



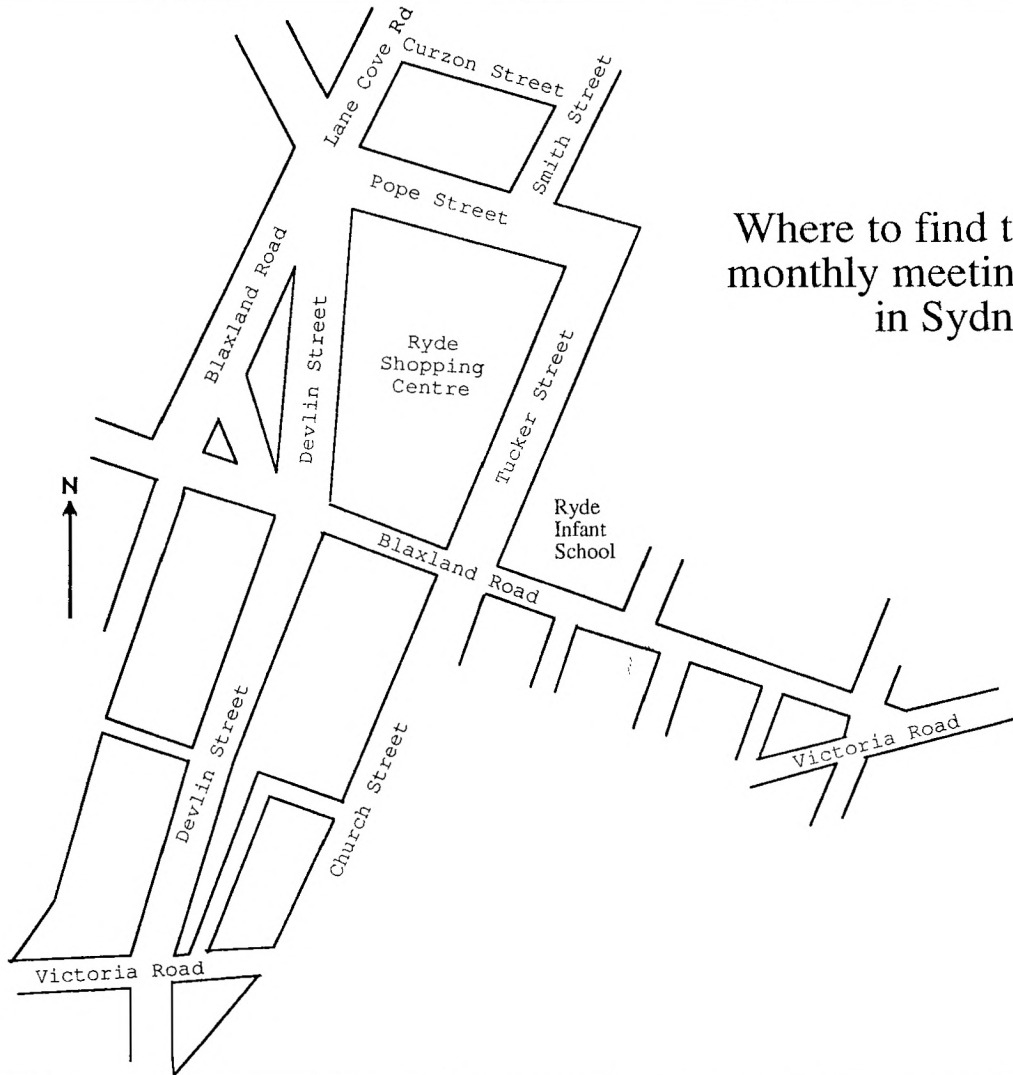
# NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 9, Number 2

March, 1990

Registered by Australia Post - Publication No. NBH5933



## TISHUG News Digest

March 1990

All correspondence to:

P.O. Box 214  
Redfern, NSW 2016  
Australia

**The Board****Co-ordinator**

Dick Warburton (02) 918 8132

**Secretary**

Terry Phillips (02) 797 6313

**Treasurer**

Geoff Trott (042) 29 6629

**Directors**

Rolf Schreiber (042) 84 2980

Russell Welham (043) 92 4000

**Sub-committees****News Digest Editor**

Bob Relyea (046) 57 1253

**BBS Sysop**

Ross Mudie (02) 456 2122

BBS telephone number (02) 319 1009

**Merchandising**

Percy Harrison (02) 808 3181

**Publications Library**

Warren Welham (043) 92 4000

**Software library**

Rolf Schreiber (042) 84 2980

**Technical co-ordinator**

Lou Amadio (042) 28 4906

**Regional Group Contacts****Carlingford**

Chris Buttner (02) 871 7753

**Central Coast**

Russell Welham (043) 92 4000

**Coffs Harbour**

Kevin Cox (066) 53 2649

**Glebe**

Mike Slattery (02) 692 0559

**Illawarra**

Lou Amadio (042) 28 4906

**Liverpool**

Larry Saunders (02) 644 7377

**Northern Suburbs**

Dennis Norman (02) 452 3920

**Sutherland**

Peter Young (02) 528 8775

**Membership and Subscriptions**

Annual Family Dues \$30.00

Overseas Airmail Dues AUS\$60.00

**TISHUG Sydney Meeting**

The next meeting will start at 2.00 pm  
on 3rd of March at Ryde Infant School,  
Tucker Street, Ryde.

Printed by  
The University of Wollongong  
Printery

**Index**

Title	Description	Author	Page No.
Co-ordinators report	General news	Warburton, Dick	2
Emulate file creation	Software hints	Schreiber, Rolf	33
Extended BASIC Tutorial	Software hints	McGovern, Tony	9
Forth to you too!	Software hints		20
From the bulletin board	Mail to all		30
From the editor	General interest	Relyea, Bob	2
From the service bench	AT disk contr, RAMdisk	Trott, Geoff	6
Game of NIT-WIT	Software review	Lang, Chris	25
Games information	Adventures	Brown, Robert	15
How does it work?	General interest	Trott, Geoff	34
Modified formatter commands	Word processing		24
More PEEKs and POKEs	Software hints	Schreiber, Rolf	7
New location for Sydney meetings	Club news	Mudie, Ross	12
Newsletter annual report	Club news	Trott, Geoff	1
Program to type in	Chemistry test	Grunge	27
Quick Run review	Software review	Shaw, Stephen	22
RAM memory tester	Assembler code	Hansen, Tor	31
Rambles from UK	General interest	Shaw, Stephen	13
REF/DEF table	Software hints	Rebel, Eric-Paul	23
Regional group reports	General interest		35
Secretary's notebook	Club news	Phillips, Terry	3
Software extra	Club software	Schreiber, Rolf	3
Sorcerer part 1	Adventure hints	Scorpio	15
Techo time	RGB inter, AT disk cont	Amadio, Lou	5
TI-Base tutorial	Data base	Smoley, Martin	11
TI-Writer hints	Word processing	Relyea, Bob	4
TI99/4A user groups	General interest	Peterson, Jim	16
Tips from the tigercub #39	Software hints	Peterson, Jim	21
TISHUG software column	Club software	Phillips, Terry	3
Younger set	Program, adventure	Maker, Vincent	29

**TISHUG Fairware Author of the Month**

The Fairware Author for this month is John Birdwell for his Program Disk Utilities. The latest version available from the shop is version 4.2. All Donations collected at the meeting and sent in will be mailed to him this month.

**Newsletter Annual Report, 1989**

by Geoff Trott

1989 was the best year yet for the TND. We put together 12 issues with a total of 408 pages and most of the time managed to get them in the mail before the meetings. We had one special issue on Hard Disks as well as the first one on the Indices from 1988. One disturbing factor was the increasing cost of printing which occurred during the middle of the year. Fortunately the printery has now in place a cheaper process for producing the plates so that costs have become lower. If our membership declines we will probably have to reduce the number of pages each month while maintaining the quality. You may have noticed a decline in the print quality as the year progressed. The daisy wheel printer has just returned from repair where it was found that the print hammer was worn out. It just lasted the year but it has pounded out a few million letters! I imagine. Anyone like to work out how many characters in 1989 TNDs?

continued on page 34

# Co-ordinator's Report

by Dick Warburton

Welcome to a new year with Tishug. The directors have been elected, and are rearing to go. There are four major considerations for 1990. Firstly, money is tight, and we will need to watch our expenditure carefully. You can help yourself and the club as well, if you buy what you need from the club shop. We will try to stock it with what you want, at a reasonable price.

Secondly, 1990 will be a year of exciting projects for our members. We will try to make kits available through the shop for the range of projects. As indicated last month, we will soon be able to proceed with fitting eproms on our ramdisks. Craig has worked out a way to access up to 1 Meg of eproms. One can now have a ramdisk with all our major operating software available, without using up our precious memory chips. It works as well as the normal ramdisk. It is fast, and the memory does not corrupt. We will need to seek permission from software authors before we can use it widely, but it shows great promise at this time. The installation instructions should be available in March. Other projects, like the interface for the analog colour monitors, are already available through the shop. Expansion of the TI will be economical, and very satisfying. You will be able to add 32K memory, build a ramdisk, build a cheap double density disk controller, and buy low cost drives etc. This is the year to get the best out of your TI, and have the satisfaction of doing most of it yourself. To help with projects, a project group will meet regularly in Sydney. The venue has not yet been decided. If you have some ideas about the venue please let me know.

A third major consideration, must be what we do to attract new members, particularly young people. A games coordinator has been chosen, and a room will be available at future meetings for those who are young at heart. One step taken to help achieve this goal, is the move to a new location. From the March meeting onwards, all main meetings will be held at Ryde Infants School, Tucker St. Ryde. Rooms are plentiful, and the cost is low. It should be possible to run all the separate activities we wish to. Fourth consideration, is the need to become self reliant in servicing our consoles and equipment. It is hoped to provide opportunities for us to learn how to fix the more difficult problems which are now arising because of age. (The console's age, not ours). The console repair group will continue to meet at Cyril's place and you are welcome. The continuing education of our members about this machine, is of primary importance. Hopefully, lots of other things will happen as well.

I am looking forward to the April meeting, which will be another buy, swap and sell day. They really seem popular. Ross tells me that he might run another tutorial on linking assembly to basic. Hopefully some new software will be written locally. I trust that we can stimulate this. Certainly, the club can make every effort to import the newest and best available software from overseas, and distribute it at the lowest cost. Our new pricing of software disks, should keep our software coordinator out of mischief for the whole year. Get in for your bargain at the next meeting. If the software you want is not available then see Rolf Schreiber, and order it for the next meeting. Catalogues of software are available on disk.

Although the annual meeting day was wet and cold, I was pleasantly surprised at the number of members who attended. I counted 40 at the meeting. I was also pleased with the willingness to take on responsibilities. The club is only as strong as its members. While people will help each other, and give something back, we will stay strong. We will continue to grow more self reliant and more confident of our own abilities and talents. You can help in some way. You can be friendly to newcomers at meetings. You can open up your home occasionally, for local meetings. You can bring a system to the monthly meetings. You can also help by being on the lookout for bargains for our members (not off the back of a truck please). You can be

on the lookout for anyone with a TI who wants help, or who wants to sell it. You can tell people about your club, and invite friends to come. You can help with the coffee, or look after a barbie, but most of all come to the meetings where possible and enjoy yourself.

Well I look forward to seeing you at the next meeting...

Dick Warburton.

## From the Editor

by Bob Relyea

This is my first official editorial since the A.G.M. so I thought that I would take the opportunity to introduce myself and to give you a bit of my background. I live in a small country town called The Oaks which is situated near Camden and about a 50 minute drive from Wollongong. I am a high school Maths and Science teacher and I have been posted at Airside High School (Campbelltown) since 1982.

My interest in computers and other calculating devices goes back many years - to my high school days. It was further strengthened by exposure to computers during my university years when I was attending engineering university in the United States. Back in those days the computer was in its infancy and a huge floor model computer would take ages to do things that a pocket calculator could easily do today (Must have been back in the dark ages! did I hear somebody shout?).

Being a family man (5 children - one has recently joined the Australian Navy) I was not able to afford a PC back when they first came out. The prices started to drop back in the early 80's, however, and in 1984 when the price of the TI/994A dropped below \$200, like a lot of other people I bought one and started to fiddle. After a lengthy phone conversation with John Robinson I finally joined the club in 1985, but did not attend any meetings until the next year when the meeting venue changed to Burwood. In fact my first meeting was a technical day run by Ross Nudie where we all sat in the big room and upgraded our computers with the 32K "match box" expansion. I did not have any intentions of participating when I went as I had not reserved a kit of components but just went along to see what happened and to get the 'feel' of the club meetings. Not being sure as to what went on at the meetings I had everything in the car, from my colour TV set to the console. I somehow had an idea in my mind that everybody sat in a room each with their own console and followed instructions given by somebody on some topic. Anyway, as I walked through the door I signed in like everybody else that day and a kind member sitting at the desk said that he had a 32K kit ordered that he was not able to use and offered it to me. I eagerly took him up on his offer and spent the rest of the day soldering up those numerous pins. Fortunately it worked the first time.

So, I bought a computer in '84, joined the club in '85, fitted the memory expansion in '86 when I attended my first meeting, expanded my system in '87 when I purchased a Peter Schubert mini-expansion system. Since then I have added a Ramdisk and with my wife, Frances, working last year I was able to 'splurge' and purchase a modest printer. I enjoy word processing and working with Multiplan and TI-Base. How I use the computer as a teacher will be the topic of a future editorial.

I am happy to be able to relieve Geoff for a year or two with the editing so that he can get on with doing things that only he and a few other club members can do such as hardware development. I would like to thank the directors and you club members for your support and trust that I am able to maintain the fine standard set by Geoff and others before me. As Dick Warburton said in a recent report - "I could never possibly give back to the club what I have gotten from it". I like attending the meetings and I am looking forward to a great year of learning and computing.

*Bob Relyea*

# Secretary's Notebook

by Terry Phillips

It was with regret that I learned of the passing of member and former Director and Treasurer, Bert Thomas, who passed away in mid January after a battle with illness for the past few months. Members, who attended meetings, will remember Bert for his cheeriness and helpful attitude. To his family and friends, sincere condolences from all at TISHUG.

The AGM, held on Saturday 3 February was reasonably well attended, I suppose, given the atrocious weather conditions on the day. Approximately 40 people attended and the agenda was gotten through at a fairly brisk rate.

A few main points to come out of the meeting were the announcements that fees will rise this year (for the first time in 5 years) from \$25 to \$30 for an annual family membership. Chief reason cited for the increase is the continuing upward spiral in production costs of the TND. Each issue, with production and mailing, costs around \$3 to produce. Alternatives considered by your board at a recent meeting were to either reduce the quality of the TND or reduce its frequency of issue. On this occasion the Board opted to stay with a high quality monthly magazine but seek a slight membership fee increase. Concurrent with the fee increase the \$5 charge for publications library access has been dropped. All members can now have access to the publications library at no additional fee. The \$5 bulletin board access fee will remain as it appears that this service will be liable to additional costs sometime during the year due to a relocation of the service.

Secondly, is the meeting venue move from Woodstock to Kyde Infants School. It is anticipated that this move will save in the vicinity of \$400 in meeting fees over a full year. More importantly, this venue will give your group room to spread out and conduct various activities as planned, and without the ever present wonder whether sufficient rooms will be available. At last Dick will be able to get his long awaited GAMES ROOM underway and hopefully attract the younger brigade along to meetings. Shane Ferrett, by the way has volunteered to be meeting games co-ordinator.

Thirdly, was the need this year to hold a ballot for the positions of Director. Although only 4 nominations had been received at the closing date, a further 3 members nominated at the AGM. Those elected to guide the club this year are Dick Warburton who will continue as Co-Ordinator, Terry Phillips continuing as Secretary, Rolf Schreiber who will hand over Treasurer duties and take on the responsibility of Software Librarian, Geoff Trott newly elected who will take over the Treasurers role and Russell Welham who will take over as Meeting Co-Ordinator. To retiring Director, Robert Peverill, goes my thanks for his support and help throughout 1989. Robert, by the way, will continue with the quality ribbon re-inking service currently being offered to members for the fraction of a price you would expect to pay for a new ribbon. Another new "body" you will see this year will be Percy Harrison who has the onerous task of running the shop. Look for bargains in this area as Percy gets into the swing of things.

There are two new members to welcome, and they are:

Peter Bestetti of Blackheath, and  
Peter Gill of Barrack Heights.

Hope you can both get along to some of the main Sydney meetings.

See you at the meetings. Here is what is planned for the next 3 months:

Feb 3, 1990 - Demonstration of new EPROM for RANC. Technical advice on how to correct problem with AT MULTI FUNCTION CARD and the kick off for the new games room with host Shane Ferrett.

APRIL 7, 1990 - Big BUY, SWAP SELL day. Come along and buy that item you are after at maybe a bargain price, or if you have a surplus of TI related items bring them along and turn them into cash. Games room will be in operation.

MAY 5, 1990 - The first of our 1990 full day workshops. Themes and topics are still being refined. Details will be provided nearer to the event. o

## TISHUG Software

Column by Terry Phillips

I have now handed over the software librarians duties to Rolf Schreiber. The transfer will be effective as from 3 March 1990 and up to that date enquiries can still be directed to me, however, after that date please direct enquiries to Rolf.

Copies of TI-ARTIST PLUS are expected any day now and will be available at the shop on the March meeting date. By all means reserve a copy before hand if you wish. Cost should remain at \$25 for new purchasers and \$15 for members upgrading from a previous version.

Do not forget that during 1990, the cost of disk software will be reduced from the present \$5 to \$2. Chief reason for the price reduction is the now lower unit costs of blank disks. The \$5 fee was set some years ago when disks were at a much higher price than they are now. It is also intended that much more publicity will be given to disk software available through the shop as well as import items that members would like to see on sale. Thoughts on this latter point would be very welcome. Demonstrations of disk software are also planned for each meeting, so you can see before buying.

My thanks to Larry Saunders for his recent donation of a disk to the software library.

See you during 1990. o

## Software Extra

by Rolf Schreiber

To those of you who were unable to attend the AGM last month, I would like to introduce myself as the new software librarian. Geoff Trott, who was elected as the only new director (replacing Robert Peverill), has taken over my old "portfolio" of treasurer. Whereas I was not the world's greatest treasurer, I believe that Geoff is right up there amongst them! Those of you that have been members of TISHUG since its inception (are there many left?) will remember that Terry Phillips has carried out the very important task of software librarian since the beginning, but he has finally decided to call it a day. I would like to take the opportunity to thank Terry on behalf of all TISHUG members for the big effort he has made over the many years to bring us all the latest software.

I believe that 1990 represents a very important year to TISHUG and its members. This is the last year that we will have any reserves left in the way of term deposits. The survival of the club in its present form (including the production of the TND in its present form) will depend very much on how many members renew their subscription in April this year, as well as on how successful the shop is at making up the shortfall on magazine production costs, currently exceeding membership dues by \$1800 p.a.

Onto more interesting matters, namely, software! I believe that there is still a mountain of new software available, both commercial and freeware, that few of us have heard of, let alone own. Most of this software originates in the USA and needs someone to import it.

continued on page 12

# TI-Writer Hints

by Bob Relyea

One of the best things about working with Geoff Trott in getting the February issue of the TND out was what I learned about using the Editor and Formatter of Funnelweb. The purpose of this article is to share a couple of them with you. For those of you who have used the TI-Writer/Funnelweb software for a long time this may be nothing new to you but to me it really opened the door to one aspect of my editing, both for personal use and for the Club.

Have you ever thought that it would be nice to view the Formatted text before you sent it off to the printer to make sure that everything worked out the way that you had planned? If such a thing could be done it would sure save a lot of printer paper that was wasted for all those mistakes! For those of you who have the TI-WRITER manual this procedure is briefly mentioned on the bottom of page 117. It is done with the usual TI brevity and unless you are a genius or have had it explained to you it is likely that you missed the importance of it. The club system that I have been using has an 80 column card and a hard disk drive. I am not sure which one is the most beneficial for doing all the editing but they are both a real blessing! Geoff has the system set up in a very professional manner and after a tutorial or two(or three?) it is really easy to get things done. With the hard drive it is easy to set up files and subfiles which can store things nicely and are easily retrievable. For purposes of editing it has an 'R' an 'E' and a 'P' file. Into the R file goes material that has been taken from disks that have been given to the Editor and 'roughly' edited. Into the E file are ones that have been fully edited with the spelling checked and all. The P file is reserved for those articles that have been passed through the text formatter, a process that I will explain a little later. All of these 'P' files can be called up by the editor(LF) and viewed at any time, and they appear just the way that they will be outputted to the printer. When you think about the fact that some articles are very long, involving quite a few pages of paper, being able to view it in advance with the intent of correcting any last minute errors, is really useful. You do not have to have a hard drive to view a formatted text or even a ramdisk, although it is often quicker. You do not even have to have a multiple disk system as I viewed a formatted text last night with just one drive to see if it would work - it did! Here are the steps that I take to view a formatted text.

- 1) After your text is ready with the formatter commands(such as FI, LM, RM and AD) in place, return to the command mode and save the file to disk(more about the method of doing this with hard drive or ramdisk files later)
- 2) Then execute Quit and Exit to return to the main menu and select 2 for Formatter.
- 3) The first prompt that comes up is ENTER INPUT FILENAME which is usually done for you. If you have not been using the editor then you will have to type in DSKn.filename of the text that you want formatted.
- 4) This is the important part! When the next prompt comes up, i.e. ENTER PRINT DEVICENAME you will have to change what the computer automatically prints there which is something like RS232.BA=4800 or whatever. If you press Enter here you go through the normal procedure of getting the computer to format the text and to automatically send it to the printer. What we want the computer to do is to format the text and send it to a disk. So, we tell it to do just that. Type in DSKn.filename and press Enter. Just press Enter for all of the remaining prompts. After pressing Enter for the last one the word 'working' appears at the bottom of the screen and remains there until the text is formatted and sent to the disk. To make sure you understand what I mean I will give an example. If you have only one drive then the edited text and the formatted text will all be on the same disk and your response to the first two prompts would be:

DSK1.filename

DSK1.filename

Make sure you use a slightly different filename for the second one otherwise you will get a file error message. So, for example, if the filename of the edited text is 'mad', then make the formatted filename 'mad1'. If you have two drives you could have the edited text on one disk(usually DSK2, as DSK1 was probably where you had the programme disk) and the formatted text on another disk(DSK1). You would, therefore, type in for the first two prompts:

DSK2.filename

DSK1.filename

In this case it would not be necessary to change the filename.

5) Now if you want to view the formatted text before you print it just load the Editor and with LoadFile(LF) call up the formatted text and have a look. You will notice that there are LineFeed symbols on each line which is just a command for the printer but does not get printed, of course.

6) If the formatted text is just the way you want it then you can print it right from the Editor using the PrintF(PF) command as long as you suppress the LineFeed by typing LF at end of your printer parameters. A Typical PF command would look like this:

RS232.BA=4800,LF

If you neglect the LF then everything will be printed with double spaces.

6) There is no need to save all the formatted texts on disks as you have the edited text, so you could delete them to save space for other formatted texts.

Now if you have a harddisk or ramdisk you would go about your file management differently. I have a Mini Expansion System Ramdisk which I have partitioned into two 'drives', drive 4 and drive 5. Drive 4 is named 'E' and drive 5 is named 'P'. When you save the edited text to the Ramdisk you type in:

DSK4.E.filename

If you do a catalog of DSK4 the text you have loaded will show up as E.filename. Using this system with the Ramdisks necessitates that you enter the following information for the first two prompts of the Text Formatter:

DSK4.E.filename

DSK5.P.filename

In both cases the filename would normally be the same, of course. The procedure for viewing the formatted text is as the case for those using a disk system. With a harddisk system you adopt a similar procedure as per ramdisks except that it is possible to keep the 'E' and 'P' files on the same drive.

The other little tidbit that I picked up was involving the use of FindString(FS). If you have used this command you have probably been as annoyed as I have been about the fact that after each request to locate a particular word(or whatever) the cursor returns to the command mode instead of finding the next place where the string occurs. You can get around this problem by using the ReplaceString(RS) command instead. Just type in RS and then for 'old string' type in the one you want to locate and for 'new string' type in anything(but short) as you do not plan on using it anyways. After pressing Enter the cursor goes to the first occurrence of the string and if it is a word then the cursor locates itself on the first letter of the word. If you type in 'N' for No then the cursor goes to the next occurrence of the string(word) instead of back to the command mode as in FindString. As long as there is at least one more occurrence of the string further ahead in the text the cursor continues to light on the first character of the string. When it has reached the last occurrence it stops on the last character. That is your signal that the last occurrence has been found. If you type in N here the N will appear on the text buffer over writing the previous character.

continued on page 12

## Techo Time with Lou Amadio

One of the more interesting projects which we hope will be available soon is Craig Sheehan's version of the horizon RAMdisk using preprogrammed 256K EPROMs to store files. This will be great for word processors, databases and any number of utility programs, which, once installed, do not need to be changed. Does this mean that there will be no more corrupted files (including the DSR) on the RAMdisk? Craig, my RAMdisk, which has been unreliable lately, is a candidate for your modification.

Recently Geoff and I spent an evening checking the head alignment on an assortment of floppy drives. We used an alignment disk kindly supplied by Bob Keast. Although the floppy was designed to operate from a PC, we were able to activate the drives independently using a disk drive tester and we read the data on an oscilloscope. This showed up an eccentricity error on one of the drives which could not be easily diagnosed by other means. Geoff will have more to say on this subject at a later date.

Geoff is also working on some hardware modifications for the Mini PE system RAMdisk to make it more reliable.

The RGB monitors offered to TISHUG members last December may only be available for one last order. This really depends on whether or not Wang will continue to trade in old systems. So if you have a need or desire for a RGB monitor at around \$110, then contact either Rolf or myself ASAP. Please note the errata on the RGB interface this month. Printed circuit boards for this interface are now available for \$12 through the club.

By the way, thank you to Werner Kanitz for his invaluable assistance in producing the artwork for the RGB interface, and also to Eric Ockenden for producing the negatives for the PCB. The net result was a significant reduction in the cost of producing the interface. This sort of involvement by members is what makes it a pleasure to belong to this club.

### Errata: RGB Interface

Now that I have built the PCB version of the RGB interface, I would like to make the following comments:

1) Sync output points (S1, S2) are incorrectly marked on the copper side of the PCB. Use the overlay diagram on the front cover of the February '90 TND for guidance. S1 is the correct sync output for the Wang monitor. S2 provides an opposite polarity sync signal for other monitors.

2) Due to the close spacing of some of the mounting holes, a number of components (resistors and diodes) have to be mounted vertically.

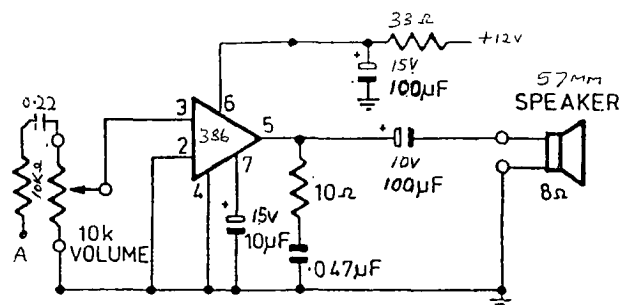
3) The silicon diode (1N914 or 1N4148 or 1N4001) used in the sync amp was left off the parts list.

4) If you are using a filtered plugpack power supply, you do not need the bridge diodes or the 1000  $\mu$ F capacitor.

5) There may not be enough room on the PCB for some of the larger electrolytic capacitors specified for the audio amplifier. The simplified circuit below shows substitution values which will work perfectly well with a 57 mm speaker.

6) Mount a conventional 5K or 10K volume control pot on the front panel of the box along with the 57 mm speaker. After marking the speaker position, drill a suitable number of holes in the top of the box before gluing the speaker in place.

7) If the volume control is too sensitive, the input resistor to the audio amp (10K) may be increased to about 56K or more.



Audio amplifier for console

8) If there is interference on the screen, particularly associated with the audio, power the audio section of the interface from the console +12 volt supply. This is done by lifting the 33 ohm resistor off the +12 volt rail on the interface PCB and connecting it to the red wire from the console DIN cable from your modulator.

### Direct I/O Interface PCBs

PCBs for the Direct I/O Interface project are now available through the club for \$22. These boards are very high quality units with plated through holes which will save a lot of work as well as making the project more reliable. Rolf has also informed me that he has transformers capable of powering both the interface as well as a floppy drive.

### 32K Memory Expansion - Part II

Once again I must delay publishing the remainder of this article due to more urgent projects. Anyone wanting information on single chip 32K memory expansion should contact me personally.

### AT Disk Controller Kits

Disk controller kits are now available through the club shop. Here is your chance to build a double density controller for your TI at a very good price. The controller may be used in a PEB or in conjunction with the Direct I/O Interface system described late last year in the TND. Contact the club shop or Rolf Schrieber for more details on price and availability. See the article below for building instructions.

### Building The AT Disk Controller

Ten Peter Schubert designed Multi Function Cards have been purchased to date. Most of the users have indicated that they want to build the disk controller. This article outlines the steps required to build the double density disk controller. Next month I will describe how to add the remaining options: ie 32K memory, RS232 serial interface and the PIO printer interface.

Please note that all TTL chips (ie 74XX series) must be "LS" types except for 74245 which must be an "ALS" type.

Using the parts list and overlay diagram:

1) Mount and solder all of the passive components (resistors, potentiometers, capacitors, diodes, LED, 8 way DIP switch). Take care with the orientation of the electrolytic capacitors and the LED. The Single In Line (SIL) resistor packs must be inserted with pin 1 towards the top of the board. The capacitor associated with the 50K pot is 27 pF.

2) Solder all of the IC sockets next. Mount them so that the notch is facing the top of the card, ie away from the I/O Edge Connector. The components marked with an "x" are not required for the disk controller function. Mount also the 34 way male IDC connector for the disk drive control/data cable.

3) Mount the active components (transistor, voltage regulator, crystal). Use a heat sink on the voltage regulator and bolt the assembly to the PCB.

4) Plug in the IC chips as indicated on the diagram. Orientate all chips towards the top of the PCB. Note that the DSR (27C256 or 27C128) and the 74LS02 chips have one lead that does not plug into the socket. The DSR chip has pin 27 connected externally to pin 28. The 74LS02 chip has pin 3 connected externally to pin 7.

5) Connect the remaining links on the board as indicated. Solder a pin connector to each of the 5 points marked "Test Point".

Double check all soldering and connections. Two potentiometers need to be adjusted prior to using the card. As an oscilloscope is required to do this, it is recommended that prior arrangement be made with myself in order that this may be done at one of the meetings.

Note: Geoff Trott has found a hardware bug in the AT disk controller design concerning compatibility with RAMdisks. Geoff will be publishing a fix in the TND ASAP.

#### Parts For AT Disk Controller

1 MFC PCB	1 74LS138
1 34 pin PCB IDC	1 7416N
1 8 way DIP switch	1 7438N
2 40 pin IC sockets	3 1N914 diodes
1 28 pin IC socket	1 7805 regulator
3 20 pin IC sockets	1 PCB heat sink for 7805
1 18 pin IC socket	1 BC549 transistor
1 16 pin IC socket	1 red LED
11 14 pin IC sockets	1 8 pin 1K SIL
1 4.0 MHz crystal	1 6 pin 470R SIL
1 27C128 or 27C256 Eprom	1 10 pin 10K SIL
1 TMS2793NL disk control	2 5K trimpots
1 TMS9901 I/O control	2 1.2K resistors
1 TMS2114-45NL	1 220R resistor
1 74ALS245N	1 330K resistor
2 74LS244K	1 1K resistor
1 74LS74	1 4.7K resistor
1 74LS04N	8 0.1uF mono caps
1 74LS32N	1 0.22 uF cap
1 74LS30	1 4.7uF 6V TAG
1 74LS10	1 33uF 10V TAG
1 74LS27	1 0.01uF cap
1 74LS122N	1 27pF cap
1 74LS20	2 22uF 10V TAG
1 74LS02	

See component overlay on page 8

#### Coming Next

Next month I will describe how to add the remaining options to the MFC: 32K memory, RS232 and PIO ports. o

## From the Service Bench

by Geoff Trott

### AT Disk Controller Bug

Unfortunately, there is a hardware bug in all the disk controllers produced by Peter Schubert. The problem relates to the fact that the registers in the disk controller chip are accessed in memory mapped mode as addresses between 5FC0h and 5FFh. Only the even addresses are used and different addresses are used for reading and writing to the same register. The bug is there because the disk controller chip is not disabled when the DSR chip is disabled. That is, the disk controller should only respond when the CRU bit whose address is 1100h is set to 1. It is possible, with the existing circuit, to change the contents of the

registers in the disk controller chip at any time by writing to the addresses between 5FC0h and 5FFh. A read does not cause a problem because there is a buffer which is disabled, except if the CRU bit at 1100h is set, but for a write, random data will be written into the registers.

Now this will not generally be a problem unless you have a RAMdisk or other such device. This is because the addresses 5FC0h to 5FFh are in the DSR address space, which is normally occupied only by ROMs which are not written to but only read from. The RAMdisk, however, uses the addresses 5800h to 5FFFh to read and write to its memory which is used to simulate disks. So the worst thing you can do with an AT disk controller and a RAMdisk, is to copy files from a floppy disk to the RAMdisk. During this time the floppy disk remains selected for all the time that the data is written to the RAMdisk and data is written to all the addresses between 5FC0h and 5FFh. What usually happens to me in that situation is that a whole track on a floppy is destroyed and with DM1000 it is almost impossible to copy more than 1 file at a time. I am sure that everyone with this combination of RAMdisk and AT disk controller has these problems.

Do not despair, the solution is relatively easy. In both cases another chip is soldered on top of U10, a 74LS32. Then a track is cut and a few connections are made and all is well. I have been testing two mini-PE systems with RAMdisks and have had no problems with file copying using any disk manager program and any number of files. If you want Lou or me to do this modification for you, see us at a meeting or ring and arrange to send your disk controller by mail. Use certified mail, as I have had one parcel lost by Australia Post. If you want to do it yourself, I will describe how in the following paragraphs.

For the mini-PE system, locate the 74LS32 next to the 74LS30 near the I/O port connectors. Take a 74LS04 hex inverter and bend out all pins except pins 2, 7 and 14. Make these pins a snug fit when the 74LS04 is placed over the top of the 74LS32. Solder pins 2, 7 and 14 to the pins on the 74LS32. On the top of the board, solder a piece of wire from pin 1 of the 74LS04 to pin 4 of the 74LS20, close by which there is also a resistor to ground connected to it. Turn the board over and locate the track that runs from pin 11 of the 74LS32 to a via (plated through hole) through the board. Make sure that you have the correct track and carefully cut it with a sharp (Stanley) knife. Connect pin 11 of the 74LS32 to pin 1 of the 74LS32 with a piece of wire. With another piece of wire, connect pin 3 of the 74LS32 to the via mentioned previously. That is all there is to it, using a spare gate in the 74LS32 plus one inverter.

For the multi-function AT card for the PEB, it is almost as easy but there is no spare gate in the 74LS32, so a 74LS00 must be used. Once again locate the 74LS32 in the middle of the board next to the 74LS30. Bend all the pins of the 74LS00 out except for pins 7 and 14 which need to be set at 90 degrees to the plastic body of the IC. Solder the 74LS00 to the top of the 74LS32 using only pins 7 and 14. Make sure all other pins are bent well out but do not break them off. Cut the track from pin 11 of the 74LS32 to a via (plated through hole) close to pin 1 of the TMS9901. Using a length of wire, connect this via to pin 8 of the 74LS00. Using another short piece of wire, connect pin 11 of the 74LS32 to pins 12 and 13 of the 74LS00. With another short piece of wire, connect pin 11 of the 74LS00 to pin 10 of the 74LS00. Finally, with another piece of wire, connect pin 9 of the 74LS00 to the via at about 4 o'clock from pin 21 of the 2793 IC (where 12 o'clock points to pin 40), the nearest unused via in that direction. There is one other via close by and not used which is further away from pin 21 and one closer to pin 21 with one lead of a capacitor in it.

### RAMdisk data loss

When the power is turned off or turned on to a

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

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

The reason for going this way is that it is only one signal which goes to all memory chips and if it is disabled then the contents of memory cannot change! The write enable pulse comes through a tri-state buffer chip from the console I/O port. This means that when the power of the console is off the signal may well be dragged low and as power is applied to the buffer chip it is hard to ensure that the buffer does not go low for a period of time. I have tried out one circuit on Bob Relyea's RAMdisk which solves the problem on turn on but has problems on turn off. This means that the RAMdisk must be turned off before the console, to save the memory contents. I shall be coming up with a circuit which isolates the write enable for both turn on and turn off soon. I still have a RAMdisk from Don Dorrington to make "safe".

At this stage (11/2/90), I think I have solved the problem. I shall elaborate in the next issue of TND.

#### Console repairs

I have had a number of consoles over the Christmas period with problems. One was very simple, just a system ROM. Another one was quite interesting as it had a black screen but the console tester ran and showed VDP memory problems in the LEDs, but only gave a coloured striped screen. Changing the VDP processor seemed to fix it except after a few hours on the Diagnostic module and memory errors appeared. These looked like all bits of some addresses were not correct which pointed to the VDP processor again. Changed it again and then the screen started to break up after a short time, which made me measure the resistance of the inductance supplying the VDP processor and sure enough it was an open circuit. A new inductor installed and no more problems. I wonder if the failure of the inductor led to the demise of the VDP processor?

I finally found the problem with John Hagart's console. It was a bi-directional buffer chip (74LS245) in the 16 to 8 bit data multiplexer de-multiplexer. I

also had another console which would not run the console tester and this one turned out to have a poor buffer on the address bus (74LS367) which has happened before. On this one the processor went into a hold or wait state when the GROM was incorrectly addressed and for a while it looked like the processor might be dead. It pays to remove the GROMs and sound chip before condemning the processor.

Another interesting fault was in a CorComp mini expansion system in which the disk controller did not appear to work (would not load a LOAD file into Extended BASIC, for example) but tested out OK using Disk Manager 2. It turned out to be the 32K memory expansion which had a bad chip (1 of 4). The annoying thing was that the CorComp memory test found nothing wrong. However, changing the chips around did get it to find an error or two.

We have also obtained some floppy disk alignment disks (from Bob Keast) and have spent a few hours looking at various drives. One of Bob Relyea's drives which was giving him trouble had a bad spider which centres the floppy disk on insertion. Four of the eight fingers had cracks in them and were not working so the floppy was not being centred correctly giving rather random results. One of my drives had azimuth problems with one head and was out of alignment. Two of Rolf's had alignment and track 00 problems. Using the drive tester unit and a CRO it is quite easy to check a drive.



## More PEEKs and POKEs

by Rolf Schreiber

The Micropendium article in the March 1989 issue dealing with peeks and pokes fell victim to a formatter bug. So, courtesy of your friendly TisHUG editorial team, here are the correct versions:

```
CALL PEEK(8192,A,B,C,D) :: PRINT 256*(C-A)+D-B
```

will show the free memory in low memory after a CALL INIT or a CALL LOAD("DSKx.filename").

```
CALL PEEK(-31866,A,B) :: PRINT 256*A+B-41023
```

will give you free program space in high memory after a program has been loaded.

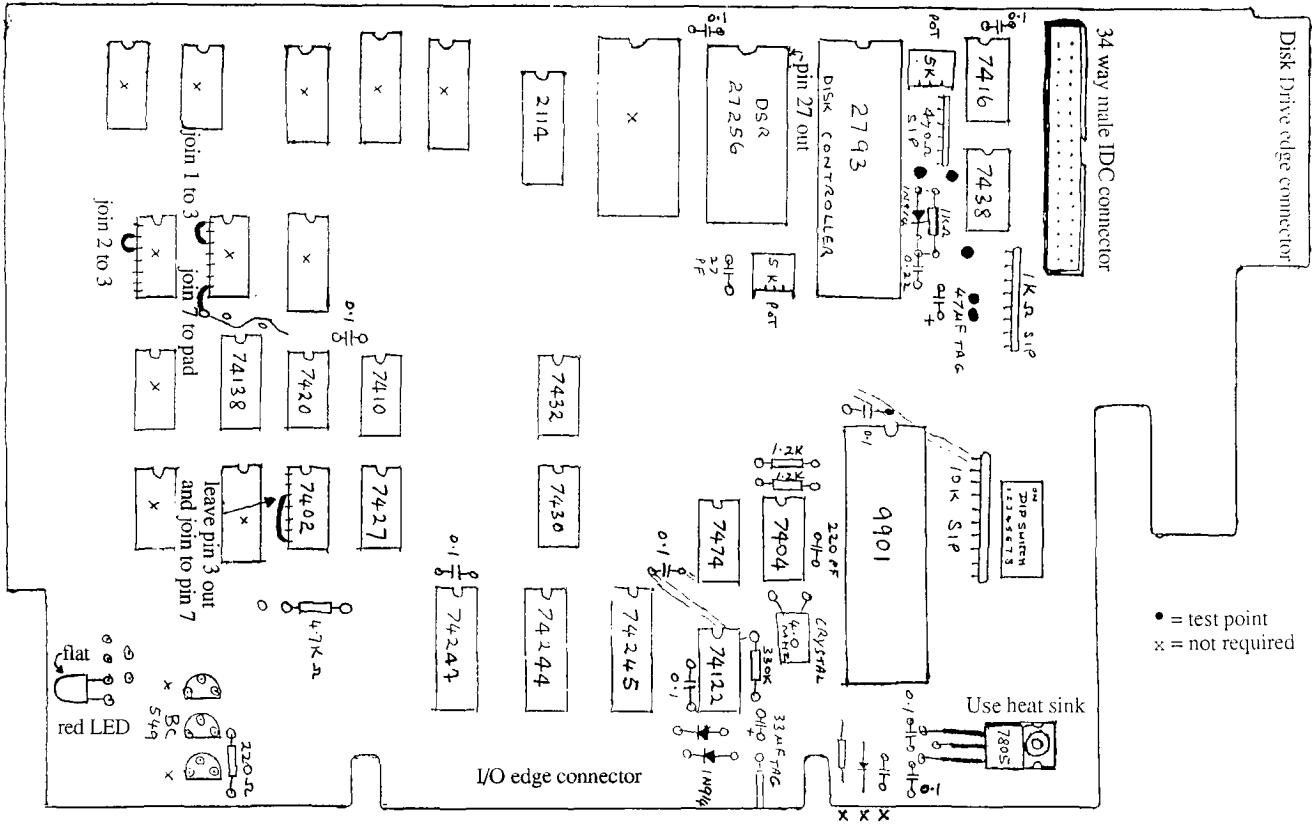
```
CALL PEEK(-31936,A,B) :: PRINT 256*A+B-2487
```

will show the exact amount of free stack space, while

```
CALL LOAD(-31873,X)
```

starts the printing at column X





AT Disk controller on multi-function Printed Circuit Board

continued from page 19

NINE T NINE USERS GROUP #109-2356 GERRARD ST. EAST TORONTO, ONTARIO M4E 2E2 CANADA (416)469-3468 3/89	TICC OF VANCOUVER 13675 HILTON ROAD SURREY, B.C. CANADA 6/87	INTERNATIONAL TI UG 96 BANBURY RD OXFORD OX2 6JT ENGLAND 12/88
OTTAWA TI-99/4A USER GROUP PO BOX 2144, STATION D OTTAWA, ONTARIO K1P 5W3 CANADA 1/89	CALGARY 99ERS. BOX 935 STA. T CALGARY, ALTA CANADA T2H 2H4 2/88	EAST ANGLIA REGION 99ERS UG SSGT D. COPELAND 2164CS PCS BOX 3596 APO NY 09755-5371 1/89
NORTH BAY 99ERS 159 DONALD DR. NORTH BAY, ONTARIO CANADA P1A 3H2 (705)474-9290 6/87	CLUB INFORMATION MONTREAL C.P. 686 STATION "C" MONTREAL, QUEBEC CANADA H2L 4L5 (514) 353-9091 8/87	TI99ER WORKSHOP RHEINLAND KARL-MARX-ALLEE 18 D-5000 KOELN WEST GERMANY 5/87
SASKATOON TICC P.O. BOX 7925 SASKATOON, SASKATCHEWAN CANADA S7K 4R6 (306)244-1394 5/87	MAURICE 99 930 DES SAULES TROIS-RIVIERES, P.Q. CANADA G8Z 2K5 (514) 374-9638 12/87	REX-SOFT UG C/O MIKE ROHRMOSER NIETZCHEWEG 16 4300 ESSEN 14 WEST GERMANY
SUDBURY 99ERS C/O JIM McLAREN SITE 1, BOX 7 WHITEFISH, ONT CANADA P0M 3E0 (705)522-5665 9/88	THE NEW 99ER USERS GROUP 216 10TH AVE. NEW WESTMINSTER, B.C. CANADA V3L 2B2 1/89	VERENIGING TI-GEbruikersGroep C/O ERIK VAN WETTE KREMERMATEN 106 7511 LC ENSCHEDE NETHERLANDS 2/88
WESTIG 2920 ST. PATRICKS WINDSOR, ONTARIO CANADA N9E 3G7 (519)966-6977 11/88	RED DEER TI 99ERS 5924-51 AVE RED DEER, ALBERTA CANADA T4N 4H9 (403)347-4183 3/89	PROGRAMBITEN (NITTINIAN) C/O SCHIBLER WAHLBERGSGATEN 9NB S-121 46 JOHANNESHOV SWEDEN 12/88
WINNIPEG 99/4 UG 1105 CHURCH AVE. WINNIPEG, MANITOBA CANADA R2X 1G1 (204)586-6889 9/88	ALMONTE 99ER UG 18 BROWN ST. CARLTON PLACE, ONT CANADA K7C 3W3	COMPUTERCLUB WEST99 PO BOX 8897 S-402 72 GOTHENBURG SWEDEN 7/88
	TI*MBS (for newsletter exchange) STEPHEN SHAW 10 ALSTONE RD STOCKPORT, CHESHIRE ENGLAND SK4 5AH 1/89	

# Extended BASIC Tutorial

by Tony McGovern, Funnelweb Farm

## VIII Extended BASIC program scrunching

Well, where do we go in the future? So far the series has had a detailed look at user SUBprograms and the ACCEPT AT statement, the two most powerful features of TI's Extended Basic, and also at the prescan switch commands lurking in the V110 manual addendum. For the next few sessions we will continue with topics which are of immediate relevance to console-only users, namely squeezing programs to fit in memory, and extracting maximum speed from XB. I can see no point, and have even less interest in writing about, say, SOUND or SPRITEs from the very beginning, as these are fairly well documented in the manuals and the subject of many books and articles. That is not to say that subtleties in using them will not come up from time to time.

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

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

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

The most obvious thing to do is to remove REMs from your program. I would suggest that this be left till later

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

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

100 CALL CLEAR

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

100 DISPLAY ERASE ALL

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

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

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

byte. I will not go into it here but you can use this general information to interpret the various time delays when you edit a line or enter a new line.

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

```
100 CALL CLEAR :: DISPLAY ERASE ALL
```

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

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

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

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

```
100 A=0
```

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

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

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

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

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

The next Tutorial will continue with the principles of program scrunching, getting more into the program writing end of things.

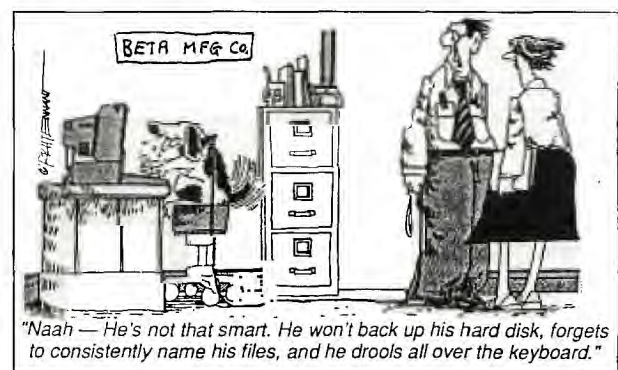
## For Sale

The following items are surplus to my requirements. They all come in their original boxes and with all documentation:

TI99/4A beige console	\$50
TI Extended BASIC module	\$40
Shuttle Modem (300 baud only)	\$40

### Wanted

PIO adapter for Peter Schubert RS232/Modem unit.  
Contact Jim Ellis on (064)92 1772 after hours.



# TI-Base Tutorial #7

by Martin Smoley, NorthCoast 99ers  
© Copyright 1988 by Martin A. Smoley

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

## \*\*\*Miscellaneous Information\*\*\*

This month I am going to try to answer some questions and include some new information from the new TI-Base manual. The first thing I would like to say is "read the manual several times". There is a lot of information in it that you will probably miss the first or even the second time you read it, "I know I did". Make the two readings at least a day apart. If you read it twice in a row, you will read in mistakes the first time through and rather than find the mistakes you will reinforce those mistakes the second time through. It is very apparent from letters I receive that some people try to start using TI-Base after just skimming through the manual. TI-Base is too complicated for you to bluff your way through. I have been working with it for many months and I still keep the manual close at hand, and refer to it or the new QUICK REFERENCE GUIDE constantly. As far as programming problems go, typographical errors are my biggest. If you enter LOCAL BLNK C 30, and later in the program you try REPLACE BLANK WITH "Good morning", TI-Base will not match BLNK with BLANK. The next big mistake I make is to try and concatenate too many characters together. I then try to jam them into a variable with too little space. If BLNK has 30 spaces available you must add up the characters you have in the fields you want to stuff into BLNK. It is easy to say REPLACE BLNK WITH FN | LN. FN and LN do not look like much but they may represent 25 characters each, for a total of 50. You must keep these things in mind or written down, the system will not give you any help with these problems. One important thing I try to remember is "do not try to write a big program in one chunk". You should be creating a library of small command files or programs. Each of these programs should do a fairly specific job, like printing labels or writing screen messages. Work on an individual command file until it works well and then use a bunch of these small programs together to do a larger job. If you are having problems with a command file, or a group of command files you are trying to use, leave the TALK ON. All the lines will scroll up the screen as you watch. If TI-Base cannot execute a line for some reason, it will place an asterisk at the very beginning of the line for you to see. If you cannot get the hang of this method, Version 2.01 has a TRACE command. It is simply TRACE ON/TRACE OFF. If you enter TRACE ON, all lines which TI-Base can execute will be sent to your printer. You can then compare that printout to your original command file and the lines that did not print out did not execute. When you find lines that WILL NOT execute, check them for variables first. See if you are trying to put too much data into a variable, or possibly the data is the wrong type. With version 2.0 and up I cannot seem to mix those C)haracter and N)umeric fields the way I could with 1.02.

I hate to admit it but on several occasions I found that I was trying to put things into a variable that I had not created or initialized at the beginning of the command file. If you are attempting to use a command or function for the first time and you do not understand how to phrase the statement, work small. Create a very small command file that does little more than test your new function. If you need to use a data base with this function, use one with only a few names in it. "I use T)NAMES, it contains only six names." That speeds up the test time and allows me to try the maximum amount of

variations in the least amount of time. When you are having problems with a command file and YOU WOULD like a hard copy to look at, there is a new command called LIST. If you type LIST DSK1.filename/C at the dot prompt, TI-Base will go to the disk number you entered and try to find the filename you want, remember the /C is needed. If it can find that filename, it will LIST it to your printer. I find this command and SNAP, which is the new TI-Base screen dump, very helpful. I seem to be able to find many small errors on a printer page that I overlook on the monitor.

## \*\*\*Printer Controls\*\*\*

Since I am already talking about printing hard copies, I would like to throw in some printer information. TI-Base Version 2.01 has built in printer control codes. They are present in the form of a data base on the PRGDISK, named PRINTER. If you LIST or EDIT the SETUP command file that comes with TI-Base you will see \*PRINTER EPSON in that file. Because there is an asterisk at the beginning of that line, TI-Base ignores the command and none of those control codes are available to you. That is because of the wide variety of printers on the market today. You must set up your own printer control codes in this system and make them available for TI-Base's access. Here is a little help on approaching this utility. We will look at the EPSON section because I am familiar with EPSON printers. Type USE DSK1.PRINTER <E>. The name PRINTER should come up in the bottom line across your screen as a normal data base in use. Next type DISPLAY STRUCTURE. Now, with your printer turned on, type SNAP <E> and hopefully the screen will be dumped to your printer. It should look like the screen below.

CREATED	CHANGED			
FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	NAME	C	010	
2	FF	X	002	
3	LF	X	002	
4	CR	X	002	
5	DS	X	004	
6	UL	X	006	
7	EX	X	002	
8	CM	X	004	
9	IT	X	004	
10	B	X	004	
11	SPS	X	006	
12	SBS	X	006	
13	HT	X	002	
14	ST	X	020	
15	NM	X	026	
16	BLANK	X	030	

•SNAP

000 1 PRINTER 00006/00007

You will notice that we are looking at a data base structure, even though it is a little abnormal. It is abnormalities are what make it so useful. TI-Base recognizes this data base as the place where it will find the control codes for your printer. The first field must be NAME. In the NAME field you would store your printer name, such as Epson. The next three fields you should recognize as Form Feed, Line Feed, and Carriage Return. These must stay exactly as you see them for TI-Base to work properly. The rest of the fields are Double Strike, Under Line, Enlarged Print, Condensed Mode, Italics, Bold, SuPerScript, SubScript, Horizontal Tab, Set Tab, Normal Mode (NM cancels all of the previous commands), and a Blank line 30 characters in length. One item of great importance is the X type field designation. The X type tells TI-Base that the item stored in this field is specifically a printer control code. It also tells TI-Base that it should interpret the normal numbers and characters you type into the X field as hexadecimal. If you USE DSK1.PRINTER and type EDIT, you will see the screen below. DIABO, "I assume for Diablo", is the first screen you will encounter. Pressing FCTN 5 will move you to the EPSON screen.

EDIT

```

NAME      EPSON
FF        OC
LF        OA
CR        OD
DS        1B47
UL        1B2D01
EX        OE
CM        OOOO
IT        1B34
B         1B45
SPS       1B5300
SBS       1B5301
HT        09
ST        1B440A0A0A0A0A0A0A00
NM        1B481B2D3014121B351B461B5>
BLANK     000000000000000000000000
    
```

OOO 1 PRINTER 0000/00007

Any of these screens can be edited to your needs. You can change the names to your own liking. For example you can change EX to ENL, for ENLARGED. You can also change the data in the fields to match the codes your printer expects, if they are not already set properly. This part is a little confusing, but not really hard. NOTE: If you change the length of a data field the data will be lost and must be re-entered. Make sure you understand what you are doing before you change any lengths. Changing the data and thus changing the control code is as simple as typing over what is currently in a particular field. Lets look at DS, which stands for Double Strike for Epson or Bold on the Star NX-10. The printer command for DS or Bold is <ESC> "G". What you see in the field is 1B47. If you wish to change this, you would merely type over it with normal characters and numbers. TI-Base, however, will interpret these to be Hexadecimal numbers.

The code for <ESC>, escape, is decimal 27, and in <Hex> or Hexadecimal, it is 1B. So the first two characters in the DS field stand for Escape (1B). The second half of the command is "G" or G. The decimal for G is 71 and the Hex for G is 47, so 1B47 is Hex for <ESC> G. In every printer handbook I can remember seeing, the ASCII, the Decimal and the Hexadecimal are given for the printer commands. You may have to search through your printer manual, but you do not have to figure these out, just look them up and type them in. One more example would be B for Bold, which actually produces Emphasized on my NX-10. My book says <ESC> "E" turns on Emphasized. Just below that it says 27 69 and below that it says 1B 45. If you check the EDIT screen, you will see that the B field contains 1B45. OK, how do you use this stuff. Well in most cases it is easier and more convenient than you might think. First EDIT the SETUP file. Remove the asterisk in front of the PRINTER EPSON. If you have an Epson, Gemini Star or a similar printer, try the EPSON command first. After using FCTN 8 to save the new command file, type DO DSKn.SETUP to rerun the SETUP file. As SETUP runs watch the bottom of your screen and you will see PRINTER come up while the EPSON commands are being installed. After SETUP finishes all traces of this operation will disappear, but the results will still be usable. Open the T NAMES data base by typing USE T NAMES. With your printer turned on type PRINT (B) <E>. If everything worked, you just set your printer to TI-Base Bold. Type PRINT ALL FN,MI,LM <E>. This should print out ALL of the names in T NAMES, in Bold Faced print. Typing PRINT (NM) <E> should return you to normal print, or DRAFT MODE. Try PRINT ALL (B) FN (CM) MI (NM) (IT) LN (NM) <E>. This should PRINT ALL the First Names in Bold, the Middle Initials in Condensed and the Last Names in Italics. Notice that the printer controls are enclosed in parenthesis and that a delimiter (comma or blank space) is needed between the fields or control codes. The <E> represents Press Enter. This type of control code use is very helpful when you are dumping a complete data base to the printer. You can type PRINT (CM) <E> and then PRINT ALL <E>. With TI-Base version 2.01 that troublesome line crawling that produced a printout that looked like a printer test pattern has been eliminated. Condensed mode now produces a nice neat columnar type

page. I hope this is enough to get you started. Remember, read that manual. There is more on printer controls in the manual and I will probably slip some of them into the tutorials in the future.

\*\*\*Geneve Myarc\*\*\*

Here is an important tip from Bob Stevens of Battle Creek, MI. Bob says he had some disastrous results on his Geneve with a Myarc Disk Controller when he tried to run command files off the PRGDISK. The cure he found was to initialize some new disks using 16 rather than 18 sectors and reset the interlace to 10. He then copied his TI-Base files to the new disks and lived happily ever after. Thanks for sharing that with us Bob, it will probably save someone in Florida a lot of aggravation. o

continued from page 3

That someone can be TISHUG, if members will show their support by purchasing NOT pirating new releases of commercial software, and generously contributing to collections for freeware authors, whenever the plate is passed around at TISHUG meetings. At the present time I have not yet taken over the software from Terry, so the "new" releases for March are based on some of my own library holdings. I would like to be able to cater for the broadest cross-section of users each month, by releasing software on a regular basis in the following categories:

- GAMES
- UTILITIES
- EDUCATION
- MUSIC
- HOME USE/FINANCE/BUSINESS/APPLICATION PROGRAMMES

If you read about any new software in a TI99/4A orientated magazine, and you think that you want to buy it, get in touch by phone or letter. We will require a small deposit upfront, then arrange to import the item for you. Since we can trade as a company, we can buy things at a dealer discount and the price to you will be less than if you imported the software yourself. o

## New Location for Sydney Meetings

from Ross Mudie

TISHUG is moving its regular meeting location to the RYDE INFANTS SCHOOL which is at the post office end of TUCKER St in RYDE.

Parking is available in the school grounds or across Tucker St in the shopping centre car park. Public transport is by train to West Ryde and then bus to Top Ryde or arrange with the directors to be picked up by car from West Ryde station.

The first meeting at Ryde will be 3rd March 1990. o

continued from page 4

I trust that the above will be of use to somebody. It has all greatly helped out my editing. For anybody interested in reading up on a few more TI-Writer tips I refer you to p. 22 of the January/February 1988 issue of the TND and p. 29 of the May 1989 TND. Both of these articles were written by Geoff Trott. Then, of course, there is the April, 1988 TND issue which was almost entirely devoted to TI-Writer. o

## For Sale

- TI99/4A Console - \$50, Speech Synth- \$50
- Mechatronics 128K memory expansion with Centronics Printer Interface - \$50
- TI Thermal Printer - \$20
- Extended Basic module - \$40
- Terminal Emulator II module - \$40
- Personal Record Keeper module - \$5
- The following modules at \$5 each: Mash, Connect Four module, TI Invaders, Buck Rogers.
- Joystick Adaptor with Joyball joystick, TI Joysticks-\$10
- Star DP8480 Printer with cable - \$50
- Contact: Jason Baker (043) 41 6073

# Rambles from the UK

by Stephen Shaw, England

Your letters and enquiries are always most welcome, especially those with questions or suggestions for Rambles! If you would like a direct response, please send a self addressed envelope and if you have not had a reply within two weeks, assume a letter has got lost somewhere! In past months two members have been sitting there for months waiting for a reply from me, and too polite to ask if I had heard from them! (Well, one member actually wrote six times, but was unaware that there was a local postal strike in his area with no letters getting out and although the strike is now over, his letters have still not appeared.)

My address is:

10 Alstone Road, STOCKPORT, Cheshire, UK, SK4 5AH.

Overseas correspondents can send two International Reply Coupons on a US\$5 note instead of a self addressed envelope .

I now have on disk and available for free copying (send blank disk and return postage!) a copy of Jim Peterson's "Tiger Cub Software" catalog and his "PD-Catalog", plus an index of Tips from the Tiger Cub (not of too much use to us as we do not have the early issues, and have edited the copy in any case!) and also from Jim, a list of TI99/4A User Groups known to him as at 3/89. A large number of them have not been heard of since 1987!

Remember my review of the Konix Speedking also known as Epyx 500XJ joystick two issues back. MICROpendium reviewed it in December 88 and gave it a final grade of A- at the US price of US\$20. In the same issue they reviewed CARFAX ABBEY by our very own David Vincent, available from our disk library, and marked it "send your money and buy it"; "four star in every way that I can tell"; not too many UK members taking the disk from our library but it is proving extremely popular in the USofA. It has a 100 room building to be explored, and Dracula to be vanquished, picking up treasures on the way.

Another "send your money in and buy it" review for Enhanced Display Package by Paragon Computing, which is also available from your disk library.

Not yet subscribing to MICROpendium? Just to help you out, they now accept credit card subscriptions, so no excuses OK? Sea mail sub is US\$25, airmail is US\$37, address is:

MICROpendium, P O Box 1343, Round Rock, TX, USA, 78680.

Refer to Barclaycard as Visa and to Access as MasterCard and give expiry date.

Fans of c99 will find some interesting articles in back issues of MICROpendium, which are available for subscribers only!

The February 89 issue of MICROpendium gave a "send your money and buy it" review for Ray Kazmer's "St Valentines Day Card", which we have in our disk library as Ray Kazmer's Maze of Grog, it is a modified version of the game of the same name, starring who else but Woodstock and further modified to play with joysticks or keyboard out of Extended BASIC.

The Bunyard Hardware Manual for the TI99/4A has already been mentioned by me, and recommended for the hard and software nut as well as anyone wanting a general read of what is, was and might have been. This is a lot more than a circuit diagram! Since last mentioned, US postal rates have gone up and they have also changed addresses. For your own copy, send US\$26, post paid, to:

Bunyard Group, PO Box 62323, Colorado Springs, CO, USA, 80962-2323.

John Guion has a number of hardware modifications for anyone prepared to delve in with a hot soldering iron, including a TI disk controller mod to allow use of a fourth drive, and access to DSK1 as a RAMdisk or dsk1 as a real disk, and also a modification for the Triton Extended BASIC module, and another mod for the RS232 card. He has also moved recently. If any of this sounds of interest send him a couple of IRCs (or a couple of \$\$\$) to:

PO Box 4628, Lubbock, TX, USA, 80223.

NEW ADDRESSES \*\*\*\* NEW ADDRESSES \*\*\*

And as we are on address changes, please note that Texaments moved a while back, and are no longer at Patchogue, but are now at:

244 Mill Road, Yaphank, NY, USA, 11980

New addresses for two Freeware authors, PLEASE NOTE:

Ray Kazmer: 8614 Foothill Boulevard, Apt 221, SUNLAND, CA, USA, 91040

Ken Gilliland: 7647 McGroarty Street, TUJUNGA, CA, USA, 91042

TIshUG in Australia reprinted a modified article taken from "Texas Weekly", April 1984, called "Death of a Computer" which details the death throes of our lovable machine. The details appear, with more detail, in The Orphan Chronicles which can be obtained from the LA group. See last issue of TI\*MES for details.

Several reports of the power supply to the PE Box cards being a little over stated voltage, stated voltage being +8, +16 and -16 volts, but measured in some cases at +11.5, +22.5 and -23.5 volts, not causing damage to the cards as they have their own voltage regulators, which generate heat. With a sparsely populated PEB there is not too much heat, so no problem, but with a PEB full of heavy duty cards, like those from Myarc, the heat generated is rather like having a 30 watt soldering iron stuck in the box! OK, it may work, but it will also work with less heat generated if you drop the voltage levels input to the cards!

One suggestion, which is easy if you can get the part required, is to use an autotransformer, with 240V input and a lower output, which is fed into the PEB, a variable output would allow you to set the auto-transformer to a level generating just the right voltages to the cards! One writer fed 180 volts into his PEB with no ill effects.

I would suggest that setting the PEB transformer for 240V input and actually sending in 210/220 volts would probably help with heat generation. The autotransformer could be quite small, 30 watts would be more than adequate.

Commercially, a cheap way to drop a voltage is with a series capacitor, try say a 20uF capacitor rated for at least 450V continuous (.4A) connected to the active line from the on/off switch. Two very important points: check the output voltage with the capacitor connected, if the measured voltage (with normal load connected) is too high, try a lower value capacitor. This method is potentially lethal to you due to the voltage which can be stored in a capacitor used this way. A very famous food processor manufacturer had to be reminded of this! You \*must\* place a resistor across the capacitor, to drain the voltage, 100k 2watt would be about right.

Also \*beware\* a capacitor used this way can make a hash of your PEB. When a capacitor like this fails, they can go !bang!, so enclose it in a separate sturdy well earthed metal box! Addition of a circuit breaker between the on/off switch and the capacitor would do no harm.

Capacitors are tricky things, and probably the most

often met failure point of consoles which have been unused for a few months. Transformer adjustments are more reliable! Modification of power supplies should be undertaken with very great care!

Infocom players may already be aware that SCRIPT and UNSCRIPT are used to direct output to a printer (and off again). Leather Goddesses has a further software switch you may wish to try. The default mode is SUGGESTIVE. The other one is LEWD. To avoid frequent use of LOOK for descriptions of locations as you REenter them, VERBOSE is another "software switch".

The Ottawa group have a circuit for building in 32K RAM using a single 32K RAM chip (and three 74 series logic chips), but Mike tells me that 32K chips cost more here than using four 8K chips.

Whatever happened to:

Richard Mitchell, publisher of The Smart Programmer, who owes subscribers several issues. He writes legal information bases for the IBM world.

Craig Miller of Millers Graphics. Raking it in with custom software for the...

A young man who had a hand in Burgertime, Demon Attack, Munchmobile, Jawbreaker, Facemaker, Treasure Island, Angler Dangler, MiniMem LBLA, and a few other things, who is not yet 30 years old, John Phillips?

Did you know that HOPPER was the only TI module to be written on a TI99/4A? The rest used a 9900 mini.

PRBase to TI-Base, from LA TopIcs.

Maximum record size is 132 bytes at a time! As TI-Base can have 5 databases in use at a time, you can split the records up a bit!

Use the report feature of PRBase to print to a disk file, those fields you wish transferred, in a tabular format. Maximum 132 characters remember!

Now use the convert feature of TI-Base.

If the report file has field one from column 1, for 28 characters, followed by a space before field 2 starts, then tell TI-Base convert that field one is 29 characters long (eg include the space!), then convert will be able to follow that field 2 starts at column 30!

And a belated welcome back to the Group to R J Bates, who asks what is the maximum size RAMdisk you can have? Any RAM device used as a RAMdisk has a maximum of 1600 sectors of disk space (400K) due to the way the disk bit map is used. In a Myarc 512k RAM card, the remaining RAM is used for a print spooler and also to replace the 32K RAM TI allowed for, or if you use Myarc's own Extended BASIC, to place the 128K operating system into! It is possible, if a device is configured to be divisible into several disk emulations, for each disk emulated to have up to the 1600 sectors maximum. Until recently if you bought an IBM PC and fitted a 60Meg hard disk, you had to divide it and pretend it was two 30Meg disks, all that IBM allowed!

The limitation of 1600 sectors is not placed by the disk controller card you have, which is usually bypassed for RAMdisk operation, but rather by the need to maintain compatibility with TI99/4A software, and of course, it reduces the amount of code the manufacturer actually has to originate.

In facing the decision to buy a RAM card, I decided on the Myarc RAM card, as it is simplicity to operate, enables me to use Myarc Extended BASIC, and with only one exception all programs run with it (Text to Speech does not like it), whereas the New Horizon RAMdisk

continues to baffle me with its operating system, and there are several programs I use that are unhappy with the HRD. However, a number of our members are happy with their HRD, so each has to make his own choice!

CALL SOUND reminder.

CALL SOUND does not produce a frequency with the tone you input! and some "musical" tones are impossible, leading to music which even the tone deaf can recognise as being ugh!

Example:

CALL SOUND(100,1657,0) will produce a note with exactly the same frequency as CALL SOUND(100,1681,0). Any frequency between these also produces the same note.

Example 2:

CALL SOUND(99,1661,0) should be note G sharp. It is 10% more sharp than it should be, while the next note:

CALL SOUND(99,1760,0) which should be A, is 15% flat. The period between the two notes is 25% out!!!

The rate of error grows as the frequency rises, due to the way in which the sound chip works. However even a few low frequencies are fairly way out, as D sharp (156 Hz), F (175 Hz) and A (220 Hz) are all 5% sharp.

So, if you hear a "bum" note in a music program, it might not be the programmer at fault, but rather the sound chip.

TI-Planner, another bug.

If you enter a formula which divides one cell by another, make absolutely certain that all the cells which are being divided have a non-zero value, otherwise the program seems to lose track of printer buffer space, and attempting to print causes a colourful lock out. Problem only applies to printing, and anyway you would not want to find out how many eights there are in zero would you.

The TI99/4A uses a TMS9900 chip which has 16 address lines and 16 data lines. The Geneve uses a 9995 chip which uses 16 address lines but only 8 data lines. Why is the Geneve, an 8 bit machine, faster than the TI99/4A, a 16 bit machine? Well, the TI99/4A operates at a leisurely 4 Mhz while the Geneve zips along at 12Mhz, and the TI99/4A has been hog-tied into using a mere 8 bit data bus for most of the time. It is possible to add your 32K RAM onto a 16 bit bus, and presumably some talented person could rebuild a TI99/4A entirely on a 16 bit bus?

Myarc Extended BASIC 2.12 bug.

```
100 DEFINT A1
110 DEFREAL A2,A3
120 A1=2000 :: A2=2000
130 A3=A1*A2
    WARNING : NUMERIC OVERFLOW
???
```

Cure:

```
130 A3=INT(A1)*A2
    Strange I know but it solves the problem.
```

## For Sale

TI99/4A console with all documentation  
Brand new NTSC TI99/4A console  
Transformer and modulator  
Peter Schubert 32K memory expansion and RS232 port  
Speech Synthesizer, Joystick adaptor, Widget board  
20 games cartridges (with documentation)  
Many tapes, 3 programming books  
\$250 the lot. Contact Paul Robinson on (043)59 1532.

# Games Info

by Robert Brown

Welcome to another games information. The time is the middle of the holidays, and I am going to do the HSC next year, so I am now writing a few articles for the future. So after this article I will not be writing any more until Dec 1990. I wish all my reader every success in 1990.

I had a phone from Evan Matchett who wanted to know some info on adventures in general. Well here we go.....

**ADVENTURELAND** - has the distinction of being the first of Scot Adams good puzzle adventure games. You have to find 13 treasures below the ground.

**PIRATE ADVENTURE** - build the pirate ship and sail to treasure island and find the treasure.

**VOODOO CASTLE** - Yours is the not so simple job of removing a curse from the Count Cristo. The game takes place in the rooms of Voodoo Castle. Your hotel, jail etc... Your mission is to find 13 treasures.

**GOLDEN VOYAGE** - you have 3 days to find the potion from the Fountain of Youth required to restore the aged King. You also receive enough gold to cover your mission. ie buy a sailing ship, needed to sail between 3 different islands. You might even meet the cyclops.

**SAVAGE ISLAND** - you try in this adventure to obtain the password that permits you to enter Savage island part II (a new program). To do this you must survive a tropical hurricane, a bear, and find a mysterious looking pirate, and do not forget to push the right buttons!

**RETURN TO PIRATE'S ISLAND** - Scot Adams also wrote this graphical adventure available in disk and module. You must try to return the island, avoiding the pirates on your way.

All the above adventures need the "Adventure Module"(except Return To Pirate's Island), disk loading and a 32K computer.

The following adventures are from Infocom. Infocom has a different style of adventures. They are again, text only, but have longer descriptions and also allow you to type more than 2 words.

**DEADLINE** - You, the detective, are called upon to investigate the apparent suicide of a wealthy industrialist. You encounter no mentor or wizards, just 7 characters. You can watch them move and talk to them as you like. You have 12 hours to unravel the mystery and arrest a suspect.

**ENCHANTER** - You, the wizard, are charged by your wizened mentor, Belboz, to seek and destroy the evil wizard Krill.

**SORCERER** - This adventure is a sequel to enchanter. It seems that your mentor, the aged and wise Belboz, has gone off on a secret mission to rid the world of an evil force. You must find him.

**SPELLBREAKER** - this adventure is the sequel to Sorcerer. It is also unpreviewed.

**INFIDEL** - You are a not to bright archaeologist bent on finding the vast treasures of an Egyptian Queens pyramid, somewhere near your camp in the sand. Beware!!! the queen has set lots of traps!!

**PLANET FALL** - After surviving an interstellar explosion (Superman??) you find yourself on a strange planet (Earth??), deserted except for a friendly robot named Floyd. The purpose of the game does not become clear until well into it, until you find a computer within a computer.

**STAR CROSS** - It is long ago and far away when suddenly the strident alarm of your spaceship advises you of an uncharted mass in space. Naturally you are curious You turn on some lights and encounter both living and dead remnants of preceding alien civilizations, advanced technology, religious superstition and past tragedy. Your adventure has just begun.

**SUSPENDED** - It starts as just another day in the life of the central mentality of a planet - that is you! Normally 3 computers maintain stability and equilibrium in the weather, food growing and transportation systems, but everything goes wrong, so you must use your wits, manual over-ride capabilities and six independently controlled robots to right the wrongs. Good luck!!

**WITNESS** - In L.A in 1938 the wife of a noted millionaire businessman has committed suicide. You are the detective assigned to investigate his concern over threats made on his life. In the midst of your investigation, he is shot! Who did it?? The butler? The daughter?? Or the wife's lover and how???

**ZORK I** - You start near a deserted house over a huge and wonderous underground cavern. 19 treasures await discovery. You will meet and deal with a troll, a thief and cyclops.

**ZORK II** - this adventure is called the Wizard of Frobozz and is more challenging than Zork I. It may have been written by terrorists as the adventurer needs to know how to use explosives and open locked doors without having to find you way around in the universe and make your way to the secret planet of Magnet.

**LEATHER GODDESS OF PHOBOS** - Rated R. Review withheld.

**LURKING HORROR** - this adventure has been made available on the TI through a transfer process. This adventure has not been released for the TI. Unpreviewed.

I hope Evan and my good readers can find an adventure to their liking.

See you soon,  
Robert Brown

## Sorcerer, part 1

A Walk Through, by Scorpia, USA

Well, you took care of Krill and returned to the Guild Hall in triumph, but things do not look good for you at the moment. Not when you are standing there facing the slaving jaws of a vicious hellhound! Fortunately for you, this time, it is only a dream, so just wait, and you will wake up in your room in the Guild Hall. Whew!

However, it is a bit dark in here, so Frotz your spellbook, get up, and head West into the hallway. Read the note on your door if you like, although it is not all that important. It just lets you know you are here alone. Now is a good a time as any to tell you that not all the spells you know will be useful in the game, nor all the potions you will find. So unless you are told specifically to take an item, you can safely ignore it.

Now, head along South down the hall until you reach the Lobby. Go West into the Store Room, and pick up the Ochre Vial and the matchbook. Open the vial and drink the potion, which will prevent you from becoming hungry and thirsty. Drop the now-empty vial. Read the matchbook if you like, then return East and place the matchbook in the receptacle.

Moving North twice, and West once, brings you to Belboz's quarters. Listen to the bird as you lift the wall hanging, revealing a key. Now examine the desk and open the drawer. Inside are several items, but the only useful one is the journal. Use the key to unlock the journal, then read it. Make \*careful\* note of the day's code!!

continued on page 20



# TI99/4A User Groups

by Jim Peterson, USA  
Updated March 1 1989

This is by no means a complete list, but is probably the only up-to-date list in existence, because all addresses are verified by newsletters or correspondence received, current listings in Micropendium and Computer Shopper, exchange mail lists furnished by UG's, etc. The last month of validation is indicated after the ZIP(POSTAL)code. Those with most recent dates are generally from return addresses on newsletters, and may be addresses of the presidents or secretaries as submitted to Computer Shopper, etc.

JIM PETERSON  
TIGERCUB SOFTWARE

NORTH ALABAMA (TI) 99/4A UG  
P.O. BOX 11204  
HUNTSVILLE, AL 35814 11/88

TI-BUGS  
1338 EDWARDS ST.  
DOLOMITE, AL 35061 11/88

SHOALS 99'ERS  
PO BOX 2928  
MUSCLE SHOALS, AL 35662 11/88  
766-2032

TIBUG BIRMINGHAM  
728 JEFFERSON BLVD  
TARRANT, AL 35127 11/88  
(205)849-5592

SMAUG  
2006 RIVERSIDE DR.  
MOBILE, AL 36605 11/88  
(205)867-7193

C.A.T.I. 99/4A UG  
PO BOX 11751  
MONTGOMERY, AL 36111 9/88  
(205) 284-1804

WIREGRASS 99ER UG  
102 AUBURN DRIVE  
ENTERPRISE, AL 36330 11/88

MOark 99ers  
2806 OSAGE DRIVE  
SPRINGDALE ARK 72764-6923  
(501)442-0856 1/89

JACKSONVILLE TI UG  
PO BOX 91  
JACKSONVILLE, AR 72076 11/88

SOUTHWEST NINETY-NINERS UG  
P.O. BOX 17831  
C/O WESLEY ENG  
TUCSON, AZ 85730 1/89  
(604) 747-5046

VAST 99 USER GROUP  
2120 S. LOS FELIZ DR.  
TEMPE, AZ 85282 2/89  
992-7668

ARIZONA USERS GROUP  
4328 E. LAPUENTE AVE.  
PHOENIX, AZ 85044 6/87

BREA 99ERS UG  
C/O KERRY GRISSOM  
16328 PLACID DR.  
WHITTIER, CA 90604 3/89  
(213)943-6631

KINGS 99/ER UG  
299 W. BIRCH AVE.  
HANFORD, CA 93230 5/87  
(209)582-1385

E.T. 99ER CLUB  
P.O. BOX 2752  
COSTA MESA CA 92628 5/87  
(714)855-0480

TI-SIG  
SAN DIEGO COMPUTER SOCIETY  
PO BOX 83821  
SAN DIEGO, CA 92138 2/89  
(619)264-6515

NOR-CAL TI UG  
P.O. BOX 112  
OCCIDENTAL, CA 95465 5/87  
(707)527-2474

NORTH COUNTRY 99ERS  
P.O. BOX 2500  
ESCONDIDO, CA 92025 11/88  
(619)747-3567

LA 99ER COMPUTER GROUP  
P.O. BOX 67A79  
LOS ANGELES, CA 90067-1079  
(213)271-6930 1/89

SADDLEBACK VALLEY UG  
PO BOX 2752  
COSTA MESA, CA 92628 3/89  
(714)855-0480

POMONA VALLEY 99ERS  
1833 E. PRINCETON ST.  
ONTARIO, CA 91764 6/87  
(714)988-9691

SAN GABRIEL VALLEY 99 UG  
1008 DORE ST.  
WEST COVINA, CA 91792 6/87  
(213)330-8240

TI RIVERSIDE USERS GROUP  
2372 ELSINORE ROAD.  
RIVERSIDE, CA 92506 2/89  
(714)686-0336

SNUG (SACRAMENTO 99ERS)  
7661 PLAID CIRCLE  
NORTH HIGHLAND CA 95660 11/88  
(916)726-8993

TI CLUB OF OXNARD  
2241 SANDERLING ST.  
VENTURA, CA 93003 1/89  
(805)984-3391

CENTRAL VALLEY TI99/A UG  
1914 REGIS DRIVE  
DAVIS, CA 95616 2/88

SAN FRANCISCO 99ers  
24816 MANGO ST.  
HAYWARD, CA 94545 11/88  
(415)786-1776

SFV 99ers  
2823 SANBORN AVE  
LA CRESCENTA, CA 91214 9/88  
(818)248-9558

AVTI USERS GROUP  
PO BOX 4101  
LANCASTER CA 93539 6/87  
(805)948-1587

SOUTHERN CALIFORNIA C.U.G.  
PO BOX 21181  
EL CAJON, CA 92021 3/89  
462-5802

THE TRI-VALLEY 99ERS  
760 WARWICK #17  
THOUSAND OAKS CA 91360 3/88  
(805)494-7246

99/4A UG OF ORANGE COUNTY  
17301 SANTA ISABEL ST.  
FOUNTAIN VALLEY, CA 92708  
(714)633-7317 9/88

SOUTH BAY UG  
PO BOX 23447  
SAN JOSE, CA 95153-3347 12/87

ROCKY MOUNTAIN 99ERS  
1825 E. 113TH AVE.  
NORTHGLENN CO 80233 11/88  
(303)289-7667

PUEBLO 99er USER GROUP  
809 EUCLID AVE.  
PUEBLO CO 81004 2/89  
(303)544-2504

FRONT RANGE 99ER COMPUTER CLUB  
PO BOX 9572 1/89  
COLORADO SPRINGS, CO 80932  
(716)976-3199

COLORADO USERS GROUP  
PO BOX 3400  
LITTLETON, CO 80161 6/87

NUTMEG NINETY NINERS  
139 GRASSY HILL ROAD  
WATERBURY, CT 06704-1234  
(203)875-1647 2/89

DELAWARE VALLEY USERS GROUP  
P.O. BOX 6240, STANTON BR.  
WILMINGTON, DE 19804 11/88  
(215)494-1372

NORTHWEST FLORIDA 99ER UG  
PO BOX 3641  
PENSACOLA FL 32516 9/87  
(904)944-2770

GREATER ORLANDO 99ERS UG  
PO BOX 1381  
MAITLAND FL 32751 11/88  
(305)774-0592

BREVARD USERS GROUP  
P.O. BOX 37211 3/89  
SATELLITE BEACH, FL 32937

BROWARD COMPUTER GROUP  
6264 NO. ANDREWS  
FT. LAUDERDALE, FL 33309  
(305)491-2886 3/89

COUNTRYSIDE (TI) UG  
48 SUGAR BEAR DR.  
SAFETY HARBOR, FL 33572 5/87  
(813)726-9192

WEST JAX 99ERS  
PO BOX 176  
ORANGE PARK, FL 32073 1/89

GREATER TAMPA BAY TI UG  
C/O PAUL WIESE  
4115 OKARA RD  
TAMPA, FL 33617 9/88

WEST PALM BEACH GROUP  
C/O STANLEY KRAJEWSKI  
103 SHOREVIEW DRIVE  
GREENACRES, FL 33463

MIAMI USERS GROUP 19301 NE 19 AVE N. MIAMI BEACH, FL 33179 (305)931-3644	11/88	N.W. SUBURBAN UG 1546 WILLIAMSBERG SCHAUMBURG, IL 60193	5/87	THE BLUEGRASS AREA UG P.O. BOX 11866 LEXINGTON, KY 48578-1866 (606)266-6392	12/88
MANASOTA 99/4A UG 3405 OAKWOOD BLVD S. SARASOTA FL. 34237 (813) 365-7234	3/89	WW99ERS 1053 CLARK ST. RANTOUL, IL 61866 (217)351-0943	9/88	KENTUCKIANA 99/4A COMP SOC PO BOX 36246 LOUISVILLE, KY 40233-6246 (502)491-2974	3/89
MARTIN COUNTY TI-99/4A UG 1200 STARFISH LANE STUART, FL 33494 (305)287-1760	11/88	MID ILLINOIS COMP RES ORGN P.O. BOX 766 BLOOMINGTON, IL 61701-766	6/87	BAYOU 99 USERS GROUP P.O. BOX 921 LAKE CHARLES, LA 70602	12/88
BRANDON TI USER GROUP 2512 LAURELWOOD AVE. VALRICO, FL 33594 (813)681-8954	10/87	KANKAKEE USER GROUP PO BOX 1945 KANKAKEE, IL 60901	6/87	BOSTON COMPUTER SOCIETY TI-99/4A UG ONE CENTER PLAZAD BOSTON, MA 02108	1/89
BRADENTON UG 5815 13TH ST. E. BRADENTON, FL 34203 (813)755-2439	5/87	FOX VALLEY USERS CLUB 34W 762 S. JAMES DR. ST. CHARLES IL 60174 (312)931-7887	2/89	MAGNETIC C/O JOYCE CORKER 30 MIDDLESEX CIRCLE WALTHAM, MA 02154 (617)326-6527	3/88
SUN COAST 99ERS UG 8421 WESTRIDGE DRIVE TAMPA, FL 33615 (813)971-9681	1/89	PEKIN UG 1904 AMERICAN ST. PEKIN, IL 61554	1/89	CLUB 99 34 FOREST ST. MAIL STATION 1-0 ATTLEBORO, MA 02703 (617)761-7245	11/87
UPPER PINELLAS 99ER UG PO BOX 3031 SEMINOLE FL 33542 (813)736-1616	7/87	THE WINDY CITY 99 CLUB 1549 WEBSTER LANE ROSEMONT, IL 60018-1423 ATTN: MICHAEL G. MICKELSON	11/87	PIONEER VALLEY 99/4A UG 111 OAKRIDGE ST. CHICOPEE, MA 01020 (413)593-6951	3/89
SPACE COAST 99'ERS P.O. BOX 35 MERRITT ISLAND, FL 32952	1/88	EAST CENTRAL USER GROUP R.J. WAHLFELDT 109 IOWA ST. DANVILLE, IL 61832 (217)446-7035	11/87	M.U.N.C.H. 560 LINCOLN ST. WORCESTER, MA 01605 (617)852-3250	11/88
DAYTONA 99ERS PO BOX 15232 DAYTONA BEACH, FL 32017	6/87	HOOSIER USERS GROUP PO BOX 2222 INDIANAPOLIS, IN 46206-2222 (317)787-4184	1/89	T.I.-BUG PO BOX 24 MARYLAND LINE, MD 21105 (301) 343-1253	3/89
ATLANTA 99/4A COMPUTER UG PO BOX 19841 ATLANTA, GA 30325 (404)231-0992	3/89	MICHIANA 99/4A UG C/O DAVID L. FLOWERS 52836 SEARER DRIVE SOUTH BEND, IN 46635	6/87	OAKLAND TI COMPUTER CLUB C/O EUNICE SPOONER BOX 3720 WATERVILLE ME 04901	2/89
COLUMBUS TI99ERS 3735 ARREL DR. COLUMBUS, GA 31909 (404)3327-3347	5/87	MIAMI COUNTY AREA 99/4A UG P.O. BOX 1194 PERU, IN 46970 (219)563-2213	11/88	GREAT LAKES COMPUTER GROUP PO BOX 7151 ROSEVILLE, MI 48305 623-7926	1/89
MACON 99/4A UG 635 VILLA CREST AVE. MACON, GA 31206 (912)781-0829	3/89	THE FORT'S USER GROUP C/O PAT MURPHY 5319 TWILIGHT LANE FT WAYNE, IN 46835	5/88	MISSISSIPPI COMPUTER SOCIETY P.O. BOX 16417 JACKSON, MS 39236-0417 (601)856-2509	3/89
SAVANNAH COMPUTER UG 128 JACQUELIN DRIVE SAVANNAH, GA 31406	6/87	CEDAR VALLEY 99ER UG 288 WINDSOR DRIVE NE CEDAR RAPIDS, IA 52402 (319)377-4073	2/89	SP 99 USERS GROUP PO BOX 12351 ST. PAUL, MN 55112 (612)533-8494	9/88
ALOHA 99/4A U.G. 92-865 PALAILAI ST. MAKAKILO, HI 96707 (808)422-7731	2/88	CENTRAL IOWA 99/4A UG BOX 3043 DES MOINES, IA 50316 (515) 263-0796	2/89	ST. LOUIS 99ERS PO BOX 260326 CRESTWOOD, MO 63126-8326	1/89
BOISE 99ERS COMPUTER CLUB 6357 NO. PORTSMOUTH AVE. BOISE, ID 83703 (208)342-1396	9/88	MID AMERICA 99/4 UG 8726 MARTY LANE OVERLAND PARK, KS 66212 (913) 648-6380	8/87	JACKSON COUNTY 99ERS 4711 NORTHERN AVE. KANSAS CITY, MO 64133-2219 (816)229-0314	6/88
CHICAGO TI USERS GROUP PO BOX 578341 CHICAGO, IL 60657	9/88	ECHO COMPUTER UG RT 1 BOX 5 CAMBRIDGE KS 67023 (316) 467-2588	3/89	OZARK 99ERS 1342 EAST STANFORD SPRINGFIELD, MO 65804 (417) 887-9471	1/89
DECATUR 99ER UG P.O. BOX 726 DECATUR, IL 62525 (217)422-6933	2/89	KAW VALLEY TI UG P.O. BOX 8424 TOPEKA, KS 66608 (913)357-0402	1/88	OMAHA TI UG 3224 LINCOLN BLVD OMAHA NE 68131 (402)896-2221	1/89
TI UG OF WILL COUNTY P.O. BOX 216R ROMEDEVILLE, IL 60441 (815)741-3272	1/89	WICHITA AREA 99ERS 9525 E. CENTRAL AVE. WICHITA, KS 67206	6/87		

CROSSROADS 99'ER UG PO BOX 293 YORK, NE 68467	6/87	TWIN TIERS C/O H. WILHELM 254 LUCE ROAD GROTON, NY 13073	4/88	SOONER 99ERS C/O GARTH POTTS 12513 HICKORY HOLLOW DR OKLAHOMA CITY, OK 11/88 (405) 949-0111
SOUTHERN NEVADA UG PO BOX 26301 LAS VEGAS NV 89126-0301 (702)647-1062	1/89	CHARLOTTE TI 99 UG 3708 COUNTRY CLUB DR. CHARLOTTE NC 28205 (704)334-8302	9/88	CENTRAL OKLAHOMA 99ERS UG 307 WEST RICKENBACKER DR. 6/87 MIDWEST CITY, OK 73110-5646
NORTHERN NEVADA 99ERS 75 COUNT FLEET CT RENO, NEV 89502 747-7848	1/89	GUILFORD 99ER'S UG 3202 CANTERBURY DR. GREENSBORO, NC 27408 (919)674-2318	9/88	SONN PO BOX 13073 SALEM OR 97309 9/88 399-0865
NEW HAMPSHIRE 99'ERS UG INC. PO BOX 5991 MANCHESTER, NH 03108-5991	2/89	MAD HUGgers 509 REEVES DR. GRAND FORKS, ND 58201 (701)772-6180	2/89	EUGENE 99/4A USERS GROUP P.O. BOX 11313 EUGENE, OR 97440 3/89 (503)998-8590
NEW JERSEY USERS GROUP C/O BRET MUSSER 60 BROADWAY RDD WARREN, NJ 07060	3/88	MISSILE RANGE 99ERS 2718 TOPLEY AVE. LAS CRUCES, NM 88005	6/87	PORTLAND USERS OF NINETY-NINES PO BOX 15037 PORTLAND, OR 97215 2/89 777-1531
NORTHERN NJ 99ER UG PO BOX 338 KENVIL, NJ 07847 (201)647-2365	2/89	CIN-DAY USERS GROUP 416 PINEWOOD AVE. PIQUA, OH 45356 (513)773-5941	1/89	AIRPORT AREA COMPUTER CLUB PO BOX 710 CORAOPOLIS, PA 15106 11/88
NEWJUGNORTH PO BOX 84 DUMONT, NJ 07628 385-7205	2/89	CLEVELAND AREA 99/4A C. G. (joint newsletter) C/O DEANNA SHERIDAN 20311 LAKE ROAD ROCKY RIVER, OH 44116	1/89	CAPITOL AREA USERS GROUP c/o DAVID HULTBERG 1407 CONCORD RD MECHANICSBURG PA 17055 6/87
NORTH JERSEY TI UG 90 BLVD PEQUANNOCK, NJ 07440 (201)595-0816	3/89	NORTHCOAST 99ERS 18909 GLADSTONE RD CLEVELAND, OH 44122 (216)286-3179	9/88	NITTANY USERS OF TEXAS INSTR. 625 WILTSHIRE DRIVE STATE COLLEGE, PA 16803 1/89 (814)238-0396
CENTRAL GARDEN STATE UG C/O GLENN G. PEARSON 61 COUNTRY LANE HAMILTON SQUARE, NJ 08690 (609) 890-9337	1/89	GOLDEN CRESCENT 324-4388		CHESTER COUNTY TI UG 111 LAKEVIEW DR. EXTON, PA 19341 6/87 (215)363-9321
UPSTATE NY 99/4 UG PO BOX 13522 ALBANY, NY 12212	11/87	TI-CHIPS UG C/O JOHN PARKEN 4172 WEST 217 FAIRVIEW PARK, OH 44126 (216)331-2830	3/89	NINETY-NINER DATA EXCHANGE P.O. BOX 7160 YORK, PA 17404 5/87 (717)993-6514
MID-HUDSON TI-99/4A UG PO BOX 7298 NEWBURGH, NY	12/88	CENTRAL OHIO 99ERS 181 HEISCHMAN AVE. WORTHINGTON, OH 43085 (614)263-5319	1/89	CENTRAL PA 99/4A UG PO BOX 14126 1/89 HARRISBURG PA 17104-0126 (717)652-3702
CENTRAL WESTCHESTERS 99ERS C/O ART BYERSU 1261 WILLIAMS RD SHRUB OAK, NY 10508 (914)592-2080	2/89	OH-MI-TI 225 S. WHEELING OREGON, OH 43616 (419)693-7394	2/89	PITTSBURGH USERS GROUP P.O. BOX 8043 PITTSBURGH, PA 15216 1/89 (412)885-1502
LITI 99ER UG 93 MYERS AVE. HICKSVILLE, NY 11801 (516)587-5462	1/89	NEW HORIZONS UG 1690 IDLEWOOD ST. TOLEDO, OH 43615 (419)537-1454	12/87	SUSQUEHANNA VALLEY COMPUTER GROUP PO BOX 90 HUMMELS WHARF PA 17831 3/88 (717)275-5791
HUDSON VALLEY 99/4A UG RD 1 BOX 359-5 KINGSTON, NY 12401	9/88	CARNATION CITY 99ERS C/O GARY CARR 25869 HARTLEY RD BELOIT, OH 44609 (216)525-5451	5/87	ERIE 99'ER USER GROUP 2812 WEST 33RD ST. ERIE, PA 16506 12/87
QB 99ERS QUEENSBORO COMMUNITY COLLEGE BAYSIDE NY 11364-1497 (516)759-9328	1/89	PENN-OHIO 99/4A HCUG (POUG) 71 ELM ST. STRUTHERS, OH 44471 755-7691	1/89	LEHIGH 99'ER COMPUTER GROUP P.O. BOX 4837 * 1501 LEHIGH ST. ALLENTOWN PA 18103 1/89
ROCHESTER USERS GROUP C/O ROBT OSBORNE 34 CALIFORNIA DR. ROCHESTER, NY 14616 (716)663-0397	5/87	GREATER AKRON 99'ERS P.O. BOX 3201 CUYAHOGA FALLS, OH 44221	2/89	WEST PENN 99ERS RD#1 BOX 73A JEANNETTE, PA 15644 1/89 (412)271-6283
WESTERN NEW YORK 99ER UG 84 GLENHAVE N. AMHERST NY 14120 (716)778-9104	3/89	LIMA AREA TI UG PO BOX 647 VENEDOCIA, OH 45894	2/89	PHILADELPHIA AREA UG C/O DON ARSENAULT 1290 BUTTOWOOD DR. LANSDALE, PA 19446 1/89

YORK AREA 99'ERS UG BOX 7160 YORK, PA 17363	1/88	FOREST LANE UG PO BOX 743005 DALLAS, TX 75247 (214)328-9257	4/88	TTC TAC USER GROUP 812 SOUTH 132ND ST SEATTLE, WA 98168 542-1046	11/87
NORTH EASTERN 99ERS C/O HANK RANDALL 209 4TH ST PROVIDENCE, RI 02906 (401)274-1329	3/89	SAN ANTONIO AREA 99ERS PO BOX 2509 UNIVERSAL CITY, TX 78148	7/87	SEA-TAC TI ORPHANS 833-9244	
SC MIDLANDS 99ERS 645 W. FIVE NOTCH RD NORTH AUGUSTA, SC 29843-9319	11/88	SUN CITY TI-99/4A COMP. CLUB P.O. BOX 6966 EL PASO, TX 79906 (915)751-1967	6/88	FOX CITIES UG P.O. BOX 2553 APPLETON, WI 54913	1/89
SIOUXLAND 99ers 4604 BLUESTEM CIRCLE SIOUX FALLS, SD 57106 (605)624-4877	1/89	DALLAS TI UG PO BOX 29863 DALLAS, TX 75229	11/87	MILWAUKEE AREA 99/4 UG 4122 NO. GLENWAY WAUWATOSA, WI 53222	8/88
K-TOWN 99/4A UG 3506 GARDEN DRIVE KNOXVILLE, TN 37918 435-1667	1/89	SALT LAKE VALLEY 99ER UG 3818 W. 6540 S. WEST JORDAN, UT 84084 (801)966-7144	2/89	MADAREA 99ERS C/O WISCONSIN BLUEPRINT CO. 437 W. GORHAM MADISON, WI 53703 (608)255-8878	5/87
MUSIC CITY 99ERS PO BOX 24886 NASHVILLE, TN 37202 (615)889-5852	1/89	OGDEN TI99/4A UG 1603 CHILDS AVE. OGDEN, UT 84404	2/89	SHEBOYGAN AREA UG P.O. BOX 1151 SHEBOYGAN, WI 53082-1151 (414)564-2511	9/88
MID-SOUTH 99 UG PO BOX 38522 GERMANTOWN TN 38183-0522 (901)358-0667	2/89	WASHINGTON DC UG (joint newsletter address) PO BOX 267 LEESBURG, VA 22075 (301) 899-3707	3/88	TI SYDNEY UG (TISHUG) PO BOX 214 REDFERN, NSW 2016 AUSTRALIA	1/89
ATHENS TI COMPUTER CLUB C/O JOE SIMMONS ROUTE #2, BOX 608 ETOWA, TN 37331 336-7475	8/87	BALTIMORE USER GROUP (301) 256-6445		TIUP (PERTH) P.O. BOX 8083 PERTH, STIRLING ST. WESTERN AUSTRALIA 6001	1/89
CHATTANOOGA USER'S GROUP PO BOX 136 CHATTANOOGA, TN 37343	6/87	HAGERSTOWN-WILLIAMSPORT UG (301) 223-8014		TICHUG (CANBERRA) P.O. BOX 610 BELCONNEN, A.C.T. 2616 AUSTRALIA	2/87
AMARILLO 99/4A UG PO BOX 8421 AMARILLO, TX 79114-8421 359-0380	2/89	MONTGOMERY COUNTY TI UG (301) 493-4502		ADELAIDE TI COMPUTER CLUB 26 SUFFOLK AVE. BRAHMA LODGE SOUTH AUSTRALIA 5109	3/88
CENTRAL TEXAS 99/4A UG PO BOX 200246 AUSTIN, TX 78720-0246	6/87	TIDEWATER 99/4 U.G. 4701 ATTERBURY ST. NORFOLK, VA 23513 (804)826-6958	1/89	HUNTER VALLEY 99ER UG 6 ARCOT CLOSE TARRO, NSW 2322 AUSTRALIA	5/87
CORPUS CHRISTI 99ERS 534 VAKY ST. CORPUS CHRISTI, TX 78404 (512)852-4874	2/89	ROANOKE VALLEY 99ER UG BOX 12522 ROANOKE, VA 24026 366-0145	2/89	MELBOURNE USER'S GROUP 18 KARRAKATTA ST. VICTORIA 3193 AUSTRALIA	6/87
PASTOR'S USERS GROUP DRAWER O HEMPSTEAD, TX 77445	1/89	OLD DOMINION 99ERS UG C/O GARY SYDNOR 585 SIGNAL HILL RD APT 6B MECHANICSVILLE VA 23111	9/88	BRISBANE USER'S GROUP PO BOX 57 ASPLEY, QUEENSLAND 4034 AUSTRALIA	6/87
JUG-JSC UG C/O JOHN R. OWEN 2321 CORYELL ST. LEAGUE CITY, TX 77573	6/88	TACOMA 99ERS USER GROUP PO BOX 42383 TACOMA, WA 98442 (206) 848-6872	11/87	TI-CLUB DE BRUXELLES SECRETARIAT P. MOTTART AVENUE DE BIOLLEY, 79 1150 BRUXELLES BELGIUM	10/87
HOUSTON USERS GROUP C/O RICHARD LUMPKIN 10910 HIGH KNOB DRIVE HOUSTON, TX 77065	1/89	NOVA (99ERS OF VANCOUVER) PO BOX 508 VANCOUVER, WA 98655 693-7070	1/89	CHANNELL 99 HAMILTON UG 77 LAVINIA CRESCENT HAMILTON, ONTARIO L9C 5S8 CANADA (416)385-5576	9/87
NET 99ER HCUG P.O. BOX 534 HURST, TX 76053	11/88	PUGET SOUND 99ERS PO BOX 6073 LYNNWOOD, WA 98036 633-2174	1/88	EDMONTON USERS' GROUP PO BOX 11983 EDMONTON, ALBERTA T5J 3L1 CANADA	8/87
PARIS 99/4A UG C/O JERRY KEISLER 2221 COLLEGE DRIVE PARIS, TX 75460 (214) 785-5694	1/89	WESTERN WASHINGTON COMP. CLUB C/O BILL HOWDER 515 S. 3RD AVE. TUMWATER, WA 98502 943-6500	9/88	MONTREAL 99ER 5290 WEST BROADWAY MONTREAL, QUEBEC H4V 2A2 CANADA	5/87
		KITSAP 99ERS UG 774 NE PINECREST DR BREMERTON, WA 98310 479-2619	3/89		

continued on page 8

# Forth to you too! Session 6

Author unknown

My original intention had been to write a few tutorials for our local 99ers to get them on their way with Forth. That seemed easier than to explain the basics over and over again. And besides, what I had been explaining had already appeared in condensed form in Millers' "The Smart Programmer" anyway. If some questions came up, they were easily resolved at our meetings or by a local phone call. Now that these tutorials have been posted on COMPU-SERVE I receive calls from all over and it looks as if there are a number of details yet to cover.

The question being posed most frequently indicates that some of you are not sure about Forth screens and programs. So let's clear this up: Think of screens simply as a means to record programs, which are, however, not limited to a single screen but may occupy as many screens as necessary. As an example, let's assume the following: You have made a copy of the TI-Forth disk, booted -EDITOR, -COPY, -PRINT and -BSAVE because these will be the only ones needed for the program you are going to write. You BSAVED your autoboot starting on screen 22. The autoboot occupies screens 22 through 31. Screens 32 to 89 are now available for your program screens. (Remember, if you follow the 'TASK 22 BSAVE with . (dot) Forth tells you the next available screen after the BSAVE.) The idea is to prepare a working Forth disk with needed load options and then put one or more programs on it. We make the assumption that none of the unbooted load options will ever be needed for the programs on this disk, and therefore we can utilize the screens they occupy. Some people seem to have the idea that one Forth disk can do it all. That simply is impossible. You can have a collection of short routines on a disk but sooner or later you will run out of room if you try to maintain the original load option screens, unless you have two disk drives and put all your routines on a disk in drive 2.

The screens of a program are linked with the --> (load next screen) word which is placed at the very end of the screen. In this manner only nn LOAD is necessary to load an entire program (nn = number of the beginning screen). You will not find --> in Brodie's STARTING FORTH, however, it is part of the Controlled Reference Word Set of Forth-83.

The line numbers are for reference only. They are not to be equated with BASIC line numbers. (Line numbers are superfluous in Forth because there is no GOTO.) However, words are compiled sequentially, i.e. starting with the first word of the first program screen and continuing down each screen line-by-line to the very last word of the last screen. Each word is added to the dictionary provided that any words within its definition can be found there. For example, : INVENTORY IN-STORE IN-WAREHOUSE + ; will not compile unless both IN-STORE and IN-WAREHOUSE have been compiled previously. (For advanced users who should not be reading this: Yes, this is not quite true, but remember that this is for beginners.)

For reasons which I do not understand it seems to be an obsession with some Forth programmers to cram their screens with utter disregard for legibility and clarity. If a program might take 9 screens they use every means to condense it to one less. I can see placing two short words on one line if (and that is a capital IF) one additional line would make it go into the next screen. But otherwise it is not good Forth style and it certainly does not make it easy for a beginning student to understand the program. In general, make it a practice to start words at the beginning of a line, indent the following lines if it takes more than one line for the definition. In long programs I even place the words being defined on each screen within the parentheses on line 0 so that I can find them easily with INDEX.

The normal number base (the one you are put in once Forth is booted) is DECIMAL. Invoke HEX and you can enter your parameters in hexadecimal numbers, but do not use > to designate them as such. You can also use binary numbers, simply put your system in that base with 2 BASE ! or better yet, define a nice word like : BIN 2 BASE ! ; I also define : DEC 10 BASE ! ; so I can go from base to base by entering only 3-letter words. In this manner you can use Forth as a handy-dandy conversion calculator. nnn HEX . will display a decimal number converted to hex. Just do not forget to reset the base with either DECIMAL (or DEC as above). In the same manner you can use any number base (Octal by 8 BASE ! etc), Forth will do the rest. Note: HEX is usually invoked within programs when putting parameters for character or sprite definitions on the stack.

Some of you report encountering problems with the words AT and TOP. They are not standard TI-Forth words, instead John J. Volk, THE elder statesman of TI-Forth whose Data Disks have been distributed nation-wide, originated them to save wear and tear on his typing fingers (just kidding, John). They should be part of everyone's dictionary:

```
: AT GOTOXY ; (AT is a lot shorter than GOTOXY)
: TOP CLS 0 0 AT ; (same as Miller's PAGE)
```

Time for one more question: How do I get out of Forth? Well, that depends. If you are through for the day, just pull out your disk (you might enter FLUSH first to make sure there are no loose ends in the buffers) and turn off your system. If you are going to continue, enter MON. This will return you to the TI colour bar screen.

\*\*\* END SESSION 6 \*\*\*

continued from page 15

Go East to the hall again, South once, and West into Helistar's room. Here you will find a scroll of Gaspar. Take and Gnusto the scroll, then head back to the hall, and South to the Lobby. By this time, you have probably heard the doorbell chime, which means the mail is in. Open the receptacle, and inside is an issue of Popular Enchanting and an Orange Vial. Get the vial, then go East into the Library.

There is a scroll of Meef here. Get and Gnusto that one. Now, it is time to leave this place. If you have played the game before, you know that if you try going out the main entrance, you will be stopped. You may also know that if you DO NOT get out, you will eventually fall asleep and find yourself in the Room of Living Death, which is MOST unpleasant, and from which there is no escape.

So, now is the time to get a move on! Head down into the Cellar, where the trunk is. There are five buttons on the trunk, and each one is a different colour. The sequence to open the trunk is keyed to the code of the day in the journal, and will change with each game. The code itself is the colours of a monster from the Infotater. Whatever monster was noted in the journal, look it up on the Infotater, and make note of its colours, and what order they are in. Save the game (just in case, because if you make even one little error, you will not be able to open the trunk!), then press the buttons in the proper order as given in the Infotater. If you have done it right, the trunk lid will spring open by itself after you push the last button. If the trunk does not open, restore the game and try again.

Ok, so now you have the moldy scroll of Aimfiz, which is one of those complex ones that you can only use once. Actually, you only need to use it once, so that is no problem. All you have to do is figure out which person you want to exchange places with. Since there are not too many choices, it will not take you long to discover it is none other than Belboz himself. So, Aimfiz Belboz, and you are out of the Hall and on your way to visiting Jeearr!

"Sorcerer" is copyrighted 1984 by Infocom

This walkthru is copyrighted 1984 by Scorpia.

# Tips from the Tigercub #39

by Jim Peterson, Tigercub Software, USA

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

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

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.00 each plus \$1.50 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.00, deductible from your first order.

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

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

\*\*\*\*\*  
Nuts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.  
Nuts & Bolts No. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.  
Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

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

For descriptions of these send a dollar for my catalog!

Answer to last month's challenge - for the longest possible one-liner, run the following "program to write a program" -

```
100 OPEN #1:"DSK1.LONG",VARIABLE 163,OUTPUT
110 FOR J=1 TO 79 :: M$=M$&CHR$(149)&CHR$(130)
NEXT J:: M$=CHR$(254)&CHR$(254)&M$&CHR$(149)
&CHR$(0) :: PRINT # 1:M$ :: PRINT #1:
CHR$(255):: CLOSE #1
```

Then enter NEW, then MERGE DSK1.LONG, then LIST - over 34 lines long! But that one does not do anything, so try this -

```
100 OPEN #1:"DSK1.LONG",VARIABLE 163,OUTPUT
110 FOR J=1 TO 52 :: M$=M$&CHR$(162)&
"X"&CHR$(130) :: NEXT J :: M$=CHR$(254)&CHR$(254)
&M$&CHR$(162)&"X"&CHR$(0):: PRINT #1:M$ ::
PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
```

Again enter NEW, and MERGE DSK1.LONG, then RUN. You will get a message BREAKPOINT IN 32510 ( do not ask me why! Can anyone tell me?) but just enter RUN again. Then LIST it - over 24 lines long! Explanation? Programs are saved in token code similar to MERGE format code. The maximum length of a record is 163 bytes - which is why MERGE files are D/V 163. The token for RANDOMIZE is ASCII 149, for the double colon is 130. Repeating that 79 times takes only 158 bytes, plus one more RANDOMIZE, the two - byte tokenized line number and the mandatory ASCII 0 to end the record, totals 162.

Here is a spooky one for Halloween -

```
100 CALL CLEAR :: CALL MAGNIFY(4):: CALL SCREEN(2)
! The Blob by Jim Peterson
110 CALL CHAR(96,RPT$("3C7EFFFFFFF7E3C",4)):: J=-1
120 FOR L=1 TO 28 :: CALL SPRITE(#L,96,16,L*4+20,10,0,
L+18):: NEXT L
130 FOR L=1 TO 28 :: CALL MOTION(#L,0,L*J):: NEXT L
140 J=J*-1 :: GOTO 130
```

Wes Johnston published an unusual sprite 2-liner in the Charleston Area 99ers newsletter. It is based on a CALL LOAD which freezes all sprite motion until they are turned loose by another CALL LOAD -

```
100 R=PI*2/28 :: CALL CLEAR:: CALL SCREEN(2):: CALL INIT
:: CALL LOAD(-31806,96):: FOR I=1 TO 28 :: CALL SPRITE
(#I,46,16,96,128,COS(I*R),SIN(I*R)):: NEXT I
110 CALL LOAD(-31806,0):: GOTO 110
```

You might like to try adding my "jewels" to that -

```
100 FOR CH=33 TO 60 :: FOR A =1 TO 4 :: X=INT(8*RND+1)::
T$=SEG$("18243C425A667E81",X*2-1,2):: A$=A$&T$ :: B$=T$&
B$ :: NEXT A :: CALL CHAR(CH,A$&B$):: A$,B$="" ::
NEXT CH
110 R=PI*2/28 :: CALL CLEAR :: CALL SCREEN(2):: CALL
INIT :: CALL LOAD(-31806,96)::FOR I=1 TO 28 ::
CALL SPRITE(#I,32+I,INT(14*RND+3),96,12 8,COS(I*R)
,SIN(I*R)):: NEXT I
120 CALL LOAD(-31806,0):: GOTO 120
```

Also try CALL MAGNIFY(2)

And, here is a companion program to the TAKE AWAY in Tips #35 -

```
100 CALL CLEAR :: CALL TITLE(5,"ADD & CARRY")
!by Jim Peterson
110 DISPLAY AT(3,10):"COPYRIGHT":TAB(10);"TIGERCUB
SOFTWARE":TAB(10);"FOR FREE":TAB( 10);
"DISTRIBUTION":TAB(11);"SALE PROHIBITED"
120 CALL PEEK(-28672,A@):: IF A@=0 THEN 160
130 DATA FINE,NO,GOOD,UHOH,RIGHT,TRY AGAIN,
YES,THAT IS NOT RIGHT
140 FOR J=1 TO 4 :: READ RIGHT$(J),WRONG$(J):: NEXT J
150 FOR D=1 TO 1000 :: NEXT D :: CALL DELSPRITE(ALL)
160 CALL CLEAR :: CALL CHAR(95,"FFFF"):: CALL MAGNIFY(2)
:: RANDOMIZE :: CALL SCREEN(14):: FOR SET=5 TO 8 :: CALL
COLOR(SET,16,1):: NEXT SET
170 CALL CHAR(120,"E700420018007E0000E700420099423CE7004
20099423C00E7004218003C4200")
180 CALL CHAR(124,"0E000401000708007000208000E01000")
190 DISPLAY AT(3,8):"ADD AND CARRY" :: CALL CHAMELEON
200 CALL COLOR(14,2,2):: CALL HCHAR(4,4,143,2):: CALL
HCHAR(5,4,143,2):: CALL SPRITE (#25,120,11,25,25)
210 T=T+1 :: IF T=6 THEN T=0 :: GOTO 250
220 Z=INT(8*RND+2):: IF Z=Z2 THEN 220 ELSE Z2=Z
230 Y=INT(Z*RND):: IF Y=Y2 THEN 230 ELSE Y2=Y :: X=Z-Y
240 N=1 :: GOSUB 470 :: GOTO 210
250 T=T+1 :: IF T=11 THEN T=0 :: GOTO 290
260 X=INT(10*RND):: IF X=X2 THEN 260 ELSE X2=X
270 Y=INT(10*RND):: IF Y=Y2 OR X+Y<10 THEN 260 ELSE Y2=Y
:: Z=X+Y
280 N=1 :: GOSUB 470 :: GOTO 250
290 T=T+1 :: IF T=11 THEN T=0 :: GOTO 330
300 X=INT(90*RND+10):: IF X= X2 THEN 300 ELSE X2=X
310 Y=INT(90*RND+10):: IF Y=Y2 THEN 310 ELSE Y2=Y :: Z=X
+Y
320 N=2 :: GOSUB 470 :: GOTO 290
330 X=INT(900*RND+100):: IF X=X2 THEN 330 ELSE X2=X
340 Y=INT(900*RND+100):: IF Y=Y2 THEN 340 ELSE Y2=Y :: Z
=X+Y
```



# REF/DEF

by E.P. Rebel, Netherlands

## Program Purpose

REFDEF V3.0 will show you all the DEFs and REFs found in one of the REF/DEF tables from Editor Assembler, MiniMemory or Extended BASIC module. It offers you a convenient method to find out start names of assembly language object files (DEFs) or the missing library names (REFs) when linking files (for example from the C99 compiler) together. Whenever loading and running object files leaves to a message like "PROGRAM NOT FOUND" or "UNRESOLVED REFERENCES", REFDEF V3.0 will help you.

## History

In autumn 1985 I wrote a BASIC program that showed the contents of the REF/DEF table. It tested the module it was run from and showed the starting addresses of each DEF it found. As with any BASIC program, this first version of REFDEF was slow, but did the job quite well. In April 1986, the program was rewritten in assembly language. It had to find out if it was running from BASIC or LOAD and RUN and even started automatically with Editor Assembler or MiniMemory module. Using the user interrupt routine to start it automatically in Extended BASIC too, did not seem to work, because using KSCAN destroyed the returning information. REFDEF V1.0 did not display REFs properly either. It would not start easily when REFs were found.

REFDEF V2.0 came in January 1987. It would automatically start from all three modules, did show REFs the proper way but still had the same problems with starting when an unresolved REF was there. Like V1.0 it could be run again with a CALL LINK("REFDEF").

In June 1987 I decided to rewrite REFDEF again and after two evenings of programming and testing, REFDEF V3.0 is ready. The problems I had in finding out the REFs are gone, so now I can easily find which file I forget when linking C99 files. Restarting the program has changed and a PIO screen dump has been build in. By holding down the <CTRL> key the standard DEFs from Editor Assembler and MiniMemory can be shown too. In the future I hope to be able to change the program to work with CorComp and Myarc disk controller loaders too. But since I do not have them, you will have to wait, or change the program yourself.

## Running the Program

Because there is only one version (an uncompressed one) of REFDEF V3.0 loading and running the program is easy. REFDEF V3.0 will find out itself if it is running from BASIC (via CALL LOAD) or LOAD and RUN and which module is used, so you do not have to bother about those things. From BASIC (either Extended or TI BASIC with MiniMemory or Editor Assembler module plugged in) you type in:

```
CALL LOAD("DSKx.REFDEF")
```

When you use the MiniMemory or Editor Assembler LOAD and RUN option or even the same option from BEAX, you type in:

```
DSKx.REFDEF
```

The program will start automatically by using the user interrupt vector.

As REFDEF V3.0 is most useful in combination with other object files that contain REFs and/or DEFs, it is advisable to load these first.

## What do you See?

After loading REFDEF V3.0 it will find out which

module you are using. If it cannot find the proper flags used by MiniMemory, Editor Assembler or Extended BASIC it will display the message:

```
MODULE FLAGS NOT FOUND
```

I know Funnelweb V3.3 is one of the Editor Assembler simulators that has this problem, but another, BEAX does not. It is up to those programs to setup the >A55A flag at >2000 properly.

If REFDEF V3.0 has found one of the flags it will display its title and check the REF/DEF table. Sometimes this information may be destroyed and if so the following message will appear:

```
BAD TABLE FOUND
```

If everything is all right, (and why should it not), the contents of the REF/DEF table should be displayed. Suppose you loaded SAVE (from the E/A\*PARTB diskette) first, then the result is:

```
R SLOAD >A4E4 R SLAST >A4FO  
R SFIRST >A4EC SAVE >A3D6
```

What does this mean? As you probably will know, SAVE contains three REFs and only one DEF. REFs are displayed on the screen with an "R" before the name and DEFs without it. After the name you find a hexadecimal address. For DEFs this is the address where it resides. For REFs this is the start of a chain, but do not bother about that. Most of the time you do not want to know these addresses at all and are only interested in running a nice assembler program with a mysterious start name. One of the DEFs on the screen will be that name!

After you have found the start name you will most likely return to the loader. Be aware of the possibility of over writing code. As REFDEF V3.0 is a relocatable object file, precautions must be taken when loading AORG files before or after loading REFDEF V3.0. If a program crashes you have most likely encountered an AORG file.

If the REF/DEF table is too big to be displayed on one page, and I do know programs where it is, you can press a key to go on to the next page.

## Options

REFDEF V3.0 contains some options which can be used by pressing the right function keys. A special one is the <CTRL> key that can be hold down while loading the program. If you do so, the REF/DEF table of the Editor Assembler module will be extended with the standard DEFs. With the MiniMemory module the standard DEF table (found between >6FOE and >6FFD) will be displayed when holding down the <CTRL> key while loading. This key has no effect when using the Extended BASIC module.

Other options are:

```
FCTN[8](REDO): Rerun the program (useful when more than  
one screen is displayed or you used the  
<CTRL> key.  
CTRL[8](REDO): Rerun the program (notice that you hold  
down the <CTRL> key and the meaning of  
this you know).  
FCTN[9](BACK): Abort the program and return to the  
loader (this is useful with multiple  
screens too).  
CTRL[9](BACK): Abort the program and return to the  
loader (the difference with FCTN[9] will  
be explained later).  
FCTN[5](QUIT): You can guess the function of this key.  
FCTN[7](DUMP): Make a screen dump to a PIO printer.  
Only the names with addresses are dumped.  
This screen dump only works on standard TI  
RS232 cards because it does not use the  
DSRLNK method. Sorry about that.  
FCTN[4](ABORT): Abort the screen dump.
```

Any other key will display the next screen or return you to the loader.



### Returning and Re-starting:

As mentioned above there is a difference in using CTRL[8] and any other key to leave the program. When using CTRL[8], REFDEF V3.0 starts a small interrupt routine that checks the <FCTN> and <SHIFT> keys. If you do press them together REFDEF V3.0 will re-start. REFDEF V3.0 installs a LOAD vector as well. So after you leave the program you can re-start it by using the LOAD switch. Be sure you do not destroy the code when rerunning REFDEF V3.0.

A little bug in returning is that the last key you used will be accepted by the loader. I do not know how to avoid it, unless the program waits until you release the key. I do not want to examine it too, because I hope someone of you knows the solution and will write and tell me.

### Example

Suppose you use the MiniMemory module and loaded Debug. After this you type in DSKX.REFDEF and press <ENTER>. When loading is finished, the display will be like this:

\* MiniMemory REF/DEF TABLE V3.0 \*

Debug >BOBE

PRESS ANY KEY TO CONTINUE...

You now know where Debug starts (>BOBE) and notes this. Later on you will load a cassette program with Easybug and want to examine this program with Debug. To know which standard routines the program uses, you press CTRL[8] and FCTN[7]. You get a list of the routines with their addresses:

```

UTLTAB >7020  PAD    >8300  GPLWS  >83E0
SOUND  >8400  VDPDRD >8800  VDPSTA >8802
VDPWD  >8C00  VDPWA  >8C02  SPCHRD >9000
SPCHWT >9400  GRMRD  >9800  GRMRA  >9802
GRMWD  >9C00  GRMWA  >9C02  SCAN   >000E
XMLLNK >601C  KSCAN  >6020  VSBW   >6024
VMBW   >6028  VSBR   >602C  VMBR   >6030
VWTR   >6034  DSRLNK >6038  LOADER >603C
GPLLNK >6018  NUMASG >6040  NUMREF >6044
STRASG >6048  STRREF >604C  ERR    >6050

```

Now you quit the program and load the cassette file. To be sure you loaded the right one, you press the LOAD switch and yes, the start names NEW, OLD and LINES are displayed by REFDEF V3.0. Notice that Debug has gone and that is why you noted its start address. You leave REFDEF V3.0 and type in (you are still in Easybug) EBOBE. From this point you are in the debugger and you have to use your own fantasy what you are going to do.

### About the Source

REFDEF V3.0 is not the nicest piece of programming you can imagine. But because I tried to keep the code (object file) small, I had to use some non standard routines. I am sure this program is in the 5% range of programs that do not run on the Geneve computer. This is because I used direct access to keyboard and PIO by using the CRU commands. It makes the code short but not user friendly. If you do not possess a TI PIO card, I am sorry but hope you still like the program.

If you do not like it feel free to change the source. If you can improve it (for example by making it suitable for CorComp and Myarc loaders), please send your version to me.

### Public Domain, Freeware or Whatsoever.

This program may be duplicated in any form without notice of the author. You may distribute it via your users group or give it away to your friends. But please pass along the source and this documentation too. If you like the program do not send me \$10.00 (although I would not mind) but send me a program of your own that I

can distribute via the Dutch users group. Thank you.

### The Author's Name and Address

REFDEF V3.0 and this documentation were written by:

Eric-Paul Rebel  
Merelstraat 27  
1223 NR HILVERSUM  
The Netherlands  
Phone: 31-35832929

Do not ring me up when you do not speak Dutch and live in another continent because the time in Hilversum is different and my English, Spanish and Japanese are bad! My apologies for the bad English but I suppose you prefer it over the Dutch version.

0

## Commands for modified Formatter

by Robert Green, Canada  
typed by Larry Saunders

For those wanting to know the new commands for TI-Writer modified version of the Formatter.

```

.TX Sets text mode using your printers fonts.
.GR Sets graphics mode, allowing use of images and fonts.
.FI Fill command.
.NF No fill.
.AD Adjust command.
.NA No adjust .
.BP Break page.
.CE n Center. n= how many lines.
.IN n Indent. n= how many spaces.
.LM n Left margin. Sets the left margin to n. Note:
    starts from 0.
.RM n Right margin. Sets right margin to n. Note: ends
    at 79.
.LS Line space.
.SP n Spaces.
.PL n Page Length.
.CO Comment.
.IF Include file.
.BL n Blank Lines.
.LF Load Font.
.CS n Extra Character Spacing.
.DS ON/OFF Density On or Off (double)
.DB ON/OFF Double Width.
.TI ON/OFF Text Invert.
.II Include Image.
.IM Image Merge.
.IL n Image Left.
.IC Image Complete.
.IR Image Row. For example, .IR DSK1.LTBULB_I will
    print row of light bulbs.
.IP include Picture.
.GI ON/OFF Graphics Invert.
.FR ON/OFF Frame Graphics.
.FS n Frame Style.
.BB Begin Box.
.BE n Box End.
.BI ON/OFF Box Invert.
.OX Open External.
.CX Close External.
.IX include External.
.DF Dump File.
.KB n Keyboard entry. n=number of lines.
.ME Message to yourself.
.QU Quit. When working from keyboard.
.CD ON/OFF Condensed print.
.EX ON/OFF Expanded (double width)
.EM ON/OFF Emphasized print.
.2S ON/OFF Double Strike.
.IT ON/OFF Italics.
.NL ON/OFF Near letter quality.
.EL Elite. 12 characters per inch.
.PI Pica. 10 characters per inch.
.RS Reset. Returns all setting Defaults.

```

You can still use all CTRL[U] codes if you want. 0

# Nit-Wit

by Chris Lang, USA

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

Chris Lang,  
1905 Jackson Rd,  
Baltimore, MD. 21222 USA

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

## INSTRUCTIONS

### OBJECT OF THE GAME:

To be the first player to reach or exceed a predetermined point limit. This is a two-player game.

### EQUIPMENT REQUIRED

- 1 floppy disk or cassette tape containing the Nit-Wit game program (included)
- 1 instruction booklet (included)
- 1 TI99/4A computer console (not included)
- 1 colour monitor (or 1 r.f. modulator and a colour TV set) (not included)
- 1 Exceltec (or TI) Extended BASIC command module (not included)
- 1 disk drive (for disk version only) (not included)
- 1 disk controller card (for disk version only) (not included)
- 1 32K memory expansion card (for cassette version with disk system attached and turned on; also for disk version) (not included)
- 1 peripheral expansion box with peripheral expansion card (for disk version only) (not included)
- 1 cassette recorder with interface cables (cassette version only) (not included)
- 1 pair of joysticks (not included)

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

### PREPARATION:

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

### GAME DESCRIPTION:

Each of the two players, at the same time, traps letters from falling through drain pipes by maneuvering each of their respective stopper plugs into the pipes, trapping specific letters in succession, eventually spelling out the longest word they can make for point scores.

### START OF GAME:

After loading the program and entering the RUN command, the screen will remain blank for several seconds; and then the game's title (Nit-Wit) will appear down the right-hand side. At that time, you will hear a brief musical interlude. When the music ends, a series of questions, each requiring an answer, will appear. After answering these questions, by typing the answers on the computer and entering each answer, the game will be ready to play.

### SEQUENCE OF QUESTIONS

WOULD YOU LIKE TO WATCH SOME ANIMATION BEFORE PLAYING THE GAME? (Y/N) Y

There is an animation sequence lasting approximately 2 1/2 minutes which depicts a grid, with

letters floating in random directions above it. Also floating above the grid is a thingamajig, black in colour. This thingamajig will spell out the name "NIT-WIT" by catching each letter, in sequential order, landing on top of the letter, and then applying pressure which will smash the letter against the grid. It will then slide the smashed letter down the grid and off the screen.

When the above question appears at the bottom of the screen, simply press the enter key to view the animation, or type the letter N and press enter to skip over the sequence and play the game.

Note: If, while watching the animation, you should desire to skip over the rest of the sequence, just press any key and you will be able to start playing the game.

ARE YOU NIT-WITS READY TO PLAY THE GAME? (Y/N) Y

When the above question appears, type N and press enter if you are not ready; otherwise, just press enter.

PLEASE SELECT A POINT LIMIT FOR WINNING THE GAME (UP TO 900 POINTS IS ALLOWED).

When the above statement appears, select an amount from 1 to 900, to determine what the winning score will be, by entering the number(s) from your keyboard. The first player to reach that total will win the game.

After the above statement is answered, the game begins; so grab your joysticks and be ready to play!

### GAME SCREEN DESCRIPTION

After entering the desired point limit, the actual game screen will develop. On the screen you will see two groups of drain pipe openings. Each group of drain pipes will be grouped in a circle, six in each circle. Each drain pipe is shaped like the letter "Y" and looks as though the letter was split down the middle and separated slightly.

You should also notice that the screen is cut in half by a vertical line. During the play of the game, player #1 controls the left side of the screen and player #2 controls the right side of the screen. Likewise, joystick #1 is assigned to player #1 and joystick #2 is assigned to player #2.

In the middle of each group, or circle, of drain pipes you will see a figure shaped like a rubber stopper. The left stopper is assigned to player #1, is controlled by joystick #1, and is coloured green. The right stopper is assigned to player #2, is controlled by joystick #2, and is coloured orange.

Also, on the screen, you should notice that a group of six letters, one group for each side, or player, will begin to fall from the top, through the drain pipes, and then down off the bottom of the screen. These same letters will then wrap around the screen and begin to fall from the top again. The group of letters on the left side of the screen are assigned to player #1 and are coloured red. The group of letters on the right side of the screen are assigned to player #2 and are coloured blue.

### PLAYING THE GAME

As soon as the two groups of letters begin to fall from the top of the playing screen, each player must now try to form a word from his own 6-letter group (both letter groups contain the same six letters) in his head. After a player finds a word amongst the letters, he must then use the joystick to maneuver his stopper and position the stopper so that it plugs up the bottom of the specific drain pipe through which the first letter of the word which he has thought of will fall.

You may also have noticed that the drain pipes are arranged so that, on each side of the screen, three columns are formed, each column containing two drain pipes, one under the other. Two of the falling letters

from each group of six falling letters will fall, or pass through, the two drain pipes located in the column, or direction, of their fall. In other words, each of the three columns on each side of the screen will contain two drain pipes and two falling letters. Each of the falling letters will be falling at different rates of speed; therefore, one of the two letters in each column may pass by the other letter in that column during their fall through the drain pipes.

After positioning the stopper at the bottom of one of the two drain pipes through which the first letter of the word which a player has thought of will fall, he must then press the fire button on his joystick to plug up that specific drain pipe, thereby catching the first letter that falls into it and stopping its falling motion, trapping that letter in the drain pipe. When the fire button is pressed, a gong will sound, signalling to the player that the computer has detected a depressed fire button and the stopper will now trap the first letter that falls into that particular drain pipe. A different sound gong will be heard depending on which player's fire button was pressed.

If the fire button on both players' joysticks are pressed at the same time, the word "TIE" will appear briefly at the bottom of the screen. Whenever a fire button is pressed, both players' joysticks will remain inoperative until the computer determines the results, based on the location of the stopper on the screen. If a stopper is positioned too far below, above, or in some way, out of alignment with the bottom of a drain pipe, a low-pitched beep will be heard; and the stopper controlled by the player who pressed the fire button will return to the center of his side of the screen. That player must then try again from that center spot, or home position.

If a player has positioned his stopper successfully at the bottom of a drain pipe, and has pressed the fire button on his joystick, first a gong will be heard. Immediately, at the same time, both players' joysticks will become inoperative. Also, the other player's stopper will disappear until a letter falls into the drain pipe which was just stopped up. After the letter is stopped in its motion, it will change its colour to black; the stopper will change its colour to grey; and a number will appear above the drain pipe that was plugged up, letting the player know what position that letter will appear in the word he is trying to form. Finally, a new stopper will be assigned to each of both players and they will appear in their respective home, or center, positions.

Play now resumes with both players trying to trap more letters to form their desired words.

When a player finishes trapping letters, in order, to spell his word, he immediately presses the fire button while his stopper appears at his home position. The word he has formed will then appear under his player number near the top of his side of the screen, and a strange warbling sound will be heard signalling the end of that player's turn. At that time, a sequence of statements will appear at the bottom of the screen; each statement requiring input by the player.

Note: During each round of play, a drain pipe can be plugged only once and can contain only one letter at a time.

#### SEQUENCE OF STATEMENTS

PLAYER \_ HAS SPELLED A WORD! IS THERE A CHALLENGE?(Y/N) N

After a player has finished spelling a word, the above two statements will appear at the bottom of the screen. The first statement will tell which player has spelled his word. The second statement asks if the other player thinks that the word is illegal or spelled incorrectly. If that other player agrees, then type Y and press enter, otherwise, just press enter and the point value for that word will be added to the score of the player who spelled the word, in accordance with the scoring procedures. Both players' turns are now over

and a new set of letters will begin to fall, after the present letters and stoppers are cleared off the screen.

WHO WON THE BATTLE? (P=PLAYER; C=CHALLENGER) C

When a player spells a word and the other player challenges that word as being mis-spelled or not legal, or allowable, the above two statements will appear. Because the word is in doubt, a dictionary should be consulted to determine if spelling and usage is correct. If the word is found to be correct, type P and press enter; and the point value for that word will be added to that score of the player who spelled the word, in accordance with the scoring procedures. Both players' turns will now be over. If the word is not correct, simply type enter and two of the following statements will appear:

PLAYER \_ : YOU LOSE YOUR TURN!  
PLAYER \_ : FINISH YOUR WORD!!!  
THIS ROUND IS OVER!!!

When a word is challenged and found to be incorrect the first statement, above, will appear, telling the player that formed the word that he has just lost his turn. If the other player has not finished making his word, the second statement will also appear and that player may now finish his word without any competition from the player who just lost his turn. If, however, one player has already lost his turn during this round and the other player's word, when finished, is also challenged and found to be incorrect, the third statement, above, will appear. Both players' turns are now over and a new set of letters will fall, starting a new round of play.

----- END OF GAME -----  
PLAYER \_ WINS BY \_ POINTS!

When a player whose score reaches or exceeds the point limit that was predetermined at the beginning of the game, the above two statements will appear and the screen will turn red, signalling the end of the game.

#### SCORING

There is a score box near the bottom of each player's side of the screen. These score boxes will show, after each round of play is over, the total score, so far, for the game.

The point value for a correct word is totaled as such: the length of a correct word is multiplied by itself (squared). This point value is then added to the player's score, who completed the word, and becomes his total score, so far, for the game. Only a player with a correct, completed word will receive any point value for that round of play.

#### GETTING READY TO PLAY

For Disk Version Only: Make sure that the Extended BASIC command module is inserted into the command module port of your console. Turn on all peripherals, then turn on the console. When the master title screen appears, press any key to display the module's main menu. Now select Extended BASIC from the menu and wait until the word READY appears on your screen. Insert the Nit Wit program disk into disk drive #1. Type in the following statement: RUN "DSK1.NIT-WIT"

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

For all versions: be sure that the "ALPHA-LOCK" key on your computer's keyboard is in the off, or up, position after loading the game program.

```

90 CALL CLEAR
100 PRINT TAB(2);"**GRUNGE O
N CHEMISTRY**"
110 PRINT TAB(9);"CONTINUED"
120 PRINT : : : :
130 PRINT "WELCOME TO THE SE
COND HALF OF THIS INTELLECT
UALLY STIM-ULATING PROGRAM.
AS YOU"
140 PRINT "ALREADY KNOW, MOS
T OF THE QUESTIONS ARE APP
LICATION PROBLEMS. THEY AR
E NOT TOO"
150 PRINT "DIFFICULT, SO IF
YOU HAVE EVEN THE SLIGHTE
ST IDEA OF WHAT YOU ARE DOIN
G, YOU SHOULD DO FINE."
160 PRINT : : : :
170 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
180 CALL CLEAR
190 PRINT "TO MAKE SURE YOU
HAVE NOT LOST YOUR TOUCH,
HERE IS ONEMULTIPLE CHOICE A
S A WARM- UP."
200 PRINT "-----"
205 LET SUM=0
210 PRINT "1.METHANE HAS A
FORMULA WEIGHT ALMOST HAL
F THAT OF OXYGEN. USING THE
CONCEPT"
220 PRINT "STATED IN GRAHAM'
S LAW, HOW MUCH FASTER THAN
OXYGEN WILLMETHANE DIFFUSE?"
230 PRINT "A.1X
B.2X
C.4X
D.8X"
240 PRINT
250 INPUT "PLEASE ENTER THE
LETTER OF THE BEST ANSWER.
":B$
260 PRINT
270 IF B$="C" THEN 300
280 PRINT "YOU SHOULD HAVE C
OPIED DOWN THE FORMULA."
290 GOTO 320
300 PRINT "BRILLIANT!?"
310 SUM=SUM+1
320 PRINT : :
330 INPUT "TYPE 'ENTER' TO C
ONTINUE. ":A$
340 CALL CLEAR
350 PRINT "THE NEXT EIGHT QU
ESTIONS ARESTRICTLY PROBLEMS
(LITER- ALLY). THEY ARE N
OT THAT"
360 PRINT "DIFFICULT, BUT YO
U MUST SUP-PLY THE ANSWER. N
O RANDOM GUESSING!!!"
370 PRINT "-----"
380 PRINT "2.A GAS OCCUPIES
A VOLUME OF240 MILLIMETERS A
T 3 ATMOS- PHERES OF PRESSUR
E. WHAT"
390 PRINT "VOLUME WILL IT OC
CUPY AT 5 ATMOSPHERES OF PR
ESSURE? "
400 PRINT
410 INPUT "GIVE IT A SHOT. "
:A
415 PRINT
420 IF A<144 THEN 450
425 PRINT
430 PRINT "YOU SEEM TO HAVE
AN INBORN ABILITY TO WORK W
ITH RATIOS.CONGRATULATIONS!"
440 SUM=SUM+1
445 GOTO 460
450 PRINT "SORRY, BUDS, BUT
YOU ARE WRONG!!!!!"

```

```

460 PRINT : : : :
470 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
480 CALL CLEAR
490 PRINT "3.A VOLUME OF GAS
MEASURING 320 LITERS IS STO
RED AT 31 DEGREES CELSIUS.
IF THE"
500 PRINT "VOLUME DECREASES
TO 280 LITERS, WHAT IS T
HE NEW TEMPERATURE OF TH
E GAS? (DO"
510 PRINT "NOT FORGET TO CHA
NGE CEL- SIUS TO KELVIN DU
RING COMPU-TATIONS.)"
520 PRINT
530 INPUT "NO GUESSING! ":C
535 PRINT
540 IF C<264 THEN 550
544 IF C<=268 THEN 570
550 PRINT "I ALLOWED FOR A L
ARGE MAR- GIN OF ERROR, AND
YOU STILL GOT THE PROBLEM W
RONG. SHAME!!!"
560 GOTO 590
570 PRINT "YOU ARE A CREDIT
TO YOUR CLASS. I SHOULD N
OMINATE YOUFOR THE NOBEL PRI
ZE. MAYBE NEXT YEAR!"
580 SUM=SUM+1
590 PRINT : : : :
600 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
610 CALL CLEAR
620 PRINT "4.IF A GAS OCCUPI
ES A VOLUMEOF 200 MILLLITER
S AT 185 DEGREES FARENHEIT
, WHAT VOL-"
630 PRINT "UME WILL IT OCCUP
Y AT 203 DEGREES FARENHEIT
?"
640 PRINT
650 INPUT "LET'S TRY FOR ACC
URACY THIS TIME. ":D
660 PRINT : : : :
670 IF D<223 THEN 680
675 IF D<=224 THEN 690
680 PRINT "NO COMMENT. THE C
ORRECT AN- SWER IS APPROXIMA
TELY 224 MILLLITERS."
685 GOTO 710
690 PRINT "I AM PLEASED TO K
NOW THAT YOU CAN CONVERT F
ROM KELVIN TO FARENHEIT. GOO
D JOB!"
700 SUM=SUM+1
710 PRINT : : : : :
720 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
730 CALL CLEAR
740 PRINT "5.GIVEN 100 MILLI
LITERS OF AGAS MEASURED AT 1
7 DEGREES CELSIUS AND 380 M
ILLIMETERS"
750 PRINT "PRESSURE, WHAT VO
LUME, IN LITERS, WILL THE
GAS OCCUPY AT 307 DEGREES CE
LSIUS AND"
760 PRINT "500 MILLIMETERS P
RESSURE?"
770 PRINT
780 INPUT "THE CDRRECT ANSWE
R, PLEASE! ":D
785 PRINT : :
790 IF D<.151 THEN 810
800 IF D<=.153 THEN 830
810 PRINT "YOUR FUTURE DOES
NOT LOOK TOO BRIGHT! THE C
ORRECT AN- SWER IS APPROXIMA
TELY .152"
815 PRINT "LITERS. WHERE IS
YOUR CALCU-LATOR?"
820 GOTO 840

```

```

830 PRINT "ABSOLUTELY BRILLI
ANT! NO WORDS CAN EXPRESS
MY ADMIRA-TION FOR YOUR GRE
AT ABILITY."
835 SUM=SUM+1
840 PRINT : : : :
850 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
860 CALL CLEAR
862 PRINT TAB(4);"$FASCINAT
ING FACT$$"
863 PRINT : :
864 PRINT "TEST TUBES ARE GO
OD FOR ONLY ONE OF TWO T
HINGS:"
865 PRINT : : : :
866 PRINT "1.GROWING
BABIES
2.BLOWING
UP THE"
867 PRINT " CHEMISTRY
LAB"
868 PRINT : : : :
870 CALL CHAR(96,"404020100F
000000")
880 CALL CHAR(97,"02020408FO
000000")
890 CALL CHAR(98,"4040404040
404040")
900 CALL CHAR(99,"0202020202
020202")
910 CALL CHAR(100,"010000000
0000000")
920 CALL CHAR(101,"FF4040404
0404040")
930 CALL CHAR(102,"FF0202020
2020202")
940 CALL CHAR(103,"800000000
0000000")
950 CALL CHAR(104,"070000000
0000000")
960 CALL CHAR(105,"820404040
2010000")
970 CALL CHAR(106,"040202020
408F000")
980 CALL CHAR(107,"384482828
2443800")
990 CALL CHAR(108,"070810202
0201008")
1000 CALL CHAR(109,"80402010
10102041")
1010 CALL CHAR(110,"84483000
0000F008")
1020 CALL CHAR(111,"02010000
00000000")
1030 CALL CHAR(112,"00008040
201F0000")
1040 CALL CHAR(113,"00000102
04F80000")
1050 CALL CHAR(114,"44820100
00000000")
1060 CALL CHAR(115,"00000081
7E000000")
1070 CALL CHAR(116,"20408000
00304884")
1080 CALL CHAR(117,"1C224141
41221C00")
1090 CALL CHAR(118,"04040404
04040404")
1100 CALL CHAR(119,"20202122
24242424")
1110 CALL CHAR(120,"7E810000
00000000")
1120 CALL CHAR(121,"04038040
20202020")
1130 CALL CHAR(122,"20C00000
00000000")
1140 CALL CHAR(123,"00000000
01020404")
1150 CALL CHAR(124,"1F204080
00000000")
1160 CALL CHAR(125,"F8040201
00000000")

```

```

1170 CALL CHAR(126,"00000000
80402020")
1180 CALL CHAR(127,"03040810
10101008")
1190 CALL CHAR(128,"C0201008
08080810")
1200 CALL VCHAR(8,11,123)
1210 CALL VCHAR(8,12,124)
1220 CALL VCHAR(8,13,125)
1230 CALL VCHAR(8,14,126)
1240 CALL VCHAR(8,16,127)
1250 CALL VCHAR(8,17,128)
1260 CALL VCHAR(9,11,118)
1270 CALL VCHAR(9,14,119)
1280 CALL VCHAR(9,15,120)
1290 CALL VCHAR(9,16,121)
1300 CALL VCHAR(9,17,122)
1310 CALL VCHAR(10,11,111)
1320 CALL VCHAR(10,12,112)
1330 CALL VCHAR(10,13,113)
1340 CALL VCHAR(10,14,114)
1350 CALL VCHAR(10,15,115)
1360 CALL VCHAR(10,16,116)
1370 CALL VCHAR(10,17,117)
1380 CALL VCHAR(11,13,107)
1390 CALL VCHAR(11,14,108)
1400 CALL VCHAR(11,15,109)
1410 CALL VCHAR(11,16,110)
1420 CALL VCHAR(12,14,104)
1430 CALL VCHAR(12,15,105)
1440 CALL VCHAR(12,16,106)
1450 CALL VCHAR(13,13,100)
1460 CALL VCHAR(13,14,101)
1470 CALL VCHAR(13,15,102)
1480 CALL VCHAR(13,16,103)
1490 FOR X=1 TO 8
1500 CALL VCHAR(13+X,14,98)
1510 CALL VCHAR(13+X,15,99)
1520 NEXT X
1530 CALL VCHAR(22,14,96)
1540 CALL VCHAR(22,15,97)
1550 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
1560 CALL CLEAR
1570 PRINT "6.CONVERT TO STA
NDARD CONDI-TIONS: 1000 MILL
ILITERS OF GAS AT -23 DEGRE
ES CELSIUS"
1580 PRINT "AND 700 MILLIMET
ERS PRESSURE(EVEN IF A GAS A
T THIS TEM- PERATURE IS SLIG
HTLY RARE). "
1590 PRINT
1600 INPUT "USE THOSE BRAIN
CELLS! ":E
1610 PRINT : : :
1620 IF E<1000 THEN 1640
1630 IF E<=1010 THEN 1660
1640 PRINT "STUPID MISTAKE,
RIGHT? YOU SHOULD HAVE COME
UP WITH AN ANSWER BETWEEN 1
000 AND 1010."
1650 GOTO 1680
1660 PRINT "WONDER-CHILD! I
AM GLAD TO KNOW THAT THE OD
D TEMPERA- TURE DID NOT DIS
TURB YOU."
1670 SUM=SUM+1
1680 PRINT : : : : :
1690 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
1700 CALL CLEAR
1710 PRINT "7.SEVENTY MILLIL
ITERS OF OXYGEN ARE DISPL
ACED BY MER-CURY IN A EUUDIOM
ETER. THE "
1720 PRINT "MERCURY LEVEL IN
SIDE THE TUBE IS 30 MILLI
METERS HIGH-ER THAN THE LEVE
L OUTSIDE"
1730 PRINT "THE TUBE. IF THE
CURRENT TEMPERATURE IS 2
0 DEGREES CELSIUS AND THE
CURRENT "
1740 PRINT "PRESSURE IS 715
MILLIMETERS,WHAT IS THE NEW
VOLUME AT STANDARD CONDITI
ONS? "
1750 PRINT
1760 INPUT "SHOW ME HOW SMAR
T YOU ARE. ":F
1770 PRINT
1780 IF F<58 THEN 1800
1790 IF F<=59 THEN 1820
1800 PRINT "WHAT DO YOU THIN
K THAT GRAY MATTER IN YOUR H
EAD IS FOR? THE CORRECT ANSW
ER IS ABOUT 58.5."
1810 GOTO 1840
1820 PRINT "NO SMART REMARKS
THIS TIME... I REALLY AM G
LAD YOU GOT THAT PROBLEM RIG
HT."
1830 SUM=SUM+1
1840 PRINT : :
1850 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
1860 CALL CLEAR
1870 PRINT "8.SOME NITROGEN
IS COLLECTEDBY DISPLACEMENT
OF WATER IN A GAS-MEASURING
TUBE. HERE"
1880 PRINT "ARE THE ESSENTIA
L FACTS: GAS VOLUME, 45 M
ILLILITERS; LIQUID LEVELS IN
SIDE AND"
1890 PRINT "OUTSIDE THE TUBE
ARE THE SAME; TEMPERATUR
E, 23 DE- GREES CELSIUS; B
AROMETER"
1900 PRINT "READING, 732 MIL
LIMETERS. CONVERT THE VOLU
ME TO THAT OF DRY GAS AT ST
P."
1910 PRINT
1920 INPUT "I HOPE YOU DID N
OT MAKE ANY CARELESS MISTAKE
S. ":G
1930 PRINT
1940 IF G<38 THEN 1960
1950 IF G<40 THEN 1980
1960 PRINT "SO THAT WAS A LI
TTLE COM- PLEX? I HAVE SEE
N MUCH WORSE! CORRECT A
NSWER: ABOUT39."
1970 GOTO 2000
1980 PRINT "CONGRATULATIONS!
JOE-CHEMISTHAS INVADDED THE
COMPUTER SCIENCE LAB."
1990 SUM=SUM+1
2000 PRINT : :
2010 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
2020 CALL CLEAR
2030 PRINT "*****ELEBRA
TE*****"
2040 PRINT
2050 PRINT "YOU ONLY HAVE ON
E MORE PROB-LEM TO WORRY ABO
UT!!!"
2055 PRINT
2060 PRINT "*****
*****"
2070 PRINT "9.AIR IS COMPOSE
D OF THREE GASES: NITROGEN,
OXYGEN, ANDARGON. OUT OF 10
0 PARTICLES,"
2080 PRINT "78 ARE NITROGEN,
21 ARE OXY-GEN, AND 1 IS AR
GON. WHAT WOULD BE THE PAR
TIAL PRES-"
2090 PRINT "SURES OF THE NIT
ROGEN IN AIRAT THE ATMOSPHER
IC PRESSURE OF 760 MILLIMETE
RS?"
2100 PRINT
2110 INPUT "THINK ABOUT THIS
PROBLEM. (HINT: RATIOS) "
:H
2120 PRINT
2130 IF H<592 THEN 2150
2140 IF H<=593 THEN 2170
2150 PRINT "OK, MAYBE THAT W
AS A TINY BIT TRICKY, BUT
NOT REALLY. THE CORRECT ANSW
ER IS 592.8."
2160 GOTO 2190
2170 PRINT "DALTON WOULD BE
THRILLED, TO SAY NOTHING O
F BRONIS!"
2180 SUM=SUM+1
2190 PRINT
2200 INPUT "PRESS 'ENTER' TO
CONTINUE. ":A$
2210 CALL CLEAR
2220 PRINT "WHAT BETTER WAY
TO END THIS TEST THAN WITH A
MULTIPLE- CHOICE QUESTION?"
"
2230 PRINT "-----
-----"
2240 PRINT "10.WHICH OF THE
FOLLOWING CAN ACCOUNT FOR
THE FACT THAT YOU CAN SME
LL A SKUNK"
2250 PRINT "EVEN THOUGH YOU
MAY NOT BE ABLE TO SEE IT?"
2260 PRINT "A.GASES ARE IN R
APID MOTION B.GASES MAY BE C
OMPRESSED C.GAS MOLECULES
ARE FAR APART"
2270 PRINT "D.MOLECULAR VELO
CITY IS DIRECTLY PROPO
RTIONAL TO TEMPERATURE
E.A AND C"
2280 PRINT "F.A AND D
G.A,B,C,D"
2290 PRINT
2300 INPUT "FOR THE LAST TIM
E, ENTER THEBEST ANSWER. ":I
$
2310 PRINT
2320 IF I$="E" THEN 2350
2330 PRINT "THAT IS NOT THE
WAY TO END ATEST, DOUGHEAD!
THE CORRECT ANSWER IS 'C'."
2340 GOTO 2370
2350 PRINT "WAY TO GO! AT LE
AST YOU GOT THE LAST QUESTIO
N RIGHT!"
2360 SUM=SUM+1
2370 PRINT
2380 INPUT "PRESS 'ENTER' TO
FINISH!! ":A$
2390 CALL CLEAR
2400 CALL SCREEN(14)
2410 FOR SET=1 TO 8
2420 CALL COLOR(SET,16,1)
2430 NEXT SET
2440 LET A$="*DONE*"
2450 FOR X=1 TO 50
2460 PRINT TAB(2);A$;TAB(8);
A$;TAB(14);A$;TAB(20);A$
2470 NEXT X
2480 FOR DELAY=1 TO 2000
2490 NEXT DELAY
2500 CALL CLEAR
2510 CALL SCREEN(4)
2520 FOR SET=1 TO 8
2530 CALL COLOR(SET,2,1)
2540 NEXT SET

```

continued on page 32

# Jenny's Younger Set

DATE: 1/1/80

The program enclosed will give the music and lyrics to, "I FEEL THE EARTH MOVE", a song currently sung by DARLINA. I hope it is OK for the younger set.

Vincent Maker

```

100 CALL CLEAR
110 FOR I=0 TO 12
120 CALL COLOR(T,16,4)
130 NEXT I
140 DISPLAY AT(4,3)BEEP:"** I FEEL THE EARTH MOVE **"
150 DISPLAY AT(9,7):"Music and Lyrics by CAROLE KING."
160 DISPLAY AT(15,7):"Program by VINCENT MAKER."
170 FOR I=0 TO 500 :: NEXT T :: CALL CLEAR
171 DISPLAY AT(3,7):" I FEEL THE EARTH MOVE".
172 DISPLAY AT(6,1):"I FEEL THE EARTH MOVE UNDER MY
FEET. I FEEL THE SKY TUMBLING DOWN;"
173 DISPLAY AT(10,1):"I FEEL MY HEART START TO
TREMBLE WHENEVER YOU'RE AROUND."
180 CALL SOUND(83.5,220,0)
190 CALL SOUND(83.5,196,0)
200 CALL SOUND(187.5,175,0)
210 CALL SOUND(500,175,0,220,1)
220 CALL SOUND(500,175,0,220,1)
230 CALL SOUND(62.5,196,0)
240 CALL SOUND(62.5,175,0)
250 CALL SOUND(62.5,147,0)
260 CALL SOUND(125,175,0)
270 CALL SOUND(62.5,220,0)
280 CALL SOUND(62.5,196,0)
290 CALL SOUND(62.5,175,0)
300 CALL SOUND(250,220,0)
310 CALL SOUND(125,294,0)
320 CALL SOUND(125,220,0)
330 CALL SOUND(125,196,0)
340 FOR W=0 TO 62.5
350 NEXT W
360 CALL SOUND(750,175,0)
370 CALL SOUND(83.33,220,0)
380 CALL SOUND(83.33,196,0)
390 CALL SOUND(83.33,175,0)
400 CALL SOUND(250,220,0)
410 CALL SOUND(125,196,0)
420 CALL SOUND(125,175,0)
430 CALL SOUND(125,196,0)
440 CALL SOUND(125,175,0)
450 CALL SOUND(500,196,0)
460 CALL SOUND(125,147,0)
470 CALL SOUND(250,175,0)
480 CALL SOUND(125,196,0)
490 CALL SOUND(250,175,0)
500 CALL SOUND(250,220,0)
510 CALL SOUND(125,220,0)
520 CALL SOUND(375,220,0)
530 CALL SOUND(125,233,0)
540 CALL SOUND(125,220,0)
550 CALL SOUND(500,196,0)
560 CALL SOUND(125,294,0)
570 CALL SOUND(125,220,0)
580 CALL SOUND(250,294,0,175,1)
590 CALL SOUND(125,175,1)
600 CALL SOUND(250,220,0)
610 CALL SOUND(125,110,0)
620 CALL SOUND(125,147,0)
630 CALL SOUND(250,175,0,147,1)
640 CALL SOUND(250,147,1)
650 CALL SOUND(250,175,0)
660 CALL SOUND(125,175,0)
670 CALL SOUND(250,196,0)
680 REM PAGE TWO
681 DISPLAY AT(5,7)ERASE ALL:"MELLOW AS THE MONTH OF
MAY, OH, DARLIN', I CAN'T STAND IT WHEN YOU LOOK AT
ME THAT WAY. I FEEL THE EARTH "
682 DISPLAY AT(10,7):"MOVE UNDER MY FEET; I FEEL THE
SKY TUMBLING DOWN."
690 CALL SOUND(62.5,220,0)
700 CALL SOUND(62.5,196,0)
710 CALL SOUND(62.5,175,0)
720 CALL SOUND(62.5,147,0)
730 CALL SOUND(125,175,0)
740 CALL SOUND(125,196,0)
750 CALL SOUND(250,175,0)
760 CALL SOUND(500,220,0,147,0)
770 CALL SOUND(250,294,0)
780 CALL SOUND(250,220,0)
790 CALL SOUND(250,262,0,165,1)
800 CALL SOUND(250,165,1)
810 CALL SOUND(500,220,0)
820 CALL SOUND(125,131,0)
830 CALL SOUND(125,147,0)
840 CALL SOUND(250,175,0)
850 CALL SOUND(312.5,147,0)
860 CALL SOUND(62.5,147,0)
870 CALL SOUND(62.5,175,0)
880 CALL SOUND(62.5,220,0)
890 CALL SOUND(250,196,0)
900 CALL SOUND(500,175,0)
910 CALL SOUND(125,196,0)
920 CALL SOUND(125,175,0)
930 CALL SOUND(250,220,0,147,1)
940 CALL SOUND(125,294,0)
950 CALL SOUND(187.5,196,0)
960 CALL SOUND(62.5,220,0)
970 CALL SOUND(62.5,196,0)
980 CALL SOUND(62.4,147,0)
990 CALL SOUND(500,220,0,175,1)
1000 CALL SOUND(500,220,0,175,1)
1010 CALL SOUND(62.5,196,0)
1020 CALL SOUND(62.5,175,0)
1030 CALL SOUND(62.5,147,0)
1040 CALL SOUND(125,175,0)
1050 CALL SOUND(62.5,220,0)
1060 CALL SOUND(250,196,0)
1070 CALL SOUND(500,175,0)
1080 CALL SOUND(250,220,0)
1090 CALL SOUND(125,294,0)
1100 CALL SOUND(125,220,0)
1110 CALL SOUND(125,196,0)
1111 REM page three
1112 DISPLAY AT(5,7)ERASE ALL:"I FEEL MY HEART TO
TREMBLE WHENEVER YOU'RE AROUND. OOH,
DARLIN',WHEN I'M NEAR YOU AND YOU "
1113 DISPLAY AT(10,7):"TENDERLY CALL MY NAME,"
1114 REM VERSE 4
1115 DISPLAY AT(12,1):""
1116 DISPLAY AT(13,7):"I KNOW MY EMOTIONS ARE SOMETHING
I JUST CAN'T TAME. I JUST GOT TO HAVE YOU BABY.AH!
AH! YES, I FEEL THE"
1117 DISPLAY AT(18,7):"EARTH MOVE"
1120 CALL SOUND(750,175,0)
1130 FOR T=0 TO 62.5
1140 NEXT T
1150 CALL SOUND(83.33,220,0)
1160 CALL SOUND(83.33,196,0)
1170 CALL SOUND(83.33,175,0)
1180 CALL SOUND(250,220,0)
1190 CALL SOUND(125,196,0)
1200 CALL SOUND(125,175,0)
1210 CALL SOUND(125,196,0)
1220 CALL SOUND(125,175,0)
1230 CALL SOUND(500,196,0)
1240 CALL SOUND(125,147,0)
1250 CALL SOUND(250,175,0)
1260 CALL SOUND(125,196,0)
1270 CALL SOUND(250,175,0)
1280 CALL SOUND(250,220,0)
1290 CALL SOUND(125,220,0)
1300 CALL SOUND(375,220,0)
1310 CALL SOUND(125,233,0)
1320 CALL SOUND(125,220,0)
1330 CALL SOUND(750,196,0)
1340 CALL SOUND(125,220,0)
1350 CALL SOUND(125,196,0)
1360 CALL SOUND(750,175,0)
1370 CALL SOUND(125,196,0)
1380 CALL SOUND(125,175,0)
1390 CALL SOUND(250,165,0)
1400 CALL SOUND(125,294,0)
1410 CALL SOUND(125,220,0)
1420 CALL SOUND(250,262,0,165,1)
1430 CALL SOUND(500,175,0)

```

continued on page 30

# From the Bulletin Board

MAIL TO : ALL  
MAIL FROM : PETESAKE  
SENT ON Monday 04/12/89 at 14:06:44

I have a Mechatronics EPROM PROGRAMMER for sale. These are quite rare here and I am also including my extensive collection of ROMS, DSR'S and MODULE Roms on Diskettes with it. It has very easy to use menu-driven software and I will also include a few EPROMS for you to practice on. Asking \$150 for the lot. Call me on 358 5602. Peter Schubert.

-----  
MAIL TO : ALL  
MAIL FROM : NERK  
SENT ON Sunday 03/12/89 at 21:14:26

FOR SALE: One single sided single density disk drive in excellent condition. Also two console to cassette recorder I/O cables and the following modules: HOUSEHOLD BUDGET MANAGEMENT, PARSEC, ALPINE, TI INVADERS, DISK MANAGER, EXTENDED BASIC, all in original packing with manuals. Also original TI-WRITIK module and disk master in top condition.

Regards, NERK.

-----  
MAIL TO : ALL  
MAIL FROM : ABEL  
SENT ON Tuesday 09/01/90 at 20:38:16

\*\*FOR SALE\*\*

ONE HORIZON STYLE RAM DISK WITH 192K  
IN WORKING ORDER ALSO WITH ROS EPROM  
\$280.00,  
\$260.00 WITHOUT EPROM CONTACT 044 214274  
EVENINGS.

-----  
MAIL TO : ALL  
MAIL FROM : LOU  
SENT ON Saturday 06/01/90 at 23:09:16

I have a TI system available, complete with EXPANSION BOX. As I already have two complete systems I would like to sell one of them for \$400.  
I also have one spare console complete with POWER SUPPLY and MODEM for \$100. Also quite a number of cartridges.

Phone me at 626 8855 and ask for LOU.

-----  
MAIL FROM : LOU  
SENT ON Saturday 06/01/90 at 23:13:49

Can anybody advise me how I can transfer T.I.files to an ATARI 1040ST?  
I believe it is possible, but nobody can give me any information. As the ATARI 1040ST does not have any memory language built in, I thought it might be possible to transfer the memory from the TI BASIC and the EXTENDED BASIC into disk form to run TI programs on the ATARI 1040ST.

PHONE ME AT 6268855 ASK FOR LOU

-----  
MAIL TO : ALL  
MAIL FROM : LOU  
SENT ON Sunday 28/01/90 at 18:06:01  
For SALE  
TI CARD RS232 FOR EXPANSION BOX.  
MODEM CARD FOR TI EXPANSION BOX.  
RAM DISK 720 K  
TI KEYBOARD INCLUDING POWER SUPPLY AND RF CONVERTER  
EXTENDED BASIC CARTRIDGE  
TERMINAL EMULATOR CARTRIDGE  
PHONE LOU (B) 6398888 (H) 6268855  
NO REASONABLE OFFER REFUSED.

MAIL TO : ALL  
MAIL FROM : GREGAR  
SENT ON Saturday 27/01/90 at 12:20:18

For sale, for sale!! 1 fully expanded TI syst including KS232, CLOCKCARD with print buffer, 500 baud (with phone attached) modem, Corcomp DS/DD drive controller, 32K expansion, 3 DS/DD 5.25 drives (2 attached and 1 spare), 3 keyboards, heaps of software including Legends II and also the SCI-FI BBS program (2 different versions all working from Techie to TI-ML1) which also includes GAMEROCK, plus L/A, nTINA-ML1 and HEAPS of other progs on disk. Disk progs also include a cracked version of 4A/TALK, the premier term emulator prog for the TI. No reasonable offer will be refused. Please either contact me via TEXPAC BBS, SCI-FI BBS (646-4865) or by writing to Greg Hope, 90 Harrow Rd., Auburn, 2144.

Rgs, Greg.

-----  
continued from page 29

1440 CALL SOUND(500,220,0)  
1450 CALL SOUND(125,131,0)  
1460 CALL SOUND(125,147,0)  
1470 CALL SOUND(250,175,0,147,1)  
1480 CALL SOUND(250,175,0,147,1)  
1490 CALL SOUND(125,175,0,147,1)  
1500 CALL SOUND(375,196,0,147,1)  
1510 CALL SOUND(62.5,220,0)  
1520 CALL SOUND(62.5,196,0)  
1530 CALL SOUND(62.5,175,0)  
1540 CALL SOUND(62.5,147,0)  
1550 CALL SOUND(125,175,0)  
1560 CALL SOUND(125,196,0)  
1570 CALL SOUND(250,175,0)  
1580 CALL SOUND(500,220,0,147,1)  
1590 CALL SOUND(250,294,0)  
1600 CALL SOUND(250,220,0)  
1610 CALL SOUND(250,294,0,165,1)  
1620 CALL SOUND(500,220,0,165,1)  
1630 CALL SOUND(125,131,0)  
1640 CALL SOUND(125,131,0)  
1650 CALL SOUND(250,175,0)  
1660 CALL SOUND(312.5,147,0)  
1670 CALL SOUND(62.5,147,0)  
1680 CALL SOUND(62.5,175,0)  
1690 CALL SOUND(62.5,220,0)  
1700 CALL SOUND(250,196,0)  
1710 CALL SOUND(497,175,0)  
1720 CALL SOUND(125,196,0)  
1730 CALL SOUND(125,175,0)  
1740 CALL SOUND(250,220,0,147,1)  
1750 CALL SOUND(250,147,1)  
1760 CALL SOUND(125,196,0)  
1770 CALL SOUND(125,220,0)  
1780 CALL SOUND(62.5,262,0)  
1790 CALL SOUND(62.5,220,0)  
1800 CALL SOUND(62.5,196,0)  
1810 CALL SOUND(62.5,175,0)  
1820 CALL SOUND(375,175,0,131,1)  
1830 CALL SOUND(125,131,0,147,0)  
1840 CALL SOUND(500,131,0,147,0)  
1850 FOR Y=0 TO 497  
1860 NEXT Y  
1870 CALL SOUND(250,147,0)  
1880 FOR T=0 TO 250  
1890 NEXT T  
1900 CALL SOUND(250,175,0,117,1)  
1910 CALL SOUND(250,196,0,117,1)  
1920 CALL SOUND(250,220,0,131,1)  
1930 CALL SOUND(250,175,0)  
1940 CALL SOUND(125,220,0,131,1)  
1950 CALL SOUND(125,175,0,131,1)  
1960 CALL SOUND(250,196,0,117,1)  
1970 CALL SOUND(250,117,1)  
1980 FOR TY=0 TO 83.33  
1990 NEXT TY  
2000 CALL SOUND(83.33,220,1)  
2010 CALL SOUND(83.33,196,3)  
2020 CALL SOUND(83.33,175,5)  
2030 END

continued on page 33

# RAM Tester for Memory expansion

by Tor Hansen, Canada

\*\*\*\*\*

\* RAM TESTER FOR VARIOUS  
\* MEMORY EXPANSION UNITS  
\* BY TOR HANSEN  
\* STARTED 3/31/87  
\* FINISHED 4/7/87  
\*\*\*\*\*

```

DEF START
AORG >2FOO
START B @AA *Program start
STA EQU >837C :Status and
MYWS EQU >8300 :workspace
HEXWS EQU >8320 :equates
GPLWS EQU >83E0 :
ENT BYTE 13 :ENTER Key
ONE BYTE >31 :ASCII 1
TWO BYTE >32 : " 2
TRES BYTE >33 : " 3
TITL1 TEXT 'R A M T E S T E R '
TITL2 TEXT 'Select'
TITL3 TEXT '1. Expansion RAM (32K Only)'
TITL4 TEXT '2. Mini Memory Cartridge'
TITL5 TEXT '3. Other Command Module RAM'
MESS TEXT 'ENTER to Quit'
AA LWPI MYWS :Actual start
BL @TXTMD :Set Text Mode
AB BLWP @CLRSCR :Call Clear
BLWP @VMBWD
DATA >031,TITL1,19
BLWP @VMBWD
DATA >0D7,TITL2,6
BLWP @VMBWD
DATA >195,TITL3,27
BLWP @VMBWD
DATA >1E5,TITL4,24
BLWP @VMBWD
DATA >235,TITL5,27
BLWP @VMBWD
DATA >37C,MESS,13 :Above writes Title
KEYS CLR @STA :Screen
BLWP @KSCAN :Call Key
MOVB @>8375,R1 :Get Keypress
CB @ENT,R1 :Is it ENTER?
JEQ OUTS :
CB @ONE,R1 :Is it 1?
JEQ ONES :
CB @TWO,R1 :Is it 2?
JEQ TWOS :
CB @TRES,R1 :Is it 3?
JEQ TRES :
JMP KEYS :If none, do again
ONES BL @AA1
JMP AB
TWOS BL @AA2
JMP AB
TRES BL @AA3
JMP AB
OUTS LWPI GPLWS :Exit Program
LI R6,>6892 :
B @>60 :To E/A
SCRE3 TEXT 'System RAM'
AA1 MOV R11,@SAV3 :Save Return
BLWP @CLRSCR :Call Clear
BLWP @VMBWD
DATA >05C,SCRE3,10
BLWP @VMBWD
DATA >1E0,SCRE9A,10
BLWP @VMBWD
DATA >230,SCRE9C,23
BLWP @VMBWD
DATA >39D,SCRE9B,29 :Write Sub Screen
KEY1 CLR @STA
BLWP @KSCAN :Call Key
MOVB @>8375,R1 :Get Keypress
CB R1,@ENT :Is it ENTER?
JEQ TEST6 :
CB R1,@RET :Is it FCTN[9]?
JNE KEY1 :If none, do again
MOV @SAV3,R11 :
B *R11 :Go Back

```

```

TEST6 LI R3,>A000 :High Mem Add.
LI R0,>1EA :Screen addr.
BLWP @HEXOUT :Print on screen
MOV R3,@ADBUF :save addr.
CLR R3
LI R0,>247 :Screen addr.
BLWP @HEXOUT :
MOV R3,@CNTBF :Save Counter
LI R2,>FFFF :Stop addr.
MOV R2,@STOP :Save it
BL @TEST :Do Test
B @AB
SCRE2 TEXT 'Mini-Memory Command Module'
AA2 MOV R11,@SAV3
BLWP @CLRSCR :Call Clear
BLWP @VMBWD
DATA >054,SCRE2,26
BLWP @VMBWD
DATA >1E0,SCRE9A,10
BLWP @VMBWD
DATA >230,SCRE9C,23
BLWP @VMBWD
DATA >39D,SCRE9B,29 :Print sub Screen
KEY9 CLR @STA
BLWP @KSCAN :Call Key
MOVB @>8375,R1 :Get Keypress
CB R1,@ENT :Is it ENTER?
JEQ TEST5 :
CB R1,@RET :Is it FCTN[9]?
JNE KEY9 :Do again if none
MOV @SAV3,R11 :
RT :
TEST5 LI R3,>7000 :Start of M.M. RAM
LI R0,>1EA :Screen addr.
BLWP @HEXOUT :Put on Screen
MOV R3,@ADBUF :Save addr
CLR R3 :Zero Counter
LI R0,>247 :Screen addr
BLWP @HEXOUT :Put on Screen
MOV R3,@CNTBF :Save Counter
LI R2,>FFFF :End of M.M. RAM
MOV R2,@STOP :Save it
BL @TEST :Do Test
B @AB
SAV3 DATA >0000
SAV1 DATA >0000
ADBUF BSS 4 :Start RAM Buffer
CNTBF BSS 4 :Count Buffer
STOP BSS 4 :End RAM Buffer
RET BYTE 15 : FCTN[9]
SCRE9 TEXT 'Cartridge Port RAM'
SCRE9A TEXT 'Address: >'
SCRE9B TEXT 'ENTER To Start/FCTN[9] To Menu'
SCRE9C TEXT 'Total Bytes Available:>'
AA3 MOV R11,@SAV3
BLWP @CLRSCR :Call Clear
BLWP @VMBWD
DATA >05A,SCRE9,18
BLWP @VMBWD
DATA >1E0,SCRE9A,10
BLWP @VMBWD
DATA >230,SCRE9C,23
BLWP @VMBWD
DATA >39D,SCRE9B,29 :Print Sub Screen
KEY3 CLR @STA
BLWP @KSCAN :Call Key
MOVB @>8375,R1 :Get Keypress
CB R1,@ENT :Is it ENTER?
JEQ TEST2 :
CB R1,@RET :Is it FCTN[9]?
JNE KEY3 :
MOV @SAV3,R11
B *R11
TEST2 LI R3,>6000 :Super Space RAM
LI R0,>1EA :Screen Loc
BLWP @HEXOUT :Print
MOV R3,@ADBUF :Save Start
CLR R3 :Init Counter
LI R0,>247 :Screen Loc
BLWP @HEXOUT :Print
MOV R3,@CNTBF :Save it
LI R2,>7FFF :End of RAM
MOV R2,@STOP :Save it
BL @TEST :Do test
B @AB

```



```

TESXT TEXT ' ..Executing..
TEST  MOV R11,@SAV1
      BLWP @VMBWD           :Put TESXT
      DATA >39D, TESXT,29 :On Screen
      MOV @ADBUF,R3        :Get Start
      LI R4,>FF             :TEST Value Ones
      MOVB *R3,R5          :Get Byte of RAM
      MOVB R4,*R3          :Write TEST Byte
      MOVB *R3,R6          :Read Byte in RAM
      CB R4,R6             :The Same?
      JNE FAIL            :No, go to Sub
      LI R4,>00            :TEST Value Zeros
      MOVB R4,*R3          :Write Byte
      MOVB *R3,R6          :Read Byte
      CB R4,R6             :Compare
      JNE FAIL            :No, go to Sub
      MOVB R5,*R3          :Restore RAM
      INC R3                :Start+1
      LI R0,>1EA           :Screen Loc
      BLWP @HEXOUT        :Print it
      MOV R3,@ADBUF        :Save it
      MOV @CNTBF,R3        :Get Counter
      INC R3                :Count+1
      LI R0,>247           :Screen Loc
      BLWP @HEXOUT        :Print it
      MOV R3,@CNTBF        :Save it
      CLR @STA
      MOV @ADBUF,R2        :Get RAM addr
      COC @STOP,R2         :Finished?
      JNE TEST             :If no, do more
      JMP BACK
FAIL1 TEXT 'There Is No RAM At This Address'
FAIL2 TEXT 'FCTN 9 To Main Menu
FAIL3 TEXT '
FAIL  BLWP @VMBWD
      DATA >2FC,FAIL1,31
      BLWP @VMBWD
      DATA >39D,FAIL3,29
      BLWP @VMBWD           :No RAM Message
      DATA >3A1,FAIL2,19
      JMP KEY7
BACK1 TEXT 'All RAM Good'
BACK  BLWP @VMBWD
      DATA >32D,BACK1,12 :RAM OK Message
      BLWP @VMBWD
      DATA >3A0,FAIL2,19
KEY7  CLR @STA
      BLWP @KSCAN           :Call Key
      MOVB @>8375,R1        :Get Keypress
      CB R1,@RET           : FCTN[9]?
      JNE KEY7
      MOV @SAV1,R11
      RT
VMBWD DATA HEXWS,S1WRTE :VMBW with DATA
S1WRTE BL @SIDCLL
S1VW  BL @S1VWRT
S1WBYT MOVB *R1+,@>8C00
      DEC R2
      JNE S1WBYT
      RTWP
SIDCLL MOV *R14+,R0
      MOV *R14+,R1
      MOV *R14+,R2
      RT
S1VWRT ORI RO,>4000
S1VRD  SWPB RO
S1VLOC MOVB RO,@>8C02
      SWPB RO
      MOVB RO,@>8C02
      ANDI RO,>3FFF
      RT
VWTR  DATA HEXWS,S1WTR :VWTR Sub
S1WTR MOV *R13,R0
      ORI RO,>8000
      SWPB RO
      MOVB RO,@>8C02
      SWPB RO
      MOVB RO,@>8C02
      RTWP
CLRSCR DATA HEXWS,S1CLR :CALL CLEAR Sub
S1CLR  CLR RO
      LI R2,960
S1BLNK LI R1,>2000
S1BCPY BL @S1VWRT

```

```

S1PBYT MOVB R1,@>8C00
      DEC R2
      JNE S1PBYT
      RTWP
TEXTMD LI RO,>01F0       :TEXT MODE Sub
      BLWP @VWTR
      SWPB RO
      MOVB RO,@>83D4
      LI RO,>07F4
      BLWP @VWTR
      RT
UWS    BSS 32
KSCAN  DATA UWS,SOSCAN :CALL KEY Sub
SOSCAN LWPI GPLWS
      MOV R11,@UWS
      BL @>000E
      MOV @UWS,R11
      LWPI UWS
      RTWP
VSWB   DATA UWS,SOVSWB :VSWB Sub
SOVSWB MOV *R13,R0
      BL @S1VWRT
      MOVB @2(R13),@>8C00
      RTWP
HEXOUT DATA HEXWS,S1RHEX :HEX no. on
S1RHEX LI R7,4             :Screen Sub
      MOV *R13,R0
      MOV @6(R13),R3
S1LP2  MOV R3,R1
      ANDI R1,>F000
      SRL R1,4
      CI R1,>900
      JLE SINUM
      AI R1,>700
SINUM  AI R1,>3000
      BLWP @VSWB
      INC R0
      SRC R3,12
      DEC R7
      JNE S1LP2
      RTWP
END    START              :Auto START

```

continued from page 28

```

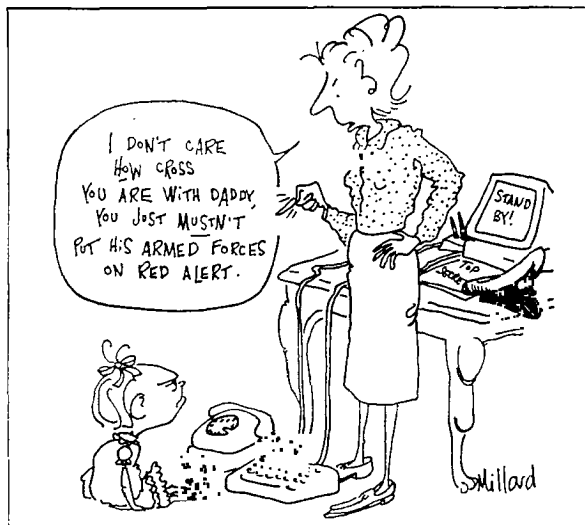
2550 PRINT "SORRY, BUT I HAD
TO GET THATOUT OF MY SYSTEM
! BEFORE YOUOUT, I SHOULD IMA
GINE YOU"
2560 PRINT "WOULD LIKE TO KN
OW YOUR SCORE."
2570 PRINT : :
2580 PRINT "#####
#####
#####
#####
#####"
2590 PRINT

```

```

2600 PRINT "YOU GOT ";SUM;"Q
UESTIONS CORRECT OUT OF A P
OSSIBLE TEN."
2610 PRINT
2620 PRINT "#####
#####
#####
#####
#####"
2630 PRINT : : :
2640 INPUT "NOW YOU CAN GO.
I HOPE THIS PROGRAM WAS HELP
FUL. BYE!":A$
2650 CALL CLEAR

```



# Emulate File Creation

by Rolf Schreiber

An emulate file is a special type of DSK1 emulation that is available with the Myarc HFDC. Although any number of emulate files can be present on any sub directory, only one can be active at any time. Once activated, an emulate file displays all the features of a floppy disk in drive 1. It can be accessed by any disk manager programme such as DM1000 or Disk Utilities, and programmes can be loaded from or saved to the emulate file from BASIC or XB, using the normal 'OLD DSK1.filename' (or 'SAVE DSK1.filename') convention. The only drawbacks to using emulate files is that the full size of the file is reserved at the time of creation, which can be wasteful of hard disk space, and only one emulate file may be in use at any time.

Setting up an emulate file on a hard disk is easy using the Myarc Disk Manager programme. However, it does mean having attached to the computer a disk drive of the required size. For example, if you wanted a DSQD (ie 2880 sector) emulate file, you would need to have an 80 track drive attached to the computer. So, using MDM5 to create emulates on a hard disk does restrict you somewhat! Based on an idea that I had, together with advice from Geoff Trott, I have developed a method of setting up emulate files of any size (up to a maximum of 3200 sectors) without having that size of floppy drive attached. All that is required are:

- TI-99/4A computer
- Peripheral Expansion System
- Myarc Hard Disk Controller Card (HFDC)
- Hard Disk (at least 16Mb capacity)
- Hardmaster(c) hard disk utility

The latest version of Hardmaster is version 2.14, and is available from Col Christensen of the Brisbane TI Users Group. Owing to a quirk or some bug in the programme, only hard disks bigger than 16Mb (65536 sectors) can be accessed by this programme, hence the minimum hard disk size mentioned above. At some later stage, if Hardmaster is modified, it should work with any size of hard disk.

## Step 1:

The first step is to reserve the required number of sectors on the hard disk, by setting the relevant bits in the bit map to indicate that those sectors are used. If you wanted a DSDD emulate, you would need to reserve 1440 sectors. I will demonstrate the creation of a DSQD emulate, equivalent to an 80 track DSDD floppy disk, with 2880 sectors (720 KB). The largest emulate file that can presently be created with this method is 3200 sectors (800 KB), the limitation being the size of the emulate file's bitmap (as distinct from the hard disk's bitmap), which only allows for 1600 allocation units, each AU representing 2 sectors in the bitmap.

Reserving sectors in the bitmap is easily achieved from BASIC or XB by RUNNING the following programme.

```
100 OPEN #1:"WDS1.DSQD",RELATIVE 2880, INTERNAL, FIXED
255
110 CLOSE #1
```

The programme takes about 1 second to execute, and you will notice the hard disk being accessed as information about the file is written to the root directory, and the bitmap is updated. If you now catalogued the hard disk using MDM5, you will find an INT/FIX 255 data file called 'DSQD' (or whatever name you chose) present in the root directory, 2882 sectors in size, but empty of any information.

## Step 2:

The next step is to change the file parameters of this data file into those of an emulate file of equivalent size. This will require the use of a hard disk sector editor such as Hardmaster.

Boot up Hardmaster from a floppy drive or your hard disk, and follow the on screen instructions. Select the TRee option and send the information to the screen. Take note of the sector number of the file descriptor record (FDR) associated with the data file you have just created. Return to the menu by pressing FCTN 9 and EDIT that sector by typing ED xxxx <enter>, where xxxx is the sector address (in hexadecimal) of the FDR associated with our file. Making sure that the screen is in HEX mode (not ASCII), change the values at bytes >0C and >0D to a 3100. Note down the values at bytes >28 to >2B before writing the changed sector back to the hard disk.

The 2 byte value at bytes >28 and >29 points to the allocation unit (each AU equals 2, or possibly 4 sectors) which is the start of the block we reserved in step 1. To find the corresponding sector number, multiply the value by 2 (or 4 if your hard disk is bigger than 32MB). Edit that sector and make the changes shown in the table below. This sector will become sector 0, the Volume Information Block (VIB), of your emulate file, and will look like the VIB of a floppy disk.

Byte(hex)	Value(hex)	Explanation
00-09	any ASCII	Diskname
0A-0B	0B40	# of sectors
0C	12	# of sectors/track
0D-0F	44534B	Hex for 'DSK'
10	20	Protection flag
11	50	# of tracks/side
12	02	Double sided
13	02	Double density
14-37	00	Reserved
38	01	Start of bitmap
39-EB	00	Bitmap ends at EB
EB-FF	FF	Not used

Write the changes back to the hard disk by pressing FCTN 9, changing the prompt to a 'Y', and pressing <enter>.

The last change that has to be made is to the next sector (ie increment the sector number by 1). This sector becomes sector 1, the File Descriptor Index (FDI), also known as the Directory Link Map, of your emulate file. Initially this sector will contain all zeros, indicating that there are no files in the emulate. Make this change and write the sector back to the hard disk.

If you followed all the directions correctly, you should now have an EMULATE file 2882 sectors in size of which 2880 sectors are useable and 2 sectors are used for the directory entry on the hard disk. By changing the size of the file in step 1, and making the corresponding changes to bytes >0A and >0B on the VIB sector, we can create an emulate file of any size between a practical minimum of 360 sectors up to the maximum of 3200 sectors, including non standard sizes such as 736 sectors (same as a 184KB Horizon Ramdisk) or 1600 sectors (same as the Myarc 512KB Ramcard). o

continued from page 30

Dear Jenny,

I bring some hints to ADVENTURE #5,"THE COUNT" this time...

- 1)You might find the answer to not being bitten a third time in the mail.
- 2)Solar ovens will not work if there's no sun.
- 3)DRACULA will not stay hidden if you use your file.
- 4)If you are a pretty good lock pick, you will have to reverse this tale to prevent DRACULA from pinching your stake.

It may be helpful also to save the game as you go along.

Crocodile Jones

continued from page 1

During the year a large sum of money was spent to upgrade the Editor's work place. The hard disk and 80 column card and monitor make life so much easier and more pleasant. I do not think that it would have been possible to produce the number of pages that we did without all that. Tony McGovern's help in producing an 80 column version of Funnelweb and in allowing for hard disk filenames to be saved in the mailbox of Funnelweb were also invaluable contributions. The only other thing that I would have liked to try during the year is Press. It promises so much but is so long in coming. I would like to thank all members who supported the upgrade of the Editor's facilities and hope you are satisfied with the results.

I did not produce the TND on my own by any means. Firstly there is Rolf who has been doing the layout for almost 3 years now. He has done it for Bob Montgomery, as well as for me. It is his skill that determines how all the articles fit together with as small wasted space and continued parts as possible. If you think that is easy I am sure that he will let you have a go at it. His long suffering wife has become used to losing the use of the lounge room floor for a few days each month as the 34 (average) A3 size pages slowly fill up with printing. It is lucky that Rolf is a night owl, otherwise it would be impossible.

The other people who are essential to the production of TND are all the contributors. Without them there would be no TND. I have been most fortunate to have a large number of TISHUG members who have answered the call and sent in something. They range from the regulars, Dick and Terry with their regular columns from the directors point of view, Lou with all his technical items which have caught the imagination of everyone to Ross who has provided the articles from the BBS on disk and arranged for people like John Ryan to type in articles from other magazines as well as his own contributions. I am sure to miss someone if I try to name everyone so I will stop there with names and simply say thank you to all who contributed anything at all to the TND in 1989. Without you all I would not have anything to do and everyone would be that much worse off. If you look at the other magazines that come into the library then you would quickly realize that we have a very superior product, both in production quality (the editorial staff will bow here) and also in the quality and quantity of material contributed locally (all contributors take a standing ovation). Quite a few newsletters are almost entirely written by one person, the editor. If we were like that the TND would quickly fold so keep sending in your contributions no matter how unimportant they seem to you.

As a final note, I asked towards the end of the year for volunteers to take over the Editor's job. I am pleased to announce that there was one such brave person and that he has taken over the role from just before Christmas. He is Bob Relyea from The Oaks and he is having fun learning to use the modem, spelling checking and the hard disk. Rolf and I will still be involved in the layout, printing and posting while Bob will keep us supplied with material. So please make sure that you know who Bob is and he knows who you are and have a chat to him about how you can contribute to your TND in 1990.

## How Does It Work?

by Geoff Trott

Talking to a group on the last tutorial day made me realise that most people do not know how the TI99/4A works and yet they would like to know. I thought then that I should try to explain, in as simple language as I can, the workings of our computer. This will take me quite a while as the TI99/4A is quite a sophisticated computer, so I will not rush it and give you time to tell me if I have not been clear in any of my explanations. I will also try and remember to sum up at the start of each article what has gone before and to explain things as many ways as I can.

So, where to start. Well, I guess we all know what happens when power is turned on. There may be a bit of noise which is cut off, the screen goes light blue (cyan actually) and then the title screen appears and there is a beep. On the screen are two colour bars, one at the top and the other near the bottom of the screen. There is some writing in large capital letters, the TI logo and a copyright symbol. Where does all that come from? Could we program it to do the same thing?

Just to explain what happens from the power being applied to the appearance of the title screen is going to take a few articles and in the process we will have to look at all the parts that make up the console. Let me start by considering what happens when the power switch is turned on. The power switch on the console is on the small power supply board inside the console. It applies AC power from the transformer to the power supply board which converts the AC to DC supplies of 5 volts, 12 volts and -5 volts. These voltages are then supplied to the electronics on the motherboard. All these ICs and transistors then start working at their own rate and doing their own thing, in particular the sound chip starts making a noise. In order for there to be some order in the computer, there is some circuitry which waits for a little while and then generates a pulse which is then used to reset all the important devices to a known state. This reset circuit contains a capacitor which is connected to the DC supply through a resistor which delays the rate at which the voltage across the capacitor increases and so a delay is generated. The reset must wait until all ICs have received voltage and settled into their operating mode. The main devices which are reset are the processor (9929A), the video display processor (9929A) and the peripheral interface adaptor chip (9901). These are the 3 complicated ICs in the console and so need to be put into a known state on power up. Of these, the one we will look at first is the processor.

The processor is the biggest IC in the console. It is obviously important to the operation of the computer but, as we will see, it does not have very much intelligence. It is designed to perform a set of simple operations. It needs a list of instructions which it can perform one after the other much like an automatic washing machine. The washing machine has a set program to perform for each type of wash, fill, soak, empty, fill, wash, empty, rinse, spin. There are a number of different programs possible depending on the settings. The processor has a simpler cycle of get the instruction (Fetch), perform the instruction (Execute) and it does not stop at the end of its cycle but goes straight on to get the next instruction. It does not have to have someone empty its bowl and fill it up again, or take the results of its execution of the instruction and supply the next instruction. So it can work much faster than the poor old washing machine. However it is still following a simple cycle of operation and even the execution of each instruction would only involve a simple operation of perhaps add two numbers together. Nothing as complicated as cleaning clothes!

If the processor is happily fetching and executing when the power is applied, it does not care if what it is doing makes sense or not, or if the instructions are legal and valid instruction even, it just mindlessly continues fetching and executing. The reset pulse however, causes the normal cycle to be interrupted, no matter what the processor is doing. The processor then enters a special cycle (like filling the washing machine with the last used soapy water rather than fresh) which causes it to get the location of the first instruction it should execute in a special location. In other words, the reset pulse enables the computer to start running at a defined place in a list of instructions, no matter what it was doing before the reset pulse came along.

In the way of all good serials, that is a good place to stop, leaving you with the question of what happens next? Stay tuned and I will carry on with this incredibly exciting story next month!

# Regional Group Reports

## Meeting Summary For March

Banana Coast	11/03/90	Sawtell
Carlingford	15/03/90	Carlingford
Central Coast	10/03/90	Saratoga
Glebe	08/03/90	Glebe
Illawarra	12/03/90	Keiraville
Liverpool	09/03/90	
Northern Suburbs	22/03/90	????
Sutherland	16/03/90	Jannali

### BANANA COAST Regional Group (Coffs Harbour area)

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

### CARLINGFORD Regional Group

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

### CENTRAL COAST Regional Group

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

### GLEBE Regional Group

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

### ILLAWARRA Regional Group

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

### LIVERPOOL Regional Group

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02)644 7377 (home) or (02)642 7418 (work) for more information.

### NORTHERN SUBURBS Regional Group

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

Come and join in our fun. Dick Warburton.

### SUTHERLAND Regional Group

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

The first regional meeting for 1990 was again well attended and proved to be an interesting night.

Ron Stewart brought along all of his new Hard disk components, including his recently acquired drive. It looks like a few more nights solid reading before the system is finally up and running.

The latest version of TI Artist Plus was demonstrated but not without our fair share of initial problems.

The software refused to run from a floppy disk drive due to the presence of my two Ramdisks in the P.E.box. The problem was traced to the partitioning of the Ramdisks. It seems that certain combinations of disk numbers will create this problem.

After a quick renumber of the Ramdisks, all returned to normal.

All meetings are held on the third Friday of each month at 51 Jannali Avenue, Jannali unless otherwise advised.

Peter Young Regional Co-ordinator

### TISHUG in Sydney

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

Russell Welham (Meeting coordinator).

continued from page 22

A public domain demonstration disk is available (from the Group library) with five demonstration programs saved with Quick Run. I only have two of these in the original Extended Basic, and these compare as follows:

VENUS (music by Sam Moore):

Time from RUN "DSK.." to first note:

TI Extended BASIC = 32 seconds

Quick Run = 22 seconds.

Not a huge saving is it? But it is a saving.

AXEL F(Music).

TI Extended BASIC = 91 seconds.

Quick Run = 23 seconds.

Not bad eh? The Extended BASIC program had to load data from two external disk files, which the Quick Run program did not need to do.

I tried the utility out on a particularly bothersome program, from way back when, written in TI Basic back in 1981, a version of Fiddler On The Roof with a very good animated graphic. In order to run it in Extended BASIC it was necessary to merge in the VDPUTIL which is on one of the disk library utility disks.

Running in Extended BASIC, the original program, which had to do a pre-scan, load the machine code of VDPUTIL, and read data from a disk file, took all of 127 seconds!

Running with QUICK RUN, this program is up and running in just 25.5 seconds! although it now occupies 92 disk sectors rather than the original 76 sectors.

As this is an even better demonstration than the ones Asgard have made available, you can obtain from the group disk library both the original and the snapped version of FIDDLER by asking for QUICK RUN DEMO DISK 2.

An especially nice thing about quick run, the RUN part is placed in the public domain so you can distribute programs saved with the utility without bothering about royalties or piracy. It is also possible to chain Extended BASIC programs, just as before, but with the advantage of the rapid start, and as before you can specify a line number to run from. An Extended BASIC program saved and then loaded with QUICK RUN can be LISTED in the normal manner.

Potentially a good time saving utility. Nice one.