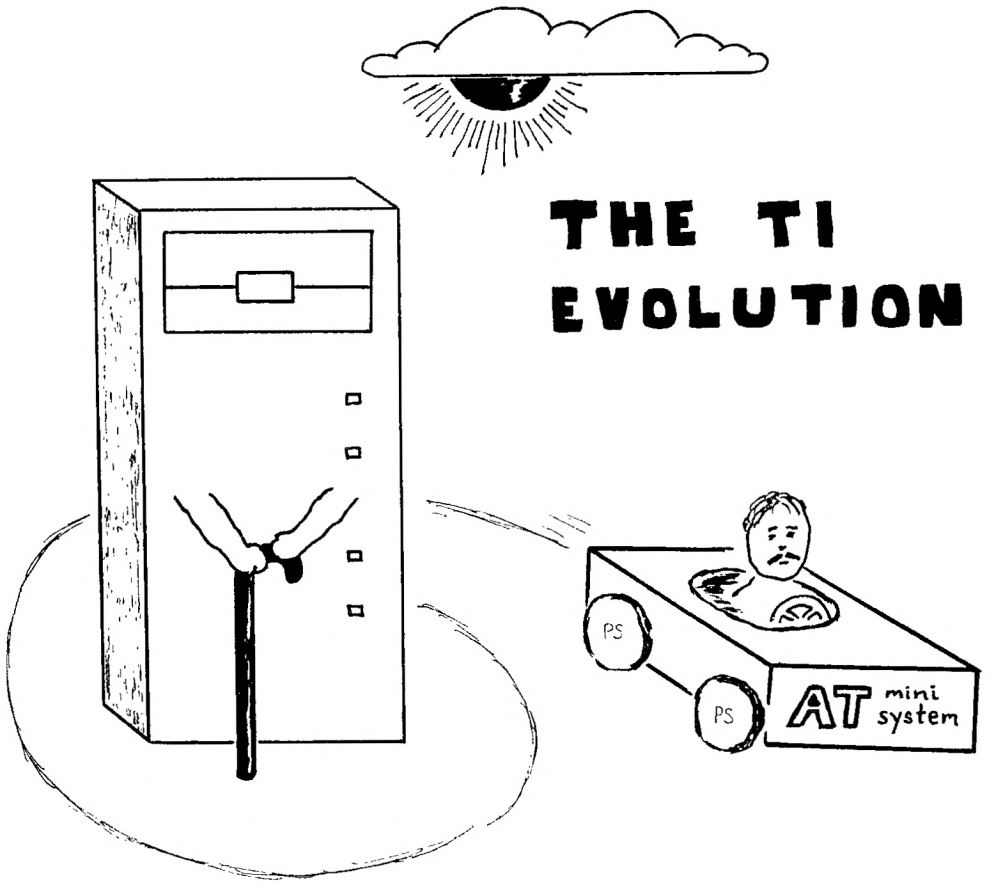
TISHUG

NEWS DIGEST

Focusing on the TI-99/4A Home Computer

Volume 7, Number 8 **September 1988**

Registered by Australia Post - Publication No. NBH5933



P.O. Box 214, Redfern, New South Wales, Australia, 2016 **$2**

TIsHUG (Australia) Ltd.

TIsHUG News Digest

September 1988

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board

Co-ordinator
Chris Buttner (02) 871 7753
Secretary
Terry Phillips (02) 797 6313
Treasurer
Percy Harrison (02) 808 3181
Directors
Cyril Bohlsen (02) 639 5847
Russell Welham (043) 92 4000

Sub-committees

News Digest Editor
Geoff Trott (042) 29 6629
BBS Sysop
Ross Mudie (02) 456 2122
Merchandising
Steven Carr (02) 608 3564
Publications Library
Warren Welham (043) 92 4000
Software library
Terry Phillips (02) 797 6313
Technical co-ordinator
John Paine (02) 625 6318

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
Bob Montgomery (042) 28 6463
Liverpool
Larry Saunders (02) 644 7377
Northern Suburbs
Dennis Norman (02) 452 3920
Sutherland
Peter Young (02) 528 8775

Membership and Subscriptions

Joining fee	$5.00
Annual Family Dues	$25.00
Overseas Airmail Dues	AUS$50.00
	or £22.00
	or US$30.00
Publications Library	$5.00
Texpac BBS	$5.00

TIsHUG Sydney Meeting

The next meeting will be at 2 pm on 3rd September at Woodstock Community Centre, Church Street, Burwood.

Printed by
The University of Wollongong
Printery

TIsHUG News Digest ISSN 0819-1984

Index

They're off

by Geoff Trott

It is nice to report that we are sending out 260 copies of TND each month so that our membership has not dropped as much as was feared. I hope that our efforts in generating this document each month has contributed to your continuing interest in the club. We are printing some material this month which Shane obtained from the USA by logging onto various BBSs in 1985. It is interesting how material that old can still be interesting today. That, I am sure, is part of the appeal of the TI99/4A that remains long after it should have been dead and buried. There is so much to learn about the machine because TI built a good operating system into the console. This has made it so easy (once you know the tricks) to add new devices to the computer, and even helped Myarc, when they developed the Geneve, to keep all the same peripherals as the TI99/4A used.

continued on page 30

Geneve 9640, The Alternative?

by Garry Christensen, TIBUG

Should I buy a Geneve? A lot of people have asked that question. Many have resolved it, others have not. Many have expressed their opinion. Of those, some have been informed and others have been mislead. It seems that in recent times, the original reason for the Geneve was lost.

We all support the TI99/4A home computer. If you do not, you will not be reading this. The Geneve was released as a TI99/4A compatible computer, an upgrade for those who wanted more from the TI99/4A.

Due to the MSDOS-like operating system used in the native environment, the Geneve was compared to the IBM clones. Was it better? Why use this operating system? Where is all the software for this computer? Why cannot Myarc release all the hardware details yesterday? Is all this really any thing to do with the Geneve?

Let me say again that the Geneve was devised to be an upgrade from the present TI99/4A. If you want something to be like an IBM, buy a PC clone. The Geneve allows the TI99/4Aers an opportunity to enhance their system, the same system that they have known and loved for all these years.

Perhaps I am biased. I have already decided what I expect to get out of the Geneve. I could have upgraded in a different way and still have a similar system. This is what my Geneve is to me. For the moment let us ignore the native mode and consider only the TI99/4A emulation mode - GPL mode.

First the physical aspects. I am now using a full 101 key keyboard instead of the little one that is on the TI99/4A. Rave 99 have a keyboard interface that will allow me to use this same keyboard with the TI.

It costs US$150, or about A$280 by the time I get it. The price of the keyboard varies but about A$100 would be a good average.

I also have a GRAM emulator. This allows me to load my modules from disk, write GPL code (if I was that way inclined) and edit the GROMs in the modules or the console. This could also be done with the Gram Kracker or a GramKarte from Mechatronics. I see the GramKartes advertised for US$200. That is around A$350.

Then there is the 9938 video processor. It gives me access to higher resolution graphics and an 80 column screen. Digit Systems also produce a peripheral that will give the humble TI99/4A these features. This costs A$380.

The GPL mode automatically configures a 180K RAMdisk and a print spooler. Allow say A$200 for the RAMdisk and A$100 for the spooler.

Here is the all up cost of doing the same as the Geneve using normal means:

Rave interface	$280
Keyboard	$100
Gram emulator	$350
Video card	$380
Ramdisk	$200
Spooler	$100
Total	$1410

The above system still has not given me 3 times the speed of the TI99/4A computer. I have nearly all the features but not the speed.

There is further developement being done in the GPL environment. Perhaps soon all of the 512K of memory will be available to the programmer.

Now it is time to look at the native mode of the Geneve. I have the features of MDOS (some like it, some do not). There are programmes being released so I can use them where I would not be able to, using an expanded TI99/4A.

Well there it is. The Geneve is an upgraded TI99/4A and a little bit more.

I have all the programmes that I need to use the Geneve. I have been using them all along. Did IBM have such a selection of programmes available when they released their first personal computer?

Is the Geneve good value? That depends entirely on what you want. If you are considering upgrading, have a good look at the Geneve, then have a good look at the alternatives and make up your own mind. Too many people have their minds made up for them by others. Sometimes those decisions are right. Sometimes, o

TiSHUG Software Column

by Terry Phillips

Here is a list of some of the new disks added to the software library over the past few weeks. Most of this software came via Stephen Shaw from the UK User Group.

LIBRARY DISK No. A240: STAR UTILITIES - Nice collection of useful utilities - X/D/32K

LIBRARY DISK No. A241: BANNERMAKER - Print out banners - X/D/32K

LIBRARY DISK No. A242: CHARTMAKER 2 - Graphs package - X/D/32K

LIBRARY DISK No. A243: LABELLER - Produce labels with graphics - X/D/32K

LIBRARY DISK No. A244: CHARACTER DESIGN - The best of its kind currently around - X/D/32K

LIBRARY DISK No. A245: TELCO V2.0 - Latest version - X/D/32K (This is an archived and compressed version)

LIBRARY DISK No. A246: GAMES - A collection of 25 games many of which are old favourites and sure to retain interest for a long time

LIBRARY DISK No. A247: GRAPHICS DISK - High resolution plotting is possible with this routine. The disk contains separate help files. X/D/32K

LIBRARY DISK No. A248: OH MUMMY - Good assembly action game - X/D/32K

LIBRARY DISK No. A249: COMIC ANIMATOR - Great utility for animating pictures. Contains a number of demonstrations and help files - X/D/32K.

LIBRARY DISK No. A250: UTILS16 - A great collection of assorted utilies - X/D/32K

LIBRARY DISK No. A251: UTILS19 - Another good collection. Includes a file to enable the printing out of the vocabulary of Infocom adventures - X/D/32K

LIBRARY DISK No. A252: SEGREGATION - A hard and complex game to play, very well done - X.

LIBRARY DISK No. A253: GAMES - Contains many good games, mainly in French, but simple to play. Contains a good demonstration version of Space Station Pheta - X

LIBRARY DISK No. A254: CARFAX ABBEY - An excellent graphics adventure. The disk also includes 3 good games in Extended Basic Boxes, Snakes and Ladders and Taskforce. - Carfax Abbey requires a disk and 32k while the remaining games require XB.

LIBRARY DISK No. A255: JP HODDIE GAMES - A good collection of games, some in assembly and some in XB

LIBRARY DISK No. A256: GAMES - A nice collection of games mainly in German but most are reasonable easy to work out how to play. Most require XB.

LIBRARY DISK No. A257: GAMES - A collection of games that have appeared in various magazines that used to support the TI99/4A. Most in XB.

LIBRARY DISK No. A258: EDP - A display enhancement package demonstration disk. Write to the author for the full routines. Looks good. - X/D/32K

From this list released at the September meeting will be:

DISK A247 - High Resolution Graphics, which will amaze you at the complexities of the designs possible.

DISK A249 - Comic Animator, with fantastic demonstration files.

DISK A254 - Carfax Abbey is the main program on this disk and it is a very well written adventure of the Tunnels of Doom type. It takes a while to set up but is well worth the waiting time.

DISK A255 - JP Hoddie Gamesdisk - some excellent assembly material is on this disk, including a very good version of Asteroids.

I hope that should be enough to keep everyone satisfied for a month.

More news on the software scene next month and remember if you have a need for a specific program or programs from the library, write or give me a call. o

Secretary's Notebook

by Terry Phillips

Two new members to welcome this month and they are:

Joanne Corrie - Airds
P M Condon - Molong

Hope that both of you can make it to some of the main meetings, but if you cannot then remember to write or phone me for any help you may need.

Last month I mentioned a games writer pack which Warren Welham was to review. Well Warren has completed his review and advised that while the concept of the pack is good, it is really only suitable for very new members, and would probably not go all that well through the shop if we were to import copies for sale. The copy we have has been placed in the library, so if you would like to borrow it and see for yourself if it is of any benefit to you, see Warren.

News from Chris Bobbitt of Asgard Software to the effect that he is producing a new publication for the TI99/4A. This publication ASGARD NEWS, will be published quarterly, and is available from Asgard Software, PO Box 10306, Rockville MD 20850 USA, at a cost for overseas subscribers of US$12 per annum. Alternatively, for a bulk subscription, where all copies are sent to the same address for collection, the overseas subscription rate is US$9. Any members interested in the bulk subscription rate should see me at the next meeting, or if you cannot get to the meeting contact me in writing. Subscriptions in advance would be required before entering into the bulk order. There were some very well written articles in Issue 1 with a table of contents being, New Products, New Versions, Reduced Prices, Bug Report, Ask Asgard, Program Focus, An Introspective Look at the 9640, Miscellaneous Stuff, The News and Next Issue. The next issue, by the way, promises the complete scoop on the status of Picasso and who will be selling what how when and where, as well as more inside news from the TI99/4A and 9640 software and hardware industry. Good value in my opinion, so get hold of the copy placed in the library and see for yourself.

Despite the cold August winds there was a sizeable turnout at the August meeting and it was surprising the amount of used software and hardware that was on display for trade or sale. Hope you all managed to sell or buy whatever you wanted. I noticed one gentleman going home with an Acoustic Hood for his printer and a stand alone disk drive, as well as other bits and pieces. Hope the wife was not too upset, Russell.

And speaking of meetings here is the list of topics for the remainder of this year.

SEPTEMBER 3 - SOFTWARE DEMONSTRATIONS AND PURCHASE

At this meeting some of the latest software as advertised in MICROpendium will be demonstrated and hopefully we will have imported copies for sale direct to members at reasonable cost. Feedback from members on what type of software they would like to purchase or see demonstrated would be very welcome.

OCTOBER 1 - JOINT MEETING WITH HUNTER VALLEY 99'ers

Contact has been made with our friends in Newcastle and a verbal response suggests that this meeting will go aheadon October 1. It is envisaged that there will be a computer meeting in the morning with a social afternoon. There will be more details on this as we get closer to the actual meeting date.

OCTOBER 8 - NORMAL MEETING WITH A SOFTWARE COPYTHON.

Be at Woodstock at 2pm to be able to get some of the latest software. Other minor events, yet to be finalised, are also planned for this afternoon.

NOVEMBER 5 - FULL DAY TUTORIAL WORKSHOP.

Here the themes and other activities are yet to be finalised, however, like all full day events in the past, this is guaranteed to be a fun day. As usual there will be a luncheon BBQ at very reasonable cost.

DECEMBER 3 - CHRISTMAS PARTY.

Given a fine day this will be one of the major events of the year, with plenty of food and drink available and a great chance to chat with fellow members in a social relaxed environment. There will be plenty of software released for this meeting so you will have plenty to keep you occupied over the Christmas/New Year holiday break.

[illegible] is it [illegible]

o

Letters to the Editor

Dear Sir,

I wish to comment on matters raised by Lou Amadio in his open letter to the Directors published in the August edition of the TND.

This letter seems to be based on emotions rather than logic. To say the market for 1Mb RAMdisks no longer exists is like saying Ford no longer has a market for Fairmont Ghias because Bond is selling Hyundais. The "market" in each case is entirely different and exists in its own right. Whatever genuine; and I emphasise the word genuine; market existed for 1Mb RAMdisks has not yet been satisfied. I can only assume there is more to Peter Schubert's decision not to proceed that we are not privy to. If Lou (and others) require such a card, obviously they should tell Peter Schubert. I suspect the 1Mb RAMdisk is not yet dead provided there are sufficient buyers willing to pay the price!

In any event, much of what was said in the letter is a "red herring". When you strip ANY Ramdisk to the essentials, it is nothing more than a mass storage device. The real alternative is obviously the new hard disk controller. True, the combined cost of both controller and hard disk will be greater but then the storage capacity is at least 20 times greater with an option to increase this even further at a later date at a substantially reduced cost. Perhaps Peter and others have indeed done their sums and decided the $/Kbyte of such a large RAMdisk is now not justified.

Chris Buttner, Carlingford. o

TISHUG Shop with Cyril

Due to the pressures of work, Bob Bunbury, our shop manager, is relinquishing his position with the club shop for the time being. As from the September meeting Steven Carr has volunteered to manage the shop for the remainder of the year.

Steven can be contacted by phone (02)608 3564 between the hours of 4:30pm and 9pm only on week days, or on the BBS as "SHOP", and by post addressed to the shop care of the secretary.

Any member with unfilled orders from the shop should contact Steven and give him full details of requested items outstanding. This will enable him to supply these items and get things operating smoothly.

Now something for the new members as well as the older members. The shop has been left with a large stock (about 500 copies) of Micropendium magazines, these issues range from June 1986 to January 1988, in all 20 issues. As there are not equal numbers of each issue, you will have to order early if you want a full set, or if you want specific issues.

The sacrifice price for these issues will be:-

single issues	$1.00 each
12 or more issues	$0.50 each

NOTE:- This year's issues of Micropendium will remain at $ 3.00 per issue as we have reduced the number per issue that we are importing.

Also available are RAM card PCBs at $35.00 each

6264LP-12 RAM chips at	$5.50 each
Console writer modules	$25.00
Viatel 1 modules	$25.00
Viatel 2 modules	$25.00
Diagnostic modules	$25.00

Plus all the regular items such as :-

Blank disks	$12.00 per box
Club software disks	$ 5.00
Club software tapes	$ 3.00
E/A manuals	$25.00
Tech manuals	$15.00
TI-modules education	$8.00
Spare parts	cheap
Peter Schubert's cards	

See you at the September meeting. o

Techo Time - Printer head repairs

by Lou Amadio

Recently the head on my printer failed. More specifically, the top pin did not fire, making most text unreadable. I enquired with my local printer retailer who informed me that a new head would cost $150, and a quote from a local computer equipment repairer was $200 plus fitting!

A few days later, when I recovered from the shock, I decided to pull out the old head and have a look at why it failed. After all, I had nothing to lose. I noticed that the head was assembled with screws so out came the jewellers screwdrivers and I slowly pulled the head to pieces.

As it turned out, the head was quite complex, with a large number of parts all intricately assembled, I thought, by a very patient Japanese worker. Nevertheless, it had to come apart, if only to satisfy my curiosity. Unfortunately I managed to disassemble it in the wrong order, and a large number of parts fell out onto the table! I then spent the next 2 hours working out how all those little parts went back together again.

During this procedure, I was able to determine why one of the pins had stopped working. Naturally, one cannot go out and buy a 10c part to fix a $200 printhead, so the next step was to find another unit (deadhead?) from a similar printer. Well, a few days later, I did manage to find one, which I was able to disassemble and replace the faulty part in my print head. Granted, the whole process took two full evenings, but when you do not have $200 to spare, there are not too many other choices.

In conclusion, therefore, any owners of Amust, Sakata or DSE printers (BMC, CPA80) who have a head problem, remember that the fault may be repairable. You may contact me on (042)28 4906 if you need any advice on the above.

Note: The head in question may be recognised by a silver sticker on a black finned body. The sticker has the following message printed in red:

<C A U T I O N>
"Hot Surface"
Avoid Contact

o

Magazine review

by Warren Welham

This I hope will become a fairly regular feature, with it appearing every couple of months. The idea of this article, is to review overseas magazines stored in the publications library.

The first one I thought I would do is a new one I have just received from Asgard Publishing, called Asgard News. Asgard News is published quarterly and is fairly thick at 15 pages long. There are very few advertisments and very informative articles. Some regular columns include:

1)New Products: As you may realise Asgard is a company which produces software and therefore this column is dominated by their products.
2)New Versions: This column concentrates on updates of programs and what has been updated in the program. This column is good as you can see if they have made updates that interest you, so you can either buy this version or wait for the next version.
3)Bug Report: This column reports on bugs found in Asgard software and how to fix them.
4)Ask Asgard: This is for readers to write in and ask a question about Asgard software.
5)Program Focus: This column picks a program, describes it and shows advantages and disadvantages of the program.
6)Feature Article: This is the main part of the magazine covering just over 3 pages. This issue's Feature article is on the Myarc Geneve 9640.
7)Miscellaneous Stuff: This section is devoted to things that really do not quite fit elsewhere or do not have anything to do with Asgard but are interesting.
8)News: This is for items not appearing in other publications like Micropendium, that Asgard thinks should not have missed out appearing.
9)Current Versions: This column lists all Asgard software and which version it is up to and when it was last updated.

All in all I believe this to be a good publication and an essential publication to get if you ever have or will have any contact with Asgard software.

It is possible for the club to get a bulk subscription to Asgard News where they are all sent to the one address. This will cost US$9.00 instead of US$12.00 for individual subscription. This US$9.00 covers a year's subscription of 4 issues. The first issue of Asgard News will be available to look at, at the publications library and if you wish to subscribe through the club in a bulk system, please see me at the library or fill in the subscription form below and send to me and I will record your name. Remember we need people to fill the numbers so do not send any money if you are sending the form below, we will contact you later.

TIsHUG Bulk Subscription Form For Asgard News.

Please include my name on the TIsHUG subscription list for Asgard news. I realise that the yearly subscription rate in bulk is US$9.00

My Name is:_____

My Address is:_____

My Town is:_____ My Postcode is:_____

My Phone Number is:_____

My BBS User Name is:_____

I Would like ___ subscription(s) to Asgard news to be reserved for me.

---

o

Genealogy anyone?

by Margaret Moore, USA

This article was given to me by Margaret Moore, the wife of our Librarian. She is aware of my interest in Genealogy and is also aware of the fact that I am of Italian heritage. So I pass this on to any and all interested parties.

Thomas E. Militello, M.D., 6932 Ctrest Road, Rancho Palos Verdes, CA 90274, USA, is now offering a computerized database called "Pursuing Our Italian Names Together" (POINT). In its first two months, POINT has recruited sixty-five members who have entered over 700 Italian surnames, representing forty Italian provinces of origin. Anyone working on Italian Names can write to POINT giving his or her name and address, the SURNAMES being researched and the Cities or Towns and provinces where the Surname originated, if known. A #12 SASE with 90c postage will obtain a print out of all those who have written, along with a list of the Surnames being researched and a small newsletter called POINTers.

This is the article as given to me. I wrote to Dr. Militello, included the SASE and $1.00 (stamps went up 25c) and received .75 stamps back and new information. The response has been so overwhelming that Dr. Militello can no longer take on the expense of photocoping, mailing etc. As of today he has 315 members who have submitted 3300 surnames. Hence a quartly Newsletter will be printed and circulated every Summer, Fall, Winter and Spring including all updates. For this and other priviledges (you may submit as many surnames as you wish) he must now charge $20.00 p.a. Or you can pay $1 per name searched. If a match is found he will send you the name and the address of its contributor.

If you wish to join contact the good Dr. direct. (Is ROOT beer a drink for genealogists ?)

o

How to Read and Write to Individual Disk Sectors

by Todd Kaplan, USA

I give permission to reprint this article in any form printed, electronic or otherwise.

This article is, as the title suggests, how to access individual sectors of a floppy diskette with a TI99/4A computer. The actual programs that perform single sector I/O are contained in the Disk Drive Controller's DSR (Device Service Routine) ROM. I will attempt to explain how to access that program. If you are not very familiar with the DSR routines on the TI99/4A and just want to easily access sectors, you can skip this part and just use the subroutines provided at the end of this article .

If you are at all knowledgable of how to use the DSRLNK utility, you know that you must set up a PAB (Peripheral Access Block) in VDP (Video Display Processor) RAM. That PAB must contain the device name and you must write the address of the device name length to a pointer in CPU RAM (Main Memory). The address of that pointer is at >8356. You should also know that the DSRLNK utility needs some DATA for it to function properly. In a typical DSR access, you usually see is the following:

```
* DSR ACCESS
     LI   R6,PAB+9 POINTER TO NAME LENGTH IN VDP RAM
     MOV  R6,@>8356 STORE POINTER TO NAME LENGTH
     BLWP @DSRLNK EXECUTE DSR
     DATA 8
```

A funny thing about the above example (or typical if you are used to TI) is that we need the 'DATA 8' part, but nowhere in the Editor Assembler (E/A) manual does it say why. The '8' is a pointer for where the search of DSR ROM begins. There is a pointer at location >4008 in all of the DSR ROMs that points to the beginning of where the search for the device name is to begin. At address >400A in the ROMs, there is a pointer for the beginning of Subprogram Search List. The point that I am driving at, is that the single sector routine in the DSR ROM is a Subprogram. That Subprogram has a name just like DSK1 is the name for Disk 1. The name of that subprogram is best represented as the hexadecimal digit >10. Yes that is a single character which would be the equivalent of CHR$(16) in BASIC. Now, all we have to do to access this program is to put the name into the PAB and execute a DSRLNK with a DATA of >10 for the start of the search, and we can access single sectors. Right? Wrong. The Single (un-married) Sector Program needs parameters. These parameters are the sector number, disk drive number, VDP buffer address and whether to read or write. Below is a Table that shows the addresses of the parameters and describes each one's function:

Word Addresses	Function
>834E	VDP I/O Buffer Address. (The buffer should be at least 256 bytes long.
>8350	The number of which sector is to be accessed.
>8356	Pointer to the Name Length in the PAB in VDP RAM.
Byte Addresses	**Function**
>834C	The number of the disk drive to access.
>834D	R/W Flag. >01 = Read; >00 = Write.

Here is what the PAB should look like:

```
* PAB DATA
PDATA DATA >0110 Name-length of 1, Name of >10
```

Here is what the Link should look like:

```
* DSR ACCESS
     LI   R6,PAB POINTER TO NAME LENGTH IN VDP RAM
     MOV  R6,@>8356 STORE POINTER TO NAME LENGTH
     BLWP @DSRLNK EXECUTE DSR
     DATA >A (or DATA 10)
```

The following is a program that reads one sector of disk drive one (DSK1) and writes the data to a different sector on the same disk.

** WARNING ** This is only an example and you run the risk of damaging the data on your diskette if you run this program.

```
*************************************
* START OF EXAMPLE USING SUBROUTINES *
*************************************
MYREG  BSS  32 WORKSPACE
*
* Warning! Do not Run this example, you may damage the
                Data on DSK1.
START
       LWPI MYREG Load WORKSPACE
       LI   R0,25
       MOV  R0,@SECTOR - SET UP TO READ SECTOR 25 ON
                DSK1
       LI   R3,BUFFER GET ADDRESS OF CPU BUFFER
       BL   @RESECT READ THE SECTOR INTO
* Now Sector 25 is in the CPU Buffer
       INC  @SECTOR GO TO NEXT SECTOR
       BL   @WRSECT WRITE the CPU Buffer to Sector 26
*
LOOP   LIMI 0
       LIMI 2
       JMP  LOOP WAIT FOR QUIT KEY.
*
*
*
************************************
* SINGLE SECTOR DISK I/O SUBROUTINES *
************************************
* Written by Todd Kaplan
* 2/11/85
*
*
       REF  DSRLNK,VMBW,VMBR,VSBW
       REF  KSCAN
BUFFER BSS  256 SECTOR BUFFER
KKEY   EQU  >8375
STATUS EQU  >837C GPL STATUS BYTE
PABADD EQU  >8356 VDP ADDRESS FOR DUMMY PAB
BUFADD EQU  >834E INPUT BUFFER ADDRESS
SECTOR EQU  >8350 # OF THE SECTOR TO ACCESS
DRVFLG EQU  >834C MSB - DRIVE #; LSB - FLAG: >01=READ
                >00=WRITE
K0     DATA 0
K2     DATA 2
K10    DATA 10
DUMPAB DATA >0470 VDP PAB ADDRESS
KD1RD  DATA >0101 DRIVE 1 and READ FLAG
KD1WR  DATA >0100 DRIVE 1 and WRITE FLAG
DUMDAT DATA >0110 CONSTANT FOR SECAC (DUMMY DATA)
DUMBUF DATA >1000 VDP BUFFER ADDRESS
***************
* READ SECTOR SUB-ROUTINE
* FROM DRIVE #1
* BUFFER IS THE CPU BUFFER USED TO READ TO
* INPUT:
*
* @SECTOR = SECTOR NUMBER OF SECTOR TO READ FROM
* R3 = AI··   S OF INPUT BUFFER
* BL @RES
*
* Output: The Buffer pointed to by R3 is filled with
                the data from the sector
* pointed to by @SECTOR.
RESECT
* SET UP DUMMY PAB
       LI   R0,DUMPAB ADDRESS OF DUMMY PAB
       LI   R1,DUMDAT DATA FOR DUMMY PAB
       LI   R2,2 2 BYTES TO WRITE
       BLWP @VMBW
       MOV  @KD1RD,@DRVFLG DRIVE #1, READ
       MOV  @DUMBUF,@BUFADD BUFFER IN VDP AT >1000
       MOV  @DUMPAB,@PABADD ADDRESS OF PAB
       BLWP @DSRLNK
       DATA 10
       MOV  @BUFADD,R0 ADDRESS OF VPD BUFFFER
       MOV  R3,R1 CPU BUFFER ADDRESS
       LI   R2,256 256 BYTES LONG.
```

```
      BLWP @VMBR
      RT
* END OF RSECT - DATA IS IN BUFFER
********** WRITE SECTOR
* WRSECT * SUBROUTINE FOR DRIVE 1
**********
* INPUT:
* R3 = CPU BUFFER
* @SECTOR = SECTOR TO WRITE TO
*
* BL @WRSECT
*
* OUTPUT: The in a buffer in CPU RAM,pointed to by R3
                  is written to the Sector
* pointed to by @SECTOR
*
*
WRSECT
*
* SET UP DUMMY PAB
      LI   R0,DUMPAB ADDRESS OF DUMMY PAB
      LI   R1,DUMDAT DATA FOR DUMMY PAB
      LI   R2,2 2 BYTES TO WRITE
      BLWP @VMBW
* WRITE CPU BUFFER TO PAB
      MOV  @DUMBUF,R0 ADDRESS OF VPD BUFFFER
      MOV  R3,R1 CPU BUFFER ADDRESS
      LI   R2,256 256 BYTES LONG.
      BLWP @VMBW WRITE TO BUFFER
      MOV  @KD1WR,@DRVFLG DRIVE #1, WRITE
      MOV  @DUMBUF,@BUFADD BUFFER IN VDP
      MOV  @DUMPAB,@PABADD ADDRESS OF PAB
      BLWP @DSRLNK
      DATA 10
      RT
* END OF WRSECT - DATA IS IN BUFFER
      END
```

Note: If you have any questions or comments, please leave them on one of the TI-West BBS and I will get back to you. I am also available via Compuserve's EMAIL at 74036,3215.

o

For Sale For Sale For

Dot Matrix Printer $200
Speech Synthesizer $60
Extended BASIC $40
Terminal Emulator II $20
Household Budget Managemnet $10
Personal Record Keeper $10
Contact Lou Amadio, Ph (042)28 4906 after hours

Wanted

Games modules - contact Lou or George at the next TISHUG meeting.

Publications Library Report

with Warren Welham

Well it has been a record month in the way of the library receiving books. My thanks to the people donating them to us. Somewhere in this issue you will find a review on a new overseas publication called Asgard News. I believe this is a good magazine and well worth getting. As you read the review you may notice that I have said I would like a review to become a regular feature. To do this I need volunteers to take home a magazine and read through it and review it, the way I have done this month. If anyone would like to do this, please see me at the library next month or write to me.

Overdue books: please search at home for any library books and bring them to the next meeting or some how let me know you know you have got them, as I will have to start sending letters to people who have not returned their books.

New TNDs arrived this month

JULY'88

Appeal:Anyone who has any TNDs(SND) they do not read anymore, please think about donating them.

New Books arrived this month

Code	Title	Author
00225	TI-99/4A Software catalog from Texaments 1988	Texaments
00226	Computer and Video Games May'83	C & V Games
00227	A Computer of your own	Goodhead Pub
00228	Exploring Forth	Owen Bishop
00229	The Complete Programmer-a guide to programming in Basic	Mike James
00230	Wire Accessary Interface for TI	Ross Mudie
00231	Learning TI99/4A Assembly Language Programming	Ika McComic
00232	Stimulating Simulations for the TI99/4A	C.W.Engel
00233	Stimulating Simulations for the TI99/4A	C.W.Engel
00234	Sams Basic Tricks for the TI	Allen Wyatt
00235	Video Games (with cartridge)	T.Instrument
00236	Game Writers Pack 1 (with tape)	T.Instrument
00237	99'er HCM Vol.2 No.4 Feb.83	HCM
00238	99'er HCM Vol.2 No.5 Mar.83	HCM
00239	99'er HCM Vol.2 No.6 Apr.83	HCM
00240	99'er HCM Vol.2 No.7 May.83	HCM
00241	99'er HCM Vol.2 No.8 Jun.83	HCM
00242	99'er HCM Vol.2 No.9 Jul.83	HCM
00243	99'er HCM Vol.2 No.10 Aug.83	HCM
00244	99'er HCM Vol.2 No.11 Sep.83	HCM
00245	99'er HCM Vol.2 No.12 Oct.83	HCM
00246	99'er HCM Vol.2 No.13 Nov.83	HCM

New Arrivals of Overseas Publications

Group	Publications Name	Date
Asgard Publishing	Asgard News	May 88
Channel 99 Hamilton	TI Focus	Jun 88
Hunter Valley 99ers User Group	Hunter Valley News	May 88
Hunter Valley 99ers User Group	Hunter Valley News	Jun 88
Micropendium	-	May 88
Micropendium	-	Jun 88
Pittsburgh Users Group	The Pug Peripheral	May 88
Pittsburgh Users Group	the Pug Peripheral	Jun 88
Orange County User Group	ROM	May 88
Sacramento 99ers User Group	Network	May 88
Sacramento 99ers User Group	Network	Jun 88
Sacramento 99ers User Group	Network	Jul 88
Central Ohio Ninety Niners inc.	Spirit of 99	May 88
Central Ohio Ninety Niners inc.	Spirit of 99	Jun 88
Tacoma 99ers Users Group	The Tacoma Informer	Apr 88
Tacoma 99ers Users Group	The Tacoma Informer	Jun 88
Club Information Montreal	CIM 99	May 88
Club Information Montreal	CIM 99	Jun 88
Ottawa TI99/4A Users Group	Newsletter	May 88
Ottawa TI99/4A Users Group	Newsletter	Jun 88
Adelaide Computer Club	ATICC	Jun 88
Northern New Jersey	-	Apr 88
Northern New Jersey	-	May 88

o

TI-BASE - a Relational Database for TI99/4A

by Chris Buttner

Overview

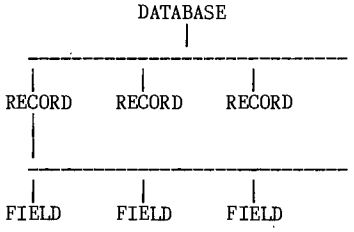
Those members who attended the July main club meeting were fortunate enough to see Shane Ferrett's demonstration of TI-BASE. If you missed the demonstration, I hope this review will help you evaluate the program's suitability for your needs.

The version under review is 1.01. The program is supplied as a two disk (360K per disk) set and comes complete with a manual and paper keyboard template from Inscebot. The Disks are called simply TI-BASE and TUTOR. The manual is printed in blue ink on grey paper which is not the easiest to read, particularly when the quality of printing leaves something to be desired. Despite this, all the information you will need to get started on either the program or Tutor disk is included. There is a command file TUTOR which when called, runs a tutorial which lasts about 20 minutes. This arrangement makes a well presented, comprehensive introduction for newcomers.

For the technically minded, the database limitations are:

255 characters/field;
17 fields/record;
16129 records/database.

The following diagram shows graphically the relationship between the database, records and fields.



The main program is capable of controlling 5 active databases so you have access to a total of 85 fields (although 4 would be repetitive 'index' or 'key' fields). Unless you need more than 81 fields at any one time, you will not outgrow the database.

Each active database is allocated to a 'slot' (numbered 1 to 5). Access to the particular database file is achieved by accessing the slot.

The program is organised on the overlay principle with the various functions being pulled in as needed. Unfortunately, most files are still not in program form so loading is a lengthy process. The program can be loaded from the following environments:

Editor/Assembler
Mini-memory
Extended BASIC.

Once the program has been loaded, you are prompted to enter the date in MM/DD/YY format. In large measure, what happens from this point on is entirely up to you. As supplied, the setup file (which is a command file executed by the program after accepting the date) displays the system status.

You have the ability to:

> interchange and manipulate data in fields of related records;
> perform mathematical operations on fields;
> use logical and relational operators in querying the database;
> find related data;
> manipulate data strings by joining (concatenating), trimming etc.;
> control the display and print format;
> create your own applications with the command language.

Operation

You know you have control of the program/database when you are at the DOT prompt which is the period. The computer is waiting to accept your command. The first step in setting up your own database is to define its structure with the create command.

As you name each field, you declare its type; Character, Date or Numeric. If you choose numeric, you must also set the number of decimal places. It is possible to modify the database, but in the current release, this should not be done once data has been entered.

The next logical step is entering your data into the newly created database with the append command. The 'entry fields' act as windows: if the field size is large, the entry will scroll from right to left in the window until you reach the maximum declared size. If you continue to enter after this point, the last character will be overwritten with each keypress. The figure on the far right of the screen and in line with your field will count the number of character positions used.

Did you make a mistake entering data? Do not worry, individual records can be edited at any time with the edit command.

Two steps are required to remove a record. The first involves deleting the record, with the delete command. This 'marks' the record for deletion. Final and irreversible removal does not take place until you reorganise the database with the pack command. This removes all marked records and rebuilds the index file if you have one. If you deleted the wrong record at step 1, the process of marking is reversed with the recall command.

More often than not, we want to see our precious data in some particular order. This is accomplished with the sort command. In each database file, you have only one field on which to sort. Ideally, you ask for the database to be sorted on a particular field before entering data. In this way the sorting is done dynamically as new records are added.

Printing and Displaying information are essentially the same (with output directed to a different device). This is one of the major departures from other databases with which you are probably familiar. You must say which fields you want displayed and in what order. This is one of the most flexible parts of the program but of course you have to think what you want. You can have field 3 from the database file in slot 5 as your first "column" followed by field 5 from the database file in slot 1 as your second "column" and so on.

An added bonus is the ability to concatenate (join end of one to start of another) fields before they are output by using local variables. Depending on your fancy, you might decide to combine say three fields with trailing blanks stripped and a word or two of your own inserted between the second and third fields. Sounds fancy does it not? It is!.

In the example I have just given for concatenating three fields, it would be a tedious process to enter the program commands from the keyboard each time you wanted to see or print the information. Fortunately, this is not necessary. Everything can be combined in a command file which is stored on disk. When you want to see the information, execute the file with the DO command. This can be programmed to get the database files, put them in the various slots then proceed to print the selected information; very neat.

Once you have mastered these basic steps you will be drawn to creating your own command files with the text editor. This is invoked with the Modify Command command. The editor is easy to use and the commands which are reminiscent of Pascal provide such constructs as DO WHILE / ENDWHILE; CASE / ENDCASE; IF / ELSE / ENDIF. For good measure you also have the ability to catalog a drive, copy and delete files and so on. These disk file commands are most often used from the dot prompt for carrying out housekeeping.

Summary

This program is a complete departure to what most people have become accustomed. The lack of an initial menu screen is the most obvious example. The flexibility of the program however, allows you to design your own menus and you can have a different menu

for each of your applications. At present, there is no facility for graphics but the text is easilly handled; it can be displayed anywhere on the screen and input can be equally as flexible. You can have one program call another which gives you the option to have multiple menus within your application. (If you do this, make sure your minor menu applications always return control to your initial menu.)

There is enormous potential for reducing the number of keystrokes when entering data. For example you may want to catalog a record collection and include a field for music types such as Classical, Rock and Roll, Opera, Country and Western etc. These categories need be entered only once in a second file (if you had the categories above, there would be just four entries with two fields each - the code such as the number 1 to 4 and the description). Your main file then makes reference to the code only. To see all the information, make both files active and link the data with the common field (code) when displaying or printing. The more entries you have, the greater the savings. You are now seeing the advantages of being able to use a relational database.

I hope this has been sufficient to whet your appetite. Now comes the difficult part. I will try to be objective in listing my likes and dislikes. Please bear in mind this is a very personal review and I do not claim to be infallible.

The things I like are:

(1) The flexibility in getting information to the screen or printer;
(2) The ability to use relational files. I have tried not to dwell too much on this aspect because I feel it is something you appreciate only with use. The more familiar you become with the program the more likely you are to use smaller related files. These can be used as "tables" and thus save you one heck of a lot of keystrokes.
(3) Speed. Yes it is fast, both in entry and retrieval of records. Make no mistake about it; once the beast is loaded, it moves!
(4) Support. Originally version 1.00 was up for review but the distributors sent version 1.01 (June 1988) to replace the main disk. I suspect the program is being refined as income is generated from sales but with upgrade support such as this, there is no reason to hold off.
(5) The ability to custom tailor applications with command files.

The things I dislike are:

(1) Slow loading if you are not using a RAMdisk. This may be overcome if the files are converted to a program form.
(2) The restriction of one sort field per database file. This restricts the ability to have a secondary sort field in your main file or a related file. (Two sort fields would make the program so much more elegant.)

In addition there are some things which do not fit neatly into the likes and dislikes and may be related to specific hardware. While I have experienced the problem, you may not. This maybe problem is that when working with three files, I can access only two for output when using them on a floppy disk. The moment I put everything on a RAMdisk, everything works perfectly. Lastly, if you try the TEST2 program on the tutorial disk it may not work properly. The fault is in the command file and the answer is in the printout of the file in the manual. See if you can find it. This is a sure way to quickly understand some of the program commands.

For my money the positive features far outshine the others, which in time (and with a little encouragement directed to the developers) may disappear. o

I Can Write to and Read From Files

by Werner Kanitz

Big deal! Have you ever thought, time to be serious? It is time to wade through the games tapes, disks and cartridges, past the cobwebs, shovel the dust off the trap door and lower yourself into the cavernous depths of the archives room. The repository of all that is serious in TI99/4A computing.

Serious computing on the TI99/4A? Oh come now!

Does this response sound familiar?

Do you know how to write to and read from a file?

Sure mate TI-Writer !

Any other way?

One of the database packages!

Do you realise that you can write to and read from a file using BASIC?

Yeah really? Bullshit! wots BaSiC?

Sure thing person (an attempt at being antisexist), there are plenty of f'rinstance programmes around, if you do not know how to start by yourself.

There certainly do seem to be an abundance of example programmes around each in its own way describing how to make (expletive deleted) little (expletive deleted) phone books. Are you getting the impression of a failure to be impressed? Perhaps I am a litle unfair. Suffice it to say starting from scratch is is not a bad move.

By the by, this was all brought on through my attempts to create my own database, specifically to help me keep track of my income and expenditure. I did not want to use someone elses, I wanted to make my own, so there.

My little money manager was to allow me to enter the data, store it on a file then let me browse it on screen and produce hard copy reports. The data for each record consisted of a date, four income codes and associated values and eight expenditure codes and values. This gives a total record length of 220 characters. A fairly simple exercise it would seem.

The TI99/4A file handling facility seems to kill with kindness, all but cooks your tea. This is not so, as you must be carefule of the defaults.

```
OPEN #[1-255]:"DSK[1-3].DATA"
PRINT #[1-255]:{print list}
```

These two lines of text seemed to be all you need and you are off and running. True, if you are building a "phone book"

TI99/4A file manager sets the defaults. First mistake, defaults are not tailored to the data. The defaults are:

```
organisation=SEQUENTIAL
   data type=DISPLAY
operating mode=UPDATE
   record type=FIXED 80
```

The 80 character default length is not quite long enough to hold 220 characters and wrap around did not seem to be the right solution. Obviously we have to do some reading. Change the record length easy, well documented.

```
OPEN #1:"DSK1.DATA",FIXED 220
```

The change did not work! The data was written to the file but there were problems in reading.

Consternation!

Try reading reading the Extended BASIC manual. This gives no help as the syntax appears OK, so why cannot I read the file. Take one step further back. Read the TI99/4A User's Reference Guide. This is a much better manual, but still no help. For the first time I realised the Extended BASIC text is only like an addendum to the User's guide. I still cannot read my file.

Diarrhoea!

I read Computes guide to Extended BASIC Home Applications on the TI99/4A. (Expletives deleted) phone books again no help, I still cannot read my file.

Desperation sets in!

I am going to do this alone. There is a manual which comes with the Disk Memory System I started reading. This stuff all looks very familiar. Bingo!!

"(The length of a DISPLAY-type record is limited to approximately 150 bytes)"!!!!

The solution was simple. The file was made an INTERNAL-type file (I do not care if I cannot browse the bastard with TI-Writer {it is too long anyway}). This means I will have to write my own browsing and error correction facility.

After all, I can write to and read from files.

Thought for the month:

Never put into one reference source that which you can distribute throughout the library.

Is it not the truth though? o

Trials of a Disk Fixer

A personal example of the uses of the "Disk Fixer" Module

by Robert Peverill

Some time ago, when SSSD full height drives for the TI99/4A were still quite expensive, I was able to obtain a DSDD 80 TPI full height drive at a good price. Unfortunatly the "Disk Manager II" module could only do 40 TPI DSSD. Not to worry, at the price 40 track DSSD was better than nothing. Happiness followed, as now I had twice as much storage capacity for storing and developing my programs, but, Murphy's Law was lurking not far away in the shadows, waiting for the most inopportune time to strike and create disaster. Then it happened, "BRRR" "BRRR", "BRRR" "BRRR", the new drive crashed, taking with it a program from the "HOME COMPUTER MAGAZINE" that I had been debugging and other programs I had also been developing at the same time. Why had I not backed them up you ask? Simple, my differing disk formats made it slow to back up, plus it would require twice as many disks for the backing up of the double sided drive, and as it was at a time when disks were costing around $40-$50 per box, I say no more.

As I said this was quite some time ago, so long ago in fact I had forgotten what programs had been lost. In the mean time the drive had been repaired, new compatible DSDD 40 TPI drives purchased and all odd format disks transferred over. But still the old damaged disk lingered on. I obtained a "NAVARONE DISK FIXER MODULE" and after a long period of "fits and starts" I had dumped various sectors of "good" and "bad" disk Directories, Bit Maps, Directory Link Maps, Directory Entries and various sectors of program data.

I read the Navarone booklet but found it a little confusing in parts, then read the series of articles by Bruce Caron and found these cleared up a majority of the hazy areas.

So with listings, articles, pencils and "Oh my aching head, what am I in for?" set about the task of finding out what happened to the bad disk and how to fix it.

The symptoms of the disk failure were as follows:-

When using Disk Manager II to catalog the disk, it would start cataloguing normally then produce garbage for a number of lines, missing some programs before completing the catalog. When trying to back up the disk using DM II only some of the programs catalogued could be copied, many could not.

The Fix

Start by reading sector 1 (Directory Link Map).
R 0001,D (R 1,D)

```
ONE IND. *** DISK FIXER V2.0 ** SECTOR DUMP SECTOR ADDRESS 0001
  = 01 23 45 67 89 AB CD EF  INTERPRETED
0000 = 0013 0014 0010 000B 0012 000A 000F 0011  ****************
0010 = 0009 000E 0008 0007 0006 0003 0004 0005  ****************
0020 = 0002 000C 000D 0000 0000 0000 0000 0000  ****************
0030 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0040 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
   . =              0000 0000 0000 0000 0000  ****************
   . =              0000 0000 0000 0000 0000  ****************
0070 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0080 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0090 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00A0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00B0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00C0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00D0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00E0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00F0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
```

Count the number of 1 word (2 bytes) Directory Entry sector addresses, add 2 (1 for Link Map and 1 to be sure) and convert to hexadecimal. This gives the number of sectors to print. Now print out all used directory sectors for evaluation.

P 0002,D,N (P 2,D,N)

The last sector printed should contain ">E5"s (intepreted as "e") in every byte, if not, print some more sectors. Do not exceed sector >21 (33) as this is the last standard Directory Entry sector. If your Link Map contains more than 30 Directory Entry addresses then you will have to read the Link Map and print all the additional sectors indicated.

Now the good part begins. With print out and pencil in hand do the following. Using the Directory Link Map, create a listing of all file names in the order in which they appear. This is done by taking the first sector address, going to that sector and writing down the interpreted file name found at that sector. Return to the Link Map for the next sequential address to check, and so on until all file names have been written down. Where a scrambled entry occurs just put a line or mark in the listing to show a fault occured. Check that all files are in some sort of alphabetical order.

Having done that the next thing to do is get some sector details from the entries. Go to sector 2.

```
NAVARONE IND. *** DISK FIXER V2.0 ** SECTOR DUMP SECTOR ADDRESS 0002
ADDR = 01 23 45 67 89 AB CD EF  INTERPRETED
0000 = 542F 4D41 4E20 2020 2020 0000 0100 000E  T/MAN      ******
0010 = 6300 0000 0000 0000 0000 0000 22D0 0000  c**********"P**
0020 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0030 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0040 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0050 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0060 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0070 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0080 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
0090 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00A0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00B0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00C0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00D0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00E0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00F0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
```

Bytes 0-9 = File name.

Byte >C = Indicates the file type. Use information to decode the file details and write it down next to the directory entry sector (in my case 01, a BASIC program).

Bytes >E->F = The total numbers of sectors used to store the file.

Byte >10 = Indicates where the end of file ((EOF) marker (>AA program, >FF variable files) occurs in the last sector used for the file.

Bytes >1C->1E = When decoded using information mentioned previously, gives the first sector address used for the file and the number of consecutive sectors used in the file block (not counting the first sector). Write these values down.

Continue this process for all remaining Directory Entry sectors.

For the file indicated in sector 2, the starting sector should be >22. Now take the details of bytes of >1C->1E (sector 2), decode the start sector and file length, add them together and write it down. This is the last sector address of the file. Do the same for the remaining good sector entries. Now take sector 2 start and finish sector addresses and add 1 to the finish sector address (another way is to add the value at byte >F to the start sector of entry). Compare it with the start sector address of the next file shown in sector 3. They should be the same. Continue doing this for all good entries. Now go back to any bad entries found previously. Look at the entry prior to the bad one, take the finishing sector of that entry, add 1 to it and write the number down as the start sector address next to the bad entry. Look at the entry following the bad one, take the starting sector, subtract 1 and write it down as the finish sector of the bad entry. Subtract the new start address from the new finish address, add 1 and write it down as the new total file length. Print or read the last sector as indicated. Search for the EOF indicator (>AA or >FF) and write the byte address down next to the bad entry. Now we have a probable START, FINISH and LENGTH of the file, also where the file finishes in the last sector. If the damaged entry is not too bad then read it in, otherwise read a good entry of a similar file type. Use the Alter command to change the hexadecimal values at bytes 0-9 to the appropriate file name. Ensure byte >C has the correct bits set for the file type. Alter bytes >E->F to show the total number of sectors used for the file. Alter byte >10 to show the new EOF

marker position. Encode the new start and finish sector addresses and Alter bytes >1C->1E with the new values. Now write this sector back to the disk. This concludes the reconstuction of the File Directory Entry.

Now to fix a directory that would not print out. Simple, read in a Directory Entry similar to the missing entry, alter it to suit the new values found using the steps previously described , then write that sector back onto the unreadable sector address and WULLAH! (Well it worked for me).

Quit Disk Fixer and insert Disk Manager II and catalog the disk. In my case the above procedures worked and I have now recovered all my lost files from the bad disk.

And now for some interesting details. In attempting to recover my disk I looked at both good and bad disk listings. My files were mainly BASIC programs which did not always fill the last sector, and because of this I found some interesting things.

They are as follows.

All BASIC program files end with >00 >AA >3F >FF >11 >03. >00 is the program end of line "symbol", >AA is the EOF marker, >3F >FF >11 >03 is unknown but is always there.

The 8th byte after the EOF marker contains the sector address of the Directory Entry for the file.

The 10th byte after the EOF marker contains the file type details.

Immediately after the file type byte or the 11th byte after the EOF marker, the file name begins and continues for 10 bytes including blanks, however, even though the interpreted ASCII shows the correct spelling of the file name the hexadecimal value of the first byte is always offset by >80. Eg. "A"=>41 is shown as >C1. Why? I do not know.

Bytes 6 and 7 after the filename give the total number of sectors used by the file.

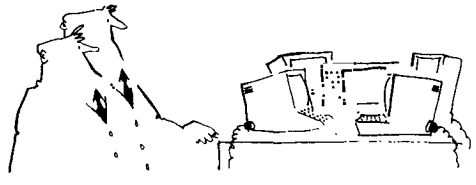
Bytes 39 to 41 after the EOF contain the encoded start and finish sector address.

These details only appear on a less than fully used program file final sector but can help simplify reconstructing a Director Entry.

```
NAVARONE IND.  *** DISK FIXER V2.0 ** SECTOR DUMP    ·. ·· ADDRESS 002F
ADDR = 0 1  2 3  4 5  6 7  8 9  A B  C D  E F  I·  i . ED
-------------------------------------------------------------------------
0000 = 54BE C801 3000 084D 4554 BEC8 0130 0008  T>H*0**MET>H*0**
0010 = 5358 BEC8 0233 3100 0946 4946 BEC8 0233  SX>H*31**FIF>H*3
0020 = 3100 0846 54BE C802 3331 0007 54BE C802  1**FT>H*31**T>H*
0030 = 3331 0007 53BE C802 3331 0007 46BE C802  31**  ..·31**F>H*
0040 = 3331 0009 9DC8 0543 4C45 4152 0015 9A20  31**·I·  LEAR***
0050 = 494E 4954 4941 4C49 5A45 2056 414C 5545  INITIALIZE VALUE
0060 = 5320 00AA 3FFF 1103 0000 0002 0001 D42F  S **?**********T/
0070 = 4041 4E20 2020 2020 0000 0000 000E 0000  MAN      ********
0080 =      0000 0000 0000 0000 22D0 0000 0000  **********"P****
0090 = ·  · 0000 0000 0000 0000 0000 0000 0000  ****************
00A0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00B0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00C0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
00D0 = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
·  · = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
·  · · = 0000 0000 0000 0000 0000 0000 0000 0000  ****************
```

Why these details occur and how accurate they are, I am not sure, but they do. Only the person or persons at TI who created the TI99/4A Disk Operating System (DOS) know for sure. Maybe they could be compelled to feel obliged to disclose the full details of the "DOS", after all, it has been 4 or 5 years since TI pulled the plug on the 99/4A. Besides, with Corcomp and Myarc running their own "DOS", it could not hurt to give it up now. Failing TI giving up the details, there may be others who do know the exact "ins and outs" of the TI99/4A "DOS" and maybe it is time they let the "cat out of the bag".

The above details and procedures have been for non-fractured BASIC/Extended BASIC program files. o



'They've called their first stop-work meeting'.

PRK and PRG

by Daniel Harris

I suppose most users have tried to use Personal Record Keeper to do sorts, and I have more or less succeeded in doing that and all the correspondence is out and generating interest, so now is the time for a little conservative skiting about how the monster was created.

Firstly, and worstly, if you give the PRK a lot of files to crunch it will take a long time, decide two files are not worth bothering about and move them to the end of the list. Once this happens, one is apt to swear at it, maybe use the system as a football or something, or otherwise do one's block. The worst thing one can do is tell the system to re-sort. It will burble away for a day or so! Instead, sort on another key, this will only take half to three-quarters of an hour. Then tell it to do the sort you want. The wise move is to save the other-key sort as well, as it took half an hour or so to generate, and might be useful. The re-sort will take only half an hour. It is a little less strenuous than watching shards of plastic carom off the walls and ceilings, takes a bit longer, but will get the information the way you told it to go. Those bodgy files will now be where they should have been in the first place. Now your printer will be enough for any subsequent work, and you will not be using scissors and glue to patch up a bodgy listing.

The next thing that can happen with the PRK is that you try too many keys, or make the fields too big. Try the number 12,000 as a rough guide. The number of files times the characters (or numeric fields) in each should be a co-factor of an integer to give the product 12,000. This means that if you have 15 fields of 15 characters, you are limited to 12,000 divided by 15, then again divided by 15. If you want to use the Personal Report Generator it may be a good thing to keep to about 60% of that, or 12,000 divided by 4, times 3, or 9,000. So for a file subsequently to be manipulated by PRG, you count all the characters in all the fields (including numeric fields) and divide THAT total into 9,000 instead of 12,000. This will save DATA ERROR when you try to put it into PRG to chop files out, after the sorting. The PRK will chew through most sorts in 35 minutes, UNLESS you try to re-sort something that is already nearly sorted. For some reason that takes longer. For a list of 60 files or less it only takes a minute or two. I have successfully got it to chew up, swallow, and digest 363 files, which is what took it the 35 minutes, doing alpha suburb, then adding the postcodes which are a snap once the thing is in alpha suburb order like the postcode book.

That reminds me of another thing; PRK does not like blank fields when sorting. It is apt to move files with blank fields to the end of the list. Back to the sort; I could then do numeric postcode sort, although the files I was using put it in as characters, otherwise the PRG would add the postcodes and insist on letting me know their totals, likewise telephone numbers have to be characters not integers or you get total telephone numbers and digital overflow messages and whatever befalls of trying to add lists of telephone numbers!

The files which were postcode-sorted could then be chased through a street directory as easily as the alpha-suburbs could be chased through the postcode book. Having found all the bodgy suburbs, access-wise, it was simple enough to do the deletions, then resave the purged files. These could then be category-sorted, printed, and chopped for saving as separate categories using PRG. One has to use a printout to plan this, the PRG makes it all too easy to obliterate files! Separately-saved category files can then be augmented to take all the addressing details. It is then simple to put envelopes into a printer, having got the format as a report in PRG, and press the button for each envelope. One learns to count the fields and take the printer off-line at the end of each addressing. This will not be a major problem I am sure. The category-sorted addresses then get the message for each category plus the address as per the envelope. At 180

continued on page 18

Review of Advanced Diagnostics

by John Craven, USA

a product of...

MILLER'S GRAPHICS
1475 W. Cypress Ave.
San Dimas, CA 91773
Price: $19.95

Advanced Diagnostics is a well done program that is typical of the products that we have come to expect from Miller's Graphics. The program, written by Steve Mildon, executes flawlessly, provides a few important hardware diagnostics, several diskette diagnostics, and contains some nice diskette utilities.

Various parts of the program are executed by issuing commands, and therein lies a powerful feature of the program. The program can execute a single command or multiple commands that are entered into a two line portion of the display screen, or it can execute a large sequence of commands that have been placed in a command file. The commands, which are two letter mnemonics or two complete American words, can be placed in command files by any DIS/VAR 80 text editor, such as TI-Writer. Several good examples of command file usage are included with the program. One example command file prompts for ten diskettes to be inserted into a diskette drive (one at a time), names each diskette, and proceeds to format each.

Well, so much for preliminaries. As you can probably tell, I like the product and I feel that I have received my money's worth. But it is probably impossible for a person who has purchased a product with their own money to give an unbiased review, so I shall try to fight this bias as I proceed.

I will begin with hardware considerations. The program requires a diskette drive, memory expansion, and software (firmware?) that can load and run assembly programs. Both TI and Corcomp disk drive controllers are supported. Miller's Graphics states that the program requires either the Extended BASIC cartridge, the Editor/Assembler cartridge, or the Mini-Memory cartridge. But my copy of the program runs fine from the file utilities menu of the Disk Manager program that is part of the Corcomp disk drive controller software.

Two hardware diagnostic commands that I found useful are Check Memory and Motor Speed. The Check Memory command performs a thorough memory test on CPU Scratch PAD RAM, VDP RAM, Memory expansion RAM, and Mini-Memory RAM (if present). Program progress is monitored on the video display as the test proceeds, so one is not left to wonder about "Just what in the World is going on" at this point. The current area of memory being tested (VDP RAM, Memory Expansion RAM, etc.), the test being performed (memory refresh or bit-shift), and the address range of the memory being tested is reported. If an error is found, the memory location of the error is reported, helping one to isolate a bad memory chip. But more data is needed to turn this into information that can be acted upon. Suppose that Advanced Diagnostics finds a memory location that tests bad, and reports "Bad Value at >xxxx". How does one locate the chip that is bad? A good question. One that I have no answer for. It seems to me that it would have been useful, and relatively easy, for the program to report "Check U108 on the System Board", or something to that effect. Supporting information such as component location diagrams might also be included in the user manual. Alas, at the present time this information is simply not there, and I feel that this is a weak point of the product - one which could be easily fixed.

The Motor Speed command will check the RPM of the selected diskette drive and report it on a nice colorful bar graph. The display has a moving pointer that is updated according to a user selectable sample rate parameter. If a drive motor tests good, this pointer will vary around 300 RPM in a green coloured area of the graph. If it tests bad, the pointer will jump into a red coloured area. As with the Check Memory command, more supporting information is needed if the user intends to adjust the motor speed on his drives. But, given the fact that TI99/4A owners use many different manufacturer's diskette drives, I do not think that it is reasonable to expect Miller's Graphics to provide instructions for performing this adjustment.

Other hardware related commands include Seek Track (to test the stepping function of diskette drives), and Use DSR; a command that acts as a toggle switch to either use the Device Service Routine ROM on the disk controller card or to use Advanced Diagnostics to set the time that it takes for a disk drive's stepper motor to step from track to track. Corcomp owners will find Head Step to be a useful command in that they can use the Advanced Diagnostics program to determine the best head step time for each of their diskette drives. They can then use this information to set hardware DIP switches on their controller card for each diskette drive according to the instructions provided with that product. The usefulness of these two commands (Use DSR and Head Step) for the TI controller user is not as evident. The commands can be used to determine the best head step times but how would one use this information to set up a hardware configuration is not clear to me. Again, a little more documentation on the use of the information that can be gleaned from operating the program would be useful.

In the area of diskette diagnostics and utilities, Advanced Diagnostics really shines. The appendices in the users manual contains the best information that I have seen on the TI99/4A disk format with only one important omission that I could detect. The contents of Sector 0, Byte 16 appears to have been inadvertently omitted from the diskette map. Therefore, if you have the product, you should fill in this information on page 29 of the users manual. Sector 0, Byte 16: >50 diskette is backup protected, >20 diskette is not backup protected.

Diskette related commands that are very useful include Check Disk, Copy Read, Copy Write, Edit Sector, Write Sector, Format Disk, and Find File. The only additional command that I wish were included would be a Find String command. But let me not be picky.

Check Disk reports on diskette usage and notes any bad sectors that are found. More importantly to me, it lists "fractured files" on good diskettes. Fractured files generally arise because files are deleted, updated or otherwise modified from day to day under normal usage. And they tend to become scattered all over the diskette surface. This causes the stepper motor to search all over the place for portions of files to read, or for "holes" upon which to record parts of new files. Thus, the stepper motor has to search a bit to find all of the file that it is trying to access. This causes undue wear on the stepper motor (as well as the nerves, since one has to listen to the stepper motor chatter away as it goes about its business of earning a living). To clean up a diskette that is badly fractured, one copies the diskette, file by file, to a new diskette. The files on the new diskette then reside in contiguously numbered sectors and can be copied back to the original diskette using a sector by sector copy program (for speed).

A sector by sector copy program? No problem. If you do not own one, you can use the Copy Read and Copy Write commands to create one. Copy Read reads in up to 36 contiguous sectors to a copy buffer, whereas Copy Write writes them to the diskette.

Now this is nice. But do you know what would be really nice? It would be really nice if a file by file copy utility were in Advanced Diagnostics. As it is, you must change to the Disk Manager cartridge (TI), or the Disk Manager program (Corcomp) to do a file by file transfer. The Edit Sector command allows one to read in a sector from the diskette, display it in hexadecimal or ASCII (DISPLAY), and edit the sector using a built in full screen editor. The edited sector can then be written to diskette. Of what use is this feature? A great big whole bunch of use. I had an occasion to use this feature as I was writing this review. It seems that my TI-Writer_Multiplan working diskette catalog could not be accessed (probably because I had screwed up in trying out a new communications program). Anyway, I had TI-Writer, Multiplan, a letter to my mom, and a previous version of this review on the diskette, and I simply did not want to lose them (one gets sloppy as they become 'experienced'). The Check Disk command indicated that

sector 18 of my diskette was mapped as being "bad". Being a doubting Thomas, I tried to access sector 18 and found out that I could not. Next, I looked at sector 1, the file descriptor index record, and I noticed that I had a pointer to sector 18; therefore sector 18 should have contained a file descriptor record. I also noticed that the file descriptor record that should have been residing in sector 18 was, alphabetically, the last file on the diskette. At this point, I took a chance and replaced '18' in sector 1 with '00' (using the Edit Sector and Write Sector commands). I then tried to access sector 18 again. It worked! I could access sector 18. Sector 18 contained all standard formatting information, so the thought struck me that, perhaps, this $49 wonder did not recognize that anything had ever been recorded in sector 18. At this point, I looked at the allocation bit map of sector 0, and noticed that sector 18 was not marked as having been used. (Note that if I had known what I was doing, I would have checked this out before modifying sector 18!). Well, It looked like everything was in order so I tried out the Disk Directory command and I SAW A NICE DISK CATALOG! Right before my eyes! TI-Writer worked, Multiplan worked, and my REVIEW was INTACT! Folks, I felt good. Advanced Diagnostics paid for itself right there.

If you think that the exercise mentioned above took years of accumulated experience, etc., forget it! It did take a little bit of mental exercise, but all of the requisite information for solving my problem was contained in a five page section of the Advanced Diagnostics user's manual. This MANUAL is great, and it is only 34 pages long. It is not intimidating at all.

Closing out this segment of the review, the Format Disk command can be used to format diskettes in all diskette formats from single sided, single density; to double sided, double density. The ability to use all of these features depends on your disk controller and your diskette drives, of course.

The Find File command is useful for locating files on diskettes. This command indicates the starting sector and ending sector of the file if the file is located in contiguous sectors, or various starting and ending sectors of fractured files.

The remaining commands fall into the class of "those things that make the program a little more useful and life a little nicer" category. These commands include Beep, Change Colours, Select Drive, Output Device, Output Width, Pause, and Time Delay.

Well, the review would not be complete without a few more words on command files. Command files are executed by the command Command File 'DSKx.xxxx', where DSKx.xxxx is a DIS/VAR 80 file that contains any of the commands that I have discussed (and all commands that I have missed, inadvertently). The best way to gain experience in the use of command files is to run a few of the examples that are distributed with the product. Then view the example command file on your video display (TI-Writer, remember?), or obtain a printout of the command file and study it a bit. If you examine the Command File 'DSK1.DEMO1', you will discover that command files can run other command files if the next command file is the last command in the command file that calls it (WHEW!).

I expect that most purchasers of Advanced Diagnostics will have to change one command file immediately upon receipt of the product. The command file "DSK1.DIAGCONFIG" executes imediately upon entering the program. This file, as distributed, contains the command OD "RS232.BA=4800. etc...". So, if you have a parallel printer, you will probably want to change this command to OD "PIO".

In summary, I enjoy using this product and it is worth $19.95 to me. But I cannot call the review complete without mentioning a pet peeve. This product is copy protected. I presume that the copy protection is of a particularly nasty form. Since we novices are told in all of the TI99/4A Manuals to make sure that we have a backup copy of all important programs and data, and to NEVER USE THE ORIGINAL, I think that it is less than forthright of Mr. Miller to sell this program without noting the fact that he intends to force you to disobey the laws of common sense.

So, if you have $19.95 that you can afford to lose, and you do not intend to become dependent on this product, I recommend this program highly. On the other hand, if your pocketbook can not withstand the (potential) loss, or if you have a tendency toward flying into a rage over the suffrage of moral indignity, I would advise you to keep your money in your pocket. Many of the functions that this program provides are in other programs that are in the public domain, and I am quite sure that all of them will be, eventually. o

Programming Hints

by David Caron, Ottawa

The following hints refer to the Extended BASIC programming environment.

1. Have you ever noticed that when you are in the immediate mode and have created a sprite, that it only lasts for about half a second? Well, now you can keep that sprite there forever while you are in the immediate mode. You can edit, list, change and delete program lines without affecting the lifespan of that sprite. In fact this is also true for character colour changes and character definitions as well as screen colour changes.

Let us say you have gotten tired of looking at your Extended BASIC program in black letters on a white screen and wish to change the foreground colour to some other colour. Here are the commands:

```
FOR CC=1 TO 14 :: CALL COLOR(CC,16,1) ::NEXT CC::CALL
    SCREEN(5)
```

Unfortunately this effect will only last for a very short period of time, but if you add to the statement:

```
ACCEPT AT(24,1):A$
```

to the end of the first line (with the :: naturally) and then press BREAK after typing RUN, you will be returned to immediate mode without Extended BASIC resetting the colours. This strange phenomenon will continue as long as you do not try to execute any commands other than for LIST or RUN. You can still edit or change lines. Unfortunatly this effect does not work while ACCEPT AT is being executed in the running mode.

2. If you have ever wanted to produce an everlasting sound (one longer than 4.15 (5 for us) seconds, it is possible using the ON ERROR routines, This is however rather complicated. Let us say we wish to have the following CALL SOUND, play endlessly:

```
CALL SOUND(-4150,117,1,116,1,-3,1)
```

The following program does the trick.

```
10 ON ERROR 100
20 CALL SOUND(-4150,117,1,116,1,-3,1)
30 CALL SOUND(-10000,110,1)
40 END !(or to the rest of your program)
100 RETURN NEXT
```

When line 30 is executed an error occurs, since -10000 is an invalid duration. Line 100 is executed because of the error. RETURN NEXT then causes execution to start at the statments after the error. By this time the computer has "forgotten" to turn off the sound generator after 4.150 seconds and therefore the sound continues on endlessly. It may be stopped simply by executing another CALL SOUND or pressing FCTN[4] (clear) causing a break in a program which will produce a low tone which overrides the previous sound.

3. This last hint is not something I discovered but something Art Green told me would save memory in that it would reduce the size of a program. The idea is to replace all of the constants used in a BASIC program with variables. Therefore if you had a statement:

```
10 CALL HCHAR(1,1,32,736)
```

memory would be saved by using this technique.

```
1 @1=1
2 @32=32
3 @736=736
10 CALL HCHAR(@1,@1,@32,@736)
```

Although it takes memory to represent the line @1=1 , if many "1"'s were replaced with @1, then you would end up benefitting from extra free memory.

These are all the interesting things I have come across that I have not seen mentioned in previous newsletters. o

Games Information

by Robert Brown and Stephen Judd

Welcome to yet another Games Info. In this month's column, we have a review of Midnight Mason and a new section, which looks at questions coming in by mail.

First of all GAMES wishes to apologise for creating some of this month's article on a MacIntosh, but we strongly advise that they are a health hazard. Now, Midnight Mason is an arcade style game by Software Specialties.

This game throws together a mason and four ghosts that cannot die. The mason must use speed, strategy and be alert to avoid the ghosts or succumb to the inevitable.

You must take on the role of the mason, trying to collect his tools from the haunted building (probably going on strike tomorrow). Being a thick Mason you have left your tools all over the building. To collect your tools you have to just touch them, and all the ghosts have to do is to touch you.

On each game level there are 7 tools, and once collected the ghosts will give up and vapourize and you go onto the next level, only for four new ghosts to start chasing you. What can I say, life is hard! There are 6 different buildings and 4 levels of difficulty, with the ghosts becoming more intelligent each time. Once you have completed the fourth level their intelligence decreases and speed starts to increase, i.e. they are slow and smart and get fast and thick.

The timer begins at 900 and goes down by 10's and when the timer gets to 0 your little mason becomes a dead tradesman. The timer begins to flash near 0, but you rarely see it.

You start with 3 men and get another man for each 5000 points scored. Although right from the start the ghosts move faster, you being a mason, you can chop a hole anywhere in the floor at any time, or repair one anytime. Another thing is ghosts love to climb ladders, even in hot pursuit of your little mason. Another way to escape is to fall through one of your holes. You cannot fall to your death so do not be scared to jump.

An annoying aspect is that when your mason gets to the top of a ladder he occasionally will hesitate causing his demise. The graphics, music and sound effects are all done well to suit this type of game. The game also takes a little time to load between levels. When the mason has collected his tools his completion effect is quite good. The mason also is unable to jump off ladders, as in TI Runner (which it is quite reminiscent of). If this game sounds easy you are wrong, although once you have completed a few screens, it gets incredibly mind-numbingly BORING.

Here is our new section, which we will try to continue each month. It is called the MAIL BAG.

Now to answer some of the questions from Games increasingly huge bag of mail. Names and addresses have been withdrawn for protection.

Dear Robert and Stephen,

Have you got any future plans to do more on the Scott Adams and Infocom adventures, as many of my attempts at completion have ended quite quickly. Thank you.

Games :

Our forthcoming reviews are quite numerous and yes, we have lots of more adventures coming, especially ZORK I, II and III.

Dear Robert and Stephen,

How do you pick up objects for example, water etc, in MOONMINE?

Games :

Well, it is quite easy. First of all, when the object comes onto the bottom screen, you press the space bar. This puts a little man on the screen. Now you can walk over to the object and press fire and when you change colour, you have picked up the object. To return to the ship, you must shoot the monster presently on the screen. I hope this answers your question.

Although all Games' fan-mail is read thoroughly, many letters are not published as only very few will fit into our jam-packed publication.

Well, that is it for this month. I hope you have enjoyed this column as you normally do. Just remember, if you have any problems, please send mail to GAMES on the BBS or write to the following address.

GAMES INFORMATION
141 BEECROFT ROAD,
BEECROFT, 2119 NSW

Until next month, goodbye.

CHOCKY AND TICTACMAN

Submarine Commander

by Robert Brown and Stephen Judd

Turn monitor, peripheral system and computer on in that sequence. Insert module (either Extended BASIC, Editor/Asembler or Mini Memory), select LOAD AND RUN and type in DSK1.GAMES. A menu will be presented. Select Submarine Commander. The game will load and begin to execute.

The controls

Joystick: optional

To surface: pull the joystick towards you (down), this pushes the submarines nose up.

To dive: push the joystick away from you (up), the nose goes down.

Rudder Control: move the joystick to the left or right. Use in conjunction with the compass setting (see instrument panel).

Keyboard

Active keys are:

E-> surface
X-> dive
S-> move left
D-> move right
1-> abort mission
2-> start game
3-> report
4-> crash dive
5-> blow ballast
6-> forward
7-> reverse
8-> fire torpedo
9-> map
0-> sonar screen
=-> periscope
[SPACE]-> pause

To surface: press up arrow [E]

To dive: press down arrow [X]

Rudder: left arrow [S] to move left and right arrow [D] to move right.

Direction: forward [6] or reverse [7] to change direction and speed.

Crash dive [4] or Blow Ballast [5] to rapidly change your depth. Dive or surface to neutralize either of these directions.

How to play

1. After the title, the computer asks you to choose a skill level (1-3). Press the space bar for the number you want. Press start [2] when you are ready.
2. The map in the centre of the screen shows your position (black cross) and the positions of the enemy convoys (white dots). The object is to track down the convoys and sink all the ships. The enemy is composed of the following: Destroyer, Tanker and Freighter. You score tonnage points for sinking ships, with more points for tankers and freighters than for destroyers. The enemy ships are armed with shells and depth charges and can cause you heavy damage. If they detect you they may attack or take evasive action to try to lose you. To attack the target, you have the following equipment:

MAP: Press map [9] to establish your position relative to land and convoys at any time.

SONAR: Press sonar [0] for a picture of the sea around you. Ships within range show up as blips on the screen.

continued on page 18

```
160 RANDOMIZE
170 CALL CLEAR
180 DISPLAY AT(2,12):"BEELI
NE"
190 CALL SCREEN(11)
200 M$=RPT$(CHR$(124),28)
210 OPTION BASE 1
220 DIM V$(2),Q$(2,18)
230 V$(2)=RPT$(CHR$(117)&CHR
$(116),7):: V$(1)=RPT$(CHR$(
119)&CHR$(118),7)
240 CALL CHAR(96,"0000102810
000000",97,"",98,"",99,"")!R
EM OPEN FIELD BEE
250 CALL CHAR(104,"010101010
03C43B3",105,"03030303030300
",106,"80C0C0C0B03CE4E0",107
,"E0E0606060206030")
260 CALL CHAR(108,"010303030
13C2303",109,"07070303020303
06",110,"80C0C0C0803CE2E0",1
11,"E0E0606060606000")
270 CALL CHAR(112,"000000000
00B1C08")!REM FLOWER BEE
280 CALL CHAR(116,"F00804020
1020408",117,"01020408F00804
02",118,"F0F8FCFEFFFEFCF8",1
19,"0103070FFF0F0703")
290 CALL COLOR(11,2,11)
300 FOR L=4 TO 20 :: DISPLAY
 AT(L,1):V$(1-(L/2=INT(L/2))
)&V$(2+(L/2=INT(L/2))):: NEX
T L
310 CALL CHAR(114,"442838382
8382810")! REM HIVE
320 CALL CHAR(120,"FF00FFFF0
0FFFFE7")! REM HIVE BEE
330 CALL CHAR(124,"CC33CC33C
C33CC33")! REM SMOKE
340 CALL CHAR(126,"3F4985858
585493F",127,"FC92A1A1A1A192
FC")! REM THUMB
350 CALL CHAR(129,"6EFFFF7EF
FFFFF76")! REM MYSTERY FLOWE
R
360 CALL CHAR(130,"183C7EFF7
E3C1800")! REM POLLEN
370 CALL CHAR(136,"FF00FFFF0
0FFFFE7")! REM BLUE HIVE
380 DISPLAY AT(23,5):"INSTRU
CTIONS? (Y/N)"
390 CALL MAGNIFY(2):: CALL S
PRITE(#1,114,2,210,210,-4,-4
)
400 CALL KEY(0,K,S):: IF K=8
9 OR K=121 THEN 1060 :: IF K
<>78 AND K<>110 THEN 400
410 CALL COLOR(10,11,1,13,14
,1,12,16,15,13,11,1,14,5,15)
420 DISPLAY AT(6,5)ERASE ALL
 BEEP:"START LEVEL? (1-9)"
430 CALL KEY(0,K,S):: G=K-49
 :: IF G>8 OR G<0 THEN 430 E
LSE DISPLAY AT(6,25):STR$(G+
1):: G1=G+1
440 DISPLAY AT(12,5)BEEP:"JO
YSTICKS? (Y/N)"
450 CALL KEY(0,K,S):: IF K=7
8 OR K=110 THEN 470
460 IF K<>89 AND K<>121 THEN
 450 ELSE JS=-1 :: DISPLAY A
T(22,5):"CHECK ALPHA LOCK" :
: GOTO 480
470 JS=0
480 CALL DELSPRITE(ALL):: R=
40 :: C=125 :: Z=0 :: GOTO 7
80
490 FOR L=1 TO 18 :: Q$(1,L)
=V$(2):: Q$(2,L)=V$(1):: NEX
T L :: IF G>10 THEN 990
500 CALL HCHAR(3,1,32,704)::
 CALL SCREEN(4):: C9=0
510 CALL VCHAR(4,24,120):: C
ALL VCHAR(4,8,136):: CALL MA
GNIFY(1)
520 CALL SPRITE(#5,96,2,R,C)
:: IF J>4 THEN 860
530 CALL DELSPRITE(#9):: F1=
60+INT(RND*130):: F2=25+INT(
RND*215):: J=4 :: L=0
540 GOSUB 1030 :: CALL MOTIO
N(#5,-Y,X)
550 CALL POSITION(#5,B1,B2):
: S=INT((ABS(F1-B1)+ABS(F2-B
2))/17.2):: CALL SOUND(-500,
RND*6+110,S)
560 IF ABS(B1-32)<6 AND((ABS
(B2-187)<6)+(ABS(B2-60)<6))T
HEN 660
570 IF L>0 THEN 880 :: SC=SC
-G :: GOSUB 910 :: IF S>0 OR
 Z=25 THEN 540
580 CALL DELSPRITE(ALL)
590 CALL MAGNIFY(2):: CALL S
PRITE(#10,129,7,122+RND*62,R
ND*230+1):: CALL SPRITE(#6,1
12,11,100,120)
600 GOSUB 1030 :: CALL MOTIO
N(#6,-2*Y,2*X)
610 CALL COINC(ALL,N):: SC=S
C-G-N*G*4 :: CALL POSITION(#
6,R,C):: IF R<60 THEN 650 ::
 IF N THEN 640
620 GOSUB 910 :: CALL SOUND(
-500,RND*B+110,3):: IF RND<.
90-G*.015 THEN 600
630 CALL LOCATE(#10,90+RND*7
0,10+RND*230):: GOTO 600
640 Z=Z+1 :: CALL SOUND(-300
,120+Z,5):: IF Z<21 THEN 620
650 CALL POSITION(#6,R,C)::
CALL DELSPRITE(ALL):: GOTO 5
10
660 C9=(ABS(B2-187)<6):: H=2
+C9 :: CALL DELSPRITE(ALL)
670 CALL MAGNIFY(2):: CALL S
CREEN(2):: CALL HCHAR(3,1,32
,704):: IF J>4 THEN 690
680 R=165 :: C=124
690 CALL COLOR(11,2,2):: FOR
 L=1 TO 18 :: DISPLAY AT(L+4
,8):Q$(H,L):: NEXT L :: CALL
 COLOR(11,11,16)
700 J=4-(RND<.1+G/25)*19 ::
P=1+2*(J=23):: CALL SCREEN(6
-C9*9):: CALL SPRITE(#7,114,
2,R,C)
710 GOSUB 1030 :: CALL MOTIO
N(#7,-Y,X)
720 CALL POSITION(#7,R,C)::
IF ABS(99-R)>75 THEN 760 ::
IF RND>.99-G*.01 OR(J>4)*(J<
23)THEN 800
730 CALL KEY(2,K,S):: IF K=1
8 THEN 740 ELSE IF K=11 THEN
 1300 ELSE 710
740 R=INT(R/8-2+(R>167)):: I
F Q$(H,R)=V$(H)OR C9*(Z<7)OR
 H*Z>=42 THEN 710
750 Q$(H,R)=V$(H):: DISPLAY
AT(R+4,8):Q$(H,R):: Z=Z+7+C9
*14 :: SC=SC-7*G*C9 :: GOSUB
 910 :: GOTO 710
760 CALL DELSPRITE(ALL):: R=
40 :: C=56-C9*128 :: L=0
770 L=L+1 :: IF C9=0 OR Q$(H
,L)=V$(2)THEN 500 :: IF L<>1
8 THEN 770
780 G=G+1 :: DISPLAY AT(1,4)
SIZE(9):"LEVEL:  "&STR$(G)::
 CALL SOUND(200,660,4)
790 CALL SOUND(400,990,2)::
SC=SC+G*150 :: GOSUB 910 ::
GOTO 490
800 J=J+P :: DISPLAY AT(J,1)
:M$
810 IF SGN(J-INT(1.5+R/8))<>
SGN(P)THEN 710 ELSE CALL MOT
ION(#7,0,0):: CALL HCHAR(5,1
,124,576):: L,Z=0
820 GOSUB 910 :: U=41+RND*8*
(16-G):: R1=RND>.5 :: CALL S
PRITE(#18,126-R1,10,U,235+R1
*225):: U=INT(1+U/8)
830 GOSUB 1030 :: CALL MOTIO
N(#7,-2*Y,2*X):: CALL COINC(
ALL,N):: L=L+1 :: IF N THEN
850 ELSE IF L<50-G*2 THEN 83
0
840 FOR L=U TO U+G-1 :: Q$(H
,L-4)=V$(2):: NEXT L :: CALL
 POSITION(#7,R,C):: GOTO 660
850 CALL SOUND(200,115,3)::
SC=SC+5*G :: GOSUB 910 :: CA
LL POSITION(#7,R,C):: GOTO 6
60
860 CALL MAGNIFY(3)
870 CALL SPRITE(#9,108,7,24+
G,C+SGN(90-C)*15,2-INT(RND*3
+1)*(RND<G/10),SGN(90-C)*3):
: J=4 :: L=0
880 CALL COINC(ALL,N)
890 IF N THEN 900 :: L=L+3 :
: CALL PATTERN(#9,104-4*(INT
(L/2)=L/2)):: IF L<100-G*2 T
HEN 540 ELSE 530
900 CALL SOUND(300,990,3)::
SC=SC+20*G :: GOTO 530
910 DISPLAY AT(1,22):SC :: I
F Z=0 THEN 920 :: DISPLAY AT
(2,4):RPT$(CHR$(130),Z):: GO
TO 930
920 CALL HCHAR(2,5,32,28)
930 IF SC<=0 THEN 940 :: RET
URN
940 CALL DELSPRITE(ALL):: CA
LL CLEAR
950 CALL SOUND(500,140,2)::
CALL SOUND(1000,110,3):: DIS
PLAY AT(10,6):"LEVELS COMPLE
TER:  "&STR$(G-G1)
960 DISPLAY AT(4,5):"OOPS -
BEE EXHAUSTED!"
970 DISPLAY AT(15,5):"REPLAY
?  PRESS REDO" :: DISPLAY AT
(17,5):"TO END PRESS CLEAR"
980 CALL KEY(0,K,S):: IF K=6
 THEN 420 ELSE 980
990 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL SOUND(300,4
66,2):: CALL SOUND(300,587,2
):: CALL SOUND(800,784,1)
1000 DISPLAY AT(13,1):"YOU H
AVE EARNED YOUR WINGS"
1010 DISPLAY AT(16,7):"AND A
LONG REST!" :: DISPLAY AT(20
,7):"SCORE:  "&STR$(SC)
1020 DISPLAY AT(23,6):"LEVEL
S COMPLETED:  "&STR$(G-G1)::
 STOP
1030 IF NOT JS THEN 1040 ELS
E CALL JOYST(2,X,Y):: RETURN
1040 CALL KEY(1,K,S)
1050 X=4*((K=4)+(K=2)+(K=15)
-(K=6)-(K=3)-(K=14)):: Y=4*(
(K=15)+(K+1=1)+(K=14)+(K>3)*
(K<7)):: RETURN
1060 DISPLAY AT(4,1):"USE TH
E ARROW KEYS AND THE"
1070 DISPLAY AT(5,1):"W,R,Z
AND C KEYS TO FLY THE" :: DI
SPLAY AT(6,1):"BEE. TO COLLE
CT POLLEN"
1080 DISPLAY AT(7,1):"FOLLOW
 THE BUZZING UNTILL IT" :: D
ISPLAY AT(8,1):"GETS LOUDEST
-THEN TOUCH"
1090 DISPLAY AT(9,1):"THE MA
GIC FLOWER. YOU MAY":"ALSO S
TEAL POLLEN BY ENTER-"
1100 DISPLAY AT(11,1):"ING T
HE BLUEHIVE AND PRESS-":"ING
 FIRE(Y) TO PICK IT UP."
1110 DISPLAY AT(13,1):""
```

continued on page 17

```
10 REM TEXAS INSTRUMENTS CON
TRIBUTED SOFTWARE
20 ON ERROR 8000
50 GOTO 10000
100 REM
110 CALL CLEAR :: DISPLAY AT
(12,9):"ONE MOMENT"
120 ALL_0$="0000000000000000
"
130 HNSL$="01030000000000000
1000000000000380C0000000000
000C000000000000060"
150 FNHUL$="00080901"
175 FNHL$=FNHUL$&ALL_0$&"000
0008080001010"
190 SET_TORSOL$="00000000FOD
0C0C0"
200 TORSOUL$="000000080F0101
01"
225 TORSOL$=TORSOUL$&ALL_0$&
SET_TORSOL$&ALL_0$
230 PANTSL$=ALL_0$&"00010101
01010100"&ALL_0$&"00C0C04040
4040"
240 APEAD$="010303010F1F1F1B
1B1B1B1B0306060E80C0C080F0F8
F8D8D8C8C8C8C0606070"
250 APEAU$="39131B19191F1F03
0303030303060E80C0C080F0F8
F8D8D8D8D8D8C0606070"
252 APEBAU$="39131B19191F1F0
3030303030306060E98D8D89898F
8F8C0C0C0C0C0C0606070"
255 RUNOR$="0000000F1327470F
0F0D0CF880000000C0C080C0C0A8
10000080C040402020 30"
260 RUNOL$="0303010323150800
0001030202040 40C000000F0C8E4
E2F0F0B0301F01"
265 RUNMR$="000000070B17170F
0F0B0B0409122233C0C080C0C0D0
600000808080"
270 RUNML$="03030103030B0600
0001010100000000000000E0D0E8
E8F0F0B0B020904844CC"
275 RUNCR$="000000070B0F0707
0703030303030303C0C080C0C080
4000000000000000080"
280 RUNCL$="0303010303010200
0000000000000001000000E0D0F0
E0E0E0C0C0C0C0C0C0C0"
285 RUNUP$="01030301070B0B0B
0B0B03020202070290D0D090F080
8080808080808080808080"
290 RUNDOWN$="01030301070B0B
0B0B0B0302020207028 0C0C080F0
9090909080808080808080"
291 SLANT1$="808080804040404
0"
292 SLANT2$="202020201010101
0"
293 SLANT3$="080808080404040
4"
294 SLANT4$="020202020101010
1"
295 BANANA$="000000000008070
3"&ALL_0$&"0000000000 10E0C0"
&ALL_0$
296 PIE$="70F8F8F870"&RPT$(A
LL_0$,3)
297 FILLER$=ALL_0$
298 FIELD_FILL$=RPT$(CHR$(91
)&CHR$(92),8)
299 RETURN
300 REM
301 CALL MAGNIFY(3)
302 CALL CHAR(40,PIE$,36,BAN
ANA$):: CALL COLOR(2,12,6)
303 CALL CHAR(48,"FEFEFE00FE
FEFE00"):: CALL COLOR(3,10,1
6)
304 CALL CHAR(88,SLANT1$,89,
SLANT2$,90,SLANT3$,91,SLANT4
$,92,FILLER$):: CALL COLOR(8
,6,6)
305 CALL CHAR(44,RUNUP$,56,R
UNDOWN$)
307 CALL CHAR(49,"FF81818181
8181FF")
310 CALL CHAR(60,APEBAU$)::
CALL CHAR(64,"FF")
320 CALL CHAR(96,HNSL$)
330 CALL CHAR(100,TORSOL$)
340 CALL CHAR(104,PANTSL$)
350 CALL CHAR(108,FNHL$)
355 CALL CHAR(112,APEAU$)
357 CALL CHAR(116,APEAD$)
360 CALL CHAR(120,RUNOR$)
365 CALL CHAR(124,RUNMR$)
370 CALL CHAR(128,RUNCR$)
375 CALL CHAR(132,RUNOL$)
380 CALL CHAR(136,RUNML$)
385 CALL CHAR(140,RUNCL$)
399 RETURN
400 REM
410 CALL SPRITE(#6,108,11,10
4,100,#5,104,5,104,100,#4,10
0,7,104,100,#3,96,2,104,100)
420 CALL SPRITE(#7,120,2,168
,100)
430 RETURN
500 REM
517 FOR R=13 TO 15 :: CALL H
CHAR(R,1,48,32):: NEXT R
540 R=16 :: FOR C=1 TO 31 ST
EP 2
542 CALL VCHAR(R,C,88):: CAL
L VCHAR(R+1,C,89):: CALL VCH
AR(R+2,C,90):: CALL VCHAR(R+
3,C,91)
544 CALL VCHAR(R,C+1,92,4)::
 NEXT C
546 R=20 :: FOR C=2 TO 32 ST
EP 2
547 CALL VCHAR(R,C-1,92,4)
548 CALL VCHAR(R,C,88):: CAL
L VCHAR(R+1,C,89):: CALL VCH
AR(R+2,C,90):: CALL VCHAR(R+
3,C,91)
549 NEXT C
550 CALL HCHAR(24,1,64,32)
560 CALL COLOR(8,16,3)!FLASH
 ON FIELD
599 RETURN
700 REM
701 CALL CLEAR :: CALL SCREE
N(6)
703 MSG$="PIES  LEFT" :: FOR
 R=1 TO 10 :: DISPLAY AT(R,1
):SEG$(MSG$,R,1):: NEXT R
707 FOR R=1 TO 12 :: IF R=6
THEN 720
710 READ C :: IF C=999 THEN
720
715 CALL VCHAR(R,C,49):: GOT
O 710
720 NEXT R
725 DATA 5,6,7,8,9,10,11,12,
17,18,19,20,21,22,23,24,25,2
8,29,30,31,32,999
730 DATA 5,8,10,11,17,20,22,
25,26,28,29,999
735 DATA 5,6,7,8,10,11,14,15
,17,18,19,20,22,25,27,28,29,
31,32,999
740 DATA 5,10,11,17,19,22,25
,28,29,32,999
745 DATA 5,9,10,11,12,17,19,
21,22,23,25,28,29,30,31,32,9
99
750 DATA 9,10,11,12,13,14,15
,16,17,20,21,22,23,24,25,26,
27,999
755 DATA 9,13,16,17,20,21,24
,25,28,999
760 DATA 9,10,11,12,13,16,17
,20,21,22,23,24,25,28,999
765 DATA 12,13,15,16,17,20,2
1,24,25,28,999
770 DATA 9,10,11,12,13,14,15
,16,17,18,19,20,21,24,25,26,
27,999
775 DATA 16,999
780 RETURN
1000 REM
1010 CALL CLEAR :: DISPLAY A
T(12,1):"NEED INSTRCTIONS? (
Y/N) "
1015 DISPLAY AT(24,8):"PRESS
 ENTER!"
1020 ACCEPT AT(12,25)SIZE(1)
VALIDATE("YN")BEEP:A$
1030 IF A$="N" THEN RETURN
1035 CALL CLEAR
1040 M$="THE OBJECT OF THE G
AME IS TOHIT THE MAN AGAINST
 THE     BRICK WALL WITH A P
IE.      SOUNDS SIMPLE, DOES
N'T IT?" :: GOSUB 2000
1050 M$="YOU MAY CHOOSE A SK
ILL      LEVEL: 'EXPERT', 'S
O-SO',   OR 'ROOKIE'" :: GOS
UB 2000
1060 M$="A  'ROOKIE'  GETS 1
0 PIES   A  'SO-SO '  GETS
8 PIES   AN 'EXPERT'  GETS
5 PIES" :: GOSUB 2000
1070 M$="IF YOU RUN OUT OF P
IES, YOU LOSE!!!" :: GOSUB 2
000
1080 M$="YOU CONTROL YOUR MA
N USING  THE 'J', 'K', AND '
L' KEYS  LOCATED ON THE RIGH
T SIDE OFTHE KEYBOARD" :: GO
SUB 2000
1090 M$="'J' MOVES HIM TO TH
E RIGHT  'K' MAKES HIM THROW
 A PIE   'L' MOVES HIM TO TH
E LEFT" :: GOSUB 2000
1100 M$="WATCH OUT FOR THE N
ASTY     GORILLA... HE'LL TH
ROW      BANANAS AT YOU. IF
YOU GET  HIT, YOU LOSE SO MO
VE       QUICKLY!!"
1105 GOSUB 2000
1110 CALL CLEAR :: PRINT "PR
ESS THE SPACE BAR TO STARTTH
E GAME                   PR
ESS THE 'I' KEY TO SEE THEIN
STRUCTIONS AGAIN"
1120 FOR I=1 TO 8 :: PRINT :
: NEXT I :: DISPLAY AT(20,4)
:"G O O D   L U C K !!!"
1130 CALL KEY(0,K,S):: IF S=
0 THEN 1130
1140 IF K<>32 AND K<>73 THEN
 1130
1150 IF K=73 THEN 1035
1160 RETURN
2000 REM
2010 CALL CLEAR
2020 PRINT M$
2030 FOR I=1 TO 10 :: PRINT
:: NEXT I
2040 DISPLAY AT(24,2):"PRESS
 ANY KEY TO GO ON!"
2050 CALL KEY(0,K,S):: IF S=
0 THEN 2050
2060 RETURN
8000 REM
8010 ON ERROR 8000
8020 RETURN NEXT
9000 REM
9010 CALL CLEAR
9020 DISPLAY AT(6,8):"SKILL
 LEVEL" :: DISPLAY AT(7,8):"
-----  -----"
9030 DISPLAY AT(10,3):"1. EX
PERT ( 5 PIES)"
9040 DISPLAY AT(12,3):"2. SO
-SO  ( 7 PIES)"
9050 DISPLAY AT(14,3):"3. RO
OKIE (10 PIES)"
```

```
9060 DISPLAY AT(24,8):"PRESS
 ENTER!"
9065 DISPLAY AT(19,1):"YOUR
CHOICE?"
9070 ACCEPT AT(19,13)BEEP VA
LIDATE(DIGIT)SIZE(1):SL
9080 IF SL<1 OR SL>3 THEN 90
70
9120 RETURN
10000 !*****************
10001 !* THE DRIVER    *
10002 !*****************
10005 GOSUB 1000 !INSTRUCTIO
NS
10007 GOSUB 9000 !GET SKILL
LEVEL
10010 GOSUB 100 !SET PATTERN
S
10020 GOSUB 300 !ASSIGN TO N
UMBERS
10030 GOSUB 700 !TITLE
10040 GOSUB 500 !DISPLAY
10060 RANDOMIZE
10070 GOSUB 400 !PUT SPRITES
 ON SCREEN
10075 CALL VCHAR(1,4,40,10)
10080 IF SL=1 THEN PIE_COUNT
=5 ELSE IF SL=2 THEN PIE_COU
NT=7 ELSE PIE_COUNT=10
10085 B_FLAG$,PIE_FLAG$="OFF
"
10090 GOSUB 11010 !KEYBRD
10120 GOSUB 12010 !TARGET
10150 GOSUB 13010 !APE?
10180 IF DIR_IND$="R" THEN C
ALL PATTERN(#7,124)ELSE IF D
IR_IND$="L" THEN CALL PATTER
N(#7,136)
10195 IF B_FLAG$="ON" THEN G
OSUB 15010 !BANANA COINC
10210 IF PIE_FLAG$="ON" THEN
 GOSUB 14010 !PIE COINC
10220 IF PIE_COUNT=0 THEN GO
SUB 19010 ELSE CALL VCHAR(1,
4,32,10-PIE_COUNT)
10240 IF DIR_IND$="R" THEN C
ALL PATTERN(#7,120)ELSE IF D
IR_IND$="L" THEN CALL PATTER
N(#7,132)
10270 IF B_FLAG$="ON" THEN G
OSUB 15010 !BANANA COINC
10300 GOTO 10090
10330 STOP
11000 !*********************
11001 !* CHECK KEYBOARD    *
11002 !*********************
11010 CALL KEY(0,K,S):: IF S
=0 THEN 11020 ELSE 11040
11020 CALL MOTION(#7,0,0)::
CALL PATTERN(#7,44):: DIR_IN
D$="U"
11030 IF PIE_FLAG$="OFF" THE
N CALL POSITION(#7,Y,X):: IF
 X>244 THEN RETURN ELSE CALL
 SPRITE(#1,40,10,Y-4,X+10)
11040 K=K-73 :: IF K<1 OR K>
3 THEN RETURN
11045 ON K GOTO 11080,11100,
11050
11050 IF DIR_IND$="U" AND PI
E_FLAG$="OFF" THEN CALL DELS
PRITE(#1)
11060 CALL PATTERN(#7,120)::
 CALL MOTION(#7,0,3):: DIR_I
ND$="R" :: RETURN
11080 IF DIR_IND$="U" AND PI
E_FLAG$="OFF" THEN CALL DELS
PRITE(#1)
11090 CALL PATTERN(#7,132)::
 CALL MOTION(#7,0,-3):: DIR_
IND$="L" :: RETURN
11100 IF DIR_IND$<>"U" THEN
CALL MOTION(#7,0,0):: CALL P
ATTERN(#7,44)
11110 IF PIE_FLAG$="ON" THEN
 RETURN
11120 CALL POSITION(#7,Y,X):
: IF X>250 THEN RETURN ELSE
CALL SPRITE(#1,40,10,Y-8,X+1
0)
11125 CALL SOUND(100,1000,0)
11130 CALL MOTION(#1,-4,-1):
: CALL PATTERN(#7,56):: PIE_
FLAG$="ON" :: RETURN
12000 !*********************
12001 !* TARGET DIRECTION*
12002 !*********************
12010 IF DIR_IND$="R" THEN C
ALL PATTERN(#7,124)ELSE IF D
IR_IND$="L" THEN CALL PATTER
N(#7,136)
12020 RDIR=INT(RND*1+.5):: I
F RDIR=0 THEN DIR=-2 ELSE DI
R=2
12030 CALL POSITION(#3,Y,X)
12040 IF X<32 THEN CALL MOTI
ON(#3,0,2,#4,0,2,#5,0,2,#6,0
,2):: GOTO 12070
12050 IF X>212 THEN CALL MOT
ION(#3,0,-2,#4,0,-2,#5,0,-2,
#6,0,-2):: GOTO 12070
12060 CALL MOTION(#3,0,DIR,#
4,0,DIR,#5,0,DIR,#6,0,DIR)
12070 RETURN
13000 !*********************
13001 !* APE TIME, FOLKS   *
13002 !*********************
13010 IF DIR_IND$="R" THEN C
ALL PATTERN(#7,128)ELSE IF D
IR_IND$="L" THEN CALL PATTER
N(#7,140)
13020 AP=INT(RND*4)+1 :: IF
AP=3 AND B_FLAG$="OFF" THEN
13030 ELSE RETURN
13030 CALL POSITION(#7,Y,X):
: IF X<3 THEN RETURN ELSE CA
LL SPRITE(#8,112,2,81,X)
13040 CALL SPRITE(#2,36,10,7
3,X-3):: CALL MOTION(#2,14,0
)
13045 CALL SOUND(100,120,0)
13050 CALL PATTERN(#8,116)::
 B_FLAG$="ON"
13060 RETURN
14000 !*****************
14001 !* PIE COINC     *
14002 !*****************
14010 CALL POSITION(#1,Y,X):
: IF Y<96 THEN CALL DELSPRIT
E(#1):: PIE_FLAG$="OFF" :: P
IE_COUNT=PIE_COUNT-1 :: RETU
RN
14020 CALL COINC(#1,#3,8,HIT
):: IF HIT=-1 THEN GOSUB 160
10 ELSE RETURN
15000 !*******************
15001 !* BANANA COINC? *
15002 !*******************
15010 CALL POSITION(#2,Y,X):
: IF Y>175 OR Y<40 THEN CALL
 DELSPRITE(#2,#8):: B_FLAG$=
"OFF" :: RETURN
15020 CALL COINC(#2,#7,8,HIT
):: IF HIT=-1 THEN GOSUB 170
10 ELSE RETURN
16000 !*******************
16001 !* WIN ROUTINE    *
16002 *******************
16010 CALL MOTION(#1,0,0,#3,
0,0,#4,0,0,#5,0,0,#6,0,0)
16015 CALL DELSPRITE(#1)
16020 CALL SOUND(-99,-5,0)::
 CALL MOTION(#7,0,0):: CALL
PATTERN(#7,56)
16030 IF B_FLAG$="ON" THEN C
ALL DELSPRITE(#2,#8)
16040 DISPLAY AT(19,7)SIZE(-
16)BEEP:"WOW!! A WINNER!!"
16045 CALL MOTION(#3,-10,0,#
4,-10,0,#5,-10,0,#6,-10,0)
16050 SN=262 :: INC=50 :: FO
R I=1 TO 200 :: SN=SN+INC ::
 CALL SOUND(-500,SN,0):: CAL
L POSITION(#6,Y,X)
16060 IF Y<8 THEN CALL DELSP
RITE(#3,#4,#5,#6):: GOTO 161
90
16070 NEXT I
16190 GOSUB 18000 ! PLAY AGA
IN
16200 RETURN
17000 !***************
17001 !*LOSE ROUTINE*
17002 !***************
17010 CALL MOTION(#2,0,0,#7,
0,0,#3,0,0,#4,0,0,#5,0,0,#6,
0,0)
17020 CALL SOUND(-99,-5,0)::
 CALL PATTERN(#7,44)
17025 CALL DELSPRITE(#2,#1)
17030 IF PIE_FLAG$="ON" THEN
 CALL DELSPRITE(#1)
17040 DISPLAY AT(19,7)SIZE(-
16)BEEP:"OH NO! A LOSER!!"
17050 CALL MOTION(#7,2,0)
17060 FOR I=260 TO 110 STEP
-8 :: CALL SOUND(-500,I,0)::
 CALL POSITION(#7,Y,X):: IF
Y>194 OR Y<8 THEN 17080
17070 NEXT I
17080 CALL DELSPRITE(#7)
17090 CALL POSITION(#8,Y,X)
17100 FOR I=1 TO 7
17110 CALL LOCATE(#8,Y-4,X):
: CALL PATTERN(#8,60):: CALL
 SOUND(-100,200,10)
17120 CALL LOCATE(#8,Y,X)::
CALL PATTERN(#8,116):: CALL
SOUND(-100,150,10)
17130 NEXT I
17200 GOSUB 18000 !PLAY AGAI
N
17210 RETURN
18000 !*****************
18001 !* PLAY AGAIN?? *
18002 !*****************
18010 FOR I=1 TO 750 :: NEXT
 I
18020 CALL DELSPRITE(ALL)
18030 DISPLAY AT(6,6)SIZE(-2
1):"'A' AGAIN  'Q' QUIT "
18050 CALL KEY(0,K,S):: IF S
=0 THEN 18050
18060 IF K<>65 AND K<>81 THE
N 18050
18065 IF K=65 THEN 18070 ELS
E 18080
18070 DISPLAY AT(6,6)SIZE(24
):" " :: DISPLAY AT(19,7)SIZ
E(-16):FIELD_FILL$ :: GOTO 1
0060
18080 CALL CLEAR :: CALL CHA
RSET :: DISPLAY AT(12,12):"B
YE!" :: CALL CLEAR :: RUN "D
SK1.LOAD"
18090 RETURN
19000 !*********************
19001 ! OUT OF PIES, PAL *
19002 !*********************
19010 CALL MOTION(#7,0,0)::
CALL PATTERN(#7,56):: DIR_IN
D$="U"
19015 CALL VCHAR(1,4,32,10)
19020 GOSUB 13030 !SEND THE
KILLER BANANA
19030 FOR I=1 TO 50 :: GOSUB
 15010 !CHECK FOR THE HIT
19040 NEXT I
19050 RETURN
```

```
100 REM CHRIS,CHERYL&BETH DE
MERS (12,14,1981)
110 CALL CLEAR
120 DIM A$(10),B$(26),AA$(10
,60),KN(26),KT(26),LN(10),QQ
$(26),XXX$(26)
130 INPUT "How many lines in
 the       cryptoquote? ":NL
140 CALL CLEAR
150 FOR J=1 TO NL
160 PRINT "Type line";J;"of
cryptoquote:"
170 INPUT A$(J)
180 PRINT
190 LN(J)=LEN(A$(J))
200 NEXT J
210 FOR J=1 TO 26
220 KN(J)=0
230 NEXT J
240 FOR J=1 TO NL
250 FOR K=1 TO LN(J)
260 FOR L=1 TO 26
270 IF SEG$(A$(J),K,1)<>CHR$
(L+64)THEN 300
280 KN(L)=KN(L)+1
290 GOTO 310
300 NEXT L
310 NEXT K
320 NEXT J
330 FOR J=1 TO 26
340 MX=-1
350 FOR K=1 TO 26
360 IF KN(K)<MX THEN 390
370 KS=K
380 MX=KN(K)
390 NEXT K
400 B$(J)=CHR$(KS+64)
410 KT(J)=MX
420 KN(KS)=-99
430 NEXT J
440 GOSUB 1120
450 FOR J=1 TO NL
460 FOR K=1 TO LN(J)
470 AA$(J,K)=""
480 NEXT K
490 NEXT J
500 FOR J=1 TO NL
510 PRINT A$(J)
520 PRINT
530 NEXT J
540 INPUT "Enter substitutio
n (cipher  letter first,deco
de second) ":X$,Y$
550 ZP=0
560 ZP=ZP+1
570 IF KT(ZP)=0 THEN 590
580 GOTO 560
590 FOR J=1 TO ZP-1
600 IF X$=B$(J)THEN 630
610 NEXT J
620 GOTO 540
630 IF Y$="" THEN 720
640 FOR J=1 TO 26
650 IF QQ$(J)=Y$ THEN 740
660 NEXT J
670 IF X$<>XXX$((ASC(X$))-64
)THEN 700
680 RE=POS(GG$,QQ$((ASC(X$))
-64),1)
690 GG$=SEG$(GG$,1,RE-1)&SEG
$(GG$,RE+1,LEN(GG$)-1)
700 XXX$((ASC(X$))-64)=X$
710 IF Y$<>"" THEN 740
720 RE=POS(GG$,QQ$((ASC(X$))
-64),1)
730 GG$=SEG$(GG$,1,RE-1)&SEG
$(GG$,RE+1,LEN(GG$)-1)
740 QQ$((ASC(X$))-64)=Y$
750 FOR J=1 TO NL
760 FOR K=1 TO LN(J)
770 IF SEG$(A$(J),K,1)=X$ TH
EN 780 ELSE 790
780 AA$(J,K)=Y$
790 NEXT K
800 NEXT J
810 CALL CLEAR
820 FOR J=1 TO NL
830 PRINT A$(J)
840 FOR K=1 TO LN(J)
850 IF AA$(J,K)="" THEN 860
ELSE 870
860 AA$(J,K)=" "
870 PRINT AA$(J,K);
880 NEXT K
890 PRINT : : :
900 NEXT J
910 GOSUB 1250
920 PRINT "FREQ(F), SUB(S),
ERASE(E)"
930 CALL KEY(0,KEY,STATUS)
940 IF STATUS=0 THEN 930
950 IF KEY=70 THEN 1400
960 IF KEY=83 THEN 540
970 IF KEY=69 THEN 990
980 GOTO 930
990 GG$=""
1000 Y$=""
1010 FOR J=1 TO 26
1020 XXX$(J)=""
1030 NEXT J
1040 GOTO 440
1050 PRINT "Do you wish to s
olve anothercryptogram (Y/N)
"
1060 CALL KEY(0,KEY,STATUS)
1070 IF STATUS=0 THEN 1060
1080 IF KEY=89 THEN 130
1090 IF KEY<>78 THEN 1060
1100 PRINT "Hope you enjoyed
 doing the  cryptoquote,try
another one tommorow !!!"
1110 END
1120 CALL CLEAR
1130 PRINT "**FREQUENCY OF O
CCURRENCE**"
1140 PRINT : : : :
1150 FOR J=1 TO 26
1160 PRINT B$(J);KT(J);"   "
;
1170 NEXT J
1180 PRINT : : :
1190 GOSUB 1250
1200 PRINT :
1210 PRINT "When review is c
omplete,     press ANY key"
1220 CALL KEY(0,KEY,STATUS)
1230 IF STATUS=0 THEN 1220
1240 RETURN
1250 FOR K=1 TO LEN(GG$)
1260 IF Y$=SEG$(GG$,K,1)THEN
 1380
1270 NEXT K
1280 IF LEN(GG$)=0 THEN 1370
1290 FOR I=1 TO LEN(GG$)
1300 IF Y$>=SEG$(GG$,I,1)THE
N 1330
1310 GG$=Y$&GG$
1320 GOTO 1380
1330 IF Y$>=SEG$(GG$,I+1,1)T
HEN 1360
1340 GG$=SEG$(GG$,1,I)&Y$&SE
G$(GG$,I+1,LEN(GG$)-1)
1350 GOTO 1380
1360 NEXT I
1370 GG$=GG$&Y$
1380 PRINT GG$
1390 RETURN
1400 GOSUB 1120
1410 GOTO 540
```

continued from page 14

```
1120 DISPLAY AT(14,1):"   PO
LLEN IS PUT INTO THE":"WHITE
 HIVE THE SAME WAY. THE"
1130 DISPLAY AT(16,1):"GOAL
IS TO FILL ALL TEN HIVES":"L
EVELS BEFORE EXHAUSTING THE"
1140 DISPLAY AT(18,1):"BEE.
YOU MUST WATCH FOR THE":"BEE
KEEPER AS HE WILL PUFF"
1150 DISPLAY AT(20,1):"SMOKE
 THROUGH THE HIVE AND":"TRY
TO TAKE THE POLLEN OUT."
1160 DISPLAY AT(23,5):"PRESS
 ANY KEY"
1170 CALL KEY(0,K,S):: IF S=
0 THEN 1170
1180 CALL CLEAR :: DISPLAY A
T(2,1):"WHEN YOU SEE THE SMO
KE,GET":"OUT THE HIVE,S TOP
OR BOTTOM"
1190 DISPLAY AT(4,1):"AND ST
ING THE BEEKEEPER FOR":"BONU
S POINTS. IF YOU ARE"
1200 DISPLAY AT(6,1):"CAUGHT
 IN THE SMOKE YOU LOSE":"ANY
 POLLEN YOU HAVE AND MUST"
1210 DISPLAY AT(8,1):"TRY TO
 BITE THE BEEKEEPER,S":"THUM
B OR RISK LOSING POLLEN":"FR
OM THE HIVE."
1220 DISPLAY AT(11,4):"THE O
BJECT IS TO FILL THE":"TEN L
EVELS OF THE HIVE WITH-"
1230 DISPLAY AT(13,1):"OUT E
XHAUSTING THE BEE.":"A ZERO
SCORE EXHAUSTS THE ":"BEE."
1240 DISPLAY AT(17,1):"THE B
EE IS MORE EASILY TIRED":"AN
D INVIGORATED AT HIGHER ":"L
EVELS."
1250 DISPLAY AT(20,1):"ENTER
 HIVES AT THEIR BOTTOM."
1260 DISPLAY AT(21,1):"THE B
EE HOLDS 3 POLLEN ROWS.":"PR
ESS P IN HIVE TO PAUSE."
1270 DISPLAY AT(24,9):"PRESS
 ANY KEY"
1280 CALL KEY(0,K,S):: IF S=
0 THEN 1280
1290 GOTO 410
1300 CALL KEY(2,K,S):: CALL
SOUND(-99,INT(RND*8+110),10)
:: IF S<>1 THEN 1300 ELSE 71
0
```

"We used to have a real problem with other corporations raiding our software people."

continued from page 13

PERISCOPE: Press periscope [=] when you are at a depth of 40 feet and you will get a view of any ship within range that is in your line of sight. Use this mode to prepare for an attack.

TORPEDOES: When you are at a depth of less than 30 feet, press the fire button on the joystick, or fire [8] to release a torpedo. Aim it ahead of the target ship.

The following is a mock up of the instrument panel

```
AAAAA  HHHHHHHHHHHHHHHHHH IIIII
AAAAA  H                  H IIIII
AAAAA  H                  H
AAAAA  H                  H JJJJJ
       H                  H JJJJJ
BBBBB  H                  H
BBBBB  H                  H KKKKK
       H                  H
CCCCC  H                  H LLLLL
CCCCC  H                  H LLLLL
       H                  H LLLLL
DDDDDD H                  H LLLLL
              H
EEEEEE H                  H N MMM
       H                  H N MMM
FFFFFF H                  H N MMM
       H                  H N MMM
GGGGGG HHHHHHHHHHHHHHHHHH N MMM
```

The diagram above shows the relative position of the gauges on the panel and the description below tells what each are used for. At the beginning of each mission, you are allocated quantities of torpedoes, fuel, air, and battery charge (see skill levels for details). You must be careful not to run out of any of these. Your speed is initially set at nil, and you have to press either forward [6] or reverse [7] to start moving.

In detail, the controls are as follows:

A....Attitude: move joystick left and right or use arrow keys to change course.

B....Compass: the compass reading determines the course you steer.

C....Clock. This times how long missions last.

D....Torpedoes: the two digits on the left are the number of torpedoes remaining. Next to these are the torpedo status indicators. These can be any of the following colors: (a) light blue (cyan) = priming, (b) dark blue = loading (c) red = ready to fire (d) yellow = tube out of action.

E....Fuel supply

F....Battery charge reading: to re-charge, you must surface.

G....Speed: when the gauge is in the blue (cyan) area, the submarine is moving forward. When it is in the green area, the attitude dial causes the compass and depth dials to move in the opposite direction.

H....Sonar screen: shows the enemy ships as white blips. Your position is the black cross in the center which remains static. When an enemy ship is correctly lined up, the blip will appear directly above the cross, or at some point between it, and the edge of the screen.

I....Depth in feet.

J....Hydrophone chart: this is the longest range dial on the screen. The nearer a ship is to your submarine, the closer the relevant peak is to the left of the dial.

K....Tonnage sunk: the amount recorded is in thousands of tons (Ktons).

L....Depth below keel: be careful when diving. This shows the water below you.

M....Damage indicators: These show damage as a square ranging from empty (no damage) to full (severe damage), to (C = controls), (I = instruments), (H = hull) or (E = engines). You may carry on without waiting for repairs, but beware of unexpected malfunctions. (Your submarine is repaired automatically. However, repairs are carried out more rapidly on the surface). If your hull damage becomes severe, it may crack, with a watery grave awaiting you.

N....Air supply: can be renewed by surfacing.

Skill Levels

The higher the skill level, the more challenging the game becomes. You will, in your encounters with the enemy, have to attack and destroy more convoys with less fuel, charge and torpedoes.

Tactics

You constantly need to assess the pros and cons of moving on the surface or underwater. Running on the surface is quicker, but if you get too close to an enemy convoy on the surface, you will be seen and attacked by the escorting warships. To warn you, a bell sounds as you approach the enemy's range of vision on the surface.

Your greatest advantage over the enemy is your ability to travel underwater. However, this does use up air and batteries, and you can still be detected if you fire a torpedo.

As your initial working tactics, try approaching an enemy convoy on the surface using the map [9] mode. As you draw near, dive and locate the ships with your sonar [0] and hydrophone chart. Then come to a depth of about 25 feet and press periscope [=] for a view of the surface. Aim your submarine at the target, press the trigger on the joystick, or fire [8] to release a torpedo, aiming it ahead of the ship. Hits are recorded on the 'tonnage' sunk chart on the right. If you are under heavy attack from the surface ships and wish to lose them, dive deep and wait until the attack is over.

End of Game

The game ends when any of the following occurs: (a) all enemy convoys are sunk; (b) all oxygen is used; (c) all fuel and charge are used; (d) there is severe damage to the hull. The game can be terminated by pressing abort [1]. You still receive a rating of your abilities as a Submarine Commander.

Rating

At the end of each game, you receive a rating (score), with points awarded for tonnage sunk and the ultimate achievement of a Submarine Commander, by the elimination of all convoys. Points are deducted for fuel and torpedoes used, and damage sustained.

Summary of factors affecting your rating. Points are given for:

(a) tonnage sunk; and

(b) sinking all convoys proportional to skill level.

Points are deducted for:

(a) fuel used;

(b) torpedoes used;

(c) damage incurred; and

(d) being destroyed. o

continued from page 10

plus envelopes, plus four printed foolscap pages each, that was 180 times 5 equals 900 plus bits of printing to do. I suppose I might have been a teensy bit balked if I had worked that out before doing it!

So there we have the way to use PRK for really massive chunks of data. Firstly, use very conservative little files to sort the material and enable you to delete what will be too inconvenient to work with, get the supplementary data in (such as postcodes) re-sort, edit, save, and re-sort to categories, and use the PRG to split the files before augmenting them to take addressing or other data using the PRG to add fields, and to delete files when not part of the list being worked on. The data needed can then be added out of PRK and printed as address labels using PRG. (PRK equals Personal Record Keeper and PRG equals Personal Report Generator.)

The other topic is; DIABOLICAL RIBBONS. I wonder if any other users have a Diablo Printer? It lives up to its name if you try non-genuine ribbons, the point of essence being a little slot in the right of the cartridge with a brass or copper leaf spring in it which takes up the torque during printing and changing directions. Without it, and the substitutes tend to lack it, the ribbon slips up or is thrown up, and only the tops of letters are printed! Genuine Diablo Ribbons cost $18 whereas the substitutes cost only $9, but it would seem as if the "pact" must be taken seriously! Watch it if a dealer tries to sell you a cheapo ribbon. It might just work if it has that slot on the right with the leaf spring. o

Tutorial 1, the beginning

by Mack McCormick, USA

Here are the objectives of this first tutorial:

1. To introduce you to the Hexadecimal and Binary Number systems.
2. To introduce you to the assembler instruction format.
3. To introduce you to addressing modes.
4. First program. Adding two numbers and displaying them on the screen.
5. How to assemble.

Just a few words on Assembly language before we begin. It is not as difficult as you may believe. You will be communicating with the microprocessor at the first level above machine language, assembler. As you know, the machine actually communicates in binary 0's and 1's, on or off. Assembler allows us to talk to the machine in a language we can understand (Although I am sure the uninformed would disagree).

With these tutorials I will assume no prior knowledge of assembler or other number systems. Please bear with me, I will not insult your intelligence and things will become more complex soon. Stick with the tutorials. Read every book about assembler you can get your hands on. I will publish a bibliography of books soon. Do not get discouraged! Compuserve is a difficult medium through which to provide assistance. I promise to answer your questions and if I do not know the answer, I will find someone else who can. Please make this an nteractive process, as we grow and learn with each other.

Numbering Systems: hexadecimal (HEX) and binary are merely different base numbering systems for counting. It is important we understand both of these systems in addition to base 10 or the decimal system because assembler uses all three. I will tell you up front that I use a calculator designed for these numbering systems usually, but we need to understand the principles also. If you want to get a calculator, and I recommend that you do, there are several inexpensive models on the market. I use the Casio solar powered fx-451 scientific calculator for $35. It supports HEX, OCT, BIN, LOGICAL OPERATORS, and all scientific functions. Works great!

Craig Miller and others have also published programs which will allow you to use your computer but this has the disadvantage of requiring you to load another program every time you need to make a calculation, a real pain.

Binary Number System: As already mentioned, binary is the native language for your computer. Everything eventually gets converted to binary. Let us look at a decimal number first. As you know decimal means powers of 10. Each number represents a power of ten. For example 4175:

```
10|3=1000:4 X 1000 = 4000
10|2= 100:1 X 100  =  100
10|1=  10:7 X 10   =   70
10|0=   1:5 X 1    =    5
                      ----
                      4175
```

Binary numbers can be 1 or 0 only, hence base 2. The individual number is called a bit. A group of eight of these is called a byte. To convert the binary number 00001011 to decimal follow the same procedures you used with the decimal number: Ignore any leading zeros.

```
2|3=8:1 X 8=8
2|2=4:0 X 4=0
2|1=2:1 X 2=2
2|0=1:1 X 1=1
  ---
   11
```

To make it easier to communicate with the computer we most often use HEX. From now on I will use a > to indicate a number is in hex. Hex is base 16. That is a number may be 0 through 15. To represent numbers greater than 9 we use letters of the alphabet. 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Just remember to use >A for 10 and count to 15 ending with >F. Let us convert >394F to decimal:

```
16|3=4096:3 X 4096=12,288
16|2= 256:9 X 256 =  2,304
16|1=  16:4 X 16  =     64
16|0=   1:F X 1   =     15
                    ------
                    14,671
```

The largest number you may represent in one byte is >FF or decimal 255. The largest value in a word (two bytes) is >FFFF or 65,535. Enough on numbering systems for now, we will cover minus numbers (two's compliment) and additional points as we encounter them in programs.

Assembler Instruction Syntax.

Like every computer language there are certain rules we must follow for inputting instructions. Unlike BASIC, assembler will not give you a warning or error until you assemble the program. Here is the general syntax:

LABEL OPCODE OPERAND COMMENT

LABELs must begin in the 1st column and may be up to 6 characters long. One or more spaces follow. Next is the OPCODE. This is the actual instruction to be performed followed by one or more spaces. Next are one or more OPERANDs or data for the instruction to operate on followed by one or more spaces. Finally is an optional COMMENT which may extend to column 80. Each time we use a new instruction I will fully explain it.

Addressing Modes.

There are five general addressing modes and one special addressing mode used for assembler instructions. We will examine each one in detail as we encounter them in a program. There is one type of addressing we need to look at now. We are going to be operating on individual bits, bytes, and words of memory. Think of the computers memory as a series of consecutive small pieces of memory laid out end to end. We can address any single byte of memory by hanging a label on it but frequently we must address a byte of memory some distance from that label. Think of it like an array. To get to the 5th byte from the label we could say LABEL+4. We used 4 instead of 5 because we must start counting from 0. Think of it like OPTION BASE 0 in BASIC. A lot more on this later.

First Program.

I strongly recommend you enter the program manually by typing it in instead of just cleaning it up using TI-Writer or Editor/Assembler. The only way to gain experience programming is to practice. I have placed the program separately in DL3 to make it easier to read. Its name is PRO1.ASM.

Program explanation

These comments supplement the comments contained in the program itself. Any statement with an * in column 1 is a comment and you may enter anything else on that line. One fairly unique thing about the 9900 microprocessor in the TI99/4A is the ability to designate your own workspace registers anyplace in memory or more than one set at a time. Think of registers as 32 consecutive bytes of memory that are used as your scratch paper for calculations. Thirty-two bytes is of course 16 words of memory. Because this is a 16 bit (1 word) machine (something many of your friends cannot brag about) that gives us 16 registers to use for our computations. We place an R in front of the number to designate that we are refering to a register. For example, R0 is register zero and R15 is register fifteen (really the 16th word of memory because we started counting with zero). Here is the detailed explanation of the program:

DEF START

DEFines the entry point of the program so the computer may find it. Places the name START in the Reference/Definition table. More on this next time.

REF VSBW,VMBW

REFerence refers to console routine the program will use. In the advanced tutorials we will create our own utilities.

WSREG BSS >20

WSREG is the label we decided on for our workspace registers. Could have been any label 1 to 6 characters long. Block Starting Symbol (BSS) sets aside a block of memory, unititialized for use as our workspace. >20 means set aside >20 or 32 bytes (16 words) for R0 to R15.

```
X     DATA 10
```

Initializes one word of memory to 10 (>000A). Hangs the label X on that word.

```
START LWPI WSREG
```

Load Workspace Register Immediate (LWPI) tells the computer to use the 32 bytes of memory for our workspace which begin at WSREG. START is the entry point (beginning) of our program.

Here is the program that goes with the previous tutorial. It is just a short routine that shows how to put characters on the screen in assembler.

```
*****************************************
*
* This is the first program for the
* beginner assembler tutorial.
* It clears the screen, adds two numbers
* together and displays the sum in the
* center of the screen.
* Here is what it would look like in
* Extended BASIC:
*
* 10 CALL CLEAR
* 20 X=10
* 30 Y=27
* 40 X=X+Y
* 40 X=X+Y
* 50 DISPLAY AT(12,15):X
* 60 END
*
*****************************************
* This part of the program is the initialization
*
      DEF  START    the program name is START
      REF  VSBW,VMBW    console routines we are going
                    to use
WSREG BSS  >20      sets aside a block of 32 bytes for
                    use as workspace registers
X     DATA 10     could have used >A instead of 10 (like
                    X=10)
Y     DATA 27     (Y=27) could have also said Y EQU 0027
TEN   DATA 10
ANS   DATA 0    word to put answer in init to 0.
* Program begins here
*-------------------------------*
START LWPI WSREG      load workspace pointer immediate.
                    point to our workspace.
* Clear the screen
      CLR  R0       clears R0 to zero (beginning of
                    screen image table)
      LI   R1,>2000     load immediate R1 with >2000.
                    VSBW routine writes the left byte
LOOP  BLWP @VSBW      byte in R1 always.  in this case
                    >20 or 32 or SPACE char
      INC  R0       add 1 to R0
      CI   R0,767       compare immediate R0 to 767
      JLE  LOOP     if it is less than or equal JMP
                    (GOTO) LOOP
* Add the numbers together and convert to ASCII
*-------------------------------*
      A    @Y,@X      adds X to Y and places result in X
      MOV  @X,R6      move what is at X to R6
      CLR R5      clear R5
      DIV  @TEN,R5      divides 10 into 37.  quotient in
                    R5, remainder in R6
      AI   R6,>30 ADD     immediate ASCII offset to R6
      MOV  R6,@ANS      move contents of R6 to the word
                    ANS
      MOV  R5,R6      move contents of R5 to R6
      CLR  R5       clear R5
      DIV  @TEN,R5      divide 10 into R5, R6
      AI   R6,>30       add immediate ASCII offset >30 to
                    R6
      SLA  R6,8     shift left arithmetic R6 8 bits.
MOVB R6,@ANS      move MSbyte R6 to R1
*-------------------------------*
* Display on the screen at row 12 column 15
      LI   R0,366       position on the screen is 366
      LI   R1,ANS       load R1 with the address of ANS
      LI   R2,2     two bytes to write
      BLWP @VMBW
      JMP  $      prevents the program from ending so
                    you may see the result
*-------------------------------*
* Return to the calling program
      CLR  @>837C       clear the status byte
      LWPI >83E0      load GPL workspace registers
      B    @>0070       branch to the calling program
      END
```

This is a tutorial on using SBUG to debug and "step through" Assembly language programs. The example program is in the next note.

Using SBUG to DEBUG your Assembly Programs

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

As I have already mentioned there are several very good debug programs available for the TI. Here they are briefly. DEBUG is the program supplied with your editor assembler cartridge. It has many helpful features particularly the ones that allow you to set CRU bits, move memory, and compare memory. My main criticism of DEBUG is its inability to single step through instructions. BUGOUT written by Gregg Wonderly is also very good. It offers multiple fields of information on the screen at the same time. Many advanced features like dumping to disk. Without a doubt the most advanced that I have used. My main criticism is that it maps the VDP to text mode which interferes with VDP mapping for the screen image table and color table if you need to inspect these locations in your program. The one I use the most is SBUG. This is the super debugger released by TI to users groups after the pull out.

SBUG allows you to operate from graphics or bit map mode, single step, and output to printer or disk. Let us get down to how to use it. First you should print out the instructions supplied with SBUG on the disk. Since this is a display/variable 80 file you may print it out with TI-Writer or Editor/Assembler editor. This help file covers all of the instructions vailable. I intend to cover only the ones of greatest interest to us. It is important to know where your assembly program loads into memory. With some few exceptions which will not be discussed now, your program will load into high memory expansion at >A000. That is the first address in high memory. You can make your program load in other places but that is beyond the scope of this tutorial. SBUG is what we call relocatable object code. It loads on top (not over) of the last program entered. In other words, if your program is 500 bytes long it would load from >A000 to >A500. SBUG when loaded will load beginning at A502.

I have included a brief program in this tutorial which puts characters on the screen. We will use it to illustrate how to use SBUG. Go ahead at this time and type it in using the editor, save it, load the assembler except this time at the LIST FILE NAME prompt enter your printer description. Remember if you are using a PIO printer it should be entered with a period following PIO. At the OPTIONS prompt enter RLS. R to tell the assembler that you used R in front of your registers in the source code, L to give you a source listing to your printer, and S to give you a table of the symbols you used. This source list is necessary for using SBUG effectively. Let us examine the source listing columns or a moment.

Here is an example line from the source listing:

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

Here is what that tells us. 0049 means this statement is number 0049 in our source code. That means if we received the message invalid register in line 0049 when we were assembling the code we could look at line 0049 and see what was wrong with R1. The

next field over is the most important one to us when using SBUG. This gives us the exact location in memory where this instruction resides. If this was the first program you loaded into memory then this instruction would reside at >A06C. Remember I told you that the program loads at >A000 by default? In other words >A000 plus >006C=>A06C. We will come back to this field shortly. The next field >0201, is the machine language mnemonic for Load Immediate (LI). Look on page 5 of your blue editor/assembler card. See the opcode >0200? This tells us the instruction for this OPCODE is LI. The 1 in >0201 tells us to load immediate R1 with the value in the next word of memory. In this case Load Immediate R1 with >5500. See how easy that is? For a detailed description see section 15.4 in the E/A manual. We are ready to get down to business. First load the object code for the program I have given you into the computer. You should *always* load your program before you load SBUG. Next before you press enter after you have loaded the demo program load SBUG.

One note here. As long as you are not using the Extended BASIC cartridge to load with, it is much faster to load the compressed version of SBUG or SBUGC as it is listed on your disk. Both do the exact same job. Press enter and for the program name enter SBUG. The SBUG title screen should come up. The message on the screen will ask you if you are using a bit map screen? Enter N at this prompt. Next you will be asked for your list device. Enter your printer description here (you could enter DSK1.FILENAME instead if you wanted output to disk). The input prompt is a '.'. At the prompt enter L to turn off the list device.

Our program runs from SBUG's control so we need to set up SBUG to run our program so we may interact with it. To do this we must tell SBUG where the first executable instruction (entry point) to our program is and what address we are using for a workspace. You may remember there are three hardware registers used by the 9900 CPU in the TI99/4A. These are the program counter, workspace register, and status register. The program counter always points to the next instruction to be executed in memory. The workspace register points to the current 32 bytes (16 words) of memory we are using for workspace. The status register contains the current status of the computer as a result of the last instruction executed. Given that, we now need to find the entry point to our program.

There are two basic ways to do this. Perhaps the easiest is to look on the source listing at the label (in the example program START) we placed in the DEF statement at the beginning of the program and get the address directly from the source listing. The alternate way is to use the M command of SBUG to examine memory. The memory we need to examine is the Reference/Definition (REF/DEF) table approximately from >3F30 to >4000. This table lists the entry point to the program and the utilities we referenced.

Here is how to examine this table.

M 3F30,3FFF <enter>

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

3F30=5354 4152 5420 START

3F36=A058 ....

Press FCTN[X] to abort the listing or any command. Since labels may be up to 6 characters (bytes) long, the seventh and eighth bytes contain the program entry point. In this case >A058. This should be the same address on the source listing. *Remember* SBUG always displays and only accepts HEX numbers. Next we need the workspace address. We obtain that from the source listing. In this case it is >A05C. We get this by finding the label for our workspace, in this case WS. We look at the second column and there is the beginning of the workspace.

o

Aidkey

© by Arto Heino, 1987

If you get a copy of this somehow please send the Author $10.

ARTO HEINO
TH 35/8 GUERNSEY AVE.,
MINTO , 2566 ,
N.S.W. , AUSTRALIA

REQUIREMENTS-XB/32K/DISK

This program is intended for the Extended BASIC programmer who needs aid!

Loading Aidkey

```
CALL INIT
CALL LOAD("DSK1.AIDKEY")
CALL LINK("AIDKEY")
```

The cursor should now be changed,this means Aidkey is now ready when you need it.

Using Aidkey

You can initiate Aidkey while:

In command mode
In Edit mode
Listing to screen
Program is running

Just press FCTN[7]

Caution

Do not create a character while your program is running.

If you use the characters 48-90 in your program, you will not be able to read the menu.

Make sure you put the correct device name.

Aidkey menu

The Menu is actually a window you can move around using FCTN[D,S,X,E]

1. CREATING CHAR
   This function will put a HEX string 16 characters long after the cursor in Edit mode. To move character window around use FCTN[D,S,X,E]. To move cursor use D,S,X,E To change ON/OFF press SPACE bar. Press ENTER to return to menu
2. RLE/BIT SCREEN DUMP
   This function is useful only if you have the MAX/RLE disk. It will do a bit sceen dump so you can use your BASIC screen with Graphx or TI-Artist or as a file for another computer.
3. PRINTER SCREEN DUMP
   This will bit dump your screen to a EPSON compatable printer.
4. TURN OFF AIDKEY
   It will return your cursor to normal and FCTN[7] will be inactive. To activate type in command mode:
   CALL LINK("AIDKEY")
5. RETURN TO BASIC
   This will return you to where ever you were before you pressed AID. If you created a character, your string of HEX will be printed on your line

o

From the Illawarra Regional Group

by Lou Amadio

Approximately 20 persons attended the July meeting. Those who saw the PRBASE demo were probably wondering why it did not work as intended. If you recall, the computer locked up, with an interesting display of graphics, at the end of the initial data disk scan on start up. Geoff discovered the problem, which turned out to be a faulty data disk, later that week. Using a sector editor, the faulty disk was compared to a backup of the same data base. The editor showed that there was an odd character in one of the bytes in sector zero. When we attempted to correct the fault, the disk would not accept the edit. So scratch one disk. The moral here is to always backup your work, particularly if you have put a lot of time into it.

George, Rolf and myself went to the August TISHUG swap and sell meeting. There were quite a few bargains in both hardware and software. For those wanting to expand to disk drives, there were some very attractively priced items. We are looking forward to the next swap/sell meeting.

Our August meeting was held in the home of Phil Thompson as he forgot to get the key for the school. There was a smaller turnout than July but Rolf brought along his Geneve which caused interest for all those familiar with DOS in its various forms. Thanks to Phil for allowing us to use his home.

Programming Tips

This program first appeared about 4 years ago. It needs Extended BASIC. What does it do?

```
100 CALL CLEAR
200 CALL SCREEN(2)
300 CALL MAGNIFY(2)
400 FOR X=1 TO 26
500 CALL SPRITE(#X,64+X,INT(X/2)+3, 10,10,20 ,104-8*X)
600 NEXT X
700 GOTO 700
```

Experiment with the values in line 500.

PRBASE Print Tip

From Micropendium, August 1987.

You may encounter problems when trying to print out a report. The problem is caused by the way that the data disk is initialised in that there are >E5's written to all the sectors. When you enter PIO for your printer name, >E5 is attached to the printer name and causes an error. There are two ways around this:

1) Enter your printer name then press FCTN 3 to clear the rest of the field, or
2) Enter PIO.EC. Apparently the ".EC" is ignored and the printer behaves normally.

TI-Writer Tips

Find String (FS) in TI-Writer can also be used with a column specification, eg:

2 15/debits/ will look for the string "debits" in columns 2 through to 15 inclusive.

If you have a tabular document, do not forget to deactivate Word Wrap (CTRL[O]) prior to using Replace String (RS), as the automatic Reformatting feature of TI-Writer will make a disaster of your table.

Lastly, if your printer does not have a slashed zero, use the following Transliteration:

".TL 48:48,8,47"

It prints a normal zero (48) then backspaces (8) and prints a slash (47).

Extended BASIC Colours

(Adapted from Micropendium)

I have long wished for a means of permanently changing the default colours in Extended BASIC from the black on cyan provided by TI. The following program will do just that. You may simply load and run it, or, if you save it as "LOAD" on DSK1, then it will auto-execute when you select XBASIC.

The program requires 32K memory expansion. The colours are set in line 110, B for background and F for foreground. Check page III-6 in your User's Reference Guide for other high resolution colour combinations.

Note that if you try to LIST the program after RUNning it, you will get an error message. This is because line 170 automatically erases the program from memory (like entering NEW). Line 170 could also be used in any program where this feature is desirable. If you do use it in another program, do not forget to CALL INIT first.

If you have a RAMdisk, save it on your boot drive (ie the one with the MENU program), as "XBASIC", then make it one of your menu selections. You will find that it will get you into Extended BASIC quickly without looking to drive 1 first.

```
100 CALL CLEAR
110 B=5 :: F=16
120 C=16*(F-1)+(B-1)
130 CALL INIT :: CALL LOAD(9984,C,C,C,
    C,C,C,C,C,2,0,7,15+B,4,32,32)
140 CALL LOAD(9999,48,2,0,8,0,2,1
    ,39,0,2,2,0,8,4,32,32,36,2,0,8,8,4)
150 CALL LOAD(10021,32,32,36,2,0,8
    ,16,4,32,32,36,2,0,8,24,4,32,32,36 ,4,91)
160 CALL LOAD(-31804,39,8)
170 CALL LOAD(-31952,255,231,255,231)
```

From the Bulletin Board

MAIL TO : ALL
MAIL FROM : RAJAH

To whoever bought my MBX at the August meeting, I forgot to give you the Power Supply. Please ring me on 449-4273 and I will arrange its handover-sorry about that.

MAIL TO : ALL
MAIL FROM : DOBELL

I need a copy of the documentation to the extended busines graphs program. Is there any one out there with the documentation who can help me?

DOBELL

MAIL TO : ALL
MAIL FROM : GOWFAR

This is just a short message to let you all know that SCI-FI BBS is still alive and kicking and available at 300/300, 1200/1200 or 1200/75, on 8 data bits, null parity, 1 stop bit or 7 data bits, even parity, 1 stop bit (VIATEL setup), on (02)646-4865, 24 hours a day (and other "times" too). Why is SCI-FI BBS different to others? Have a look at my ON-LINE RPG, "GAMEROOM" and see for yourself. This RPG was the FIRST, in AUSTRALIA to incorporate users playing against (and killing if you do not like your opponent), other USERS as opposed to just playing the computer. You have the choice of PROVING yourself against another user, raiding his castle (and stealing his money), jousting a user or fighting any of the over 200 monsters available (which are straight from the D and D manual). In addition, once you have played and won enough money, there is the chance of making more (or LOSING it), by going to the CASINO and playing games of chance. Please note that NEW users access is updated within 24 hours, so it takes till the SECOND call before you can get into the GAMEROOM. On the BBS are areas dealing with SCI-FI, BOOK REVIEWS, MAILROOM, D and D (adventure type games - RPG's) and the odd "HIDDEN" section. Want to know more? Then ring the BBS and enrol, then ring back next day and SEE!

EPROM Data:

Pin connections and programming information

by Ben Takach

EPROMS

Table 1.

Type TMS-	Organisation (words x bit	Size kb	Addr.Lines LSB-MSB	No.	Address Tot. (DEC.->HEX.)	Package J=.600"
2716	2k x 8 bits	16	A0 - A10	11	2k=0-2047 >0->7FF	24 pin
2732	4k x 8 bits	32	A0 - A11	12	4k=0-4095 >0->FFF	24 pin
2764	8k x 8 bits	64	A0 - A12	13	8k=0-8191 >0->1FFF	28 pin
27128	16k x 8 bits	128	A0 - A13	14	16k=0-16383 >0->3FFF	28 pin
27256	32k x 8 bits	256	A0 - A14	15	32k=0-32767 >0->7FFF	28 pin
27512	64k x 8 bits	512	A0 - A15	16	64k=0-65535 >0->FFFF	28 pin
2532	4k x 8 bits	32	A0 - A11	12	4k=0-4095 >0->FFF	24 pin
2708	1k x 8 bits	8	A0 - A9	10	1k=0-1023 >0->3FF	24 pin

TYPICAL DATA - EXTRACTS

- Maximum Access/ Minimum Cycle times dependent on the type. It will vary between 150 and 450 ns.

- Power down mode reduces the maximum active current from 150ma to 35ma in case of conventional N-channel Silicon devices and from 40ma to 500mic.a (TTL-level inputs) or 250mic.a (CMOS-level inputs) by applying a high TTL signal to the E-(bar) pin. In this mode all outputs are in the high impedance state.

- Erasure. EPROMS may be erased by exposing the chip to to high intensity UV light (wavelength 2537 angstroms). The min. expos. dose is 15W/s/sqcm. A 12mW/sqcm filterless UV lamp will erase the device in apr. 25 minutes. The chip should be placed 25 mm below the lamp. After erasure all bits are in the high state. Sunlight also contains the same wavelenght UV radiation. The window of programmed EPROMS should be covered with a non transparent label to prevent errasure by ambient light.

- Pin compatibility. CMOS EPROMS are pin compatible with the conventional type EPROMS as well as ROM-s.

- General remarks and handling CMOS chips. CMOS EPROMS use less current and lower programming voltage, but are not necessarily faster than the Silicon-gate types. CMOS devices are sensitive to electrostatic charge, and therefore should be handled only in static-free environement.

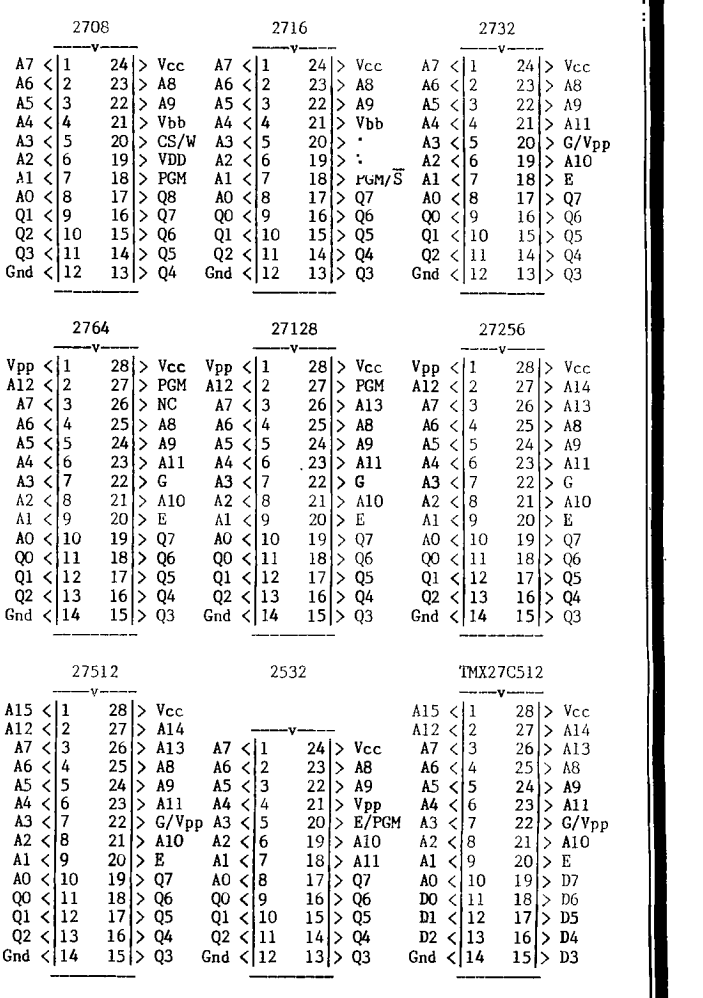
EPROM PROGRAMING VOLTAGE CHART

Table 2.

Type No.	Manufacturer	Volts
2708	AMD, Motorola, National, Intel, Texas Instr.	25
2716	Texas Instr., Mostek, Fuji	25
2532	TI (TMS), Motorola (MCM), Hitachi, National, OKI	25
2732	AMD, Fujitsu, NEC, Hitachi, Intel, Mitsubishi National	25
2732A	Fujitsu, Intel, NEC, AMD, Texas Instr.	21
TMS2564	Texas Instr.	25
2764	Intel, Fairchild, OKI, NEC, Hitachi, Fujitsu, Toshiba, AMD, Texas Instr.	21
2764A	Intel, AMD	12.5
TMS27C64	Texas Instruments	12.5
MCM68764 / ...6 / ...'64	)Motorola	25
27128	Intel, AMD, Fujitsu, NEC, Toshiba, Texas Inst.	21
TMS27C128	Texas Instruments	12.5
M5L27128K / M5M27C128K	)Mitsubishi	21
HN4827128G	Hitachi	12.5
27256	Intel, Atmel	12.5
27C256	National, Texas Instruments (TMS)	12.5
27256	Fujitsu	12.5
27C256	Fujitsu	21
TMM27256D	Toshiba	21
HN27256G	Hitachi	12.5
M5L27256K	Mitsubishi	12.5
27512	Intel, AMD	12.5
27513	Intel	12.5
TMX27C512	Texas Instruments	12.5

- CMOS EPROMS use 12.5v power source to burn in the program (Vpp) The programming signals are TTL level. After erasure all bits are in logic '1' state. During programming logic '0's are burned into the desired locations. The programming voltage, Vpp is critical. It must be maintained at 12.5V (+-0.5V). Higher programming voltage will destroy the CMOS Eprom.

CONNECTION DIAGRAMS



PIN NOMENCLATURE

A0 - A15	Address Inputs	Q0 - Q7	Outputs
		Vcc	+5 V Power S.
$\overline{E}$	Chip Enable (Power Down)	Vpp	12.5 or 21 or 25V Power S.
Gnd	Ground, Common	$\overline{G}$	Output Enable
		NC	No Connection
$\overline{PGM}$	Program		

STORING EPROM PROGRAMS ON DISK

The EPROM programs read by the Mechatronic EPROMmer can be saved on disk. These will be on 33 sector program files.
16k, 32k and 64k EPROM programs fill up only some of the 33 sectors. The remaining sectors will remain blank:
16k EPROMS fill up 8 sectors,
32k EPROMS fill up 16 sectors,
64k EPROMS fill up 32 sectors.
The save command saves about 8k of the buffer. This is adequate for the above 3 EPROM types (e.g. 8k x 8bits= 8kbytes), but the 128k series has to be saved in two steps (e.g. DSKn.FILE1 and DSKn.FILE2).

INSPECTING AND/OR EDITING EPROM PROGRAMS

The EPROM program files can be best inspected by one of the disk sector editor programs. The location of the wanted program is recorded on the file header sector in the form of a 3 byte block link.
Block link translation:
the 3 Bytes of block link____ 12 34 56
to be rearranged in the order of: 412 563
The first 3 hex digits indicate the first sector of the file, the last 3 hex digits tell the number of sectors following.
Example: 20 F0 01 = >020 START + >01F sectors ... >020 - >03F

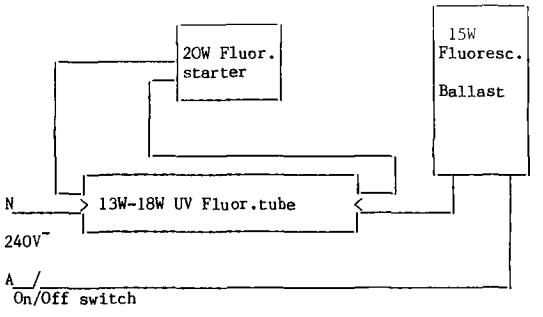
```
   e.g.. >020 +
         >01F
         -----
         >03F
```

The CHANGE BUFFER option provides an on screen program edit facility. This however is a rather slow process, one has to single step through the buffer starting with the nominated starting address at 4 hex digit steps. Using one of the disk sector editors, one can step through the program, and view one full sector at the time.

CONSTRUCTION OF AN INEXPENSIVE EPROM ERASER

Photocopiers, germicide lamps and UV-light inspection units generally use a 15W UV fluorescent tube. An inexpensive erasing device can be made using such tubes. Erasure times are around 15 minutes.
CAUTION. MAINS POWER OPERATED DEVICES ARE LETHAL IF WIRED INCORRECTLY. CONSTRUCTION SHOULD ONLY BE ATTEMPTED BY PERSONS QUALIFIED TO DO SO. HOME MADE UNITS SHOULD BE CHECKED BY A LICENSED ELECTRICIAN.

The wiring diagram:



Any enclosure of appropriate dimensions may be utilised to accomodate the components.

EPROM OPERATION MODES

Function (pins)	Mode					
	Read	Output disable	Power down (stand by)	Fast Progr.	Progr. verif.	Inhibit programming
$\overline{E}$ (20)	LO	HIor LO	HI	LO	LO	HI
$\overline{G}$ (22)	LO	HI	HI or LO	HI	LO	HI or LO
$\overline{PGM}$ (27)	HI	HI	HI or LO	HI	HI	HI or LO
Vpp (1)	Vcc	Vcc	Vcc	Vpp	Vpp	Vpp or Vcc
Vcc (28)	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
Q0-Q7	Q	HI-Imp.	HI-Imped.	D	Q	HI-Imped.

Comparison of Computers

by Shane Ferrett

Comparison Charts
Based on maximum configuration of each machine

Machine	CPU RAM	VDP RAM	Floppy drives	Hard drives	Mouse port
TI99/4A	32Kb	16Kb	4|DS/QD	3|134Mb	No
Myarc 9640	2024Kb	192Kb	4|DS/QD	3|134Mb	Yes
IBM PC	640Kb	64Kb	2|DS/DD	2|30Mb	No
Atari ST 520	960Kb	64Kb	2|DS/DD	1|40Mb	Yes
Atari ST 1040	960Kb	64Kb	2|DS/DD	1|40Mb	Yes
Apple 512 Mac	448Kb	64Kb	2|DS/DD	1|80Mb	Yes
Apple Mac plus	4032Kb	64Kb	2|DS/QD	1|80Mb	Yes
Amiga	896Kb	128Kb	2|DS/QD	1|80Mb	Yes

Note: 1 DS/DD Drive=360Kb; 1 DS/QD Drive=720k

Machine	Video output	Fixed keyboard	RS232 ports	PIO ports	$1000's
TI99/4A	Composite/Mod	Y	4	2	.4
Myarc 9640	Composite/RGB	N	no limit	no limit	1.1
IBM PC	TTL RGB	N	2	1	2.0
Atari ST 520	RGB/mono	Y	1	1	1.2
1040	RGB/mono	Y	1	1	1.5
Apple 512 Mac	mono only	Y	2	1	2.0
Apple Mac plus	mono only	Y	2	2	4.5
Amiga	RGB/mono	Y	1	2	2.2

Machine	Low-Res	Colours	High-Resolution	Colours
TI99/4A	32x24	16	256x192	16
Myarc 9640	256x192	256	512x424	16(512)
IBM PC	160x100	16	320x200	4
Atari ST all	320x200	64(512)	640x400	2(512)
Apple Mac all	512x384	2	512x384	2
Amiga	320x200	64	640x200	16(4096)

Note: Number in () is total pallet

Machine	Text	Quality	Output type
TI99/4A	40x24	2/3	Composite/modulator colour
Myarc 9640	80x26	3/4	Composite and RGB colour/ monochrome
IBM PC	80x25	2/3	TTL RGB colour/monochrome
Atari ST all	80x48	3/4	Only Atari monitors
Apple Mac all	80x26	4	Built in
Amiga	80x25	1	RGB colour/monochrome

Output Ratings: 1-Poor; 2-Good; 3-Very good; 4-Excellent

Tips from the Tigercub #49

by Jim Peterson

Distributed by Tigercub Software to TI99/4A 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 120 original programs in BASIC and Extended BASIC, available on cassette or disk, Now reduced to just $1 each, plus $1.50 per order for cassette or disk and postage and handling. Minimum order of $10. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Colour Programming Tutor, are no longer available on cassette. Descriptive catalogs, while they last, $1 which is deductable from your first order.

Tigercub Full Disk Collections, reduced to $5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

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

NUTS & BOLTS DISKS

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended BASIC. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS #1 has 100 subprograms, a tutorial on using them, and 5 pages of documentation. NUTS & BOLTS #2 has 108 subprograms, 10 pages of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pages of documentation. Now just $15 each, postpaid.

Tips from the TIGERCUB

These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions. TIPS (Vol. 1) contains 50 original programs and files from Tips newsletters No. 1 through No. 14. TIPS VOL. 2 contains over 60 programs and files from Nos. 15 through 24. TIPS VOL. 3 has another 62 from Nos. 25 through 32. TIPS VOL. 4 has 48 more from issues No. 33 through 41. Now just $10 each, postpaid.

```
*****************************
*        NOW READY          *
* TIPS FROM TIGERCUB VOL.5  *
* Another 49 programs and   *
* files from issues No. 42  *
* through 50. Also $10 ppd  *
*****************************
```

TIGERCUB Care disks #1,#2,#3 and #4.

Full disks of text files (printer required). No. 1 contains the Tips news letters #42 through #45, etc. Nos. 2 and 3 have articles mostly on Extended BASIC programming. No. 4 contains Tips newsletters Nos. 46-52. These were prepared for user group newsletter editors but are available to anyone else for $5 each postpaid.

Another one for the teachers and their students -

```
100 DIM K$(17):: DIM B$(185) :: DIM C$(18,2)
110 GOTO 150
120 SET,CH,K,S,K$(),J,B$(),C$(J,1),Z$,
    Y$,X$,Q,X,Y,W$,PL$,A,Q$
130 CALL CLEAR :: CALL COLOR :: CALL SCREEN :: CALL
    CHAR :: CALL KEY :: CALL PLURAL :: CALL SOUND
140 !@P-
150 CALL CLEAR :: FOR SET=0 TO 14 :: CALL
    COLOR(SET,2,8) :: NEXT SET :: CALL SCREEN(5):: FOR
    CH=127 TO 129 :: CALL CHAR(CH,"0"):: NEXT CH
160 CALL CHAR(64,"3C4299A1A199423C"):: DISPLAY
    AT(3,2):"PLURAL ENDINGS Version 1.1" :: GOSUB 250
170 DISPLAY AT(5,1):"@ Tigercub Software for free
    distribution.  No price or copying fee may be
    charged." !written by Jim Peterson 20 Nov.  87
180 DISPLAY AT(12,1):"DO YOU WANT TO:":" (1)TAKE A
    TEST" :" (2)FIND PLURALS":"":"  TYPE 1 OR 2"
190 ACCEPT AT(16,15)VALIDATE("12"):Q :: IF Q=1 THEN
    DISPLAY AT(12,1):"":"":"":"":"" :: GOTO 240
200 DISPLAY AT(3,1)ERASE ALL:"This program has been
    pro-  grammed with all the rules  for forming
    plurals, but  there are quite a few irreg-"
210 DISPLAY AT(7,1):"ular plural forms in Englishso the
    answer it gives may not always be right."
220 DISPLAY AT(15,1):"Your word?" :: ACCEPT
    AT(15,12)VALIDATE(UALPHA):W$ :: CALL
    PLURAL(W$,PL$,A)
230 DISPLAY AT(17,1):"The regular plural form is";PL$
    :: DISPLAY AT(20,1):"      I"&SEG$(K$(A),6,255)&
    RPT$(" ",28):: GOTO 220
240 DISPLAY AT(12,8): "GETTING READY...." :: GOTO 440
250 CALL KEY(5,K,S)
260 K$(1)="No, if the word does not end in
    E,F,H,N,S,X,Y or Z just add S"
270 K$(2)="No, if the word ends in IFE, change it to
    IVES (FIFE is   an exception!)"
280 K$(3)="No, if a word ends in E but not FE, just add
    S"
290 K$(4)="No, if a word ends in F,    (except EF or
    FF) change it to VES"
300 K$(5)="No, if a word ends in CH or SH, add ES"
310 K$(6)="No, if a word ends in H but not CH or SH,
    just add S"
320 K$(7)="No, if a word ends in S, X  or Z, add ES"
330 K$(8)="No, if a word ends in AY,   EY, OY or UY,
    just add S"
340 K$(9)="No, if a word ends in Y not preceded by a
    vowel, change the Y to IES"
350 K$(10)="No, if a word ends in N but not in MAN,
    just add S"
360 K$(11)="No, if a word ends in MAN,  change it to
    MEN"
370 K$(12)="No, if a word of Latin      origin ends in
    US, change   it to I"
380 K$(13)="No, the plural of this word is the same as
    the singular"
390 K$(14)="No, some words ending in UM change the UM
    to A"
400 K$(15)="No,if a word ends in EF or  FF, just add S"
410 K$(17)="No, many kinds of fish have the plural the
    same as the  singular"
420 RETURN
430 !@P+
440 DATA CAT,DOG,COW,MONKEY, PARROT,WHALE,PLATE,CUP,
    FORK, SPOON,DISH,WATCH,HOOK,PEA,APPLE
450 !@P-
460 DATA CUFF,CLIFF,SKIFF,RUFF,CLEF,
    CHEF,CHIEF,DONKEY,COMIC
470 DATA LIMB,HAND,SOLO,SEA, CLOUD,ROAD,BOY,GIRL,
    CORNCOB, ARC,TREE,PIG,TANK,BALL,DRUM,
    GUN,HARP,CAR,BOOT,SHOE
480 DATA MOTH,SLOTH,MYTH,LATH,DEATH
490 !in the next line, key in CTRL B before each word
500 DATA CARP, MACKEREL, SUNFISH, PIKE, SALMON
510 DATA SAW,WINDOW,HOUSE,BAY,GUY,TOY, GOAT,CAN,AUTO,
    TRUCK,BRA
520 DATA WIFE,LIFE,KNIFE,LOAF,CALF, HALF, SCARF,ELF,
    LEAF,WOLF,PELF, SELF,WHARF,HOOF
530 DATA GAS,MISS,KISS,LASS,
    TRUSS,BOSS,GLASS,CLASS,IRIS
540 DATA LATCH,WITCH,BATCH,ROACH,LEECH, PEACH,ARCH,
    BRANCH, BIRCH,MULCH,BROOCH,POUCH
550 DATA SASH,CRASH,FLASH,VARNISH, WISH,FETISH,
    RADISH,BUSH,RUSH
560 DATA BAY,BOY,DAY,RAY,TRAY,HIGHWAY, GUY,ALLOY,BUOY,
    KEY,MONKEY,TURKEY
570 !in the next line, key FCTN V before each word
580 DATA RADIUS, FUNGUS, CACTUS, GLADIOLUS, OCTOPUS
590 DATAMAN,WOMAN,FIREMAN,POLICEMAN, FOREMAN,
    CHAIRMAN,POSTMAN, CHARWOMAN,MIDWIFE
600 DATA LADY,CANDY,BUDDY,BABY,ORGY, DOILY,PONY,
    PUPPY,STORY,POSY, PARTY,COVY
```

```
610 DATA TALLY,ARMY,NAVY,FOLLY,PANSY, ARRAY
620 DATA BOX,FOX,TAX,WAX,SEX
630 DATA SPA,GURU,POTATO,TOMATO,ZEBRA, SKI,OPERA,
    CIRCUS,PLUS,MINUS, BUS
640 !in the next line, key CTRL , before each word
650 DATA PANTS, SCISSORS, SQUID, DEER, SHEEP, SWINE,
    MOOSE, BISON, GROUSE, SERIES, STAIRS
660 !in the next line, key CTRL A before each word
670 DATA DATUM, MEDIUM, CURRICULUM, PLANETARIUM,
    SOLARIUM
680 DATA I,WE,HE,THEY,SHE,THEY,THIS, THESE,THAT,
    THOSE,CHILD, CHILDREN,TOOTH,TEETH
690 DATA MOUSE,MICE,LOUSE,LICE,GOOSE, GEESE,OX,OXEN,
    FOOT, FEET,CRISIS,CRISES,APPENDIX, APPENDICES
700 DATA ROOF,ROOFS,FIFE,FIFES,PROOF, PROOFS,THIEF,
    THIEVES
710 FOR J=1 TO 185 :: READ B$(J):: NEXT J
720 RESTORE 680 :: FOR J=1 TO 18 :: READ
    C$(J,1),C$(J,2) :: NEXT J
730 FOR J=1 TO 185 :: Z$=Z$&CHR$(J):: NEXT J :: Y$=Z$
    :: X$=SEG$(Z$,1,18):: DISPLAY AT(12,1):""
740 RANDOMIZE :: Q=INT(203*RND+1):: IF Q<186 THEN 770
750 X=INT(RND*LEN(X$))+1 :: Y=ASC(SEG$(X$,X,1))::
    X$=SEG$(X$,1,X-1)& SEG$(X$,X+1,255) :: IF LEN(X$)=0
    THEN X$=SEG$(Z$,1,18)
760 W$=C$(Y,1):: PL$=C$(Y,2) :: A=16 :: K$(16)="No,
    this word has an        irregular plural form.  It
    is"&PL$ :: GOTO 790
770 RANDOMIZE :: X=INT(RND*LEN(Y$))+1 ::
    Y=ASC(SEG$(Y$,X,1)):: Y$=SEG$(Y$,1,X-1)&
    SEG$(Y$,X+1,255):: IF LEN(Y$)=0 THEN Y$=Z$
780 W$=B$(Y):: CALL PLURAL(W$,PL$,A)
790 DISPLAY AT(12,14-LEN(W$)/2):W$ ::
    DISPLAY AT(15,1):"Type the plural form" ::
    DISPLAY AT(18,1): "" ::
    ACCEPT AT(18,14-LEN(W$)/2):Q$
800 IF Q$=PL$ THEN CALL SOUND(50,523,5) ::
    DISPLAY AT(20,1):"":"":"" :: DISPLAY AT(20,11):
    "CORRECT!" :: DISPLAY AT(12,1):"" :: GOTO 740
810 CALL SOUND(200,110,5,-4,5)::
    DISPLAY AT(20,1):"":"": "" ::
    DISPLAY AT(20,1):K$(A) :: GOTO 790
820 PRINT K$(A):: GOTO 780
830 !@P+
840 SUB PLURAL(W$,PL$,A)
850 GOTO 880
860 Y$,W$,PL$,A
870 !@P-
880 Y$=SEG$(W$,LEN(W$)-1,2):: IF ASC(W$)=127 THEN
    PL$=SEG$(W$,2,LEN(W$)-3)&"I" :: A=12 :: SUBEXIT
890 IF ASC(W$)=128 THEN PL$=SEG$(W$,2,255):: A=13 ::
    SUBEXIT
900 IF ASC(W$)=129 THEN PL$=SEG$(W$,2,LEN(W$)-3)&"A" ::
    A=14 :: SUBEXIT
910 IF ASC(W$)=130 THEN PL$=SEG$(W$,2,255):: A=17 ::
    SUBEXIT
920 ON POS("EFHSXYZN",SEG$(W$,LEN(W$),1),1)+1 GOTO
    930,940,960,970,980,980,990,980,1000
930 PL$=W$&"S" :: A=1 :: SUBEXIT
940 IF SEG$(W$,LEN(W$)-2,3)="IFE" THEN
    PL$=SEG$(W$,1,LEN(W$)-2)&"VES" :: A=2 :: SUBEXIT
950 PL$=W$&"S" :: A=3 :: SUBEXIT
960 IF Y$="EF" OR Y$="FF" THEN PL$=W$&"S" :: A=15 ::
    SUBEXIT ELSE PL$=SEG$(W$,1,LEN(W$)-1)&"VES" :: A=4
    :: SUBEXIT
970 IF (Y$="CH")+(Y$="SH"[illegible]N PL$=W$&"ES" :: A=5 ::
    SUBEXIT ELSE A=6 :: G ·  950
980 PL$=W$&"ES" :: A=7 :: SUBEXIT
990 IF (Y$="AY")+(Y$="EY")+ (Y$="OY")+(Y$="UY") THEN
    PL$=W$&"S" :: A=8 :: SUBEXIT ELSE
    PL$=SEG$(W$,1,LEN(W$)-1)&"IES" :: A=9 :: SUBEXIT
1000 IF SEG$(W$,LEN(W$)-2,3) <>"MAN" THEN A=10 :: GOTO
    930 ELSE PL$=SEG$(W$,1,LEN(W$)-3)&"MEN" :: A=11 ::
    SUBEXIT
1010 !@P+
1020 SUBEND
```

Here's another tinygram -

```
100 CALL CLEAR :: CALL CHAR(47,"000000007C") ::
    DISPLAY AT(2,1): "TIGERCUB ONE-FINGER FIGURER"
110 DISPLAY AT(4,1):" Add and subtract with
    one":"finger while the other hand keeps track in a
    column -   you can type the minus sign without the
    shift key!"
120 ACCEPT AT(12,10)VALIDATE (NUMERIC,"/"):A$ :: ON
    ERROR 130 :: A=VAL(A$):: GOTO 150
130 ON ERROR 140 :: A=-VAL(SEG$(A$,2,255)):: RETURN 150
140 CALL SOUND(100,110,5,-4,5)::
    DISPLAY AT(18,1):"ERRONEOUS INPUT!" :: RETURN 120
150 T=T+A :: DISPLAY AT(18,1):"Total is";T :: GOTO 120
160 DISPLAY AT(18,1):"Total is";T
```

The new Super Extended BASIC offers CALL KEY input with validation. Now you can have it too. This subprogram will accept only one of the characters listed, ABCD in this case, and the value returned in K will be the position of the input in the validation string.

```
100 CALL KEYVAL(K,"ABCD"):: PRINT SEG$("ABCD",K,1)::
    GOTO 100
10000 SUB KEYVAL(K,V$)
10001 CALL KEY(0,K,S):: IF S=0 THEN 10001 ::
    K=POS(V$,CHR$(K),1):: IF K=0 THEN CALL
    SOUND(200,110,5,-4,5):: GOTO 10001
10002 SUBEND
```

CALL FLASH(L,R,C,T,K), where L is the number of DATA items, R and C are DISPLAY row and column, T is the flashing speed and J is the number of the item selected, will display options alternately until a key is pressed.

```
100 DATA FCTN 7=AID,FCTN 8=START OVER,FCTN 4=QUIT
110 CALL CLEAR :: CALL FLASH(3,1,8,15,J):: ON J GOTO
    120,130,140
120 PRINT "AID" :: STOP
130 PRINT "START OVER"::STOP
140 PRINT "QUIT"
10000 SUB FLASH(L,R,C,T,J):: FOR J=1 TO L :: READ
    M$(J): : NEXT J :: J=1
10001 DISPLAY AT(R,C):M$(J):: FOR A=1 TO T :: CALL
    KEY(0,K,S)
10002 IF S<>0 THEN SUBEXIT
10003 NEXT A :: J=J+1+(J=L)*L :: GOTO 10001
10004 SUBEND
```

Memory full..... Jim Peterson o

continued from page 27

The Forth Column

```
SCR #33
 0 ( SCREEN EDITOR 090CT84     )
 1 BASE->R
 2 0 VARIABLE ER     0 VARIABLE EC
 3 0 VARIABLE BLINK  0 VARIABLE OKEY  0 VARIABLE CC
 4 10 CONSTANT RL   50 CONSTANT RH  0 VARIABLE KC RH VARIABLE RLOG
 5
 6 : HCHR ( X Y CNT CH ---  )
 7   >R >R SCRN_WIDTH  * + SCRN_START  + R> R> VFILL ;
 8 : GCHR ( X Y --- ASCII ) ( COLUMNS AND ROWS NUMBERED FROM 0 )
 9   SCRN_WIDTH  * + SCRN_START  + VSBR ;
10
11 : CURSOR  ( n --- ) >R EC  ER  1 R>  HCHR ;
12
13 : RKEY BEGIN ?KEY -DUP 1 BLINK +! BLINK  DUP 60 < IF 30 CURSOR
14  ELSE CC  CURSOR ENDIF    120 = IF 0 BLINK ! ENDIF
15 -->
SCR #34
 0 ( SCREEN EDITOR 090CT84     )
 1
 2        IF ( SOME KEY IS PRESSED )    KC    1 KC +!  0 BLINK !
 3          IF ( WAITING TO REPEAT )   RLOG   KC   <
 4            IF ( LONG ENOUGH ) RL RLOG ! 1 KC ! 1 ( FORCE EXT)
 5            ELSE OKEY  OVER =
 6              IF DROP 0    ( NEED TO WAIT MORE )
 7              ELSE 1 ( FORCE EXIT )    DUP KC !   ENDIF
 8            ENDIF
 9          ELSE ( NEW KEY ) 1 ( FORCE LOOP EXIT )  ENDIF
10        ELSE ( NO KEY PRESSED) RH RLOG ! 0 KC !   0
11        ENDIF
12   UNTIL DUP OKEY ! DUP 31 >
13   IF DUP CURSOR ELSE CC  CURSOR ENDIF ;
14
15      R->BASE  -->
```

o

The Forth Column

Forth Forum <5>

by George L Smythe

This month I wrap up the discussion of the editor. In the Forum section I am presenting an extension to the 40 column editor which will allow auto-repeat of all functions. I still have not been able to figure out why this was not included in the original system, as it is part of the 64 column editor which I feel is not seriously useable. I hope to present later, a couple of modifications to the editor, as there are a few things that still bother me. I guess that I will never be totally satisfied with the present situation, but I think that that is also a function of the fact that I can do something about it. The other languages I have, BASIC, Assembler, and Logo, all have problems with the editor that I have overlooked because I am locked out of any software modifications. Luckily, the Forth editor is available to play with and modify to one's own personal tastes. If you agree that auto-repeat is important, you will want to modify the editor. If not, you may want to look at the modification as a learning tool. Whatever, the choice is up to you because only you know what you want. The flexibility is there!

Using the editor

Remember I mentioned that the system refers to the screens as BLOCKs? Well, the word that loads a screen into memory is BLOCK (n ---). The word BLOCK expects to find the screen number to be loaded on the stack, loads the screen into memory, and leaves an address (which we will not bother with now) on the stack. Since the naming of user defined words is the prerogative of the programmer, there is no standard name for the procedure which writes a screen into memory and marks it as updated. My definition is:

: BDU (n ---) BLOCK DROP UPDATE ;

BLOCK loads the screen number on the stack and leaves an address, DROP removes the unneeded address, and UPDATE sets a flag indicating that this particular screen should be written to disk when told.

All UPDATEd screens in memory can be written to disk without having to load additional screens into memory by using the word FLUSH (---). We therefore, can copy screens 30 to 34 from one disk to another by executing: '30 31 32 33 34 BDU BDU BDU BDU BDU', changing disks, and entering 'FLUSH', which writes the five UPDATEd screens in memory to the new disk. Before Doug Smith presented the club library with his three pass copier written in Forth, the single disk copiers used this idea to make backups. Unfortunately, this took 18 passes. For moving a few screens, however, this method is quite handy.

Last month we were working on screen #31. We accessed that screen and exited it by pressing 'FCTN[9]' (BACK). To reenter the screen we could type '31 EDIT', but there is an easier way. ED (---) will return you to the editor and place your cursor in the upper left hand corner of the screen you exited. Go ahead and try this and then we will continue.

Pressing 'FCTN[1]' deletes characters. This works the same as the editing function in BASIC. 'FCTN[2]' is the insert function, but it does not work like the similarly named BASIC function. This key combination moves the contents of the line to the right one space, starting where the cursor is placed. The advantage of doing things this way is the ease of programming the function. The disadvantage is that any characters that are on the end of the line will "fall off" if they exceed the 64 column limit. If this happens, there is no recovery, and the characters must be entered on the next line. Here, an 80 column display would really be handy, but if you are careful this will not happen. Type some letters in, move the cursor around with the arrow keys, and try inserting and deleting the characters to get familiar with these functions.

The easiest way to move to the next or previous screen is to use 'FCTN[4]' or 'FCTN[6]' respectively. Since you are editing screen #31, press 'FCTN[4]'. The drive will turn on and you will now be ready to edit the next screen, screen #32. To go back to screen #31 press 'FCTN[6]'. Notice that the drive was not necessary to access this screen. That is because screen #31 is in the disk buffer area we discussed earlier.

Entering 'FCTN[3]' or 'FCTN[7]' picks up information on the line the cursor is on. 'FCTN[3]' picks up the entire line, whereas 'FCTN[7]' picks up information on the line from the cursor to the end of the line. The information is placed in an area called "PAD". The contents of this area can be placed back on the line the cursor is on by entering 'FCTN[8]'. These functions can be used to move lines and editor and place your cursor in the upper left hand corner of the screen you exited. Go ahead and try this and then we will continue.

Pressing 'FCTN[1]' deletes characters. This works the same as the editing function in BASIC. 'FCTN[2]' is the --> (---). This word indicates that during a LOAD operation the compiler should continue by compiling the next screen. When this word is placed at the end of all screens in an application with the exception of the final screen, the application can be loaded by loading the first screen only. All following sequential screens will be loaded via the --> instruction. The word ;S (---) halts compilation of a screen if you have a need to do so before the end of screen. This word is often used when an author wants to place some remark statements at the end of a program without enclosing them in parenthesis or taking up compilation time.

There are a few other functions of the editor that you will find in the manual, but I think that these few commands combined with the ones presented last month will suffice as far as beginning to program using the available screens is concerned. Above all, experiment! What happens if you press a back arrow when the cursor is on column 1 of line 4? Can you wrap a definition around a line? I could list a number of other questions, but you will learn more and become more familiar with this editor by trying things and noting their consequences.

Next month I will start going over a conditional structure. This will be the BASIC equivalent of the IF..THEN construct. I will also quickly go over the numbering systems available. Forth allows not only decimal and hex, but also binary, octal, and any other one you can dream up.

The Forum

I mentioned last month that I would present an addition to our present editor which will allow auto-repeat. Well, following this section are a couple of screens that you will need to type in which will do just that. I have submitted this extension to the local BBSs, so if you do not feel like typing you may wish to download the file and send it through the FILTRAN program. Following are the easiest instructions I can think of to get this program up and running. I am assuming that this will be included in a BSAVE format on system disk. If not, the instructions are a bit more complicated. I will go through them only if requested. I do not know who wrote this extension, but if I can find out I would like to thank him/her.

Copy the present editor to another disk. Enter '34 32 SCOPY 34 CLEAR'. This will move the first screen of the present editor two screens lower. Now enter the two screens that follow. On screen #36, the word .CUR needs to be altered. Line 4 of that screen should read:

'ENDIF DUP EC ! SWAP DUP ER ! GOTOXY EC ER GCHR CC ! ;'

Also, on screen #38 the word 'KEY' should be changed to 'RKEY'. Now go back to screen #32 and delete line 1 (the second line). That is it! Now you can make a new system disk the same way as before with the exception that you must load -SYNONYMS separately and before the editor, and that when you load this editor you will load it by entering your work disk and type '32 LOAD'.

George L. Smyth
3017 Sylvan Dr.
Falls Church, VA 22042
(703) 533-8710

continued on page 26

Software tips #1

Rounding numbers, ON BREAK RUN LOAD,
TI-Writer II, PRBASE, Axiom Interface
from Stephen Shaw, England.

Correct to N places

Rounding is easy, just use:

```
RESULT=INT(NUMBER + 0.5)
```

But if you want to display a number correct to a certain number of digits? A little more difficult but not impossible. The GENERAL format is:

```
RESULT=INT(NUMBER * P + 0.5) / P
```

Where P is the power of ten of the number of places to be rounded.

That is easy is it not? To round to two places, P = 100 (10^2). The general format will only work for positive numbers. For numbers which may be either negative or positive, the formula becomes:

```
RESULT=INT(ABS(NUMBER) * P + 0.5) * SGN(NUMBER)
```

Here is a tiny utility subprogram for you. For TI BASIC just remove the first and last lines and GOSUB instead.

```
100 SUB PLACES(NUMBER,PLACES,RESULT)
110 P=EXP(PLACES*LOG(10))
120 RESULT=INT(ABS(NUMBER)*P+0.5)/P*SGN(NUMBER)
130 SUBEND
```

Now to find out what 23456 rounded to 3 places is you would use:

```
1 CALL PLACES(23456,3,RESULT)
2 PRINT "Result:":RESULT
3 STOP
```

From Maurice Swinnen of Mid Atlantic 99ers. 1986

Source code.

Author and original publication not known.

Assemble this little utility into a non-compressed DF80 file and you can load it into Extended BASIC with a CALL LOAD. Then it will auto-boot "DSK1.LOAD" whenever a running Extended BASIC program breaks for ANY reason - a little more powerful than ON BREAK NEXT. Try it!

```
       DEF  CHECK
CHECK  MOVB @>8344,@>8344  * >8344 IS 0 IF XB NOT IN
                           * RUN MODE
       JEQ  NORUN
       B    *R11           * EVERYTHING OK SO RETURN
                           * ELSE:
NORUN  CLR  @>83C4         * TURN THIS ROUTINE OFF
                           * (ISR HOOK)
       LI   R1,>6372       * XB GROM START
            * MAY NOT WORK WITH SOME XB VERSIONS
       MOVB R1,@>9C02      * WRITES >63 TO GROM WRITE
                           * ADDRESS REGISTER
       SWPB R1
       MOVB R1,@>9C02      * WRITE >72 TO GROM WRITE
                           * ADDRESS REGISTER
       B    @>006A         * EXECUTE XB
       AORG >83C4
       DATA CHECK          * PLACE THIS ROUTINES
                           * ADDRESS AT >83C4
       END
```

==========

INFOCOM adventures are not all fully logical. There is a random element in some of them which means that sometimes you die and sometimes you live. An interesting command to type in to your INFOCOM adventures is $VE. Try it! There is a #RAND command in Lurking Horror, which expects a number before/after it. I am not sure what it does but I think it may determine the path when you come to a random choice. HITCHHIKER has a total vocabulary of 969 words. Have you found them all yet? KILL ADAMS? Some odd commands, purpose unknown, include XYXXY and ZZMGCK, the latter may just be an end of file dummy. SUSPENDED has a vocabulary of 680 words, but you can complete it with just 35. That is real overkill!

DV80 Files TI-Writer cannot read

Any time you see a Display Variable 80 file on a disk, it is always a good idea to take a look at it with TI-Writer, as there is a good chance it is either documentation or source code, which may contain documentation. In almost every case a DV80 file should load with TI-Writer, whatever is in it, text or data or anything.

However, there are a few text files coming out of Europe that you cannot load with our version of TI-Writer. Our European friends are using a different and incompatible version of TI-Writer, Version 2.0. If their text files are saved to disk with PF there is no problem, but using SF adds tab information.

In order to produce those odd European characters, printers use ASCII codes outside the usual range of Version 1 of TI-Writer, so Version 2 was modified to accept them, and the tab information had to be modified as well. And as their tab data is outside the capabilities of our version to handle, the result is a console lock up. Curing the problem was difficult, until our membership secretary dealt with it.

In the April 88 issue of EAR99'ers newsletter, he presented a program to amend the tabs on a Version 1 TI-Writer file, largely to demonstrate the way the tabs are saved. What was interesting was that by appending a new tab set, the original set is "replaced". If it works for Version 1, why not try appending a Version 1 tab set onto the end of a Version 2 file?

It works! The following routine is a much modified form of his program. If you cannot load any DV80 file, amend it with this trifle and try again!

```
100 REM MAKE VN 2 TI-WRITER FILE ACCEPTIBLE TO VN 1
    based on an idea by Peter Walker. UK. 1988.
110 DISPLAY AT(2,2)ERASE ALL :"INPUT NAME OF FILE TO
    BE": "MODIFIED:"
120 DISPLAY AT(6,2):"DSK1."
130 ACCEPT AT(6,5)SIZE(-12): FILE$ :: FILE$="DSK"&FILE$
140 OPEN #1:FILE$,DISPLAY,VARIABLE 80,APPEND
150 A$=CHR$(128)&CHR$(134)&CHR$(128)& CHR$(212)&
    RPT$(CHR$(213),16)&CHR$(128) &CHR$(136)
160 PRINT #1:A$
170 CLOSE #1 :: PRINT "DONE"
180 END
```

==========

PRBASE hints:

Sorting and so on are based on an ASCII string and everything works according to the ASCII values of the characters. Thus while 4 comes after 2, 22 will come before 4. Use leading zeroes on numbers you are sorting. Then you will correctly find the sort as 02,04,22.

Selective Indexing search works on your input UP TO the first space, so that "good day" will only work on "good". To use the whole thing, you must insert a question mark, thus "good?day".

PLEASE will someone write tutorials for us for PRBASE and CFS!

==========

Axiom Printer Interface?

By this time some owners of Axiom printers may discover that their printers are not entirely behaving themselves. In particular, your computer may give you error messages indicating the printer is not there! The problem lies in the thin fiddly wires they used which are push fitted in their little connectors. In due course the wires will either pull out, or more likely, break near the end, and contact is lost. The solution is to remove the whole ribbon, remove about a half inch off the end, and then refit. Soldering is recommended, but CAREFULLY.

==========

TI made 250 TI99/8's. How many TI99/5's did they make? (The answer is NOT none!) A complete set of "home computers" from TI would be quite a collection, especially if you collected all the different versions of the TI99/4A as well.

Graphics: File Compatibility

from Stephen Shaw, England

This article has been prompted by a very odd chart of the various Graphics programs for the TI which I came across in a US newsletter; odd because at the end of the day it failed to tell you very much and was decidedly biased! This article also follows, in a way, from the discussion of various formats of disk file.

Each type of file is referred to by means of a short abbreviation, details of which are given in the first section below:

1. List of Formats:

```
TI ARTIST....Fonts (_F files, referred to later as
             TIAF)
             Pictures (_P and _C files, referred to as
             TIAP)
             Slides (_S files, TIAS)
             Instances (_I files, TIAI)
GRAPHX.......Clipart, includes fonts (GC)
             Pictures (GP)
CSGD.........Pictures (/DT files, CP)
             Graphics (/GR files, CG)
             Fonts-usual (/CH files, CF)
               -care: see note at end!!
             Fonts-DocuPrinter (/DP files, CD)
             Labels (/LB files, CL)
             Letterheadings (/1L files, CH)
JOYPAINT.....Pictures (JP)
             Compressed pictures (JC)
PICASSO......Pictures (PP)
             Fonts (PF)
             Icons (PI)
BITMAC.......Pictures (BP)
DRAW N PLOT..Pictures (DP)
DRAW A BIT 1.Pictures (DAB1)
DRAW A BIT 2.Pictures (DAB2)
MAX RLE......Pictures: DV80 files or DF128 files (MP)
```

NOTE: CSGD uses two different sets of /CH files. The font editor creates one set of /CH files, which then have to be converted to another type of /CH file for use. The /CH files referred to here are always the converted files. The conversion program is on CSGD Volume 1.

MUTUALITY:

This section indicates the types of file each graphics program can use from the above list, WITHOUT using an external conversion utility. The ability to both save and load can be assumed unless otherwise noted:

```
MAX/RLE.........TIAP, GP, MP
TI ARTIST.......TIAP,TIAF,TIAS,TIAI,DAB1,DAB2,DP,GP
GRAPHX..........GC,GP
CSGD 1 AND 2....CP,CF,CG
CSGD 3..........CF,CG,CH,CL and LOAD ONLY CD.
PICASSO.........PP,PI,PF,TIAP
                Can also LOAD a TI Writer text file.
JOYPAINT........JP
JOYPAINT PAL 2...JP,TIAP,JC
                Can also LOAD GP,DP
```

GRAPHICS UTILITIES including external (eg separately loaded) conversion routines on main graphics disks. Where more than one type is listed in the above section, conversions are possible as part of the main program, which is usually much faster.

THE PRINTER'S APPRENTICE: Uses its own picture and font formats, can also use TIAP.

TPA TOOLBOX: Uses TPA fonts and graphics, plus can convert into TPA format the following; TIAI,TIAF,TIAP,CF

PRINT WIZARD: Creates its own format from TIAI and TIAF

FONT WRITER 2: Uses, in various utilities, TIAF,TIAI,TIAP,CF,CG,GP. It can convert CG to TIAI, CP to TIAI, TIAI to CG and TIAI to CP.

PICASSO: can convert an XB font to PF, or load a PF into an XB program; convert BP to PP; make use of CF and CG files.

CSGD 1: can convert an XB screen into CP.

ARTIST EXTRAS (Texaments): can convert CF or CD to TIAF, CG to TIAI, and CP to TIAI.

ARTCONVERT (Trio+): can convert TIAI and TIAF to TI Writer graphics.

ARTIST ENLARGER (Asgard): works with TIAF and TIAI.

GRAPHICS EXPANDER and BIGTYPE (Genial): works with TIAF,TIAP, and TIAI.

JBM103 (Disk library): enables graphics to be loaded/saved to/from Extended Basic bit-mapped screens in TIAP format.

UTIL12 (Disk library): has a utility to convert from TIAI to Extended BASIC program format or merge file, or listing to disk or printer.

UTIL 7 (Disk Library): has a utility to convert TIAI to TI Writer graphics.

UTIL17 (Disk library): has a utility to convert a segment (5x5 chars) of a GP CG, and a utility to convert CG to TIAI and/or Extended BASIC merge file.

The de facto standard has been set by TI Artist- only graphics programs released before TI Artist lack TIA capabilities, apart from CSGD, although external utilities have been created to remedy that!

As far as printers go, all these work with Epson FX series printers or any printer which follows Epson commands- the usual commands used are:

ESC * (8 pin bit image mode)
ESC K (480 DOT 8 PIN mode)
ESC L (960 dot 8 pin)
ESC Z (1920 dot 8 pin)
ESC A n (Line spacing in n/72 inch)
ESC 1 n (Left margin setting)

A few programs allow Gemini printers to be used, but Gemini used two incompatible codings in their printers, and Gemini owners often report problems. A very few programs will support other printer codings. o

continued from page 30

BUG BYTES - July 1988 - Group announced its incorporation, report on success of Brisbane TI Fair, other groups should have a go. Report on Myarc HFDCC has a missing page, mention of HV99ers building 128K memory card, Mark Murfin reviews CALANDER MAKER 99 and rates it 9, P7 shows a TI99/4A keyboard wiring matrix, P8 has Myarc update - PAL version of Geneve arriving this month together with HFDCCs and V1.9 of MDM5. Lou Philips advises of a hardware modification to HFDCC, a zener diode is added to pins 12 and 4 of the 9223 chip. The latest version of MDOS, GPL and MYWORD are on their way. P8 has information on Geneve Special Interest Group set up to distribute software and a tip on fast loading using a file called FWB. Col Christensen on interrupt driven display of a real time clock (BASIC) and review of EZ-Keys which allows single keystroke operation of macros in Extended BASIC. P14 has a TAX REFUND CALCULATOR, COLORVISION and XB programmer's aid.

ROM Newsletter (Orange County, CA) June 1988 - Adrian Robinson adds to his modular approach on assembly language with auto repeat in "ACCEPT AT", Roger Merit reviews Certificate 99 V2.0 which makes perfect bordered certificates with text and graphics, Earl Raguse on how computers see and use numbers in Forth, Stan Corbin reviews PRBASE and how to set up mailing labels.

CIM 99 CLUB INFORMATIQUE MONTREAL is written in French - any volunteers to extract relevant information? There are articles on Multiplan, Telco, a Graphic Editor, Assembler, a Golf game and printers.

SPIRIT OF 99 (Central Ohio) July 1988 - Article on MULTIPLAN as a database with an index to stored items and a comprehensive cross-reference, Jack Sughrue on the TI Fair in Lima, Ohio together with a helpful hint on using Gemini printers. Paul Scheidemantle reviews the Star NX1000 printer; at $US180 it makes you wonder who is making all the profit here. Finally, there are hints for [illegible]. o

continued from page 1

Speaking of the Geneve, it is reported elsewhere that Rolf finally demonstrated his Geneve at the Illawarra Regional Group meeting. I bought this for Rolf when I was in Philadelphia, in July 1987, and carried it around Europe until November. Rolf first saw it last December when it showed a rather nice swan on my original TI99/4 modified TV to take NTSC composite video. Unfortunately, while Rolf was trying to get it to do more with the somewhat inadequate software provided at that time, it stopped showing the swan and instead said that there was a memory error. Disaster! Eventually Rolf sent it back to Myarc, when it looked like software was finally going to be available, and after a few weeks he was sent a good, fully tested one back. It was not his one repaired, unless they unsoldered all the chips on his board and inserted sockets, but at least it worked! He also received quite good service on it too, so that should be good news for all those who have bought one or are contemplating buying a Geneve. At the moment he has to find a suitable RGB monitor as he either has a mono monitor which has funny lines on some colour combinations, or my TV which is not too good in 80 column mode. The next release of M-DOS is awaited somewhat eagerly in Balgownie.

****

I have just built my second RAMdisk using the Horizon type card and 32K byte memory chips. My first card has been doing good service but it is only one layer of the 8K byte chips and so is only 352 sectors large. This is large enough for a lot of the programs that I use regularly, but now I have decided to use Funnelweb editor and formatter as the editor that came with my AT system seems to store sectors in the wrong place. At least something was and since I have been using Funnelweb nothing untoward has happened. That is not conclusive but I was becoming more convinced that the problem was in the editor rather than DM1000 from the symtoms. What has this to do with the RAMdisk I hear you say? Be patient, please! Using Funnelweb means that I have to go back to loading from disk. Funnelweb is quite good in that respect, but when I am going between editor and formatter it is a pain to have to twiddle my thumbs. I have been spoilt by the first RAMdisk. I thought that I could put Funnelweb on the RAMdisk, and even Multiplan and files I was editing frequently would save and load faster if they were in RAMdisk. Come to think of it, I need a huge RAMdisk, perhaps a hard disk? I would need more money first. After getting the RAMdisk built (stacking chips 3 high) and getting rid of the few problems like a wire left off, a bad memory chip and a broken track, I then tried to use it. First problem seemed to be that I was using V7.1 ROS in my original RAMdisk and V7.3 ROS in the new. Change them both to V7.3 and I can copy files to the new RAMdisk with DM1000 V3.5 but not with the Funnelweb version of the manager. Also have problems with the SD function of the Editor when looking at the RAMdisk as it fills the screen with null (>00 code) characters and >16 code characters. I shall have to write to the McGoverns enclosing a contribution and ask for help. Meanwhile I am not sure how I am going to use the RAMdisk. When I have a bit of time I will experiment.

****

Now to the Newsletters. I notice that some groups send the newsletters out to their members using a chain letter approach. I have suggested that they be copied and sent to each regional group. The best we can do for you is to summarise the contents of the newsletters received as they go into the library and then if you want to read further you have to join the library and try to borrow the one you want in competition with all the other members of the library. If too many want to read one article the system must break down. Perhaps you are not interested in reading the information obtained in the overseas newsletters. One reason that we do all the work to prepare the TND is so that we get the chance to read the newsletters. Information is the name of the game. Of course people like John Ryan provide a valuable service by retyping articles of general interest from newsletters onto the BBS and from there they go into the TND for all to read.

Lou Amadio prepared most of these notes from the newsletters for me.

There are 6 issues of Northwest Ohio 99'er news. The first is Summer 1987 and contains an article by Don Turner on an EPROM for the Horizon RAMdisk. Curtis Provance from New Hampshire talks about the power of the TI99/4A while Joe Nuvolini has a review of MASS TRANSFER. There are a number of keyboard strips if you want to photocopy them. Paul Scheidemantle has an article on designing characters, Jeff Gatlin has a program of music using the fourth voice, Jack Coleman talks about business graphs, Hank Avaro tells you what to do for stubborn console hangs and also about the diode to stop alpha lock from upsetting the joysticks. There is also a circuit for converting RS232 to parallel output. In October 1987 there are articles on the Geneve by Roger Feinauer, a program to convert Extended BASIC screens to TI-Artist instances by Terry Atkinson from Nova Scotia, on putting 32K byte chips on the 16 bit bus in the console by John Clulow and some TI-Writer and formatter hints by Tom Kennedy which I think we printed in our jumbo TI-Writer issue. There are also circuits for connecting the PIO to a Smith-Corona printer and Okidata 82 printer from William M. Lucid of the Hoosier users group. In the November 1987 issue the problems of keeping a user group going in the face of falling membership are discussed. There is another article by Roger Feinauer on the Geneve and if anyone is interested in another way to interface the real world of model trains to their computer, Rod Cook provides part 1 of a system which adds on to the PEB the same sort of system as that shown by Ross Mudie. Part 2 may be in the December issue which I have not seen. In January 1988 there is a long article on M-DOS by Chris Bobbitt of Asgard Software, an article by Don Turner on using CALL in Extended BASIC with a sample program and a listing of some Fairware programs with the names and addresses of their authors. Still on the Northwest Ohio 99'er News, although the name seems to have disappeared from the front, the February 1988 issue contains an article on the Geneve by Roger Feinauer, a program for use with the MBP analog to digital converter board, two tinygrams (defined as a single program which fits in one screen), one for 28 column listing and the other called Style a Line both by Ed Machonis. There are programs for 'kids' by Ralph Kopperman of New Jersey, for using the console as a piano keyboard from the Swedish newsletter Nittinian and to convert TI-Artist instances to TI-Writer files, modified by Lutz Winkler from the original by David Dhein of Chicago. In the March 1988 issue, there are articles on trigonometry by Bill Harms, a review of Font Writer II and a Quad column printer with assembly language support by Mike Dodd from Los Angeles.

Hunter Valley - July 1988. New committee elected headed by Al Lawrence. Rumours of 2 Mbyte RAMcard, unlimited memory for text and a voice recognition system. FUNNELWEB 4.11 has been released with some bugs removed. DM1000 will now copy all files by pressing FCTN[6]. Albert Anderson (Secretary) talks of sending out a free newsletter to entice previous members, mention of new TIUP committee, Chicago TI User Group embarking on a major software exchange program. In The News by Joe Wright - "Artconvert" by Tenex converts TI-Artist Instances to DV80 files, a Forth system which loads into Super-Cart, Rave 99 card for mounting speech in PEB (TTT Card is better value), The Italian Conection - a computerised name-source DBase and a FORTI card for music lovers. Bob Carmody submitted an interesting list of CALL LOADs to effect RUN, NEW, RES, RUN without prescan, Auto Run, VDP status register, sprites in auto motion and VDP interrupt timer. Garry Christensen reviews Myarc HFDCC with future expansion capabilities for 1.44Mb floppies! It is claimed to be faster than a RAMdisk - fills 32K in 10 mS. Tony McGovern talks of squeezing assembly language when programming with small memories, Brian Rutherford on Linking Assembly to Extended BASIC, Jack Sughrue on the Lima, Ohio Fair (no charges for anything!). Hint of the Month shows how to remove unwanted lines from graphic screen dumps, Expanding your system by Joe Wright including Neil Quigg's RAMdisk project, an article by G. Christensen on the value of a Geneve compared with alternative funtionally equivalent add-ons. Keith Amann talks about XMODEM file transfer protocol which automatically detects and retransfers faulty data on a phone line. continued p 29

Regional Group Reports

Meeting summary.

Banana Coast	11/9/88	Sawtell
Carlingford	21/9/88	Carlingford
Central Coast	10/9/88	Toukley
Glebe	8/9/88	Glebe
Illawarra	19/9/88	Keiraville
Liverpool	9/9/88	???
Northern Suburbs	22/9/88	Davidson
Sutherland	16/9/88	Jannali

BANANA COAST Regional Group
(Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Bayldon, telephone (066)53 2649.

The July meeting of the Banana Coast TI Users Group was held at the Sawtell Tennis Club House. Great interest was shown at the disk containing the finalist programs from the recent Software competition. All must be congratulated on their effort as the results were really outstanding. One thing that was learned by all as a result of placing a TV monitor on top of a disk drive for better viewing, there was two hours of wondering why the disk unit refused to read any of a number of disks. At a final attempt the TV was removed and all was well with the system. No doubt the invisible "field" was suppressing the signal. (We think).

Yours in Computing, Kevin Cox.

CARLINGFORD Regional Group.

Regular meetings are usually 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.

Meetings are normally held on the second Saturday of each month, 6.30pm at the Toukley Tennis Club hall, Header St, Toukley. 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 third Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. Contact Bob Montgomery on (042)28 6463 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TIsHUG Sydney meeting at 7.30pm. Contact Larry Saunders (02) 644 7377 (home) or (02) 759 8441 (work) for more information.

Meetings coming up.
September 9th 1988 ???

NORTHERN SUBURBS Regional Group.

Hello again, this is Dick Warburton cordially inviting you to attend the next meeting of the Northern Beaches' Group. We have gradually improved our meetings,and have a small but regular group of 5 or 6 attending.

Our meetings for the next two months are as follows:

Naturally we try to fit in as much copying as possible. We also provide something to eat and drink. We welcome visitors. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132. See you soon.

SUTHERLAND Regional Group.

Any persons interested in joining the Sutherland Group are more than welcome. The format of the meetings are very informal as more often than not the conversation digresses onto matters purely social rather than related to computerisation. The supper is not bad either. Meetings are held on the third Friday of each month. Group co-ordinator is Peter Young, (02) 528 8775. BBS Contact is Gary Wilson, user name VK2YGW on this BBS.

Future meetings will revert to the home of Peter Young at Jannali at 7.30pm.

The July meeting saw the attendance of a new Regional Group member in Bill Janssen. Bill recently purchased a complete TI system and it is hoped that we can assist him in gaining good use out of his new computer.

Joe D'Ambra's perseverance with PR Base finally paid off and we are now able to get past "square one". As with most problems of this type, the fault turned out to be a simple one, associated with the correct definition of a file name.

Peter Young

TIsHUG in Sydney

Regular meetings are normally at 2pm on the first Saturday the month, except January and possibly other months with public holidays on that weekend, at the Woodstock Community Centre, Church Street, Burwood.

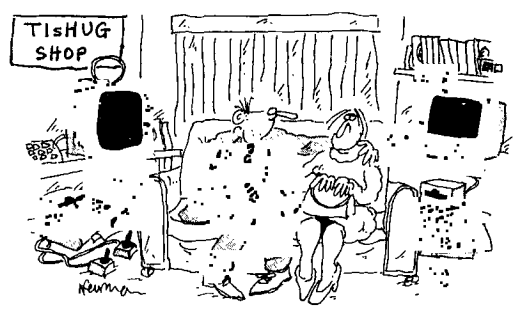
Meetings planned this year. September 3 - Software demonstrations and purchases. Again this will be a normal 2pm start. At this meeting some of the latest software as advertised in MICROpendium will be demonstrated and hopefully we will have imported copies for sale direct to members at reasonable cost.

October 1 - Joint meeting with HV99'ers. Contact has been made with our friends in Newcastle and a verbal response suggests that this meeting will go ahead on October 1. It is envisaged there will be a computer meeting in the morning with a social afternoon. There will be more details on this as we get closer to the actual meeting date.

October 8 - a software copython. Be at Woodstock at 2pm to be able to get some of the latest software. Other minor events, yet to be finalised, are also planned for this afternoon.

November 5 - Full day tutorial workshop. Here the themes and other activities are yet to be finalised, however, like all full day events in the past, this is guaranteed to be a fun day. As usual there will be a luncheon BBQ at a very reasonable price.

December 3 - Christmas party. Given a fine day this will be one of the major events of the year, with plenty of food and drink available and a great chance to chat with fellow members in a social and relaxed environment. There will be plenty of software released for this meeting so you will have plenty to keep you occupied over the Christmas/New Year holiday break. o



"How would you like to see my disk collection?"