUP will again produce the sprite.

Only by exploring and experimenting will you be able to gain proficiency with the graphics features offered by TI-FORTH. As I have mentioned, all the necessary information can be found in the manual, though not in a tutorial manner.

*** END SESSION 7 ***

*** CONCLUSION OF SERIES ***

AUTHOR'S NOTE:

Apparently, the information furnished by the above seven tutorials did the job. The flow of questions dried up and no. 7 became the final one of this series. I wish to thank all of those who contacted me and thus provided the inspiration to keep writing them. Hope you are enjoying Forth as much as I do.

continued from page 8

In Retrospect

These Tutorials were originally written about a year after TI orphaned the 99/4A. They have even been translated into Swedish since then. Now it is over 5 years going on 6 years since Black Friday in late 83 and the old 99/4A is still going strong, and the industrial strength of the expanded system has allowed all sorts of new developments, RANdisks by the megabyte if you want, a hard disk controller, and best of all now 80 column displays using the Yamaha development of the original TI video processor. The machine's potential is by no means exhausted and it remains a fascinating one to work on, and even the original console/XB remains a viable and now very low cost introduction to computers for new beginners.

(This is the final article of this very useful series on Extended Basic - thanks Tony. Ed).

Programs from TI*MES Library

by Stephen Shaw, England

Remember Personal Computing Today, Home Computing Weekly and Games Computing. They used to publish TI99/4A programs. Here are some of the better ones:

>MAG 1: Banzai Bunny, Fem on the Grid, Forklift Truck, Kitten Kong, Moth Mania, Motorway, Sea Diver, Treasure Island, Turtle Hop, Sir Prancelot, Skittles and Star Duel.

>MAG 2: Apple Scrumping, Basement Bob, Beagle Hike, Cave Attack, Channel Patrol, Debroids, Earth Defence, Electron, Eat Mince Pies (The Bosses Christmas Party), Fireman, Forest Rally, and Fruit Cocktail.

>MAG 3: Leap Frog, Lift Attendant, Mine Maze, Miss Muffet, Muncher, Nuclear Race, Robot Fire Snuffer, Sheep Dog Trials, Skiing, Slugs and Ladders.

>OLDIES BUT GOODIES. 1 and 2 on one disk. Biorhythm, Factor Foe, Hammurabi, Number Scramble, Word Scramble, Hidden Pairs, Peg Jump, TicTacToe and 3D TicTacToe. Released by TI in 1980 on two cassettes. By 1982 they were 8.25 each. The price we ask is closer to their real value perhaps! Authors John Plaster and Mary Anne Six (surname or age?). Of more historic value than usable value.

>POP DEMO Vn 1.1 from Roman Majer of Heilbronn in Germany. Not what we call pop. Four pieces, Amorada (written by Waldir de Azevedo), In the Mood (Joe Garland), Flohwalzer (?) and Charleston (Cecil Mack and Jimmy Johnson). All in machine code. See what that sound chip can do.

>SIDE*PRINT Vn 3.1 by Jim Swedlow. A program which will print Multiplan files sideways! ...

>SORT + DUMP. Two programs only! Revised March 88. Now includes Megasort96 for the Geneve, and improved sort routines for the 4A. Sort Experiment by J P Hoddie, sorts up to 1000 records in any type of disk file, with up to 8 sorting keys. NB: Memory image file and docs are revised. Source code is not revised. DUMP is by Wayne Stith and is to read and display/print any memory location: GROM, ROM, VDP, DSR. Not a very neat display but usable and supplied with source code.

>SPIELE 1. Extended BASIC games from Germany: Artillerie, Berzerk, Buddybalon, Circus (actually from COMPUTE!), Desert Flight (from Computer Kontakt), Fraggles, Two different froggers, and Indiana Jones. Rules in German but refer to line 1 for an English REM in some files, otherwise you should be able to work it out. Pretty good games.

>SPIELE 2. Again from Germany, all machine code this time, loader is not supplied. You need Editor/Assembler cartridge or FunnelWriter. OH MUMMY is my favorite as you try to make a path around the hidden treasures. When all the treasures have been discovered you may leave, not before, and do not bump into a mummy! There is also MOONFIGHT which comes with C source code, and KARATE-key FCTN[I](?) to start fight, and keys QAZ WSX to hit/kick in attack.

>TASS (Tri Artist Slide Show) Vn 1.0 A program which will read and display a mixed disk of Graphx, TI Artist and Draw-A-Bit 2 pictures automatically. Can sequence several drives. Includes a "lines" program too.

>TEXTLOADER+EA5LOAD: also from Paragon. The textloader is something many have tried to do for years, now it is done. Run TEXTLOADER and a DV80 text file is

read into the console just as though you had typed it in. You can quickly load a program on disk as text, or feed in a string of command mode instructions (do both together!). The EA5LOADER loads machine code memory image programs using Extended BASIC, and comes complete with SOURCE code.

>TI MATH. Combination of the two TI disks MATH LIBRARY and ELECTRICAL ENGINEERING LIBRARY with programs in Extended BASIC for base conversions, primes, hyperbolics, ordinary differential equations, matrices, fourier, simultaneous equations, PLL, Smith Chart, Filters and Root Locus. NB: Only one disk!!!!

>TI WRITER - EDITOR SOURCE CODE. This is mainly the original archived source code, but as two files are dated 1986, they at least are not original. Interesting comments and the names of the programmers at last. See how TI did it! Modify to suit yourself! Two disks.

>UTILITIES 15: Archiver 2.1 by Barry Boone, to pack (and unpack) several files into a single file, keeps them all together. SNAP CALC, a 13x20 Extended BASIC spreadsheet by Gary Strauss from HCM. TI Keys Vn 3.0 by Wes Johnson, instantly put up text on screen with CTRL[1] to 0 and A to Z, command mode or running; Prestel/Viditel Terminal Emulator (from Holland); Extended BASIC by J P Hoddie, a machine code program to run with Funlwriter: its the same as RUN "DSK1.LOAD" so you do not have to quit to get back to Extended BASIC; Tracker by Will McGovern, a track copy utility for owners of MYARC disk controllers; Unbasher by Barry Traver, (much revised March 1988) uncompresses those densely packed Extended BASIC programs.

>UTILITIES 16: Several character sets in Source code, Object file and Merge file formats, a File reader, a Speech Tutor Extended BASIC program, and M/COPY, the program all disk owners should have! After you have repaired any fractured files using ordinary file copy, process your disk with M/COPY. If the disk has more than 32 files, M/COPY will place all the file descriptor blocks into a single disk area, vastly cutting down access time and reducing drive wear. A must, especially for DD owners!

>UTILITIES 17: XBGC, a graphics program to translate from Graphx to CSGD and hence to TI Artist or PIO or MERGE format; 99-Calc, a small spreadsheet program, and a new Archiver (Vn 2.4 Jan 88) with a compression facility. A calendar program from MSP99 UG. CURSOR- a cursor redefinition utility.

>UTILITIES 18: One program only, CHARDES 5.2, a char/sprite design aid with a difference. LOTS of facilities, and fast to use. Can produce output as a MERGE format Extended BASIC program! Save time!

>UTILITIES 19. (243 sectors used to date): Some machine code conversion routines from LA: object code to CALL LOAD, CALL LOAD to object code, recovery of code hidden in an Extended BASIC program. A program to print graphs. And an Extended BASIC utility to give you 8 strings at the touch of one key (in command mode). Marty Krolls machine code disassembler (formerly on the FastTerm disk), and INFOLISTER which will list the vocabulary for your Infocom adventures, from the GAME1 files. and SUPERMAIL, an address data base.

EUROPA. An Extended BASIC suite occupying 343 sectors. For each country in Europe, shows the position, gives the car plate (eg GB), the flag, area, population, population density, money, capital. Slow but may have some educational value?

>XB*TOOLS Vn 1.2 by Jim Swedlow. A group of programs to help you write in Extended BASIC. They act on files saved in MERGE format to produce a reference list of variables, line transfers, subprogram calls,

DATA and DIM lines; remove REM lines; join lines together; change the names of up to ten variables at a time to names you specify; replace variable and user subprogram names with one or two letter names, delete, keep or resequence a part of a program, move a block of lines, combine DATA lines

>COMIC 1 contains the programs to create an animated picture, and an example using the Graphx walking man. The documentation file COMIC-ANL is in GERMAN and is unreadable using TI Writer. You must use DM1000 options T or P.

>COMIC 2 is an animated Graphx PLUTO, created with COMIC 1. To see Pluto breathing use LOAD AND RUN on file SHOW.

>COMIC 3. A scary learning GHOST goes boo.

>COMIC 4. A demo of animated 3D, as a wire frame cube revolves. NB: The comic disks contain two versions of SHOW. The seven sector version allows you to slightly amend animation speed by using = and / keys.

>Rude Program. Could be made into a family game by changing the words and graphics, to make it into a laser base shooting down animated monsters. It is not at present in that form and is emphatically not for family viewing. From Germany, in Extended BASIC. Unusual use of TI99/4A.

>INVENTORY MANAGEMENT: TI product PHD5024, formerly sold for US\$59! Requires the PRK or STATS modules or library disk MODUTIL. No Documentation, not sold by TI-UK. First create blank IF70 and IF80 files using the file INIT. INVENTORY is a blank PRK file. CONVERT1 transfers data from the IF80 file to the PRK file. CONVERT2 converts from the PRK file to the IF80 file. UPDATE transfers from IF70 (Main) to IF80 (Stock) file. REPORT1 and REPORT2 provide printouts. Copy master disk and use copy! The BASIC files provide a good lesson in how to use the "hidden" calls of the PRK module.

>SPELL and SORT: Very simple utilities from Software Specialities Inc, copied by TI99/4A USERS GROUP with the express consent of the copyright owner. SPELL is a spell checker for DV80 files, while SORT is a powerful general purpose sorter, which can sort ANY file on several keys. Any length, variable or fixed. An intermediate file is created, which is always FIXED, and may be quite long if the input/output files are Variable! Always specify one key. If you leave all keys set at 0 it will only remove blank fields!

>DATABASE1: A commercial database from SPC Software, who have not advertised for ages, and is not apparently on sale anywhere. I have been unable to contact them and assume they have ceased interest in their program. Two disks required. A simple database with several utilities. Perhaps best thought of as a "list processor". Satisfies the majority of my simple database requirements. Full documentation on disk.

>PRK CALLS DEMOS: A variety of programs in BASIC which REQUIRE the PRK or Stats modules or library disk MODUTIL. PRKCONVERT will convert PRK files into two files which a BASIC program can use; PRKHEADER which is IV80 and PRKDATA which is IV(n). Sample PRKDATA and PRKHEADER are supplied for a mail list program, with BASIC files ADD/DATA to use the database, B/PRKPRINT and PRKLABEL to print out to printer or screen. PRKWRITE is an inverse conversion program and will change the PRKDATA and PRKHEADER files back into a PRK file. PRKADDRESS and READ/NAMES are similar BASIC programs but use data files of IV135 format (not supplied). MAIL/LIST is a PRK file. I have added my own PRK utility, which transfers data from a PRK file to a DV80 file for TI Writer to use. Examination of these programs will show you how to use the extra calls.

>PICASSO PUBLISHER Vn 2.0 by Arto Heino. A bit map drawing program with lots of plusses. Draw on a "screen" 336 x 480 pixels. 32 brushes. 32 textures. 5 icons and icon editor. Six fonts and font editor. Can load TI Artist pictures. Can use CSGD fonts. Can load text from a TI Writer file. Very powerful. COST TEN POUNDS plus disk and postage and packing.

>PICTURE IT by Rodger Merritt. Produce 8" high banners using either TI Artist instances up to half a screen high, or the special font set included. Convert Instances to Extended BASIC Program (DISPLAY AT or SPRITES) or to TI Writer Formatter format for letter headings. Cost seven Pounds plus 3 disks and postage and packing.

==in the above two cases, payments are made to the authors, hence the higher prices==

High Resolution Graphics

Unexpanded
by Stephen Shaw, England

This is for our 50% unexpanded owners, and will allow you to sample many of the graphic routines I have (and will) present which really call for bit-map graphics. This routine allows you to plot in "pseudo" high resolution- you may program as though you had bit map graphics, but in fact what we are doing is continually redefining characters, which means the full screen area is not available, and it is a little slower than other means used! It is not possible to say when you will "run out of ink" as TI Logo, which uses the same idea, so aptly puts it. The routine is written to prevent crashes so you do run out of ink, but in itself will not check that R and C are within the screen boundaries. It is in Extended Basic and is by Gary Harding. Where in the other programs we have:

CALL DOT(1,R,C) you should use CALL PLOT(R,C,S)
CALL LINK("POINT",N,R,C) you should use CALL PLOT(R,C,S)
etc etc.
Your program must commence:

```
1 CALL SCREEN(2) :: S=31 :: CALL HCHAR(1,1,S,768)
2 FOR T=1 TO 14 :: CALL COLOR(1,16,2) :: NEXT T
```

YOUR GRAPHICS PROGRAM THEN FOLLOWS. Ignore any initialisation for other graphics languages, such as CALL LOADS and CALL LINKS or CALL GRAPHICS

Now the subroutine which should be at the END of the program:

```
31010 SUB PLOT(R,C,S)
31020 Y=INT(R/8+.875) :: X=INT(C/8+.875)
31030 H$="0123456789ABCDEF"
31040 B=C-X*8+8 :: P=2*R-16*Y+16+(B<5)
31050 IF B>4 THEN B=B-4
31060 CALL GCHAR(Y,X,H)
31070 IF H>31 THEN 31100 ELSE IF S=143 THEN SUBEXIT
31080 S=S+1 :: D$=RPT$("0000",4) :: CALL CHAR(S,D$)
31090 CALL HCHAR(Y,X,S) :: H=S :: GOTO 31110
31100 CALL CHARPAT(H,D$)
31110 N=(POS(H$,SEG$(D$,P,1),1)-1)OR(2*(4-B))
31120 D$=SEG$(D$,1,P-1)&SEG$(H$,N+1,1)&SEG$(D$,P+1,16-P)
31130 CALL CHAR(H,D$) :: SUBEND
31140 ! ROUTINE BY GARY HARDING
31150 ! FROM TIDINGS OCTOBER 1982
32000 END
```

Any programs you see which use some form of DRAW TO or LINE are merely connecting up to spots on the screen, and you can probably work out a way of connecting them by means of a looped routing with several CALL PLOTS.

This program will just cease drawing when out of ink! You must reset S=31 to reuse the routine.

c99 Function Library

by Warren Agee, USA

Function libraries are simply collections of tested functions (or subroutines) which reside in separate files from the main program. This helps the programmer to avoid reinventing the wheel each time he writes a program. There are basically two codes. The difference is that with source code the compiler has to process the code every single time you compile, while an object-code library is only compiled once.

Creating a function library using source code is the easiest of the two methods. Say you create a function `strlen()` which returns the length of a string. You could just type in the function's definition each time you need it, but a simpler way is to save the source code for the function in a separate file. If the `strlen` function is ever needed in a program, merely insert the following line at the start of your code:

```
#include "dskn.xxxx"
```

where `n` is the drive no. of where the file sits, and `xxxx` is the name of the file which contains the source code. The compiler will load in and compile the source code as if it were typed directly into the main program. The `#include` command works just like `.IF` (include file) of TI WRITER.

Creating a function library using object code is a bit more involved. You start out the same as before, with the source code of the function in question as a separate file. But, as in the case with `strlen()`, you also need the following three lines at the beginning of the file:

```
#asm
DEF STRLEN
#endasm
```

The actual definition for `strlen()` would follow these lines. The first line tells the compiler that the following code is not in C but in assembler. The second line tells the computer to make the `STRLEN` code available to another program. Even though it is defined in this program, a totally separate program (main) will also have access to it. Note 1) the leading space before `DEF` (that is important) and 2) the function name is in capital letters. The third line tells the compiler that the assembler code ends and C code begins again.

The `DEF` directive can be used to externally define many, many functions at once; just separate each function name with a comma.

Now compile and assemble your "mini-file" which contains just one function. You now have a standalone function library consisting of the `strlen()` function that can be used in ANY program. But how do you go about linking it to your main program?

The next thing to do is add three more lines to the start of your main program:

```
#asm
REF STRLEN
#endasm
```

Looks familiar! But instead of defining an external function, we are REFerencing one. This tells the computer that even though the main program will use the function `STRLEN`, it must look OUTSIDE the current program for its definition. Please note that you can REFerence more than one function as with the `DEF` directive. If you look at the `STDIO` file on the c99 disk, you will note that it contains mostly REF's!

When your program is compiled and assembled, be sure to load in the `STRLEN` file that you already compiled before you run your program. Under E/A option 3, first load

your main program, then CSUP, then any other required files, then your `STRLEN` file. Now you are all set to go!

The theory behind this is not that hard to grasp: instead of including the definition of `strlen()` within the main program, we compiled it separately as a standalone module. But without the REFs and DEFs, there would be no communication between the program module and the `strlen()` module. This momentary slip into assembly language allows us the opportunity to open a line of communication between separately compiled modules. o

Small List, Version 1.0

by E.P. Rebel, Netherlands

This file contains documentation and assembly source file for the program `SLIST/EXB` in the program menu of `TEXPAC BBS` for July-August 1989. this is a good assembly programming example of use of the User Interrupt Routine.

```
*****
* SMALL LIST *
*****
* E.P. REBEL *
* V1.0      *
* 05-05-1986 *
*****
```

With `SMALL LIST` you will be able to make listings on paper just as they appear on the screen. An interrupt routine changes the line length from 80 to 28 when you wish to do a `LIST "PIO"` or something like that. Loading `SLIST` is the same for `TI-BASIC` (with a `MINI MEMORY For EDITOR/ASSEMBLER` module) and `EXTENDED BASIC`.
`CALL INIT`
`CALL LOAD("DSK1.SLIST/OBJ")`
`CALL LINK("LIST")`
`OLD DSK1.basicprog`
`LIST "device"`
`CALL LINK("OFF")`

`CALL LINK("LIST")` installs the interrupt routine.
`CALL LINK("OFF")` removes the interrupt routine.

`SLIST` is a `PUBLIC DOMAIN` program!

Eric-Paul Rebel
Mereelstraat 27
1223 NR HILVERSUM
The Netherlands

```
DEF LIST,OFF
```

```
LIST LI RO,INTERR      set user interrupt address
MOV RO,@>83C4
RT
```

```
OFF CLR RO             reset user interrupt address
MOV RO,@>83C4
RT
```

```
INTERR BLWP @INTER     user interrupt entry address
RT
```

```
INTER DATA WSPACE          BLWP vector
DATA START
```

```
START CLR RO
MOVB @>8307,RO
CI RO,>5000
JNE END
LI RO,>1C00
MOVB RO,@>8307
END RTWP
End of interrupt routine.
```

```
WSPACE BSS 32             routine workspace
END
```

o

RAMdisk Folly

or

How not to build a solid state floppy by Forthwrite

Many months ago I decided I would like to have a ram disk as did a number of others of the Melbourne Users Group. One member arranged for the bare cards to be supplied while another contacted an electronics supplier for the memory chips which, bought in bulk, would be considerably cheaper. Well as time passed we received and paid for our cards and memory chips and went about collecting the rest of the necessary bits. Then they were all assembled and checked. Thereafter the usual poring over the construction directions and soldering between times commenced. When the card was fully assembled but no chips fitted the connections were checked with a multi-meter and found to have continuity. An abundance of humour.

Next followed the technical part with a mix of assembly and testing.

First the card had to have the operating system chips fitted and then tested to see that the indicator LEDs lit up as intended. Then one chip was fitted and the system tested again. The construction directions advised against proceeding with assembly if there were any error messages given out in the test program. The test program consistently told me I had 6k of the 8k first chip giving errors.

The same after trying three different memory chips. Humour running out fast!

Contacting the supplier of the cards and the construction directions I was told that it had to be due to open circuits somewhere. Discussion with another member of the group suggested that perhaps the contacts on the top of the card had not been made properly and that I should make sure that these connections were wetted out properly. Humour somewhat restored?

I felt that the tip of my 25 watt soldering iron was too thick for the job and set about machining a thinner bit on my model makers lathe. The tip was machined down to about 2 mm diameter and curved to allow it to touch the bottoms of the machined pin socket legs. Despite this I managed to build a solder bridge UNDER the socket which seemed impossible to break. Not appreciating that the socket was more expendable than the card I then set about removing the socket.

Yep. You guessed it. The socket came away with a number of the through card ferrules and a few millimetres of conductor. Not a pretty sight. I will admit to being a trifle cranky about the whole matter and decided to give it a rest for a week or two. Humour decidedly black!

Well, eight months later my fury had subsided somewhat and was being directed against the slower performance of my computer when compared with those fitted with solid state floppies. New supply of humour in store.

I was still not entirely convinced that the card deserved any consideration and was looking at building a mother board with the enlarged scratchpad RAM and mentioned it to a friend. He asked, too casually, if I had finished the ram disk and I had to confess that it was still in its wretched state of eight months before. In retrospect it seemed that there was a message there and I decided that the ramdisk would be fixed before the motherboard so back to work with the soldering iron and multimeter. The bridged socket was cleaned and the damaged conductor tracks repaired with fine tinned wire and the socket installed. A further run with the test program still gave the same results - 6k was giving errors. Humour converted to patience.

The multi-meter I have uses 3 volts for the lowest resistance range and then goes up to 22 volts for the higher ranges and while I was sure that I had not zapped the chips with too high a voltage I felt that a change of system chips was in order just in case. These were duly collected over some weeks. Unfortunately all suppliers did not have all the chips required so visits to various shops had to be fitted in when I was in the area.

Then came the day when I had all new system chips in the card and one memory chip in place and ran the test program: still showing 6k errors. I knew that there was continuity on all conductors right up to the chip itself. Knowing that solder joints can have their own electronic characteristics I decided that a logic probe was the order of the day to check if a pulse could not pass where a steady current could. The probe was duly obtained but the bare board had no way of generating pulses so a pulser was bought as buddy for the probe. The Tandy shop in the local shopping street did some business. So did the pulser and probe. Every conductor and connection was tested and found to be OK. A photocopy of the circuit diagram was ticked off twice in the process. So just what the heck was wrong with the jolly thing?. Patience converts to wonderment.

The pulser and probe could also tell me if the chips were working correctly but I did not know what the pinouts were so I had to invest in a couple of books to gain that knowledge. Also bought a four battery holder and leads to power up the board outside the expansion box. Learned then that most of the chips seemed to work OK.

About 1.30 one morning I had checked the dip switch connections yet again and inadvertently placed the card in the expansion box without setting any and ran the test. Hello? 6 k errors again! I took out the card altogether and ran the test program and lo and behold 6 K ERRORS. Wonderment turns to resignation!

Needless to say I was pretty dark on a few people having realised that the trauma with the socket was completely unnecessary. Also realised that at that time of the morning one has to contain one's wrath else have the neighbours complain.

After this discovery I decided to ignore the documentation supplied with the card and tried to load the operating system. This seemed to work well enough so I added a couple more memory chips and found that the loader program showed a like increase in memory. Added two more but no increase in memory appeared. Went over all the solder connections with the iron and bridged two, or so I found out after loading the operating system when the thing crashed. I fixed that then discovered that the unfortunate socket may not have wetted out its upper connections. So, I took it out and cut all conductor tracks to it. I remade all connections to the underside of the card with wrap wire - hey Presto 16 k appeared, then 32 k, then 64, then 184. Seems I was getting somewhere at last. I cautiously allowed a small seed of HOPE to be planted.

I loaded Funnelweb and played around with the operating system and menu to get the feel of it and then tried to run a program or two. Editor started to load but froze with characters that looked like Cyrillic (Russian). Formatter ran OK. DM-1000 showed 0 read write errors on loading the disk. TI-Disk Manager II showed no errors on the quick test. Resignation turns to distrust. Hope sloooooowly growing. I decided that I would not trust any tests which gave error read outs any more and further decided that I wanted to see what was happening.

Observing the ramdisk when it formatted suggested that only the sectors for the disk were assigned but no preparation of the medium was done as occurs in true floppies where each sector has orderly rows of E5 E5 printed on the tracks. This was confirmed when I looked into the sectors after 'formatting'. I found that if I formatted a floppy (no name) and copied it to the ramdisk using the sector/sector option 2 of DM-1000 then 720 sectors would have E5s printed on them. Looking into the sectors after that showed whether or not they had been written-to properly. Many had not! Distrust turns to determination.

I went through every sector and wrote down the start and finish sector where misprints were found. Printed out some sectors where the errors were particularly prominent. I analysed the results and found that errors were occurring in 32 sector blocks (8 k) which meant that the problem must be related to some memory chips. I also found that errors were occurring in the same positions indicating the address bus was at fault.

continued on page 30

From the Service Bench

by Geoff Trott

RAMdisk data loss

When the power is turned off or turned on to a RAMdisk, there is the potential for data to be lost. The memory chips are always supplied with power using a battery but some of the control lines can do strange things. Data can only be changed in static RAM memory chips if both the device select line is asserted and the write enable line is also asserted. For the 8K byte memory chips, there are two chip enable pins, one asserted low and the other asserted high. It turns out to be easier to use the asserted low pin for normal device selection and the asserted high one can then be used to make sure the chip is not selected when the power is not present by tying it to the 5 volt supply through a LED with a resistor to ground. The LED has a voltage drop which will cause the select line to go low (not selected) before the voltage reaches the point where all the logic stops working. This means that it does not matter much what the logic generating the write enable line does, or indeed the logic generating the individual device select line (asserted low).

For the 32K byte memory chips the second chip enable pin is now needed for an address line so there is only the one chip enable pin (asserted low) which now has to be used for both functions. This is solved in the steady state conditions (that is power on and power off) by using open collector logic to drive these lines with pull up resistors to the battery supply (which is powered from mains when power is applied). However, the short time when power is turned on or off can cause the logic to generate pulses on these lines. Particularly when power is turned on, the 74LS259 ICs can power up with any particular value stored in them and they then are reset by the power up reset pulse which occurs some time after power is applied (you can get many pulses in a millisecond). There is not much else that can be done with these signals, particularly as there are so many of them. I have tried the approach of using a circuit that speeds up the switch on of the power supply which seems to work some of the time, but my latest idea is to try and isolate the write enable line as soon as the power starts to drop and until after the reset has been applied to the 74LS259s.

The reason for going this way is that it is only one signal which goes to all memory chips and if it is disabled then the contents of memory cannot change! The write enable pulse comes through a tri-state buffer chip from the console I/O port. This means that when the power of the console is off the signal may well be dragged low and as power is applied to the buffer chip it is hard to ensure that the buffer output does not go low for a period of time. I have been wrestling with Don Dorrington's RAMdisk for some time now, and have arrived at a circuit which appears to work if power is applied and removed from the RAMdisk in the correct order.

The essence of the problem is to isolate the WE line from the console after the buffer so that it can be connected to the battery when power is off without causing any current drain from the battery. It is also important to disable the WE line until the reset on power on has taken effect. That should mean that no data can change even if the CE lines pulse on power on or off. I originally tried to do this with a series transistor, which worked partially. Then I decided that a FET would be better as it would act like a resistor when on. I also knew of an IC which provided an array of CMOS FETs which could be used individually as well as in pairs as inverters. This made the circuit a bit simpler.

The final circuit powers the IC from the 9 volt supply (which remains longer than the 5 volt supply on power off and appears earlier on power on), monitors the 5 volt supply to detect the state of the power with a

delay for power on and uses two inverters to sharpen the turn on and off of the control FET which is placed in series with the WE(L) line. If you have a clock, I suggest that you cut the track going to pin 23 of the clock chip and connect pin 23 to the in-board lead of the LED which comes on when power is on.

The circuit works well but still requires some care about when power is turned on and off, particularly if there is a clock on the circuit board. Power to the RAMdisk and clock must be removed before the console is turned off and then not turned on until the console is turned on. When power to the RAMdisk and clock is turned off the clock on the menu shows 00:00:00. Then the console can be turned off. After the console is turned on and the title screen appears, the RAMdisk and clock can be turned on and pressing <QUIT> (FCTN[=]) will cause the menu and clock to appear with the correct time.

Console repairs

I finally found the problem with Lou Newhouse's console. The console repair group had a look at it originally and found that when on the console tester everything looked OK except that the check sums for the system ROMs were changing all the time. They tried changing one system ROM without any joy and I checked some of the ICs in the 16 bit to 8 bit multiplexer. I decided to make a new version of the console tester program which would just loop on the ROM test so that I could look at the signals with a CRO. Eventually, I noticed that one of the address lines (A7) of the processor was not going high. I removed all load on that line but it still was always low and when the processor was removed and checked it did not work. I replaced the processor and all was OK. What a pity that a processor has to be discarded because of one address line! Perhaps a pull-up resistor would make it work! o

How Does It Work?

by Geoff Trott

So far I have discussed the application of power to the console and the generation of a reset pulse by some resistors and capacitors, some time after power is applied. This pulse is applied to three ICs in the console, the TMS9901 (peripheral interface chip, used for the keyboard, joystick and cassette interface), the TMS9929A (video processor, producing the picture on the screen) and the TMS9900 (the central processing unit, CPU). The reset pulse does different things in each of these chips, but the main thing it does in common is to put them all in a known condition. What this does to the CPU is what I would like to discuss this month.

The reset pulse interrupts the processor in whatever it may have been doing and causes it to enter a special cycle of operation. The reset pulse is also generated when a module is inserted in the cartridge port with the same effect. The processor stops what it is doing and goes to a defined place in its memory space to pick up two items of information. The first of these defines the place in memory where it is going to store its registers, while the second item is the place in memory where the first of its instructions is stored.

Well that is easy, is it not? Perhaps not to those who do not understand about memory. I shall now try to explain all about memory, as if you are having difficulty with that, then every thing from now on will be a mess. Memory is perhaps the most important part of a computer, as it the memory and what can be done with it that separates computers from calculators.

Let us go back a bit then. Memory is just that, electronic bits and pieces which can remember and hence store information. Since it is electronic, it is difficult to have a good, error free memory device which can store more than one or two pieces of information. Think of a switch which is either open or closed, magnetic material which is magnetised either north or

south pole in one direction, a capacitor with a voltage greater than a threshold or less than that threshold. It is easy to imagine more than the two pieces of information but not so easy to make it reliable. So memory is designed and built out of elements which can only store one out of two items of information. Rather than talk about items of information all the time we give these two items of information names. What names are chosen? No prizes for guessing 0 and 1. Yes that is correct, the first two numerical digits 0 and 1. What these names represent in terms of the actual physical quantity in a memory only matters to the designer of memories or perhaps to the hardware guru. For our purposes, I want you to only remember that the computer has a memory which is made up of devices or cells which can only remember or store a 0 or a 1. Using the numerical digits allows the use of numbers to represent the information that is stored in the memory.

Computers need to store more than two items of information so that the memory cells are not treated separately but grouped together. One grouping is into groups of 8 cells called bytes. Each cell (or bit) can store 2 items of information so that 8 cells can store $2^8=256$ different items of information. Or in number terms we can store the numbers from 0 to 255 in a byte. Another grouping is into words with 16 bits. This gives $2^{16}=65536$ different items of information which could be stored in a word of 16 bits. In order to store information in a group of memory cells (a byte say), the computer must be able to know where each byte is in its memory, that is, it must have some way of addressing each byte of its memory. Each byte must have its own name. Once again, no prizes for guessing that the naming of each byte or group of memory cells is done using numbers. So memory can be thought of as storing numbers in groups of cells (bytes or words) whose names or addresses are also numbers. Not surprisingly, since the logic used in computers is also two valued (true or false, 0 or 1), the best type of numbers to use for addresses are also numbers to the base 2.

What this means, is that as soon as you start to look inside digital computers, you run into binary numbers or numbers to the base 2. The trouble with these for humans is that they are contain so many digits, each of which is either 1 or 0. Numbers of this form are very hard to talk about and equally hard to remember. They are also hard to convert into decimal numbers with which we are familiar, but easy to convert into hexadecimal or base 16 numbers because of the relationship between 2 and 16, the two bases. It is easier to talk about and remember hexadecimal numbers than their equivalent binary numbers. If you are not too familiar with hexadecimal or even binary numbers then do not worry, just keep in mind that they are only a shorthand way to represent the information stored in the memory or just a convenient way to name each byte or word of the memory. As long as you can tell the difference between two hexadecimal numbers you do not need to convert them to decimal or more familiar numbers.

To get back to where I was, on the arrival of the reset pulse the CPU (TMS9900) drops what it is doing and gets two items of information from its memory. First it goes to address 0000h (4 hexadecimal digits) and gets the word (two bytes, 16 bits) from there (actually getting the bytes at 0000h and 0001h as the memory is byte addressable) and uses these bytes as the address to its register storage area in memory. Then it gets from addresses 0002h and 0003h the 16 bit word which is the address of the instruction which is to be executed next. The CPU puts the first 16 bit word into its workspace register and the second 16 bit word into its program counter. Actually, it only puts the most significant 15 bits into the program counter as it is always looking for instructions to be stored at even addresses and in 16 bit words. Then it goes into its normal cycle of fetching the word whose address is in the program counter, placing that word into its instruction decoder, incrementing its program counter (by 2, as adding 1 to the second least significant bit (bit 14) is the same as adding 2 to the 16 bit word) and executing the instruction. The execution of the instruction can cause

a lot of different things to occur but the basic cycle just keeps repeating so that the CPU slowly (quite quickly if fact but slow to talk about) works its way through the list of instructions stored in its memory.

Starting at address 0000h and going to address 1FFFh in the CPU address space (one eighth of the whole space, 0000h to FFFFh) is read only memory (ROM), which means that the contents of this memory does not change so that it is always available even after power has been turned off and on again. This is important if the computer is to be able to "come to life" when power is applied.

Well, I hope that makes some sense, as next month I will talk more about what the program stored in the ROM does, which will lead us into the world of GROMs (Graphics? read only memory)!

continued from page 28

Reviewed the whole sorry state of things and identified from the sector analysis that chips 3,5,7 and 13,14,15+17 were at fault. Recalling that I had already tried wetting out the connections twice and still apparently had faulty ones decided to wet out with extra soldering flux. This worked like a charm since only the above sockets were reheated and the disk was found to work error free. I was pleased about that!

Lessons learned:

Cards that had been lying around for some time probably had some oxidation on the surfaces. I should have ensured that the connections on the top of the card were freshly tinned before assembly. Same with the corresponding socket pins.

Wetting-out joints should be done with extra flux and not with extra solder.

Test programs will themselves have to be tested to demonstrate that they really do work.

Bouquets:

Bernie Elsner's 32 Diskmanager set out the errors in copying in easy to analyse format. Miami User Groups ROS works great.

Funnelweb must get a mention, but what more can be said about Funnelweb?.

Possible Sector Editor enhancements:

It would be convenient, I have often thought, to be able to initialize one, a few or all sectors or to fill them with test numbers, without having to copy L5s etc. sector to sector one at a time.

Tetris

A game review

Tetris is a computer program from the Soviet Union, now converted for most computers and even released as a coin-op machine, it is THAT good. Very simple in concept, it can take a moment or two to realise in play what you are meant to do—different shapes fall from above, and by rotating them, and moving them left and right, you have to do all you can to pack them so tightly that complete rows with no holes are made up—when you do the whole stack moves down as that row disappears.

The first TI version was TI TRIS, the version that follows is Tetris and comes from the September 1989 issue of MICROpendium. Just think, if you had a subscription you could have been playing Tetris some months ago! As with all programs with DATA statements, take very great care keying this in! This printout is from a program which has been up and running correctly, and can therefore be guaranteed free from bugs!

The UK disk library also has a machine code version from Germany. The original program was by Soviet citizen Alexey Pajitnov.

(The Extended BASIC version is in the March-April TEXPAC program download).

Regional Group Reports

Newsletter Update

by Bob Relyca

Meeting Summary For April

Banana Coast	08/04/90	Sawtell
Carlingford	18/04/90	Carlingford
Central Coast	14/04/90	Saratoga
Glebe	12/04/90	Glebe
Illawarra	09/04/90	Keiraville
Liverpool	13/04/90	
Northern Suburbs	26/04/90	???
Sutherland	20/04/90	Jannali

BANANA COAST Regional Group (Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month at 2 pm sharp. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Ballin, telephone (066)53 2649, or John Ryan of Mullaway via the BBS, user name SARA, or telephone (066)54 1451.

CARLINGFORD Regional Group

Regular meetings are normally on the third Wednesday of each month at 7.30pm. Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02)871 7753, for more information.

CENTRAL COAST Regional Group

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

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce St, Glebe. Contact Mike Slattery, (02)692 0559.

ILLAWARRA Regional Group

Regular meetings are normally on the second Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. Contact Lou Amadio on (042)28 4906 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. No update was received this month so members are advised to contact Larry to see if the above date of the 13th(Good Friday) is valid this month ED. Contact Larry Saunders (02)644 7377 (home) or (02)642 7418 (work) for more information.

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132.

Come and join in our fun. Dick Warburton.

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Group co-ordinator is Peter Young, (02) 528 8775. BBS Contact is Gary Wilson, user name VK2YGW on this BBS.

Peter Young Regional Co-ordinator

TISHUG in Sydney

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

Russell Welham (Meeting coordinator).

Local Newsletters

TIBUG, December, 1989: Summary of new hardware/software; Project Management(reprinted, TISHUG); The Game of 'NIM'; Review - HARDMASTER; Tips from the Tigercub #15; Let's Talk Ramdisks; 'SHUFFLING'(Random Number assortment) by Jim Peterson; Commentary on 'Non-Hackers'; Information on the "Internal Board".

MELBOURNE TI*MES(disk), December, 1989: The information I received contained mostly a summary of the Faire held in Melbourne including speeches.

OVERSEAS NEWSLETTERS

NORTHERN NJ 99er's Users Group, December, 1989: Ramdisk Recovery; Suggestions on buying Modems; TI-Keys and TI/Keys&Funnelweb(reproduced in this issue of TISHUG) by Charles Good; Tips from the Tigercub #56.

The OTTAWA T.I.99/4A Users' Group, December, 1989: A list of Coming Events and an Editorial; a 'browse' of the club's disk library: TI-Artist Plus. a review; Assembly Utility Programs for Extended Basic Part 2 by David Caron; "Hot Bug" Fairware, an article on Fast Extended Basic by Lucie Dorais.

TI FOCUS, December, 1989: News and Reviews of TI Land; Hints of creating Flow Charts; a brief review of MORE, a file typing utility developed for the GENEVE; a page of club news; some new club disks;TI-ARTIST PLUS! PLUS! PLUS! by Tom Arnold; QDAV - Funnelweb's New Enhanced Quick Directory, reviewed by Charles Good; EXEC, a Geneve Utility Loader by Eric Wicklund, and a look at GETSTR, GETKEY, and MENU also by Eric Wicklund.

ROM, December, 1989: An editorial and the 'Board Minutes'; Harrison Software Music by Earl Raguse; Write Right #5 by Siles Bazerman; Tricks and Treats, some juicy snippets regarding printers by Stan Corbin; an announcement of the Fest West software and workshop day in Feb, '90; And So Forth #44 by Earl Raguse.

TIC TOC(Rocky Mountain 99ers, December, 1989: Editorials; TI-Base Tutorial by Martin Smoley; TI Base Version 2.0, a tutorial; a list of club disks; Disk Sector 000 by W. A. Molander; Codes and Encryption with XOR.

TopIcs(LA 99ers), December, 1989: Beginning Forth #18 by Earl Raguse; Procedures to set printer by Fred Moore; Random Ramblings by Bill Gaskill; This month in TI History(interesting! ED);A Child's First Look at Multiplication by Tony Falco(includes program); various snippets by Chick De Marti; a list of software available to members.

THE PUG PERIPHERAL, Pittsburgh, December, 1989: Club News by Gary Taylor a list new products; High Resolution Graphics, Section 3: Screen Graphics(including a look at the graphics capabilities of a wide range of available software)and the 99/4A by Anne Ohein; Console Debugging service; A list of TI 99/4A and Geneve Suppliers and repairers; 'Forth Style'; Minutes of the November meeting.

TIDBITS, December, 1989: A summary of up-and-coming faire and new software releases; A report on the Chicago Faire by John Koloen; A review of 'Legends II': Multiplan with the Gram Kracker by Lynn Crow; Four-A/Talk a brief look at some new releases by Texaments, Asgard and the new version of John Johnson's 'Boot'; Star Fleet Tech Drawings II by Gary Cox; Reviews of various other interesting bits of software such as The Bit Bucket, PC-Transfer Utilities and Identifile. The latter being a different sort of disk cataloguer.

continued on page 20