TISHUG NEWS DIGEST

Focusing on the TI99/4A Home Computer

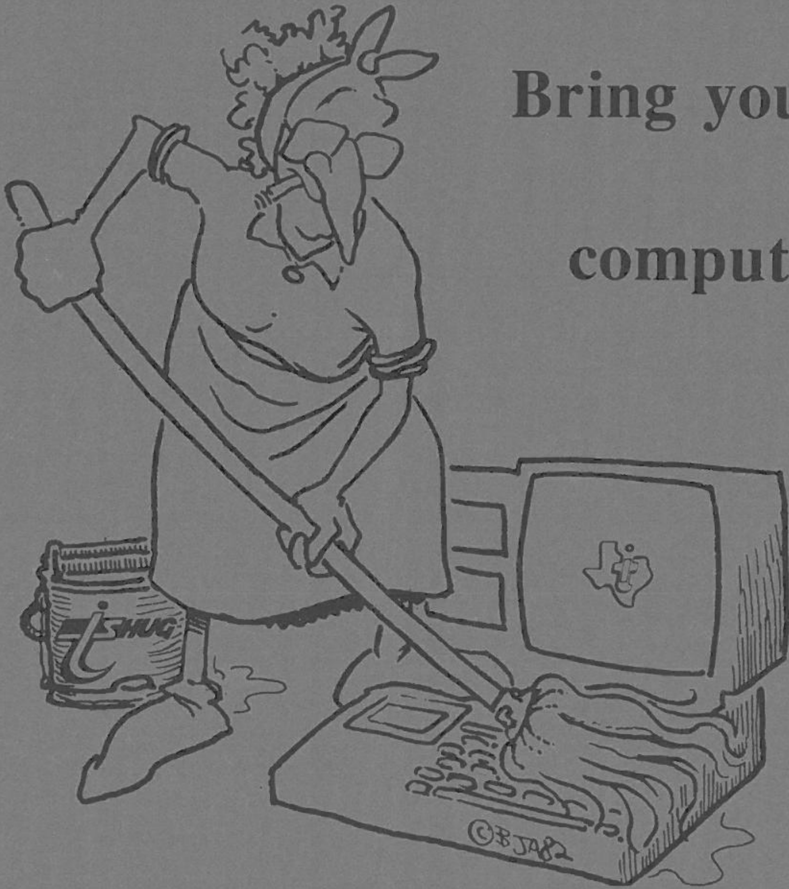
Volume 10, Number 3 April, 1991

Registered by Australia Post - Publication No. NBH5933

Clean Up while you Clean Out!

Bring your surplus

computer gear



along to the BUY, SWAP and SELL Day

P.O. Box 214, Redfern, New South Wales, Australia, 2016 $3

TIsHUG (Australia) Ltd, based in
Sydney

TIsHUG News Digest

April 1991

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board

Co-ordinator	
Dick Warburton	(02) 918 8132
Secretary	
Terry Phillips	(02) 797 6313
Treasurer	
Geoff Trott	(042) 29 6629
Directors	
Rolf Schreiber	(042) 84 2980
Russell Welham	(043) 92 4000

Sub-committees

News Digest Editor	
Bob Relyea	(046) 57 1253
BBS Sysop	
Ross Mudie	(02) 456 2122
BBS telephone number	(02) 456 4606
Merchandising	
Percy Harrison	(02) 808 3181
Publications Library	
Warren Welham	(043) 92 4000
Software library	
Rolf Schreiber	(042) 84 2980
Technical co-ordinator	
Lou Amadio	(042) 28 4906

Regional Group Contacts

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
Associate membership	$10.00
Overseas Airmail Dues	A$60.00
Overseas Surface Mail Dues	A$45.00

TIsHUG Sydney Meeting

The next meeting will start at 2.00 pm on 2nd of March at Ryde Infant School, Tucker Street, Ryde. At 12 pm, before the meeting, there will be a beginners' Editor Assembler class for all those interested.

Printed by
The University of Wollongong
Printery

TIsHUG News Digest **ISSN 0819-1984**

Index

TIsHUG Fairware Author of the Month

The Fairware Author for this month is Clint Pulley for his c99 compiler and other utilities. c99 is a subset of the C language, which is used extensively in industry. It allows the power of assembler to be used in a high level language. The shop will have version REL4.0 of c99 for sale. All donations collected at the meeting and sent in will be mailed to him.

MiniMemory batteries now available

Lithium batteries with solder tabs, suitable for MiniMemory and Supercart modules are now available for $3.50 each from the shop.

Editor's Comment

by Bob Relyea

Two members came to my assistance at the last meeting suggesting ways that I could have the computer automatically update records while running in an Extended Basic Program. That is what the club is all about. I just got back from a nice weekend at Mudgee where I attended a student's/teachers class reunion for those who left school or taught there in the early - middle seventies. It was a great time to see old faces and talk of times past but good to be back home again. After unpacking I got the Secretary's notes off the BBS and a couple of other items which well and truly finishes the editing for the month. I am hoping to get some good bargains at the next meeting. See you there.

o

Co-ordinator's Report

by Dick Warburton

Last month I emphasised the advantage that TI users have over other computer users in both cost and the convenience of the ramdisk. I made the point that expansion is now very affordable, and that with the new eprom additions to our ramdisks, we can now have quite sophisticated operating systems at a very reasonable cost. I now have all my Funlwriter files and associated utilities in eprom. They are almost instantly accessible, very easy to use, and far less complicated than other systems I have to use. I now have all the space I need to handle files. When working with Funlwriter, the files are saved and loaded so quickly and efficiently with the ramdisk,that I can see no advantage in possessing a hard disk for the TI. If I follow normal precautions, and back them up regularly to floppy, there is very little that can go wrong. In short, I am delighted with my TI + Ramdisks and Eproms. Perhaps we could call our new breed of TI, the TI+. Members may detect a slight bias or positive attitude towards using ramdisks and eproms. In this present climate of positive discrimination, I am simply doing my bit to redress the balance.

At the last directors' meeting, a number of decisions were made, some of which are quite exciting, which will set the pattern for 1991.

Firstly, a project is under consideration for development. It will involve building up a computerised train system for the club. While the mechanical side is fascinating, there is much to learn in the preparation of the software. Ross Mudie is looking at the feasibility of the project. At this stage it has the full support of your directors. Apart from the building and the computer development, I see it being used to attract people to the club, to give us publicity, perhaps in shopping centres etc. and to really interest the young at heart.

Secondly, it was also decided to go ahead with another order for ramdisk boards. Hopefully these will be available at the April meeting in the shop. The approximate cost will be about 40 dollars per board. Kits hopefully to build up a 32k ramdisk, with documentation should also be available. Kits to begin eprom additions may be available also. The approximate costs will be around $11 for 32k ram chips, and $12 for programmed eproms. Eproms can be fitted singly or in groups of up to 5. I put all my basic files on three eproms, which gives me 192k of eprom storage. Imagine, 192k of ramdisk space on eprom for only $36. Eproms can be updated very easily, if we plug them into sockets. Those who have 8k ramchips will probably be able to sell them to the club, and exchange them for 32k chips at little or no extra cost.

Thirdly, the directors gave some consideration to the format for the next two meetings. In April, we have one of our popular buy, swap and sell meetings. This will also be advertised in the trading post, and hopefully we will continue to attract new members and extra hardware for sale. Because of popular demand, there will also be a tutorial on using the ramdisk profitably. If your ramdisk does not do what you want, bring it along, and we will try to get it going well for you. The May meeting is already planned with a number of goodies organised. We hope to have a barbecue, and a variety of basic subjects of general interest. We are specially catering for newer members who would like to develop their skills on the TI. Both these meetings will be beaut. We are making every effort to get a games room up and operating. I would like to see members bring their own consoles and TVs as well as modules to play. We intend to develop competitions in some of the better games.

We are also trying to find another source for good colour monitors. If any member can help us, please let me know. We need analogue monitors, like the Amiga Monitor. If we can source good second hand ones we will.

As you can see, there is much planned for 1991 to make it a better year for TI users. Another innovation will be the copying of our disk library on to hard disk, which will be available to members on the spot. Copies of whole disks will be fast and convenient. The shop will be able to offer a wider range of software at any one time.

Well its getting late, and when you have gotter go, you have gotter go. So its goodbye for now, and

I will see you at the next meeting.

Dick Warburton. o

Letter to the Editor

by Ross Mudie

I wish to express my thanks to the members of TISHUG for the contribution made for my fairware program TI99-OPOLY, tutorials and other programs.

I wrote TI99-OPOLY in 1984 because my then 8 year old son, Peter, was monopoly mad and there was not a decent program available for the TI99/4A. The program was then the biggest that I had ever written and it became too big to run in the TI99/4A many times. It was also the challenge I needed to get me going on assembly. Up to that time I was really struggling to understand the concept of assembly and the books that I had did not help much.

The TEXPAC BBS is the biggest programming task I have undertaken, together with its continued operation and maintenance. The Wire I/O and the little train set which was developed at the request of the TISHUG directors continues to attract interest when ever it is displayed at train exhibitions or computer shows. The morse code program which was developed at the request of an Australian Amateur Radio operator has resulted in interest from England.

It is interesting to review my Link-it articles and my tutorial disks. Some of my programming efforts were (and still are sometimes) rather primitive. I had to learn it all the hard way, I had not programmed a computer prior to getting the TI99/4A at the end of 1983.

I will be providing copies of the latest versions of TI99-OPOLY (Vn 1.7 English names and Vn 2.1 American names) for the TISHUG shop. I will be putting together a disk of all the Link-Its worth reading and also a collection of other programs that I have written, as time in my busy schedule permits.

Thank you TISHUG members for a vote of confidence.

Yours Sincerely,

Ross Mudie o

continued from page 21

One or two final notes. Notice that the switch name, the operand of the ABS or DEC instruction, is a "general address" and thus can be specified in any of the 5 general address forms. Notice also that the target of the IFSW must be close since a JMP is used. You may want to modify the IFSW macro to add a "long jump" option, say by adding an "L" suffix to the ON or OFF.

Next month we will look at "calling subroutines".

"Make it possible for programmers to write programs in English and you will find that very few programmers can write in English." o

Secretary's Notebook

by Terry Phillips

Again we had a very good attendance at the March meeting. Somewhere between 50 and 60 members being present. It was unfortunate, that due to circumstances beyond our control, we were not able to have the ROCKRUNNER contest. Be assured that it WILL be held at the next meeting on April 6th.

At the April meeting it will be your opportunity to sell, swap or purchase that piece of software or hardware that you may be after, for the meeting will take the theme of a BUY/SWAP/SELL afternoon where there are sure to be plenty of goodies available at a price that is right. Already I am aware of a couple of former members who will most likely be there wishing to sell complete systems at negotiated prices and, as we are placing an advertisement in the Weekly Trading Post advising of the meeting, it is more than likely that a lot of gear will come out of the closet for sale. Make surę you are there to pick up the bargains or a little extra cash for the items you have for sale.

New memberships continue to be received and it is a big welcome to the following:

*John Senter - Northbridge California.
John is also a member of the San Fernando Valley Users.

*Bernard Smith - Albany Georgia.

*Jim Wilks - Dartmouth Nova Scotia. Jim is a member of the TI UG of Nova Scotia.

*Mrs P MacIntosh - Manly (QLD).

*Gordon Smith - Turramurra. Gordon was introduced to the group by long time member, Ron Kemp.

It was also good to see possibly our oldest member, Neal Negal make the trip down from Kimcumber to attend the last meeting. Neal brought all his equipment down as he was having trouble with his new printer and a corrupted ROS in his ramdisk. We got the lot going for him and hopefully he will not have any more problems.

MEMBERSHIP:

Excluding the overseas contingent, we currently have a total of 170 members and of this number, 112 or 66% have a common membership expiry date of 30 April 1991. If you are in this category - and you can easily tell by looking at the mailing label on your newsdigest - then please make the decision now to renew and if you are at the April meeting perhaps pay your subscription at that time.

BBS USAGE:

Our hard working Sysop, Ross Mudie (and great to see him rewarded at the last meeting) has given details of BBS usage over the past two months. Here are the details:

	JAN	FEB
Users:	20	25
Calls:	103	99
Time On:	21.33 hrs	17.11 hrs

While it is expected that BBS usage would decline over the summer months it has declined dramatically compared to the same period of the previous year. I am also mindful of the fact that some BBS users are now in the STD zone following the physical relocation of the BBS. I am currently in the process of putting together an article that will, hopefully, encourage greater BBS usage from its existing base of users and perhaps even attract other members to take out BBS subscriptions.

continued on page 8

TIsHUG BBS

by Terry Phillips

This article has been prepared to assist and advise members who may be contemplating acquiring the necessary equipment to enable them to access the bulletin board or for those members who currently use the board but not to any great extent.

Firstly, for newer members, it should be explained just what is a bulletin board service, hereafter referred to as BBS. A BBS is an information service which runs from a remote computer using the telephone network. In our case, the remote computer is located at member Ross Mudie's home and Ross is the person called the Sysop (System Operator). A Sysop's job is basically to update files and ensure that the system is restored to service as soon as possible after a system crash. His job is made a lot easier if members assist in uploading (a fancy term for sending) information to the system so that he does not have to do it all himself.

So, what is currently available on the BBS?

Firstly, there is a bulletin/news service, which at times contains up to 40 separate files with news and views from a variety of sources. If something happens in the TI99/4A world, then more than likely, this is the first place it will be heard about.

Secondly, there is a program download (a name for receiving) area, where interesting utilities and games can be obtained.

Thirdly, there is the electronic mail service which allows you to send a message to other users of the system which they will receive next time they access the BBS.

What do you need to access the system?

A minimum configuration would be a computer, some form of terminal emulator software, RS232, modem and, naturally, a telephone. You also need a user name and number, issued by the Sysop. While this configuration would be suitable, a disk drive and a printer are nearly essential for any serious BBS work. Modems, by the way, come in all shapes and sizes, some very sophisticated while others make you do all the work. The more "bells and whistles", the more you pay for them. A cheap, second hand, 300 baud (the speed of data transfer over the telephone lines)can be picked up for less than $100, while top of the range models, with auto-dial facilities and selectable baud rates, can sell for more than 10 times that amount.

An RS232 can be of the stand alone variety (not too many of these about), a card inside the PEB (multi-function, TI RS232, CorComp) or a miniPE system. The TE2 module is quite acceptable for terminal emulation software, although it does have some limitations. More suitable for serious BBS work are the specific terminal emulation programs such as Fast Term, Mass Transfer or Telco. These are available from the shop. With the right equipment you are not limited to merely accessing our BBS. A whole world of electronic communications is possible, albeit at a price.

The more users the better!

Currently, the BBS contains information relating nearly 100% to the TI99/4A, but this is not necessarily what the BBS should be exclusively for. The capacity exists on the system to include information on just about any topic that may be of interest to any group of members. As a diverse group, from widely differing backgrounds, I am sure that we could come up with a multitude of topics that would be suitable and would probably be read with interest by a lot of members. Topics that spring quickly to my mind include sports, other hobbies, education, comedy etc.

continued on page 8

TIsHUG Shop with Percy Harrison

I hope by now that you have all managed to get over the great disappointment of not being able to get your coloured monitor through the club in the forseeable future and that you have all received your payments back, except of course those who requested that I retain it for their future club purchases.

The 5.25 in. Half Height Drives proved to be very popular having sold 22 units which has cleaned us out of stock but we hope to have another lot in by the April meeting so if you missed out on the first two lots bring your money with you to the next meeting or phone me to reserve one or more for you.

Also by the April meeting I hope to have a supply of Eproms with the most frequently used programs burnt-in. The Eproms with Burnt-in Freeware programs will be $12.00 each and those with Commercial programs will cost $12.00 each plus the commercial club price of the program unless you already have paid for and received that particular Commercial program through the club in which case the cost will be $12.00 only. Please note that for each batch of four Eproms or part thereof you will require one 74SL128 IC which is available at $1.00 each.

We have a supply of 32K memory chips which are available at $12.00 each.

The good news for the month is for those users of High Density (HD) and 3.5 Disks. Because we have been able to buy at better prices than in the past we are now able to offer these items at a reduced price the new price being $13.00 and $12.00 respectively.

In addition to the club software listed in the various issues of the TIsHUG TND News the following shop items are available, but when ordering by mail please ensure that you remit sufficient funds to cover the cost of packaging and postage:

5.25 in. DSDD Disks (Boxes of ten)	$6.50
5.25 in. HD Disks (Boxes of ten)	$13.00
3.5 in. DSDD Disks (Boxes of ten)	$12.00
3.5 in. SSDD Disk Drives	$20.00
5.25 in. DSDD Half Height Drive (New)	$65.00
5.25 in. DSDD Half Height Drive (Used)	$50.00
12 Volt AC Transformer	$3.50
13 Volt Arlec Transformer	$12.00
8.5, 17 Volt Transformer	$25.00
60 VA Transformer	$20.00
MFC Printed Circuit Board	$30.00
MFC Kit (Disk Controller)	$102.50
32K Kit for MFC	$26.50
PIO/RS232 (single port) Kit for MFC	$42.50
Combined 32K and PIO/RS232 Kit	$60.00
Music Kit with PCB	$65.00
32K Memory PC Board	$7.00
TI Joystick	$12.00

COMMERCIAL SOFTWARE.

TI Sort SSSD	$15.00
Display Master	$15.00
TI Artist Plus	$25.00
Picasso Publisher Version 2.0	$14.00
Picasso Publisher Support Disks	$14.00
Picasso Applications Disk	$2.00
Nuts and Bolts #1 (DSSD)	$6.00
Nuts and Bolts #1 (SSSD)	$7.00
Genial Traveler (SSSD)	$6.00
Character Set Graphic Design I	$12.00
Character Set Graphic Design II	$10.00
Character Set Graphic Design III	$14.00
The Missing Link (TML)	$28.00
The Missing Link Companion Disk	$2.00
Page Pro 99 version 1.6	$28.00
Page Pro Utilities	$17.00
Page Pro Applications #1	$2.00
Page Pro Templates (Vols 1 to 12)	$8.00 (per vol)
Spell It! (DSDD version)	$22.00
Spell It! (SSSD version)	$24.00

Packaging and postage charges:

	Surface	Airmail
Up to 2 Disks	$1.80	$2.30
3 to 9 Disks	$2.40	$2.70
10 to 15 Disks	$3.10	$4.30
TI Artist Plus	$2.40	$2.70
Display Master	$2.40	$2.70
TI Base	$2.40	$2.70
TI Sort	$2.40	$2.70
5.25 inch half-height drive (1.25 Kg)	refer to your local post office	

Bye for now.

Assembly Class

by Ross Mudie

Unfortunately the assembly class seems to be dropping off in numbers, what has happened to all the keen ones who wanted to master assembly? If you do not put in the effort you will get nowhere!

The next class will be Saturday 6th April at 12 noon, just before the Buy/ Sell/Swap day.

I do not want to see the class go the same way as the little program which follows... By the way, why does it count down so slowly. Answers at the next class.

Extended Basic loader and linker.

```
100 ! SAVE DSK1.LOAD
110 CALL INIT
120 CALL LOAD("DSK1.O")
130 CALL LINK("QUIT")
```

Assembly source program.

```
*      IDT  'BLWP @>0'   Ross Mudie 9th March 1991
*                        Try this program; no prizes
VSBW   EQU  >2020        for guessing the outcome.
*                        Use CALL LINK("QUIT") from
       DEF  QUIT         Extended Basic.

WS     BSS  32
SAVRTN BSS  2

QUIT   MOV  R11,@SAVRTN This the normal way to start
       LWPI WS

       CLR  R0           Clear the screen
       LI   R1,>8000     Space with hex 60 already added
CLRLOP BLWP @VSBW
       INC  R0
       CI   R0,768
       JLT  CLRLOP

       LI   R0,645
       LI   R1,>9300     Starting number with offset
       CLR  R2
LOOP1  BLWP @VSBW

       DEC  R2
       JNE  LOOP1
       AI   R1,->0100    Count down
       CI   R1,>8F00
       JGT  LOOP1

       BLWP @>0          Guess where this goes?

       END
```

You Do Not Have To Have It All!

by Jim Peterson, Tigercub Software, USA

Do the conversations at your user group meeting sound like a coffee break in Silicon Valley? Are you confused by talk of GROMs and GRAMs, puzzled by references to HFDCs, intimidated by discussions of megabytes and frightened by talk of burning EPROMs? Well, join the crowd, buddy - so am I!

There are basically three types of people interested in computers. First, there are those who use a computer to run programs, to accomplish something useful or just to have fun. I believe that those people are still in the great majority, although we do not hear much from them.

Then, there are those who get their kicks out of writing programs, of creating software for others to use. There are not too many of those left in the TI world.

And finally, there are those who like to tinker with the computer, soup it up, plug in doohinkies and thingamajigs, and talk in that strange language I mentioned above. I do not know how many of those folks there are, but they are certainly the most knowledgeable, active, and interested, and they tend to dominate the conversations and the printed material in the TI world nowadays.

I presume that those fellows also do actually run programs on their souped up systems. And, some of them must be skilled programmers, because many of their hybrid hardware creations would be useless without specialized software. I am very glad that those people are around. Once in a while they invent something that I actually find useful, and they are a lifesaver when my equipment breaks down.

But, do not be intimidated by all that high-tech talk, and do not think that the computer world is passing you by. There are so many things to do with a computer that no one could possibly find time to do them all. Do your own thing and do not worry about the rest.

I have operated a TI software company for seven years, and I also spend a lot of time writing programs, using the computer as a word processor, etc. I probably spend more time on my TI than 90% of the users. So, what does my equipment consist of?

I have a console with the Extended Basic module plugged in, attached to a P-box which contains a TI disk controller, two double-sided drives, the 32k card, RS232 card, and a Horizon Ramdisk. Also plugged into the RS232 card is an old Gemini 10X printer and an Avatex 1200 baud modem.

I also have a Speech Synthesizer, a pair of TI joysticks, a TEII module and an Editor Assembler module, all of which I plug in occasionally when I need them; also, a cassette recorder and cable which has not been used in a long time.

I use Triton's Super Extended Basic module because it has some editing features which are useful when programming. It also has some limited plotting capability which I have never used - and have never heard of anyone who has. If you do not program, it would hardly pay to switch from the old TI Extended Basic. I also have the Mechatronics module but never got around to trying it.

I had a Gram Kracker but soon sold it and bought a Ramdisk instead. The Gram Kracker has fantastic capabilities if you have the skill and knowledge to take advantage of them, but most users do not seem to have done much beyond personalising the title screen.

I had a widget, and I guess it is still collecting dust around here some place. It was a nuisance, and since I use XBasic 99% of the time I did not need it. There are now widgets or "module expanders" that allow you to access more than one module from within a program. That is, if you have the skill to write such a program. I do not know that anyone has released such programs to the public domain, and I cannot think of any practical use except to access TEII speech from XBasic - but you can do that with the Text-To-Speech disk.

The ram disk is the one tool that I would not be without. In order to assemble my TI-PD catalog, I screened over 4000 programs, debugged and modified, merged in help files, conversions to XBasic and loaders, and assembled over 400 disks of programs. It took me hundreds of hours of work - without a ram disk it would have taken thousands of hours and I would not even have attempted it.

The ram disk enables me to switch from one program to another almost instantly, and with John Johnson's Boot program I can just as quickly catalog a disk or view a file. Mine has 256k of memory. I could get one with much more memory but I see no reason to do so; I have every program on it that I am apt to use even once a month, and it is only half full. That leaves plenty of room for temporary storage and downloading.

However, if you only use your computer to play games, do a little word processing and a bit of record keeping, a ram disk would be an expensive convenience rather than a necessity.

Since my ram disk is only half full, I would consider a hard drive to be about as useful as the mammalian appendages on a swine of the masculine persuasion. If I was running a BBS, sure - or if I was doing a lot of work with those memory-gobbling graphics and needed everything quickly accessible.

My old Gemini printer has been a faithful workhorse, although the hood over one sprocket wheel has lost its spring and is being held down by a loop of elastic cord. I will have to give it up soon, because the Gemini printer codes are becoming obsolete and I need to be able to write and test Epson codes. But, I hate to give up these 79-cent typewriter ribbons and start getting ripped off on $2.50 cartridges! As for a colour ribbon, the temperature will have to go way down, down under, before I pay for one of those.

Once in a while, when someone sends me a double-density diskful of stuff, I wish I had a CorComp disk controller. Otherwise, with diskettes selling for a quarter or less, it would not pay to change. If I ever get around to subscribing to GENie or Delphi, it will pay me to get a 2400 baud modem.

I cannot think of anything else I need, and I do not want what I do not need. If I really wanted to play joystick games, I would certainly get something better than the TI joystick. And if that MIDI interface cable becomes a reality, I will be sorely tempted.

I cannot see any advantage in putting the 32k under the bonnet, or anyplace other than where it is now. If I used speech a great deal, it would be nice to get rid of the synthesizer - but I know only one user who uses speech that much. I do not need a clock built in because I have a watch on my wrist. If I really did a lot of serious writing, an 80-column card would be wonderful. But then I would have to buy a monitor capable of displaying 80 columns. I certainly do not want to give up colour, and high-resolution colour monitors cost more. I would still want to use my old monitor for programming, because I like to write programs for folks who have basic equipment. I do not have room on my computer desk for two monitors, so I think I will pass.

I am a three-finger typist, so a RAVE keyboard would not speed up my typing very much. If I really wanted an IBM keyboard and 80-column capability, I would throw in a few bucks more and get a Geneve.

continued on page 8

Why I Love My Computer, but not Necessarily my User Group!

by Randy Packham, Hamiton, Ontario Canada

Sometimes I wonder why I go to meetings when I do go. Maybe it is just me, but I sometimes feel uncomfortable around a lot of people that I do not know very well. I assume some of you do too, because I have not noticed a lot of interaction. I was with my brother once and tried talking to some people but got some cool responses. Needless to say he was not impressed. Too bad, he has a complete system too!

This could be scaring some people off, but not me! I plan to remain a member, but will not be showing up at meetings since I live so far away and I am used to the isolation anyway. I will contribute what I can (having already submitted several programs), but I can do more than that. So, in response to a challenge, I have written this article for the newsletter.

Perhaps some articles could be written for specific users, if they would take the time to register their systems with the club and state what they use them for. This information could be used to find out who has the same interests and whom to call for help, or to trade accessories with (or borrow them). Maybe we can avoid having someone we do not know showing up at the meetings for the sole purpose of selling their complete system. This happened one night when I was at a meeting. The first time I saw this person was at Wentworth Supplies a year earlier, trying to find some peripherals to expand his console! Too bad! He obviously had the interest, but something must have happened to shake his faith.

It is sad to see all of our club's numbers in such decline. Surely TI99/4Aers are on the endangered species list, existing only in isolated niches in a world of IBM clones, competing, not necessarily against superior brands of computers, but against superior numbers and people's familiarity and exposure to them.

Yes, our computer is a bit of a dinosaur, but they too are mis-understood and not like most people think. They were fast and powerful and only wrenched from their position of dominance by disaster. Sound familiar? I am talking about the way Texas Instruments contributed to its (the computer's) ultimate demise.

I really love the TI99/4A. It is a physically beautiful and tough machine. One of my friends had his thrown against the wall and down the stairs when his dad trashed the house, after an argument with his sister! When he put the pieces together, it worked fine! Even the kitchen appliances did not fare that well!

Sometimes I get a kick out of the way people's faces light up when you start talking computers with them, then watch how their faces turn upside down when you tell them you have a TI99/4A. "But why not have an IBM or a Commodore?", they wimper (usually because they want to share stuff). I smile and tell it to them like it is. The old TI99/4A is like a rare muscle car in today's market. Nobody realizes the value of a classic compared to the newer cars. It comes from the era of the Voyagers and I have heard rumours that it was developed for military applications. It has a 16 bit processor, Extended BASIC (which is the most powerful BASIC I have ever seen), plus several other languages. It can handle 4 DSDD floppy disk drives, RAMdisks, hard disks, multiple RS232s (two parallel and four serial outlets with two RS232 cards), user defined functions, graphics and one of my favourites, speech! Speech synthesis is nothing to sneeze at! When kids cannot read yet, speech above all else can be vital. That is why I will not give up the TI99/4A for a Geneve.

If you think today's computers are greatly improved over the ones of yester-year, I have found a seven year old article in 99er magazine stating that the computer community expected to have real artificial intelligence by now. If computers are so advanced, where is it? The TI99/4 is nearly thirteen years old and people are still re-defining its limits. I would also like to point out that I still have less than $1000 invested in my system and there are very few things that I do not have for it!

Some people buy computers for playing games and never use them as computers because they are too difficult to program. What a waste! If people had bought Atari-2600s instead of the Commodore Vic-20s (what a joke) and the C64s (another poor joke), things might be different for the TI99/4A now.

I also laugh at them when I see their "wonderful" machines packed full of Texas Instruments' and affiliated manufacturers' chips. If they want to play the kilobyte routine, negating the TI99/4A's 16K console and adding the Kbytes the way Commodore does, our machines can have anywhere from 50K for the basic console (68K with TI Extended BASIC) to over 648K (like mine) and some people have a lot more added to their system than I do. By comparison, an IBM needs at least 640K to be of any use.

Perhaps what we can do to stimulate the proliferation of our species is something like this: if we could raise enough money, maybe in conjunction with other groups (users and special interest) we could buy up as much TI99/4A material as possible and donate it to centres for under-privileged kids or the handicapped, teach them to program and offer a program exchange wherein they can either write or modify existing programs and upon submitting them to our library, receive two or three selections in return.

Seriously! The stuff is so cheap, $50 to $60 for consoles and recorder at garage sales etc. We could increase the number of users familiar with the TI99/4A and create more demand for TI99/4A products. When I sold my first TI99/4A several years ago, it had cost me two or three times more than this one did and this time I have a lot more stuff for it.

I do not know about the rest of you, but Christmas is coming and I would not mind spending an extra $150 for a used console, recorder and TV (colour or black and white) for a classroom full of kids, if you would. It might open up opportunities for kids that may otherwise be denied them and keep them off the streets.

TIsHUG Software Column

by Rolf Schreiber

At the time of writing this column I have not received any new software from Asgard, even though it has been nearly two months since I sent the order.

Software Releases for April 1991

DISK A189A-C is the latest version, REL 4.0 of c99 by Clint Pulley. This programming language comes on three SSSD disks and includes numerous support files. It combines the power of assembly language with the ease of programming in a high level language.

DISK A211 is TI99-opoly V1.7, an Extended BASIC fairware offering from Ross Mudie. TI99-opoly closely follows the board game and your kids will just love it! The SSSD disk includes help files and a printed instruction manual is available for $2.00 extra.

DISK 228A is Ross Mudie's excellent tutorial on using speech in assembly language. This SSSD disk is a re-release of the 1986 'MUDIE 86/1' disk.

In addition to the above disks, other software will be available on the day of the meeting.

TI-Bits Number 4

by Jim Swedlow, CA USA

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

A LETTER FROM FRANCE

I exchanged letters with a TI owner in France. His English is not perfect but what he had to say is worth repeating:

"You asked me about the TI-99/4A support in France.

"The support is now very weak. The only magazine dedicated to the 4A will stop with the next issue because many TI users brought French computers over the last 3 years, often to get French educational software for children. This magazine has published fine programs, especially for assembly programmers.

"Before the Texas Instruments departure of the home computer market, the 4A was the best seller here. Many teachers used French version of TI-LOGO II. People from the French division of TI told me their division was leader in Europe, even before England division. (Over than 100,000 consoles sold in France, 70,000 in England). They had been very disappointed."

QUOTES OF THE MONTH

"The car of the future must be a car for the people ... the market for a low priced car is unlimited."
---Henry Ford (1863-1947)

"When you call me that, smile!"
---Owen Wister (1860-1938)
from "The Virginian"

PROGRAMMING TIP

Suppose you are writing a program that does a great deal of printing. There is a bug somewhere in the middle of the printing instructions. Every time you try and find it, however, you must wait while your printer wastes a lot of paper getting to the problem. What to do? If your printer is PIO, try substituting RS232.BA=9600. Unlike the parallel port, the serial port does not wait for a ready signal to return from your printer. So all of your print instructions will go out thru the RS232 port into thin air until you find your problem. Setting the baud rate at 9600 speeds things up (if you do not specify a rate, your TI will use 300 - much slower).

MORE QUOTES

"Only those who attempt the absurd achieve the impossible."
---Anon

"The technique is wonderful. I did not even dream it would be so good. But I would never let my children to come close to the thing. It is awful what they are doing."
---Vladimir Kosma Zworykin (1889)
Developer of television
on his 92nd birthday.

AN INTRODUCTION TO PRINTERS

If you are thinking about buying a printer, beware. Your choices are many as are the pitfalls.

First, you will need some things other than a printer. You need an RS232 card (stand alone or one for your P Box) and a cable. Most printers with a Centronics parallel port that will work with a standard cable (available from the houses that still support the 4A - Tenex, Tex-Comp, etc).

But which printer to buy? Epson? Star? Gorilla? Tandy? What kind? Dot matrix? Daisy Wheel?

First, lets look at the two basic types: daisy wheel and dot matrix (the others are probably out of your price range). A dot matrix printer is five to ten times faster and much more versatile. A daisy wheel gives you letter quality print while the dot matrix gives draft (poor) and 'near' letter quality (better). A tractor feed usually comes with a dot matrix printer but can be an extra cost item with a daisy wheel printer.

If 90% of your work is correspondence and you need top quality in its visual presentation, a daisy wheel is probably for you. Otherwise, for listing programs and all the other things that a printer can do, a dot matrix printer is the better choice.

Having narrowed the field, you still have to pick between the many models on the market. There are no standards in the world of printers for command structures (the codes your computer sends to the printer to tell it what to do). About the only codes two that are close to universal are Carriage Return and Line Feed. After that, anything can mean anything.

There are two 'de-facto' standards. The first is IBM. When big blue made a printer for its PC, it used a character set and command structure completely different than ASCII and just about every printer on the market. Alas, what will work with an IBM PC will NOT work on the 4A, so IBM compatibility is useless (unless you plan to defect).

The other quasi-standard is Epson. These folks developed a rather comprehensive instruction set (including graphics protocols) that some other manufactures and many software manufacturer followed. The TI impact printer is actually a bottom of the line Epson MX80. Most of the graphics programs for the TI will work with Epsons. Some of them support other printers, others do not.

A number of manufacturers make printers that follow Epson commands. Most Star (Gemini 10X, SG10, etc) and Panasonics do while the Axiom, Tandy and Banana printers do not.

Here are some suggestions to help you choose. First, see what your friends have and what they think of it. Then, in the store, have the salesman show you the draft and near letter quality print fonts. Note how long it takes to print a page (200 cps - characters per second - means different things depending on who is writing the advertising, I mean specs). Look for true descenders (is the loop below the 'g' below the line?) and the difference between the zero (0) and the letter (O). Make sure you can return it if it does not workout.

Plan to spend at least $200 (if you are buying a new printer). Any of the bargains below that normally do not have the features you will need.

My printer? A Star Gemini 10X. Its about 85% Epson compatible and has been a faithful companion.

ANOTHER PROGRAMMING HINT

When working on a program, you save it to disk often just in case your system locks up, etc. To save time, use a working name of <A> for these frequent saves. This saves up to nine key strokes. Also, if you have a load program that reads the disk directory, your working program will be at the top of the list.

Enjoy!

Treasurer's Report

by Geoff Trott

Income for February	$2526.76
Payments in February	$2729.57
Excess of expenses over income for February	$202.81

XB tips Number 5

by Jim Swedlow, CA USA

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

TEACH YOURSELF EXTENDED BASIC: This was released by TI to support the XB cartridge. It is available from our Users Group Library for the standard $2 fee.

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

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

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

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

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

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

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

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

NB: This also works in BASIC. Also, some symbols may cause strange things to happen when you run it thru the Text Formatter.

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

This program requires one disk drive and the memory expansion. In a month or so, I will do a disk menu program that does not require memory expansion.

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

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

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

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

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

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

INPUT NEEDED!!!!! I have been writing about what is of interest to me or what I am working on at at the time. This may not be what you are interested in. Questions, compliments, suggestions or even criticisms are welcome. Otherwise,

I will just keep going my own way.

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

continued from page 5

So, what about the Geneve? If I had an irresistible urge to run the few great programs that have been written for it, or if I wanted to explore its great programming capabilities, I would get one. But, I like to write programs for other people to use. When so few are interested in programs that I write for a computer that sold in the millions, why would I write programs for a computer purchased by a couple of thousand people?

I am sure that many folks will disagree with what I have written. That is why I wrote it. I hope they will disagree so strongly that they will immediately boot up Funlweb and compose a blistering reply. But do not send it to me - send it to your newsletter editor. The newsletters are badly in need of more articles by more writers!

continued from page 3

What will it cost?

If you do not currently have the required hardware to access the system, then the cost will be a major outlay. Once this is obtained, the BBS access fee is only $5 for a full year, with this fee being used to offset the upkeep of the system. A secondary cost consideration, for those outside the local call boundary, is telephone charges. These, however, can be kept to a minimum by calling the system at off-peak times.

Want further information?

Contact any of the directors or discuss BBS usage with someone you know is already on the system. Ask them to give you a demonstration. You too can join the wonderful world of computer communications.

continued from page 3

That is all for this month. See you at the April meeting.

REMEMBER - YOUR MEMBERSHIP MAY NOW BE DUE!

TML Graphics Programs

compiled by Stephen Shaw, England

While these are all presented ready to run with Extended Basic plus the disk utility THE MISSING LINK ($25 + $8 post from Texaments) the programs can be readily converted for use with any other utility that allows bit map graphics such as Triton XB, Myarc XB or the disk utility JBM103 (from disk library).

The first program although one of the smaller ones, produces some most interesting results. I have used a multiplier of 1.2, but at some stages it pays to take it a little more slowly. In effect you are backing away from a wall which is covered with circles, and as you back away, there is a repetitive effect to the patterns. The circular patterns are technically known as aliases, as the computer is not really plotting circles at all....

```
1 ! XB + THE MISSING LINK
2 !
100 REM CIRCLES
110 REM JE CONNETT/PWH MOON/S SHAW 1990
120 SIDE=20 ! 10>2000 value
130 REM
140 CALL LINK("CLEAR")
150 FOR I=1 TO 150 :: FOR J= 1 TO 150
160 X=I*SIDE/150 :: Y=J*SID/150:: C=INT(X*X+Y*Y):: D=
  C/2 :: IF D-INT(D)>.1 THEN 180
170 CALL LINK("PIXEL",I+20,J +20)
180 NEXT J :: NEXT I
187 !
188 ! next bit just stores pictures! omit if required
189 !
190 PIC=PIC+1 :: A$="DSK2."& STR$(PIC)
200 CALL LINK("SAVEP",A$)
201 !
202 !
203 ! lets look from a little farther away...
204 !
210 SIDE=SIDE*1.2 :: GOTO 140
220 END
```

The following program produces a random dot pattern on the screen and then first enlarges it AND rotates it slightly, and then just rotates it. See what effect these minor manipulations have. There are possibilities for some interesting animations here!

The program to be found in the right hand column is based on work by M. Feigenbaum in 1979 and produces some interesting pictures, depending on the limits you set when the program starts. Sometimes looking VERY closely (maximum magnification) will produce an interesting result, at other times it pays to stand back a little. Towards the right of the overall plot is chaos, on which can be found some overlying non-chaotic detail. The number of "invisible" plots can be varied- failing to plot the first few dots concentrates the display on certain features. Reducing the invisible range will produce a little more detail- if you wish to look at all dots (invisible=0) then try using the 16 colour mode and using a different plot colour for the first 14 plots say- or use all the colours at different plotting stages in the inner loop. More detail will be seen! Have fun! This program is a true fractal as the same shapes keep reappearing.

```
1 ! XB + THE MISSING LINK Stephen Shaw Aug 1990
2 ! from COMPUTERS PATTERN CHAOS AND BEAUTY by Clifford
  A Pickover
3 !
100 REM RANDOM DOT PATTERNS - MOIRE DOT PATTERN
110 REM AFTER PICKOVER
120 SEED=4 :: RANDOMIZE SEED
130 ! ANGLE OF ROTATION IN DEGREES:
140 AN=2 :: AN=AN*PI/180
150 ! SIZE OF PLOT IN PIXELS:-
160 SZ=140 :: MD=SZ/2
170 FOR I=1 TO 3000
180 RANDX=RND*SZ :: RANDY=RND*SZ
190 CALL LINK("PIXEL",RANDX+5,RANDY+5)
200 REM
210 RANDXX=1.11*((RANDX-MD)*COS(AN)+(RANDY-MD)*SIN(AN))+
  MD
220 RANDY=1.11*((RANDY-MD)*COS(AN)-(RANDX-MD)*SIN(AN))+
  MD
230 RANDX=RANDXX
240 CALL LINK("PIXEL",RANDX+5,RANDY+5)
250 NEXT I
260 REM
270 CALL LINK("PRINT",180,20,"ANY KEY FOR NEXT")
280 CALL KEY(5,A,B):: IF B<1 THEN 280 ELSE CALL LINK("
  CLEAR")
290 REM
300 ! different pattern
310 ! this time we have no enlargement, just rotation.
320 SEED=4 :: RANDOMIZE SEED
330 ! ANGLE OF ROTATION IN DEGREES:
340 AN=2 :: AN=AN*PI/180
350 ! SIZE OF PLOT IN PIXELS:-
360 SZ=140 :: MD=SZ/2
370 FOR I=1 TO 3000
380 RANDX=RND*SZ :: RANDY=RND*SZ
390 CALL LINK("PIXEL",RANDX+5,RANDY+5)
400 REM
410 RANDXX=1.00*((RANDX-MD)*COS(AN)+(RANDY-MD)*SIN(AN))+
  MD
420 RANDY=1.00*((RANDY-MD)*COS(AN)-(RANDX-MD)*SIN(AN))+
  MD
430 RANDX=RANDXX
440 CALL LINK("PIXEL",RANDX+5,RANDY+5)
450 NEXT I
460 REM
470 CALL LINK("PRINT",180,20,"ANY KEY TO END")
480 CALL KEY(5,A,B):: IF B<1 THEN 480 ELSE STOP
490 END
```

==========

```
1 ! XB+ THE MISSING LINK STEPHEN SHAW AUG 1990
2 !
3 ! FEIGENBAUM DIAGRAM Becker & Dorfler / Leon Heller :-
  FRACTAL REPORT #10 AUG90
4 !
100 CALL LINK("CLEAR")
110 CALL LINK("PRINT",1,25,"VERTICAL AXIS FROM: TO:"):: 
  CALL LINK("PRINT",21,25,"HORIZONTAL AXIS FROM: TO:")
120 A1$="1.140" :: A2$="1.171" :: A3$="2.530" :: A4$="
  2.586" :: CALL LINK("INPUT",1, 104,BOTTOM,7,A1$)
130 CALL LINK("INPUT",11,147,TOP,7,A2$):: CALL LINK("INP
  UT",31,103,LEFT,7,A3$)::CALL LINK("INPUT",31,147,
  RIGHT,7,A4$):: CALL LINK("CLEAR")
138 !
139 ! Follow these lines:-
140 IF RIGHT-LEFT<.0000001 THEN LEFT=LEFT+RIGHT :: RIGHT
  =LEFT-RIGHT :: LEFT=LEFT-RIGHT
150 IF TOP-BOTTOM<.0000001 THEN TOP=TOP+BOTTOM :: BOTTOM
  =TOP-BOTTOM :: TOP=TOP-BOTTOM
160 IF (TOP=BOTTOM)OR(LEFT=RIGHT)THEN 100
170 CALL LINK("LINE",1,1,1,240):: CALL LINK("LINE",1,
  240,192,240):: CALL LINK("LINE",192,240,192,1)::
  CALL LINK ("LINE",192,1,1,1)
180 WIDE=240 :: DEEP=190
190 CALL LINK("PRINT",3,12,"<"&STR$(LEFT)&"  BTM="&
  STR$(BOTTOM)):: CALL LINK("PRINT",2,170,">"&
  STR$(RIGHT))
200 CALL LINK("PRINT",180,20,"TOP="&STR$(TOP))
210 VISIBLE=50 :: INVISIBLE=50
220 SCALE=(RIGHT-LEFT)/WIDE
230 REM
240 FOR RANGE=1 TO WIDE
250 K=LEFT+RANGE*SCALE
260 P=0.3
270 FOR I=0 TO INVISIBLE :: P=P+K*P*(1-P):: NEXT I
280 FOR I=0 TO VISIBLE
290 ROW=(P-BOTTOM)*DEEP/(TOP-BOTTOM)
300 COL=(K-LEFT)*WIDE/(RIGHT-LEFT)
310 CALL LINK("PIXEL",ROW,COL)
320 P=P+K*P*(1-P)
330 NEXT I
340 NEXT RANGE
350 GOTO 350
360 END
```

```
1 ! XB + THE MISSING LINK
2 ! Stephen Shaw  August 90
3 !
4 ! from COMPUTERS PATTERN CHAOS AND BEAUTY 1990 by
  Clifford A Pickover
5 !
6 ! Produces patterns from a sequence of turtle like
  variables- repeats of length/angle/number of times
7 ! Random values below will often go off screen as no
  scaling is used. Plan and fill variable arrays with
  your own values!
8 !
100 CALL LINK("CLEAR")
110 CT=0
120 RANDOMIZE :: SETS=INT(RND*5+1):: CALL LINK("CLEAR")
    :: CALL LINK("PRINT",180,50,STR$(SETS))
130 REM LENGTH ARRAY
140 DIM L(10)
150 FOR I=1 TO SETS
160 L(I)=(INT(RND*8)+18)
170 NEXT I
180 REM ANGLE ARRAY
190 DIM A(10)
200 FOR I=1 TO SETS
210 ANG=RND360 :: IF ANG<12 OR ANG>350 THEN 210 ELSE IF
    (ANG>160)AND(ANG<210)THEN 210
220 A(I)=ANG*PI/180
230 NEXT I
240 REM REPEAT ARRAY
250 DIM R(10)
260 FOR I=1 TO SETS
270 R(I)=INT(RND*5+3)
280 CT=CT+R(I)
290 NEXT I
300 OLDX,X=90 :: OLDY,Y=110
310 THETA=0 :: LL=0
320 FOR I=1 TO CT*1.3333
330 REM
340 FOR J=1 TO SETS ! # SETS
350 REM
360 FOR K=1 TO R(J)
370 CALL LINK("PRINT",2,3,STR$(I)&"|"&STR$(J)&"|"&STR$(K
    ))
380 LL=LL+1 ! # OF VERTICES
390 REM
400 THETA=THETA+A(J):: IF THETA>=2*PI THEN THETA=THETA-
    2*PI ! 360 IN RADIANS
410 X=L(J)*SIN(THETA)+X
420 Y=L(J)*COS(THETA)+Y
430 CALL LINK("LINE",OLDX,OLDY,X,Y):: OLDX=X :: OLDY=Y
440 NEXT K
450 NEXT J
460 NEXT I
470 CALL LINK("PRINT",180,30,"ANY KEY FOR ANOTHER")
480 CALL KEY(5,N,B):: IF B<1 THEN 480 ELSE RANDOMIZE ::
    GOTO 100
490 END
```

```
1 ! XB + THE MISSING LINK
2 ! STEPHEN SHAW AUG 1990
3 !
4 ! How random is a sequence of numbers? This program
  uses the RND function but can take any list of
  values, ranged accordingly.
5 !
6 ! from Computers Pattern Chaos and Beauty 1990 by C A
  Pickover
7 !
8 ! If series is by chance pattern should stay on
  screen, preferably with in central circle (oval!)
9 !
100 RANDOMIZE
110 SEED=INT(RND*256):: RANDOMIZE SEED
120 CALL LINK("PRINT",3,200,STR$(SEED))
130 CALL LINK("BOX",1,1,192,240):: CALL LINK("CIRCLE",
    96,120,35)
140 X,OLDX=96 :: Y,OLDY=120
150 CALL LINK("PIXEL",X,Y)
160 REM
170 C=INT(RND*8)
180 ON C+1 GOSUB 230,240,250,260,270,280,290,300,310,320
190 CALL LINK("LINE",OLDX,OLDY,X,Y):: OLDX=X :: OLDY=Y
200 GOTO 170
210 REM
220 STOP
230 X=X-3 :: Y=Y+3 :: RETURN
240 Y=Y+3 :: RETURN
250 X=X+3 :: Y=Y+3 :: RETURN
260 X=X-3 :: RETURN
270 X=X+3 :: RETURN
280 X=X-3 :: Y=Y-3 :: RETURN
290 Y=Y-3 :: RETURN
300 X=X+3 :: Y=Y-3 :: RETURN
310 RETURN
320 RETURN
330 END
```

If you enjoyed the speed of the bifurcation plot already given, try this one, which has similarities to the affine plots we have given in earlier issues- it is slow! It is based on a code in Clifford Pickovers recent book reviewed elsewhere in this issue.

```
100 ! BIFURCATION PLOT GENERATOR - PICKOVER
110 CALL LINK("CLEAR"):: RANDOMIZE
120 MN=56 :: MX=74 ! FULL-ISH PIC WOULD BE 0 TO 125-
    CHAOS RULES FROM 59 UP
130 BETA=5 ! LOW VALUE 3 MAKES CHAOS FARTHER AWAY
    HIGHER VALUE INCREASES CHAOS
140 RS=180 ! PLOT RESOLUTION
150 XO=1.95 ! START VALUE FOR Xt [t=0]
160 N=90 ! ITERATION COUNT USE HIGHER FOR MORE CHAOS
    MORE PATIENCE NEEDED
170 RSC=RS/(MX-MN):: CS=48
180 FOR LA=MN TO MX STEP (MX-MN)/RS
190 CALL LINK("PRINT",1,120,STR$(LA))
200 X=XO
210 FOR I=1 TO N+10
220 X=LA*X*(1+X) (-BETA)
230 IF I<11 THEN 250
240 CALL LINK("PIXEL",(LA-MN)*RSC+11,X*CS)
250 NEXT I
260 NEXT LA
270 CALL LINK("PRINT",1,1,"MN "):CALL LINK("PRINT",1,18
    ,SEG$(STR$(MN),1,3))
280 CALL LINK("PRINT",1,48,"MX "):: CALL LINK("PRINT",1,
    75,SEG$(STR$(MX),1,3))
290 CALL LINK("PRINT",1,110,"B "):: CALL LINK("PRINT",1,
120,SEG$(STR$(BETA),1,3))
300 CALL LINK("PRINT",1,160,"XO "):: CALL LINK("PRINT",1
    ,180,SEG$(STR$(XO),1,4))
310 CALL KEY(5,A,B):: IF B<1 THEN 310
320 MN=RND*150 :: MX=MN+RND*100 :: BETA=3+RND*3 :: XO=RN
    D*3+.01 :: CALL LINK("CLEAR"):: GOTO 160
330 END
```

o

continued from page 19

will probably want to change the EPROMs at some time. Then you need to carefully decide what you want on the EPROMs. One suggestion is to start with Funnelweb, either a two IC set (which is rather minimal), or a three IC set (which gives the c99 compiler and a number of utilities as well). In either case you need to give us some information about how you want Funnelweb set up. It is assumed that the RAM part of the RAMdisk will be DSK5 and Funnelweb will be DSK6. I normally configure both character sets as the large character set and I use a white (16) on dark green (13, using the BASIC numbers for colours or FC using Funnelweb hexadecimal codes). The main problem is the printer. Most people have PIO as their printer, but all those with a serial printer interface need to say what they usually use, for example, RS232/2.BA=2400.DA=8.PA=N, or whatever it is. If you use Funnelweb at the moment, copy down the name which you use in the formatter or when you do a PF from the editor. If you want any changes to the following list of parameters, please bring your own list to a meeting and give it to the shop with your order for the EPROMs.

Programmers editor character set - large
Text editor character set - large
Funnelweb main disk - DSK6
Text colour - white (16)
Background colour - dark green (13)
Working disk which you use for user files - DSK5
Printer - PIO

o

Rambles

by Stephen Shaw, England

Welcome to another issue of RAMBLES and a happy new year to you all. Your comments and letters are always welcome, and your queries too -please help out with an SAE if you would like a direct response! My address, the same as the groups disk library, is:

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

I have been asked to write some summary type comments on the various XBs and also on what a brand new disk owner should be looking for.

Firstly, all versions of XB are based on TI XB, and are merely enhancements to it. All versions other than TI seem to have added some form of bit-map-graphics, where you can draw in hi-resolution.

TI VERSION 100 is very rare, being the first, and contains a number of bugs, the more serious (for TI) being the incredible ease with which a non-expanded owner can remove the proprietorial "PROTECTION" feature which you can have with XB which prevents programs being saved or listed. Version 100 also worked on the basis that you always had 28 sprites in action, and was consequently rather slow unless (being expanded) you told it otherwise with a CALL LOAD.

=======

TI VERSION 110 is the more common version and forms the standard for which almost all XB programs are written. The most serious omission being the lack of bit map graphics.

=======

Then came MECHATRONICS XB from Germany, a heavy module which tended to get a little hot. It had a very powerful but extremely complicated set up for bit map graphics, which, like TI Logo, was NOT bit map graphics but only redefined characters, consequently you could quickly run out of ink.

========

Next on the scene was MYARC XB which has been around in a number of versions, the latest and definitive version being Vn 2.12 - this has the disadvantage of REQUIRING the Myarc 512k ram card, as the ExBas itself resides in RAM. This does allow for some increase in speed. The Myarc XB has good true bit map graphics, variable screen windowing with both 32 and 40 column text modes. Myarc XB is in the form of a module which contains a loader plus ram, a disk, and a chip for the ram card. Added extras include integer and real variables- with integer variables processing a little faster than real ones. You may set a default disk drive and subsequently omit "DSK1." when using OLD and SAVE or RUN. Because more VDP is available, Myarc XB can load AND run those very long TI Basic programs that will NOT load in TI XB, AND the TI Basic extra character sets ARE available in Myarc XB.

CALL CHAR is available from 0 to 255! and you may have 32 sprites. There is a full range of bit map graphics commands- draw, drawto, circle, point, rectangle, fill, write, dcolor, and uniquely you can determine if a pixel is on or off. You may RUN many program format machine code programs directly. DF80 files written for TI XB may require you to use a utility program to reset VDP registers or may not function at all. Myarc XB allows you to run many memory image files which otherwise would require a SuperCart (EdAs with 8k ram) as the Myarc module contains ram in the correct location. Myarc has several unique abilities but does require the Myarc ram card- which is pretty handy anyway!

=======

The latest version was TRITON SUPER EXTENDED BASIC which amalgamated a number of TI XB mods prepared for Gram device owners. Thus you have such useful new commands as CALL CHIMES, CALL ALOCK, CALL ALL, a very useful CALL CAT, CALL CLOCK (runs slow in the UK), GOSUB and GOTO with variables, but perhaps its most important features would be the enhanced editing features, which allow you to move, copy, delete or resequence PORTIONS of an XB program quite rapidly- a major use of my module- and also it has included on ROM a commercial machine code bit map graphics program called DRAW N PLOT. This one allows you to plot bit map graphics, but only off screen- only when completed can you SHOW them, although there is a cursor on screen when you SHOW which allows direct editing. Graphics can be saved and printed and the format is compatible with TI ARTIST. There is a major conflict in use of VDP between the graphics and an XB program and hang ups may be frequent, especially if you PRINT the pic. Triton XB requires 32k ram for the graphics feature. There is a graphics program available (CLASS) for Triton XB.

======

With all this talk of graphics it is worth adding that the program THE MISSING LINK will work with all versions except Myarc to add true bit map mode graphics, with plotting in or out of windows. It does not bat an eye if you draw a line to a point off screen, and can dump a screen to printer at any time. Picture files on disk are compatible with TI Artist. TML is incompatible with the CLOCK command of Triton XB. TML is now my language of first choice for graphics, but suffers in not being able to tell me if a pixel is on or off.

+++++++++++

NEW DISK DRIVE OWNER?

The world is your oyster, with many programs available at very low cost which compare well with PC programs costing LOTSD more.

THE standard graphics program is TI ARTIST at about US$25, well supported with many disks of graphics, compatible with Triton XB, The Missing Link, RLE and so on.

THE MISSING LINK is well worth having especially if you have any interest in Bit Map Graphics. It too is US$25 and a demo disk is available from the Group Disk Library.

TI BASE is THE database for the TI, a fully featured database at US$25 which would cost lots more for any other computer. It is very close to dBase II. You can use it very simply or get stuck into its command language, micros, and so on. NOT easy, if you want something easy stick to the limited format PERSONAL RECORD KEEPING module!

The only significant word processor is TI Writer, which is a line orientated text processor with mail merge capability. It has been much enhanced in the form of FUNLWEB, available from the disk library, which includes also the TI Assembler plus a disk manager including sector editor. If you have a disk drive you must get Funlweb, and print out all the docs. You may not use all its features! but they are there.

Funlweb provides one "environment" for operating in- another is BOOT (disk library) which gives you an easy to configure menu for your disk, as well as the ability to catalogue and print from the menu screen.

All disk owners should have MCOPY (disk library) which will organise your disks in a more efficient manner, making disks with more than 16 files operate more quickly and with less wear and tear.

After that the choice is yours and depends on what YOUR interests (and storage space or pocket money) will run to. The group has an extensive disk library which is well worth perusing- disks are inexpensive, and the catalogue even cheaper- just send three disks and return postage!

=========

Anyone interested in videotaped tuition? My old tv set, bought specifically for use with my TI, lasted not as long, and developed a very NASTY fault which blew two ExBas modules on me- well, would not be a cheap module would it! Courtesy of Mike Goddard I now use an attractive mono monitor which has the added feature of an extra socket to link to a video recorder...

In theory I could feed the computer into a composite video link on a video, with a mike into the audio in socket, and help you all out with some of those complex programs we now have- seeing it happen in front of you is a great deal easier to take in than a letter or telephone call!

So if there is an interest - AND someone has a battered VHS video recorder which has composite video and separate sound in connections they do not want!!!!! - maybe we can be of help! At present I do not have a video with composite video in!

==============================

If you do not have a complete set of TI*MES, back issues are well worth having for reviews of old programs which are still available and which you will not know about unless you get those back issues, some of which are in rare supply (some indeed sold out) so contact Peter Walker today!!!

==============================

And if you would prefer me not to use a 9 pin dot matrix printer for Rambles, a donated daisy wheel may improve matters - I just do not need one for my personal use! A serial interface and cable would make it easier for me to use side by side with my trusty Epson...

==============================

IS TI EXPANSION EXPENSIVE?

Certainly it can cost quite a lot of money to chase down an expansion box, with prices in the region of 150-200 for a populated box. However, you must also consider the cost of the programs which you may then purchase, and compare the cost of those programs to similar (or less efficient!) programs for a new computer such as a PC. A PC may be faster and may have more memory, but do you need it? Often all the extra memory is fully occupied by programs which need the memory due to being inefficiently written and/or containing a multitude of features you will never need! Consider the software when you consider the hardware!!! The TI99/4A is considerably easier to use and especially to program for than the PC.

==============================

WHAT IS EXPANSION???

Here in the UK TI never really tried to sell anything other than the console, and extra hardware was almost never seen in the retailers. With many of our present members in the position of having console only, a brief word of explanation may be handy!

There are two forms of extra hardware- "stand alone" and "peripheral cards". The latter require that you have the Peripheral Expansion Box, or PEB, a large and very heavy steel box which has a power supply and room for up to 7 cards and a full height disk drive. Many members use two half height drives but you have to watch the power requirements if you do that!

At the "low end" you may purchase modules which have plugs coming out of them which plug directly into a printer (parallel port) for word processing or a spreadsheet. The Speech Synthesiser is a small device that plugs into the right hand socket and allows speech with a suitable module- TE2 or XB. There are "stand alone" devices which plug into the right hand socket and allow you to connect a printer (usually these are "parallel" devices, the other standard to RS232). You may find "stand alone" 32k memory expansion- and if you only add 32k ram, we can supply several machine code programs on cassette for you.

The cards for the PEB offer a wide variety, including the fairly standard printer interface card (RS232 and parallel on board). The disk drive controller card, to which you attach up to 3 single sided or double sided single density (TI controller) or single or double density (Myarc and Corcomp controllers) 40 track disk drives. The Myarc controller may also use 80 track drives. It is possible indeed to interface with almost any standard of drive, but the standard required for interchange is single density (90k single sided, 180k double sided).

The older stand alone TI disk controller will only function single sided single density. Ram cards allow you to use RAM chips as though they were disk drives, allowing much faster access- and some may allow you to store the contents even when you switch the computer off, using battery power supplies. Power drain may be high, and there is always a risk of corruption with these- always keep a disk backup! Ram cards may also allow you to use them as printer buffers (you fill the ram quickly with text and this is then sent to the printer as required while you carry on with something else). Myarc ram card allows you to use Myarc Extended Basic as well.

There is a USCD Pascal card (very slow language!) and even a FORTI music card allowing more sound channels. GRAM cards are similar to RAM cards but are configured for use with the special TI internal language GPL.

You can even buy a hard disk controller to operate with a hard disk or two! A hard disk is an expensive storage medium which has much more room than a floppy disk, typically these days 20 million bytes is the minimum compared to 90 thousand bytes for a single sided single density floppy disk! The cost per byte is low but when a hard disk becomes inoperable you of course lose rather more data than when a floppy disk decides not to work!

A MOUSE is a drawing/pointing device which requires special disk software and may -depending on make- plug into the joystick port or RS232 printer port. A TRACKBALL was once made for the joystick port, this was a special version for the TI only made by Wico. There was the SUPERSKETCH drawing tablet, which connected a drawing arm through two variable resistances to a drawing program in the module- it was possible to cut off the tablet and attach a mouse instead.

The MBX UNIT was a complex box of tricks which would have allowed use of a more complex (three plane) joystick, and for a few modules allowed rather primitive speech control. It also had a touch tablet face for switching, and offered speech without the speech synth, but only with the special MBX modules. Consoles may also be modified with the addition of a LOAD INTERRUPT switch which causes the console to tear off to a specified memory address when pressed to do a program located there. You need to put the program there and tell the console where it is! The auto-reset when you insert a module can be switched by a manual switch- you can also prevent a module resetting the console. Ask Mike Goddard for details, which are in his book of things to do to a console (10% ok Mike?).

Newsletter update

by Bob Relyea

TI UP TIt BITS (Western Australia), August, 1990: Editorial; Data Communications - Bits and Bauds; Multiplan - Joining Files; Loading Programs; Uncle Miltie's Corner (Tips & Tricks); Tips - a brief review.

October, 1990: Editorial; Verify a Formatted Disk? by Gary Cox; Handy Tips for the TI99/4A Computer; a list of newsletters received; TI-Base quick reference chart; Regena on Basic (Pyramid Solitaire); Exploring your printer - ESCape to six font sizes by Lou Borelli; The TI99/4A gets down to business by Art Byers.

TIBUG (Brisbane), January, 1991: Editorial; RGB Interface; V3.02 Extensions to TI-Base; Trading Post; What' New; New Fractals Programme; Beginning Modems; 192k Video Memory for the Geneve 9640; Specially for Cassette Users; Tips From The Tigercub #25; Blackjack; TI Multiplan V4.0 by Audrey Bucher.

MICROPENDIUM, December, 1990: Regena on Basic (Scripture Quiz); Extended Basic - A Printer Potpourri; TI-Base User Guide, program segments; c99, more on roots of a polynomial; Basic/Assembly, redefining character definitions; Getting TI's out of the closet; Gen-Tri for the Geneve; PC Pursuit, cutting the cost of logging on with your modem; European enthusiasm with Asgard Software; various Reviews; Double Column Printing; Boot Tracking.

continued on page 16

TI-Base Tutorial #9

by Martin Smoley, North Coast 99ers USA

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.

COMPLICATED DATABASE OPERATIONS AND GRAPHICS

It is time we started to do more complicated chores with TI-Base. With this tutorial I am starting a mini-series. In this mini-series we will create a set of CFs and DBs for a computer club disk librarian who handles mail order requests from members who live in other states. The system should rapidly locate a members name in the club DataBase (Db), print out the three labels you see below (To:, From:, and CAUTION) and keep track of who received disks and the club disk inventory. The system will keep us informed of the disk inventory and start telling us to order more disks at a pre-set quantity. At this point we will only keep the quantity of disks shipped to a person and the date. This will not include information on the disk name or type of disk. I want to save some information and demo how it is done, but I do not want to turn the operator into a data entry person either. I have the system working at this time, but I may make some changes as we go or by next months tutorial. I will cover as much as possible each month without stretching my page limitations too much.

```
* Copyright Martin A. Smoley 1989
*                          DSKSHP1
CLOSE ALL
    DO DSK2.PREP1
    DO DSK2.DSKSCR1
LOCAL SEL2 N 5 0
LOCAL MORE C 1
LOCAL TEMP1 C 60
LOCAL ANS N 3 0
  SELECT 5
   USE DSK2.GRF1
  SELECT 4
   USE DSK2.MSRET
  SELECT 3
   USE DSK2.DSKINV
  SELECT 2
   USE DSK2.SLSREC
  BOTTOM
  SELECT 1
   USE DSK2.TNAMES
REPLACE MORE WITH "Y"
WHILE (MORE = "Y")
  TOP
 WRITE 23,6,"ENTER NM "
 READ 23,17,SEL2
  IF SEL2 = 0
    CLOSE ALL
    RETURN
   ENDIF
     CLEAR
   FIND SEL2
  IF (NM = SEL2)
    DO DSK2.DSKNAP1
    DO DSK2.INVUPDT
  ELSE
    WRITE 23,4 "Number Not Found"
    WAIT 3
  ENDIF
 WRITE 23,4,"FIND another Y/N"
 READSTRING 23,23,MORE
 CLEAR
ENDWHILE
CLOSE ALL
 DO DSK5.SETUP
RETURN
* DSKSHP1   Save as DSKSHP1/C
* *******   Ver. 2.01 03/31/89
* Find NM  using "FIND"
* Print a label and save record
****************************
```

EDITOR'S NOTE: When you see configurations like the following in this series it usually means that there were condensed slightly from a 'wider' format. If you have need of seeing them the way they were given to me, contact me and I will arrange it.

```
                                TNAMES
REC  NM    LN              FN              MI SA
CT                   ST ZP    PH              XP    GP    ID
0002  1 Aardvark    Grant            E. 9995 State Rt. 84
Geneva         OH 44014 1-465-9876   89/02 NOCO  0717851
0003    2 Aardvark       Willard        J. No Newsletter
               OH       1-465-7689   89/09 NOCO  0717852
0005    3 Jones         Quincy    W. 37285 Burgandy Lane
Mentor-on-the-Lake OH 44060 257-1029 89/08 NOCO 0820871
0000    4 Smoley        Martin     A. 6149 Bryson Drive
Mentor         OH 44060 216-257-1661 90/02 NOCO  0713831
0004  5 Vivannovitch     Elexxie      I. 111 E. 98th. St.
Cleveland      OH 91023 541-5415     89/05 NOCO  0712881
0001   6 Whitman    Raymond (Slim)  B. 2574 East 254th.
Eastlake       OH 44094 951-2345     89/09 NOCO  0921861
```

From here on I will attempt to include all the data you will need to complete this task. This will require some redundancy that I hope will not boar the more advanced TIB users. I am printing three labels in each print cycle because I always seem to need return address and CAUTION labels. OK, let's get started.

You should recognize the database TNAMES. I have updated it and filled in some blanks. I use it because it is a direct substitute for the NorthCoast DB, but smaller. (NOTE: The only real name is mine.) Two items of importance are NM and ID. NM is an N type field with a width of 5 and 0 decimal places, while ID is a C type field with a width of 7. In order to get the desired end product I SORTed TNAMES ON LN,FN before I placed the numbers in the NM field. I then entered the NM field and SORTed ON NM. This was because I wanted the names in LN,FN order, but they had to be sorted on NM to allow me to use the FIND function on the NM field. This was all covered previously, I believe it was around Nov./Dec. 1988. FYI: A field type can be changed at will using MODIFY STRUCTURE. You can change a C type to an N type or vice versa, but do not change any field lengths or you will lose the data. Also, I do not recommend changing a C)haracter field that contains names to a N)umeric field. The field should already contain numbers before it is changed. If you want to experiment with this idea, use a database you can afford to lose.

```
                         PREP1
CLEAR
CLOSE ALL
SET PRINTER=RS232.CR.LF.DA=8
SET HEADING OFF
SET RECNUM OFF
COLOR WHITE DARK-BLUE
SET TALK OFF
RETURN
*
*   Pre-Program Preparation
*
* PREP1  Save as  PREP1/C
* *****


CLEAR
WRITE 3,9,"This section Locates a record"
WRITE 5,9,"using the NM field in the"
WRITE 7,9,"TNAMES Database. It then"
WRITE 9,9,"displays the name and address"
WRITE 11,9,"and asks how many disks to"
WRITE 13,9,"be shipped.  It also keeps a"
WRITE 15,9,"running inventory of disks"
WRITE 17,9,"in stock and shipments."
RETURN
*
*  DSKSCR1      Save as DSKSCR1/C
*  *******  Info Screen     12/1/88
```

DSKSHP1 is the main CF. To start this small system you would place the disk containing all the needed files in disk drive #2 and type DO DSK2.DSKSHP1 <E>. The TIB processor will find DSKSHP1 and start executing the commands one line at a time. The first thing TIB will do is CLOSE ALL Dbs. The next command will DO the PREP1 CF on DSK2. The PREP1 CF is my general purpose preparation file. It resets the defaults for a particular set of CFs. The only line in PREP1 that I feel is important is SET PRINTER=RS232.CR.LF.DA=8. I am using one of the original TI 99/4 Impact Printers to do the labels. This printer runs off the RS232 port and not PIO. It was also necessary to place Dip switch #1 of the SW2 set to the OFF position for graphics mode (see your printer manual), and last, it is also necessary to send the command DA=8 in the PRINTER statement. This was all necessary to set up my printer for the graphics I intend to print on the second and third label.

> *** The ability to print graphics, which can now be done with Version 2.01, will create an exciting new area of capability for the TI-Base user. ***

DO DSK2.DSKSCR1 puts up the information screen in the lower left hand corner of this page. It can say anything you want, or you can leave it out of the CF altogether. The next four lines in DSKSHP1 are not important, but then again they are important. Sounds like I am confused, "HUH". Well, I will explain. As far as I can tell the local variables are limited to about seven names with a size that is currently about 256 bytes. You can increase the size somewhat with the SET command but the point is that this space is a precious commodity. It seems like I run out of local space every few minutes. So, in the future I will try to use one or more databases to temporarily store information that TIB needs. As you will see in this tutorial the information is as readily available in a Db as is would be in LOCALs and I can store as many as I need.

CRTD 04/03/89 CHNGD 04/05/89

FIELD	DSCR	TYPE	WIDTH	DEC
1	FR	C	006	
2	TO	C	003	
3	NAME	C	030	
4	STREET	C	030	
5	CTSTZP	C	030	
6	CTN	C	020	
7	CD	C	030	
8	DNB	C	030	
9	DNX	C	030	
10	MS1	C	010	
11	MS2	C	010	
12	MS3	C	010	
13	MS4	C	010	
14	CD1	X	020	
15	CD2	X	020	
16	CD3	X	020	
17	CD4	X	020	

000 1 MSRET   00000/00001

CRTD 04/06/89 CHNGD 04/06/89

FIELD	DSCR	TYPE	WIDTH	DEC
1	GRFNM	C	010	
2	GR1	X	080	
3	GR2	X	080	
4	GR3	X	080	
5	GR4	X	080	
6	GR5	X	080	
7	GR6	X	080	

000 1 GRF1   00000/00001

CRTD 03/31/89 CHNGD 04/06/89

FIELD	DSCR	TYPE	WIDTH	DEC
1	RTOT	N	005	00
2	LDT	D	008	
3	PDT	D	008	

000 1 DSKINV 00000/00001

The statement SELECT 5 means select the number 5 work area, or as I like to call it, "slot 5". We then open the GRF1 database, which is located on disk drive #2 in slot 5, with the statement USE DSK2.GRF1. We then SELECT slots 4,3,2 and 1, and open MSRET, DSKINV, SLSREC and my old favourite TNAMES respectively. I SELECTed 1 last because that is the slot I want to be in when TIB is ready to start performing its tasks. I will get into it more later, but for now the DBs look like this. GRF1 is the DB where I will keep my graphic data to make up the OHIO and DISK graphics. MSRET is a DB that contains Martin Smoley's RETurn address and a bunch of other tricks which I will explain as we go. DSKINV, of coarse, is our DiSK INVentory, and SLSREC holds the SaLeS RECords. TNAMES holds the name and address list and must be sorted on the NM field. I have listed the STRUCTUREs for three of these Dbs above to help you understand and enter the information needed. I have squashed them slightly to make them fit in, but all the information is there. Let's look at GRF1, my graphic Db, first. It contains 7 fields. Field #1, GRFNM, is a C)haracter field with a maximum width of 10. This field will hold our graphic name, such as OHIO or DISK. This Db must be SORTed ON GRFNM to allow TIB to FIND a graphic by its name. Fields 2 through 7 are X-type fields, and will contain our graphic and other interesting stuff. I have listed the data from GRF1 at the bottom of this page. It was quite confusing to me at first, so do not expect to grasp the idea immediately. Special Note: (04/11/89), I told Dennis Faherty about these new discoveries in graphics last night and he was amazed. Until then he had not grasped the unbelievable potential of the X-type field even though he created it. The X-type field is very special. You type in normal ASCII character like 123ABC. The character must be entered in a very special pattern, but the actual entry is easy. The character must be entered in pairs because TIB will interpret them as Hexidecimal numbers. For example, the pair of characters 1B would be interpreted by TIB as Hex for the number 27 and 27 to your printer means ESC, or escape. We now have the basics for all printer control codes.

```
M0000          Database:   GRF1
GRFNM DISK
GR1
1B33181B4B1F00FFAAD5AAD5AAD5AAD5AAD5AAD5AAD5AAD5AAD5AAD
5AAD5AAD5AAD5AADFA8F80000
GR2
1B33181B4B1F00FFAA55AA55AA55AA55AA57AF5CB858B858B85CAF5
7AA55AA55AA55AAD5AAFF0000
GR3
1B33181B4B1F00FFAA55AA55AA55AA55AAD5FA351A1F1B1F1A35EAD
5AE5FAE55AA55AA55AAFF0000
GR4
1B33181B4B1F00FFAB55AB55AB55AB55AB54AB55ABFDFFFDAB55AB5
4AB55AB55AB55AB55ABFF0000
GR5
1B33181B4B1F0000000000000000000000000000000000000000000
0000000000000000000000100
GR6
0D0A000000000000000000000000000000000000000000000000000
0000000000000000000000000

0001
GRFNM OHIO
GR1
BB33181B4B2100070404040404040404040404020201010000000010
1010202020404040808101F00
GR2
1B33181B4B2100FF0000000000000000000000000000080808080000
000000000000000000000FF00
GR3
```

continued on page 23

Multiplan Exercises #2

by Herbert Schlesinger, USA

If you have not left Multiplan, and wish to continue the session: Press "T" (transfer); "C" (clear); and then "Y" to confirm. Otherwise bring up a blank screen as on page 4. We shall use this to illustrate the Functions which follow.

FUNCTIONS:

Using functions will allow manipulation of the data entered into the spreadsheet. Remember - ALL functions require an ARGUMENT. This means that the function needs something to operate on. Highlight R1C1 and type in this formula: SQRT(36).

This formula appears at the bottom of the screen, but Multiplan displays "6", the square root of 36, in the active (highlighted) cell. Here the argument is 36 which is enclosed in parentheses. Since all functions are a formula, or part of one, it is necessary to type in the equal (=) sign which will bring up the VALUE notation at the bottom of the screen. It is not necessary that the argument be a number. It can refer to another cell which in itself is a number: (Enter 100 in R1C4) and place the cell pointer in R4C1 and enter:

SQRT(R1C4) the number 10 appears in R4C1. If you now goto R1C4 and change that number to another you will notice that R4C1 changes in accordance. If the contents of R1C4 is not a perfect square, the fractional root will show in R4C1. Try all this.

The cell refered to must contain a number capable of being an argument. Functions can be used as arguments for other functions. Example - if R1C1 held the figure -100 (negative 100) and you asked Multiplan to take its SQRT - you would get an "Error message". The cell would show #NUM! indicating an error because there is no square root of a negative number. However if you asked for-

SQRT(ABS(R1C1)) Square root of Absolute R1C1

you would get 10 in the highlighted cell. Notice that every opening parentheses must have a matching closing parenthesis.

!!!! Clear the screen and bring up the file " GRADES1 "

If we place our cell pointer in a cell beneath a column of numbers, as at R8C4 and label it "AVERAGE" in R8C1,and then enter SUM(R1C4:R6C4) the sum of those six numbers would appear there. The colon (:) between the two cell names separates the beginning and the end of a Range. What that formula really said was "In this cell place the sum of all the cells from Row 1 Column 4 (R1C4) to Row 6 Column 4 (R6C4)". Do not forget that if we want the class average we must divide this sum by the number of members in the class.

IMPORTANT:

Usually when starting to use Multiplan for a spreadsheet, the first thing to do is select "Options" and change "Recalc" from yes to NO by pressing the space bar and then pressing <ENTER> so that the entire sheet will not be recalculated each time a figure is entered. If this is done, the work is recalculated by pressing FCTN 8 when finished with a series of entries or when leaving MP or saving a sheet. In these exercises this should not be done because so few cells are involved, but if the screens do not do the calculations, press FCTN 8 (the RECALC key).

LIST OF FUNCTIONS:

MATHEMATICAL FUNCTIONS

ABS(N) Returns the absolute value of a number.

EXP(N) Returns the exponent (antilog), the base of the natural logarithm to the power of the argument. If R1C5 contains the value 1, the formula EXP(R1C5) displays 2.7182818.

INT(N) Returns the integer equivilent of N, truncating (not rounding) the decimal places. Thus 123.999 becomes 123 using INT(CELL)

LN(N) Returns the natural logarithm. If R1C1 contains 2.7182818 the formula LN(R1C1) displays 1.

LOG10(N) Returns base 10 logarithm.

MOD(N1,N2) Returns the modulus (remainder) of N1 divided by N2

ROUND(N1,N2) Returns the number N1 rounded to N2 decimal places.

SIGN(N) Returns a number indicating the sign of the argument 1 is postive, 0 is zero, and -1 if negative.

SQRT(N) Returns the square root of the argument.

SUM(R) Displays the sum of a range of numbers.

BUSINESS FUNCTIONS

DOLLAR(N) Converts the argument to a dollar amount rounded to two decimal places, displayed with a dollar sign.

NPV(N,R) Returns the Net Present Value of an investment where N is the interest rate and R is a range of cash flows.

STATISTICAL FUNCTIONS

AVERAGE(R) Returns the average of a group of numbers.

COUNT(R) Counts number of cells containing numeric values.

MAX(R) Displays the largest number in a range of values. Example: If rows 1 thru 5 of column 1 contain the values 80, 62, 75, 90, and 70, the formula MAX(R1C1:R5C1) displays 90.

MIN(R) Displays the smallest value in a range. Using the example above the formula MIN(R1C1:C5C1) will display 62.

STDEV(R) Returns the Standard deviation of a range of numbers. Using the same numbers as above, STDEV(R1C1:R5C1) will display 10.526158.

LOOKUP FUNCTION:

Sometimes there is no direct mathamatical relationship between a value and the desired result. Such a case is in figuring taxes where some entries must be looked up in a table. Multiplan can LOOKUP data in a table on the spreadsheet.

LOOKUP(N,R) Looks up the value of N in the first row or column in range of cells(R)> If it can not find the exact value of N, it selects the closest lower number. Example: Columns 1 and 2 on the spreadsheet contain these values:

	1	2
1	0	0
2	10000	0.1
3	20000	0.2
4	30000	0.3
5	40000	0.35
6	50000	0.4

The formula LOOKUP(45000,R1C1:R6C2) displays 0.35, since 40,000 is the next smallest number below 45,000 and 0.35 is the associated percentage figure.

Bring the file "TAXES" to the screen.

We have a series of labels in column 1 with only "Income" entered, and a lookup table occupies columns 4,5 and 6. We will ask Multiplan to figure the tax on the income of 9500. (The $ signs and all that will come later). First we must find the "base tax" so place the pointer in R4C2; press '=', and enter the formula: LOOKUP(N,R) where N is the amount to be found on the chart and R is the range describing the chart itself. We have:

Value: LOOKUP(9500,R2C4:R12C5)

Upon pressing the down arrow or <ENTER>, 112.5 appears in cell R4C2. This is the closest to the full amount in the chart without going over. We then place our pointer to R6C2 and subtract the table amount from the full salary - thus: 9500-LOOKUP(9500,R2C4) and 500 appears in R6C2. Next place the pointer in R5C2 and use the lookup function to read the last column of our table: LOOKUP(9500,R2C4:R12C6). MULTIPLYING our shortfall (R6C2) by the percent in R5C2 is done by doing just that: Place pointer in R7C2, enter the formula (R6C2)*(R5C2) gives the amount in R7C2. By placing the pointer in R9C2 and entering the formula (R4C2)+(R7C2) gives the total tax due. All this is done by formulas which you can see if you select FORMAT: Options and select formulas:(Yes) instead of the default (No). What we have done here has already been saved as " TAXES1 " which we shall use shortly for more refining. In the meanwhile:

LOGICAL FUNCTIONS

These are used to make decisions in the spreadsheet. IF is the key. What we have is the IF function:

IF (something is true, do this, otherwise do this)

The IF function uses operators:

Operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
=	Equal
<>	Not equal to

Example: IF(R1C1>0,90/R1C1,0) -- This means "If the value in R1C1 is greater than 0, then display 90 divided by the contents of R1C1, else display zero. The commas in the argument determine the decision and the outcome.

AND(N1,N2) Determine whether or not two conditions are true simultaneously. If R1C1 contains the number 0, and R1C2 contains the number 100. The formula IF(AND(R1C1>0,R1C2>0),R1C1/R1C2,0)

Zero will be returned because while it is true that R1C2 is greater than zero, it is not true that R1C1 is greater than zero.

IF(N1,N2,N3) Makes a decision. N1 must be a comparison as: (R1C1>0 or R20C4<>100). If the comparison is true the number or formula in N2 is displayed in the cell. If the comparison is false then N3 is displayed in the cell.

OR(N1,N2) Compares two conditions to see if either is true. Example: R1C1 contains 100, R1C2 contains 0. The formula IF(OR(R1C1>0,R1C2>0,R1C1+R1C2,"both zero") will display 100 (sum of the two cells) but if both had contained the number zero, the formula would have displayed the message "both zero".

Other functions for trig, formatting, testing formulas and handling Alphas will be discussed later. Now lets do some work:

Bring the file "SALES " to the screen.

The names and amounts are already there but the total is not. To put these amounts on the screen (and in the spreadsheet); position the cell-pointer at R10C2. To get the total in this cell press = and then enter a formula: SUM(R3C2:R8C2) and the sum will appear in the highlighted cell after <ENTER> or an arrow key is pressed.

To get the statistics (the headings are already there) position the Pointer (we will call it that from now on) in C6 opposite the label "Count", press = and type in the formula COUNT(R3C2:R8C2) and then press the DOWN arrow key. Press = and type in the formula for the next item in the statistics chart:

MIN(R3C2:R8C2) press the DOWN ARROW key again and do the same for the remaining formulas which are:
MAX(R3C2:R8C2)
AVERAGE(R3C2:R8C2)
STDEV(R3C2:R8C2)

These formulas can also be written with the range : in the row portion alone, in the column portion alone or with the range in both parts. Example: STDEV(R3:8C2)

Now play around with the numbers in column 2. Each change you make will affect the total and the statistics. (If you changed the Options command to NO, the changes will not be made until FCTN 8 is pressed). It is more fun on a screen such as this to have the RECALC(Yes) on so the changes are automatic.

continued from page 12

LA 99ers TOPICS, January, 1991: Thoughts From The President; New Age/99 by Jack Sughrue; Conversion of Forth Screens to D/V 80 (and back to screens); TI - Base hints; advertisement for Fest West Faire '91; The Cracker Barrel by Chick De Marti; Funnelweb's Diskreview, a review by John Bulakowski; Heat Transfer Direction by Fred Moore.

SPIRIT OF 99, January, 1991: Index for 1990; What's Hott; Clearing House BBS Idea by Irwin Hott; Meeting Dates for 1991.

TI FOCUS, January, 1991: News and Views from CH99 by Tom Arnold; Running a Hockey Pool by Tom Arnold; Club Page by Tor Hansen; I Like Brain Games by Jim Peterson; Love-Hate (Life with the Cor-Comp Ramdisk); New Age/99 by Jack Sughrue #10; New Age Special (Funlweb Review).

UGOC ROM, January, 1991: Fest West '91 Update; UGOC Hall of Fame; In My Humble Opinion by Bill Nelson (TI-Base review).

BOSTON COMPUTER SOCIETY, Oct/Nov, 1990: Listen by Justin Dowling; Introduction to the P-Code system by Ron Williams; Basic Expanded and Expounded by Barry Ensley; Using Tips V1.7 by Deanna Sheridan.

THE FRONT RANGER, December, 1990: Pageform by Bill Gaskill; Pix Ease Software Reviews by Gene Bohot; Air Taxi Review by Jim Peterson; Plus 2.01 by Jack Sughrue; New Age/99 # 11 by Jack Sughrue; Librarian Page by Bruce Almeida.

January, 1991: TI99/4A equipment inventory for TI-Base v3.0 by Bill Gaskill (huge article!); Product Review the EN-C1 Power Center by Steve Funkhouser; New Tips Files, comments by Barry Traver.

TIDBITS, December, 1990: President's Bit and In The News by Gary Cox; Page Pro 99 Page Formatter by Bill Gaskill; State of the 4A by Steve Mickelson; GIF Mania - review by Gary Cox; Hints, Tips and Answers by Bill Sponchia.

January, 1991: President's Bit and In The News by Gary Cox; Asgard News-Online by C. Bobbit; Computers & Operations by Chick De Marti; Extended Basic by Art Byers; Funnelweb - letter from Tony McGovern; TI-Writer by Barry Peterson; Editor's Notes by Marshal Ellis.

Beginning Forth - part 4

by Earl Raguse, UGOC, CA USA

MORE EDITOR CHANGES

Last time in BF#3, I gave you an improved Editor. This month we will add a new word APPEND to Screen #41, line 7 and extend line 10 of Screen #38 to call for it. APPEND is a remarkable word for its short length.

What APPEND does, is to permit you to append a line or part thereof, deleted with FCTN 3 or FCTN 7, to the right of the cursor on any line, on any screen! All you need to do is to get to the desired screen, if not the one you are on, place the cursor where you want the moved text to start and press CTRL 9 and its done. You may execute all sorts of words or key strokes in between, as long as you do not disturb PAD. Now that is real editing power. All that because I hate to type, well not really I suppose, or I would not be doing this article, but I do hate to type anything I can make the computer do.

How did this all come about? Well one day I discussed the possibility of such a word with my friend and Forth mentor Lutz Winkler of San Diego, and he said, "That does not sound too tough, let me look at it". Almost by return mail, I had it, not only that but a new version of the autorepeat cursor and Inverse Video for the character under the cursor. After I tried that, I decided I did not like the cursor movement and the Inverse Video, although many people might think it great. I studied his technique and decided I could take just the part I originally wanted and add it to Pete Korner's version of Editor, Screen #41 from BF#3. Fortunately it worked just fine, and that is what I have attached hereto. I had to move existing words around on Screen #41 to make room for APPEND which is on line 7. Also I had to add to line 10 of Screen #38 to call for APPEND. The new line 10 is;

```
10 7F OF -TAB ENDOF 1F OF APPEND ENDOF
```

After you make the above changes, if you BSAVEd your dictionary last time as I suggested, you will have to FORGET EDITOR1, and reload -EDITOR and whatever else you had loaded, then BSAVE it again.

Last time I had my BLUNDERBUSS out again and told you to FORGET BOX. That is the word to forget allright, but its not in Forth's vocabulary, its in EDITOR1's vocabulary, hence you must FORGET EDITOR1 instead. If I make another dumb mistake in the future (I probably will), and it gives you trouble, do not despair, call me at 714/847-5875, or corner me at the meetings.

DEFINING WORDS AND LOOPS

Now that we gotten over the interruption to our plan, lets get on with it. This time we will learn to define new Forth words. We will also use some definite loop control words like DO and LOOP. I had planned to do more with loops, but the unplanned Editor changes take up too much space, so next time I will continue with +LOOP and LEAVE and some indefinite loop control words BEGIN UNTIL WHILE REPEAT.

DEFINING NEW FORTH WORDS

Defining new Forth words or redefining old Forth words is so easy you can get carried away and overdo it. When I first was learning Forth, I was tempted to define a lot of new words which just combined existing Forth words to do something which I thought at the time was very useful. Some did stand the test of time, but most did not. In the beginning one does not have much foresight, but its fun and almost harmless as long as you do not try to sell the world on your wonderful inventions that sometimes backfire.

One of the major powers of Forth is that if you do not like the way it works, or you do not like the words as written, you can change them to suit yourself. Not many other languages can do that. BASIC allows you to define Functions, but they are very slow, and you cannot call them from other programs unless you repeat them. However, once you define a word in Forth, it is on a par with other resident words, it simply becomes part of the language. You must save it to disk of course if you want it to be available after you turn off the computer.

The : (colon) is a Forth defining word, it signals Forth that the word following is to be compiled into the Dictionary along with the necessary addresses to access and execute the words following. The words following the newly defined word must have been previously defined, the definition's end is signalled by the ; (semicolon). Pretty easy, let's do one, but first let's look at a few existing Forth words.

The word CLS clears the CRT display. The word EMIT takes a number from the stack and prints it on the CRT, if it is the ASCII code of a printable character, else it prints a white blob. The word GOTOXY takes two numbers off the stack and uses them for X and Y coordinates on the CRT. The CRT has X=40 maximum characters (columns) horizontally and Y=24 maximum characters (rows) vertically.

We now define a new word STAR by entering the following, spaces are important, in the so called immediate mode, (ie Forth is booted and you have a cursor).

```
: STAR 20 12 GOTOXY CLS 42 EMIT CR ;
```

What do you suppose STAR does? Enter STAR and try it. It should go to the center of center of the screen (CRT), clear it, and print an asterisk (ASCII 42). Now was that not easy? You are right though, this is not one of the words which we are going to save for posterity.

When we do wish to save newly defined words, we should pick an empty screen, then CLEAR it if necessary. Please read Chapter 3 of the TIFM again. Then type your definition and when satisfied, enter FCTN 9 (Back) to exit EDITOR then FLUSH to save it to disk. Then to execute the word, enter the Scr# and LOAD, when the cursor returns, enter your word and hope for the best.

If it does not work as expected, EDIT the screen again. When you LOAD it again, Forth will alert you that your word "IS NOT UNIQUE" unless you remembered to FORGET it before LOADing. No problem, it just uses up memory, and I will show you a easy way to avoid this later. The last version of your word is what will execute. There is an exception, suppose STAR is called by DOIT, another previously compiled word. In that case, DOIT will execute the version of STAR in force at the time DOIT was compiled. In the immediate mode, if your word does not work right, you cannot edit it, all you can do is FORGET "word" and type the whole thing again.

STAR may be a very good word for some application, but its too limited, it does just one thing, we could make it more flexible my leaving out the numbers and entering them on the stack before executing STAR.

Suppose we defined STAR as follows;

```
: STAR ( ch r c --) GOTOXY CLS EMIT CR ;
```

The (ch r c --) specifies that CHaracter code, Row (y) and Column (x) must be on the stack when STAR is executed, and the -- signifies that these numbers will be gone after execution.

Notice that when numbers are part of a word, they are entered on the stack in left to right order, ie last on top. When you put numbers on the stack for use by a word, you must put them in correct order, first used last entered, ie a LIFO stack. Think about it, and try it out.

STAR is now a more flexible word which could print any ASCII character anywhere on a cleared CRT. The trouble with this approach is that it seems kind of dumb to call a word STAR if we are going to print Q's, K's, or even spaces with it; and what if we did not want to clear the screen every time, after all that would limit us to one character on the CRT at a time. Also are we sure we want a Carriage Return after each print? (Apparently a few lines of the article went astray here. ED)

It sure looks as if we should not combine these particular words into one new word just to be more efficient, unless, of course, our major purpose in life is to write single Stars anywhere on the CRT.

One of the commonly used Forth rewrites is to change GOTOXY to AT as follows.

```
: AT GOTOXY ;
```

You will find that AT is very frequently used, so a lot of keystrokes and screen space can be saved by it. CLS, CR and EMIT are usually used separately as needed. CLS and CR are frequently combined into words that print prompting messages. AT is what I call a Useful Forth Word, and I have attached several of my favourites. I will refer to them as UFW's.

One of the more cumbersome Forth words is EMPTY-BUFFERS, which clears all of the screen editing buffers. If a screen is already in the buffer, and you have made changes on it, but wish had not and since you have not yet FLUSHed the changes to disk, you might think you could simply enter the screen number and EDIT, to reload the old version. Well you can, but it will not do you any good because Forth will look in the editing buffers and find it there and just print that to the CRT instead of going back to the disk. If you enter EMPTY-BUFFERS first however, Forth clears all five screen editing buffers and loads in the specified screen. This long but frequently used word is often abbreviated to one of the following, ZAP, E/B, EB, MTB, etc. Take your pick. I like ZAP, which can be defined as follows;

```
: ZAP EMPTY-BUFFERS CLS 12 12  AT  ." BUFFERS
  ARE EMPTY" ;
```

We have introduced an important Forth word." (dot quote) which prints to the CRT everything to its right until another set of quotes " is found.

I have included here, copies of my favourite Forth Abbreviations as Screen #2 and my favourite UFW's as Screens #88 and #89. I will not discuss all of them this time, but I will talk more about them as we need them. If you can figure out what they do, use them, some are easy, some not. As you might expect, I make big use of E; , F; , L; , FG , SC , MV , VL and my latest UFWs MVF and LS.

I suggest you type them all and incorporate them into your dictionary. If you LOAD these screens last after all other necessary stuff, you can dump them all with FG DEC if memory runs low. FREE lets you check memory remaining. Once loaded, I predict you will have to be very desperate before you do that.

Handy as the Forth Abbreviations are, they are intended for your use only for privately conversing with Forth. If you use these things on screens which others must interpret, you will be degrading the transportability of Forth. They are great time and finger savers, and take up very little memory. I hate to type anything I can make the computer do for me.

LOOPS

Forth DO loops do the same job as FOR NEXT loops in BASIC, and are of the form;

```
L I DO . . . . . LOOP
```

Loops require a limit L and a starting value for the index I in front of the word DO, then come the words to be repeatedly executed (none are required, as is the case for WAIT below), LOOP increments the index I, tests it against the limit L and if I less than L, action returns to just after DO, else it returns control to the word following LOOP.

Because of the way LOOP works, the number of times through the loop is L-I-1. Thus you must adjust L to get the desired (continued below)

```
SCR #41
 0 ( EDITOR REPEAT KEY ROUTINE Pete Korner 12 3 84 )
 1   BASE->R DECIMAL 0 VARIABLE MY
 2 : BLINK CURPOS @ DUP VSBR MY C!
 3   3 0 DO DUP 30 SWAP VSBW LOOP MY C@ SWAP VSBW ;
 4   4 CONSTANT WW ( REPEAT SPEED) 30 CONSTANT XX
     ( DELAY)
 5   0 VARIABLE YY XX VARIABLE ZZ 0 VARIABLE OK
 6
 7 : APPEND PAD PTR C/L R/C DROP - CMOVE RELINE ;
     \ newline
 8
 9 : RKEY BEGIN ?KEY -DUP BLINK BLINK
10   IF YY @ 1 YY +! IF ZZ @ YY @ < IF WW ZZ ! 1 YY !
11   1 ELSE OK @ OVER = IF DROP 0
12   ELSE 1 DUP YY ! ENDIF ENDIF ELSE 1 ENDIF
13   ELSE XX ZZ ! 0 YY ! 0 ENDIF UNTIL DUP OK ! ;
14 R->BASE
15
```

```
SCR #88
 0 ( EGR's USEFULL FORTH WORDS #1 REV 4  7 11 87 ) 2
   CLOAD LS
 1 : BS ' TASK 20 BSAVE . ;
 2 : ZAP EMPTY-BUFFERS CLS 10 12 GOTOXY ." BUFFERS MT" ;
 3 : WAIT  ( SEC---) 7200 * 0 DO LOOP ;
 4 : MS (millisec wait) 5 * 0 DO LOOP ;
 5 : LF  ( lines to feed) SWCH 0 DO CR LOOP UNSWCH ;
 6 : FF  ( formfeed) SWCH 12 EMIT UNSWCH ;
 7 : .VL SWCH VLIST UNSWCH ; ( dictionary to printer)
 8 : P ( SCR#---) SET SWCH LIST UNSWCH ; ( print
     screen#)
 9 : .HD  ( U ) SWCH 25 SPACES ." DISK: " . CR UNSWCH ;
10 : CLRSCR ( s1 s2 ---) 1 + SWAP DO I I . CLEAR LOOP ;
11 : PRTSCR ( s1 s2 ---) 1 + SWAP DO I I . P LOOP FF ;
12 : FREE SP@ HERE - . ." BYTES LEFT" ;
13 : PS SET PRTSCR ;
14 : \ ( skip to end) IN @ 64 / 1+ 64 * IN ! ;
     IMMEDIATE
15   -->
```

number of passes through the loop. The value of L must be larger than I, unless we use +LOOP, to be discussed next time. +LOOP can increment I by any number just like STEP in BASIC FOR NEXT loops.

The word WAIT uses a very elementary loop, it serves as a delay loop in the same way you would use a FOR NEXT loop in BASIC. The limit is not a part of the definition so you can change the delay by putting a number on the stack before executing WAIT. It takes about 1/7200 of a second for Forth to execute an empty loop. Thus if a number, say 2, were on the stack, it would be multiplied by 7200, the product 14400 would become the DO limit L. We would thus get a 2 second delay. The word MS works the same way except the delay in milliseconds is put on the stack. The word CLRSCR is also a very useful looping word. It accepts a starting screen s1 the ending screen s2, adds 1 to it, SWAPs them and they become the DO limits. The loop CLEARs the Ith screen, then I . prints the screen number.

```
SCR #2
 0 ( ABBREVIATIONS & UFW'S ETC  EGR 7 11 87 )
 1 : DEC  DECIMAL ;      : DU DUMP ;         : 2/ 2 / ;
 2 : FC FORTH-COPY ;     : F;   FLUSH ;      : 2* 2 * ;
 3 : UD UPDATE ;         : FG   FORGET ;     : W; WHERE ;
 4 : CO COLD  ;          : SM   SMUDGE ;     : E; EDIT ;
 5 : VL VLIST ;          : TX   TEXT ;       : L; LOAD ;
 6 : AT GOTOXY ;         : SC SCOPY ;        : C; CLEAR ;
 7 : HOME CLS 0 0 AT ; : MV SMOVE ;          : A; ABORT ;
 8 : MVF ( move a set of screens forward to overlapping
     area )
```

continued on page 20

Spellbreaker part 4

Copyright 1985 Infocom
This walk through is by Scorpia, Copyright 1985.

Now it is time to pay a second visit to the volcano, and learn the secret of the gold box. Blorple "FIRE". Here you are in a different spot on the side of the volcano. A little further away, you can see an unmelted outcropping with a cube on it. The distance between is too far to jump, but you just might be able to throw something there.

Put the "FIRE" cube in the gold box (the design will change), then take it out again. Throw the gold box onto the outcropping...it will just make it there. Now, blorple "FIRE", and go EAST through the "impossible" exit. AHA! Now you are on the outcropping! And that is the secret of the box. If you leave it behind somewhere (except inside another cube), you can go back to wherever it is by going through the "impossible" exit of whichever cube was last in the box. This will work with any of the cubes.

Ok, pick up the cube from the outcropping. WOW! What was that??? As soon as you touch the cube, a surge of power runs through your body. This must be SOME cube...and it sure is. Write "MAGIC" in the cube. As long as you have the "MAGIC" cube, you can learn an unlimited number of spells, and some of them will be more powerful (which will come in handy shortly).

Enough dilly-dallying here! Blorple "VOID", and go through the east exit, which brings you to the inner vault. Here is a good place to save the game. For one thing, once you pass through the door into the outer vault, you WILL NOT be able to save the game!! So, learn Jindak several times (four should be enough), save now, then rezrov the door (the spell is more powerful, and the door will open). Go north into the outer vault.

Here you see two piles of cubes, labeled "X1" through "X12". Eleven of these cubes are dummies; only one is a real cube. The trick is to determine which one is the right one, in the time allowed before that obnoxious alarm fairy shows up. Also, each time you enter the vault, the right cube is determined randomly; this is why you are not permitted to save the game here...that would make things much too easy!

Now, here is one way of determining the right cube. Remember that your time is limited here, so do not waste a single move! First, cast Jindak. Both piles will glow, but one will glow brighter than the other. That is the one with the real cube.

So far, so good. Take 3 cubes from each pile (you can do this with one command), and cast Jindak again. There are two possibilities: either both piles will glow the same (in which case, the right cube is one of the three in your hand), or one pile will still be brighter (in which case, it is one of the three still left in the pile).

For purposes of example, suppose the first pile glowed brighter when you originally cast Jindak. This tells you that the right cube must be among X1-X6. So you take X1-X3 from the first pile and X7-X9 from the second (it is crucial to keep the number of cubes in each pile equal when casting Jindak). Now you cast Jindak again. Both piles glow equally bright.

Now you know that the right cube must be among X1-X3. What you need to do now is split the cubes up, so one more Jindak will indicate proper one. You also need to keep the piles balanced. So you put (say) X1 and X7 on the first pile, and X2 and X8 on the second pile. Cast Jindak again.

If the first pile glows brighter, then X1 is the right cube. If the second pile glows brighter, X2 is it, and if both piles are the same in brightness, then the cube is X3.

That was the first case. Now suppose, after you took X1-X3 and X7-X9, the first pile still was brighter. Then the right cube must be among X4-X6. The first thing to do is get rid of the cubes you are currently holding. The easiest way is to just stuff everything into the zipper (all the dummy cubes disappear anyway when you leave the vault) with a "put all in zipper" command.

Ok, now you take X4, X5, X10, and X11. You put X4 on the second pile, and X10 on the first pile, and cast Jindak. If the first pile glows brighter, you know the cube must be X6. If the second pile glows brighter, obviously it is X4. And if both glow the same, then the cube is X5.

This method will work, regardless of which pile the right cube is in. It does help to keep track of your moves on paper, so you do not become confused as you move the cubes around (which could easily happen).

Once you have the right cube, blorple it immediately, because the guards will be coming in the door (there just is not any way of shutting up that fairy). You will be in the Sand Room. Funny sort of room....kinda reminds you of an hourglass, does it not? In fact, the room could be considered a sort of time machine (by the way, this is another good place to save!).

There are two exits, up and down. You must go down first, where you will find yourself in a familiar place...the cell where you found the moldy spell book. But...the room is not a wreck, it looks new! All the furniture is in good condition, and the door is on the cell. And there is the mahogany cabinet, with the lock in working condition. What is up?

Well, for those of you who played "Sorcerer", you might remember the incident of the coal mine, where you met yourself coming and going. This is the same sort of situation. Everything must be as you found it when you first blorpled the "water" cube. So, unlock the cabinet with the key you received from Belboz.

Inside is a vellum scroll. Take that. Now comes the fun part. You must put your beloved spell book into the cabinet. So, you best memorize some spells before you do that. Actually, all you need to do is learn the blorple spell a few times, because you are getting near the end game, and it is really the only one you need now (well, almost).

Ok, the book goes into the cabinet. Lock the door, then rezrov the cell door. WHAM! The door flies open, and you can hear guards coming. Blorple the "X" cube from the vault. If you did anything wrong in the cell, you will die on your way to the cube (in which case, restore and try again), but we will hope that does not happen.

Now (save again, just in case) go through the up exit of the Sand Room. Here you are in the Ruins Room...back when it was being flooded. You see a sack on the floor. Get that and open it. Inside is a flimsy scroll with Girgol on it. Take the scroll. And just as you did in the cell, you must see to it that everything here was just as you found it.

Empty the zipper into the sack. Now, copy the Girgol scroll onto the vellum scroll. Put the flimsy scroll into the zipper, close it, and drop it. At last, you are ready for the end game and your showdown with the mysterious cloaked figure.

continued from page 22

at the time and just stuck. Running the console tester revealed a fault in the low byte system RAM (6810) which when replaced solved all the problems. The system RAM holds all the information for setting up the video processor and although the console tester runs a memory test about 400 times through the system RAM, it did not show the original fault. It just goes to show that it can be harder to find the problem in a console which is almost working than in one that is completely dead. Also, if there is an intermittent problem, it will probably get worse and easier to find given time.

EPROMs for the RAMdisk

We are now gearing up for people to add EPROMs to their RAMdisks. To do this, you need a RAMdisk with 32K RAM chips (62256 and similar) and you must have at least three of these. Then you need to install some 74LS08 ICs, one for every four EPROMs you want to add. I suggest you put in some sockets for the EPROMs as you

continued on page 10

Jenny's Younger Set

Dear Jenny,
    Here is another program I have sent in. This one is a Doctor Who quiz.
Vincent Maker

```
100 REM
110 REM BY VINCENT MAKER
120 CALL CLEAR
130 PRINT "A DOCTOR WHO QUIZ BY"
140 PRINT
150 PRINT
160 PRINT "VINCENT MAKER."
170 FOR A=0 TO 5
180 PRINT
190 NEXT A
200 INPUT "PRESS ENTER.":KLL$
210 PRINT "1. WHO LEFT THE DOCTOR TO TRAVEL UP THE
    AMAZON RIVER WITH CLIFF JONES? "
220 PRINT
230 PRINT
240 PRINT "A) JO GRANT"
250 PRINT "B) KAMELION"
260 PRINT "C) HARRY SULLIVAN"
270 PRINT "D) SARAH JANE SMITH"
280 PRINT "PRESS YOUR GUESS."
290 CALL KEY(0,J,K)
300 IF K=0 THEN 290
310 IF J=65 THEN RIGHT=1 ELSE WRONG=1
320 IF J=65 THEN PRINT "RIGHT." ELSE PRINT "WRONG."
330 INPUT "PRESS ENTER.":JKK$
340 PRINT "2. WHO DIED SAVING EARTH AGAINST THE
    CYBERMEN?"
350 PRINT "A) JO GRANT"
360 PRINT "B) THE MASTER"
370 PRINT "C) ADRIC"
380 PRINT "D) WHO KNOWS"
390 PRINT "PRESS THE ANSWER"
400 CALL KEY(0,M,N)
410 IF N=0 THEN 400
420 IF M=67 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
430 IF M=67 THEN PRINT "RIGHT." ELSE PRINT "WRONG."
440 INPUT "ENTER":KKK$
450 CALL CLEAR
460 PRINT "3. WHO TRIED TO TAKE OVER THE WORLD FROM
    GLOBAL CHEMICALS?"
470 PRINT
480 PRINT "A) B.O.S.S."
490 PRINT "B) DALEKS"
500 PRINT "C) CYBERMEN"
510 PRINT "D) WOTAN"
520 CALL KEY(0,J,K)
530 IF K=0 THEN 520
540 IF J=65 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
550 IF J=65 THEN PRINT "RIGHT" ELSE PRINT "WRONG."
560 INPUT "PRESS ENTER.":MK$
570 PRINT "4. WHOSE CAR IS CALLED,""BESSIE""?"
580 PRINT
590 PRINT "A) DR WHO"
600 PRINT "B) THE MASTER"
610 PRINT "C) JO GRANT"
620 PRINT "D) WHO KNOWS"
630 PRINT
640 PRINT "PRESS YOUR GUESS."
650 CALL KEY(0,H,J)
660 IF J=0 THEN 650
670 IF H=65 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
680 IF H<>65 THEN PRINT "WRONG" ELSE PRINT "RIGHT"
690 IF RIGHT=0 THEN A$="TERRIBLE, 0/4"
700 IF RIGHT=1 THEN A$="POOR, 1/4"
710 IF RIGHT=2 THEN A$="FAIR, 2/4"
720 IF RIGHT=3 THEN A$="GOOD, 3/4"
730 IF RIGHT=4 THEN A$="VERY GOOD, 4/4"
740 INPUT "PRESS ENTER.":MM$
750 PRINT A$
760 END
```

continued from page 23

I am going to explain some of the data in GRF1 in a condensed form to give you a quick look at what is going on. Note: The ">" is generally used to denote a Hex number. The line GR1 (under DISK) goes like this. 1B = >1B = 27 = ESC, 33 = >33 = 51 = 3, 18 = >18 = 24, or <ESC>"3"CHR$(24), 24 being 24/216", right out of the printer users manual. To my printer that phrase means, set line spacing to 24/216ths inch. This is needed for graphics. Next, 1B = >1B = 27 = ESC, 4B = >4B = 75 = K, 1F = >1F = 31, 00 = >00 = 0, or <ESC>"K"CHR$(31)CHR$(0). To my printer this says, print normal-density graphics, 31 bytes of graphic data will follow. The printer will then consider the next 31 bytes of data to be part of the graphic itself. As you can see we are taking the characters two at a time, 1B, 33, 18, 1B, 4B, etc. Each of these small units is one byte. This means that the printer will actually need 62 characters from this data field to make a total of 31 bytes. "I hope this is helping and not confusing you." So the FF, AA, D5, etc. are used to print the first line of the DISK graphic. At this point I have designated 6 lines for a graphic in the Db, but the biggest graphic is OHIO which takes only 5 lines. I have temporarily placed the characters ODOA at the beginning of lines GR6. OD = ASCII code 13, or CR, and OA = ASCII code 10, or LF. If printed, GR6 will produce a Carriage Return and a Line Feed. The reasoning is this. Printing GR1 will not force the printer into action. Apparently a LF is needed to cause the data in the printer buffer to be printed on the paper. GR6 contains both a CR and an LF, so printing out the complete disk graphic would go like this: PRINT GR1,GR6,GR2, GR6,GR3,GR6,GR4,GR6. In DISK, GR5 graphically prints a blank. Remember GR6 contains CR,LF. We also have the ability to use the TIB system CR, LF. They are entered like this: PRINT GR1,(CR),(LF),GR2, (CR),(LF),GR3,etc. This is too much typing for me, but the functions can be used together to an interesting advantage. The line PRINT GR1,(CR),GR1,GR6,GR2, (CR),GR2,GR6,etc. will produce a double struck graphic. GR1 will be printed, the (CR) will return the carriage to the beginning of that line without a line feed, the second GR1 will then be printed for a double strike affect, and the GR6 will then send a CR, LF to set up for the next graphic line. As we progress in this area I do not see any reason why it would not be possible to produce sales reports or purchase orders. These printout sheets can incorporate letterheads with graphics that could be anything from disks, as I used here, to intricate company logos. Once the graphic database is created you can use it anywhere. o

continued from page 18

```
 9 ( start# end# move#) ROT ROT DUP ROT - 1+ 0 DO CR
   OVER
10      OVER I + - OVER I - OVER OVER . ." --> " . SWAP
    SCOPY
11      LOOP DROP DROP ;
12 : PL ( page length) SWCH 27 EMIT 67 EMIT EMIT
     UNSWCH ;
13 : SET ( printer setup) SWCH 27 EMIT 33 EMIT 9 EMIT
14      27 EMIT 65 EMIT 9 EMIT UNSWCH 88 PL ;
15 : LS UPDATE FLUSH SCR @ LOAD ;

SCR #89
 0 ( EGR's USEFUL FORTH WORDS #2 REV 4  7 11 87 )
 1 : INDXHD  SWCH 24 SPACES ." INDEX FOR DISK  " . CR
     UNSWCH ;
 2 : CAT ( u1,u2,Dsk#---) INDXHD 2 LF SWCH INDEX
     UNSWCH ;
 3 : XPECT ( u--) S0 @ SWAP EXPECT 0 IN ! ;
 4 : GSTR$ ( u--) XPECT 1 PAD 72 BLANKS PAD HERE - 1 -
 5     DUP ALLOT MINUS SWAP WORD ALLOT ;
 6 : CSTR$  ( adr1 adr2 u) OVER OVER + ROT DO DROP 1+
     DUP 1-
 7   C@ I C@ - DUP IF DUP ABS / LEAVE ENDIF LOOP SWAP
     DROP 0= ;
 8 : GNUMB QUERY INTERPRET ;
 9 : PP  PAD 72 -TRAILING TYPE ; ( print the PAD )
10 : .ASK CLS 2 12 AT ." Again? Press SPACE else ANY
     other" ;
11 : PAK 12 16 AT ." PRESS ANY KEY " KEY DROP ;
12 : I> COMPILE R> ; IMMEDIATE  : >I ( a) COMPILE >R ;
     IMMEDIATE
13 : UNDO I> R> R> DROP DROP >I ;
```
o

A Look at Assembler

Switches: by Art Green, Ottawa Users Group

Most programs have switches and some have a lot of them. Most switches have only two values: YES/NO, TRUE/FALSE, ON/OFF. It is this type of switch that we will look at. Most program options are this type of switch. For example, the Assembler itself has may switches: compressed text - ON/OFF, listing - ON/OFF, cross-reference - ON/OFF, etc.

It first may seem very simple to set and test such switches, but we will see that with some thought it can be done at less cost than your first try. OK, let us look at the obvious way.

```
*
* Switch Definitions
*
ON     TEXT  'N'       Value for ON
OFF    TEXT  'F'       Value for OFF
*
SW     BSS   1         A typical Switch
*
* Setting Switch Values
*
       MOVB  @ON,@SW   Set switch ON
       MOVB  @OFF,@SW  Set switch OFF
*
* Testing a Switch Value
*
       CB    @SW,@ON   Is switch ON?
       JEQ   ISON      Jump yes
*
       CB    @SW,@OFF  Is switch OFF?
       JEQ   ISOFF     Jump yes
*
* Testing a Switch Using the IFB macro
* Generates exactly the same code as above
*
       IFB   @SW,EQ,@ON,ISON   Jump if ON
       IFB   @SW,EQ,@OFF,ISOFF Jump if OFF
*
* Reversing a Switch Value
*
       IFB   @SW,EQ,@ON,WASON
       MOVB  @ON,@SW
       JMP   ISREV
WASON  MOVB  @OFF,@SW
ISREV  EVEN
```

Before we evaluate the "cost" of doing switches this way I should point out that both the "storage" and "time" cost of a comment statement is exactly zero! You can and should use lots of them in your programs. (We defined these "costs" in last month's article.)

The storage cost of the switch itself is one byte. Setting a switch has a storage cost of 3 words and a time cost of 5 words. Testing a switch has a storage cost of 4 words and a time cost of 6 words. Reversing a switch value has a storage cost of 11 words and a time cost of 17 words.

I know you have already devised a better way to do switches, but remember I said the "obvious" way, now comes the "non-obvious" way (which will of course become obvious before we are done). You should read up on the ABS instruction, which gives the absolute value in a word of storage. When you read up on it you should notice that the STATUS bits are set according to the value of the word before the instruction is executed. That does not affect us now although it is interesting (as an aside, what other instructions set status according to the before value, what instructions do not set the status?). Note, however, that the instruction:

```
ABS   @A
```

as well as giving the absolute value of A will tell us whether the value was zero or non-zero. Also notice that if A was zero it remains zero and that if A was non-zero it remains non-zero (with its sign possibly changed). If switches had a value zero when OFF and a non-zero value when ON, then

```
ABS   @SW
JNE   ISON
```

is a better way to test switches (less cost) than our initial try. It is convenient that the 9900 has a easy way to set a word zero or non-zero:

```
CLR   @A      Sets A to zero
SETO  @A      Sets A to >FFFF, non-zero
```

Thus setting switches this way has less cost than our first try. So far we have improved on setting and testing, can we do as well on reversing the value of a switch? Consider this:

```
ABS   @A
DEC   @A
```

This is certainly getting non-obvious, but looking closely, after the ABS instruction our switch A has the value zero or +1. The DEC makes the zero into -1 (non-zero) or makes the +1 (non-zero) into zero, thus reversing the switch value. Compare the costs of reversing a switch this way to our original method -- quite a saving.

It is time for another plug. A program full of non-obvious switch setting, testing and reversing is going to be a program that is hard to read. I certainly do not like non-obvious programs. Can we use non-obvious switches that are cost effective and still have a readable program? Of course we can! We write a MACRO that hides the actual code, and at the same time reduces the number of lines we have to write and type (the third cost of a program). With the appropriate macro written we could code the following.

```
* Setting Switches
       SW    @A,ON          Set switch A on
       SW    @A,OFF         Set switch A off
       SW    @A,REV         Reverse switch A
* Testing Switches
       IFSW  @A,ON,ISON     Jump switch A on
       IFSW  @A,OFF,ISOFF   Jump switch A off
```

I probably should leave the coding of the "macro definition" as an exercise for the reader, but this time I will not. Note that the following two macro definitions look surprisingly like small programs themselves.

```
$MACRO SW             Start of macro definition
$REM   &P0 is the label field
$REM   &P1 is the switch name
$REM   &P2 is how to set the switch
$GOTO  &P2            Case: ON, OFF, REV
$LABEL ON             Case ON, Set switch ON
&P0    SETO  &P1
$EXIT                 Macro generation done
$LABEL OFF            Case OFF, Set switch OFF
&P0    CLR   &P1
$EXIT                 Macro generation done
$LABEL REV            Case REV, Reverse switch
&P0    ABS   &P1
       DEC   &P1
$END                  End of SW macro defn
$MACRO IFSW           Start of macro defn
$REM   &P0 is the label field
$REM   &P1 is the switch name
$REM   &P2 is the test value, ON or OFF
$REM   &P3 is the jump label
&P0    ABS   &P1      Test the switch
$GOTO  &P2            Case: ON, OFF
$LABEL ON             Case ON, Jump if sw is ON
       JNE   &P3      Jump sw ON (non-zero)
$EXIT                 Macro generation done
$LABEL OFF            Case OFF, Jump if sw is OFF
       JEQ   &P3      Jump sw OFF (zero)
$END                  End of IFSW macro definition
```

continued on page 2

Techo Time

with Geoff Trott

May I repeat what Lou wrote on my behalf a few months ago: if you want me to look at something for you at a meeting in Sydney, please give me a ring before the meeting so that I can come suitably equipped. If no-one contacts me, I will not bother to cart all the equipment to the meeting.

I have an appeal to make to all my readers. Can anyone help in the supply of Western Digital floppy disk controller chips used in the CorComp Micro-expansion system, the WD1773? I am trying to fix such a system for one of our members from north Queensland and I am unable to find anyone in Australia who handles these chips. I have written to a contact in Hong Kong asking him to see if they are available there. Hopefully they should be available somewhere in the world! I am interested in buying five of these (for future spares). The symptom for this CorComp system was just a blue screen on power up, which we now all know (if you have been reading this column recently) means a problem with interrupts. Sure enough, removal of the top board and insertion of a jumper caused the RS232/PIO section to work. With the top board back in place but the 9901 removed from the top board and now the 32K memory also works. Replacing the 9901 and removing the WD1773 also allowed the title screen to come up and everything except the disk controller works. I then swapped the WD1773 with another CorComp system and all works. Fortunately, all the chips are in sockets in the unit, so now I know what the problem is but cannot fix it until I can get hold of at least one WD1773! Can anyone please help?

More on Consoles

I have fixed all the consoles handed to me at the February meeting and I thought you might be interested in the faults that I found. One console had an interrupt problem which caused a blank cyan screen to be produced. This symptom is usually caused by a peripheral device malfunction, either a disk controller (as I found in a CorComp Micro-expansion system) or a RS232 card. In the case of the console, there is only one source of interrupt and that is the video processor. I actually went to the trouble of connecting a logic state analyser to the console and looking at the address lines to try to see what was happening. This told me that there was only one instruction being executed and that was at address 24h. I then looked at the addresses used when data is being read into the processor and found that these cycled round a loop of 0h, 2h and 24h. This was not correct as for an interrupt, the addresses should be 4h and 6h to read in the workspace and vector address. I then examined the data lines and found that they agreed with information on the address lines. What was happening was that the interrupt was being interpreted as a reset, so that the interrupt was never serviced and cleared. The reason for this was obvious once I realized what the problem was and measured the voltages on the interrupt input lines. Let me try and explain. The 9900 has a relatively complicated interrupt scheme which allows interrupts to be given a priority and then for the processor to vector (like a BLWP) to a different interrupt routine for each priority level. This is done by there being a single interrupt line into the 9900 (pin 32) which indicates an interrupt is pending and 4 priority lines into the 9900 (pins 33 to 36) to show the priority of the highest priority interrupt pending. The vector for each priority uses 4 bytes in memory to store the workspace pointer and the vector address of the interrupt routine, with the addresses of these vectors at memory locations at four times the priority level. So priority 1 interrupt has its vector at memory locations 4h to 7h, priority 2 at memory locations 8h to Bh and so on to priority 15 interrupt at memory locations 3Ch to 3Fh. Reset is a special case as it has its own line into the 9900 (pin 6) and its vector is stored at memory locations 0 to 3 while the LOAD interrupt has its own line into the 9900 (pin 4) and its vector is stored at memory locations FFFCh to FFFFh. Now back to the problem at hand. The TI99/4A has the interrupt priority inputs hard wired to give a priority one interrupt. That means that it does not matter where the interrupt comes from there is only one interrupt service routine and its vector is stored in memory locations 4h to 7h. When an interrupt occurs and gains the attention of the 9900, the contents of memory at locations 4h to 7h should be read in. Since this was not happening then a priority one interrupt was not being generated but a priority zero instead which uses the reset vector. Sure enough, the voltage on pin 33 was low instead of high (+5). The problem was that the connection to that pin comes from one of those white power distribution rails (which include decoupling capacitance) and the connection to that pin was broken. Surprisingly, that point only supplies pin 33 of the 9900! Once that was fixed, all was well.

Before the saga of the other two interesting ones, I was able to fix quickly two console with dead processors, one with a bad "I" key and one with a bad system low byte ROM. The dead processors were identified by their unusual patterns on the address lines and IAQ line of the 9900. The ROM was quickly identified by the console tester.

The next interesting console from last month started out with me as not working at all and I half suspected a dead 9900. On taking out the 9900 and checking it in another console, showed that it had no problem so I then started to look further. Taking out all the GROMs and sound chip and then putting one back in and the console tester started to work. However it stopped doing its checksums on the first GROM. Taking out all GROMs and the console tester ran quite happily. Much scratching of head and thinking (more hair falling out) and I decided that it must be something to do with the data bus as that is all that the GROMs are connected to. Started to take out the chips beginning with the 74LS245, 74LS373 and 74LS244. The first two tested fine in an IC tester and in another console, while the 244 failed the tester on its own but passed when in a socket (it had short leads) and when put in another console (in a socket). Interesting, I thought, but the problem must be elsewhere. I started taking out other chips which could be a problem and testing them. One by one they all tested OK. Try the next.... Soon, most of the logic chips were off the board and I re-started the brain working. That 74LS244, could it have been the problem all along? Much cursing and muttering under my breath, I started to put all the ICs back on the board along with a new 74LS244. Sure enough all worked well. That 74LS244 must have an intermittent connection inside the plastic which made contact when stressed by a socket but failed when not under stress. The IC tester used a zero insertion force socket. That was a nasty one.

The last one was the console which was almost working. It came up with a title screen with most of the writing there but the colour bars all over the place. The console tester ran flawlessly and in fact the diagnostic module, which runs tests on the console from the cartridge port, also found no errors but its screen displays were not correct and made it very difficult to read. When it was in 40 column mode or text mode everything seemed to work well. Sprites were OK. Going into BASIC looked fine but strange things happened when I tried to enter a program. For example line number 100 came out on the screen as line 11. However it was only able to be edited as line 100. Programs did not run either. When I tried Extended BASIC and did an OPEN #1:"RS232" the whole screen went strange colours and patterns. The system did not lock up, it really looked like a video problem but the 9929 was fine. There was not much I could think of as the potential problems all looked fine. The gates which generate the signals to the video processor were all working properly. The data bus was OK as everything else using it was working. The video RAM was not generating any errors no matter how it was tested. I was just about to start to take chips out and test them when the fault finally became bad enough that the console crashed. It was running the diagnostic module

continued on page 19

Regional Group Reports

Meeting Summary For APRIL

Banana Coast	14/04/91	Sawtell
Carlingford	17/04/91	Carlingford
Central Coast	13/04/91	Saratoga
Glebe	11/04/91	Glebe
Illawarra	08/04/91	Keiraville
Liverpool	12/04/91	
Northern Suburbs	25/04/91	
Sutherland	19/04/91	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.

CENTRAL COAST Regional Group

Regular meetings are 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. A variety of activities accompany our meetings. 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) 6447377 (home) or (02) 7598441 (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 monthly meetings are held at the home of Peter Young, 51 Jannali Ave., Jannali on the third Friday of each month at 7.30pm. unless otherwise advised. (02) 528 8775.

The format of each meeting is quite informal, with topics ranging from software reviews to hardware modifications with a fair sprinkling of purely social chatter in between.

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 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.

APRIL MEETING - 6TH APRIL

The April meeting will be a BUY, SWAP and SELL day. These have proved to be very popular in the past and we plan on having three such days this year. So, bring along your 'treasures' and some money - who knows what you may find?

****

The cut-off dates for submitting articles to the Editor for the TND are:

May	7 April
June	12 May
July	9 June

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest.

****

TIsHUG Meetings for Sydney, 1991

April

This is the first of the Buy, Swap and Sell days. These will all be held on the meeting days which fall in the school holidays. Mark these down on your calendar and make sure that you get to at least one of them. There will be a games centre set up for children of all ages, the RAMdisk group is planning to start its meetings and we hope the MIDI interface will be here to be demonstrated also.

May

The first all day tutorial session. Get in early and tell Russell what you would like a tutorial on. We can arrange for tutorials on any subject as long as you give us some warning. Suggested topics include: hardware clinic to explain how the bits, bytes and words go together and to answer questions; word processing for beginners; basic Extended BASIC; RAMdisks with or without EPROMs; how to modify your RAMdisk to take EPROMs; assembly class; copying software; and browse the library.

June

Demonstrations of the latest software or hardware. Watch this space for details.

July

The second Buy, Swap and Sell day. Bring things for sale and also your money for the terrific bargains. You have to be early to beat Rolf to a bargain!

August

Demonstrations of the latest software or hardware. Watch this space for details.

September

Demonstrations of the latest software or hardware. Watch this space for details.

October

The third Buy, Swap and Sell day. Your last chance this year to get some money for Christmas presents or to get an early present for yourself.

November

The second all day tutorial session.

December

The Annual General Meeting followed by some festive eats and drinks. Make sure you attend and give your support to all the workers in the club. o

continued from page 14

```
1B33181B4B2100FF00000000FF818181FF00FF101010FF00FF00FF8
18181FF00000000000106F800
GR4
1B33181B4B2000FC06020201010000000000000000000000000000
3020608102020408000010000
GR5
1B33181B4B200000000000000080808040402040408070080830C00
0000000000000000000010000
GR6
0D0A000000000000000000000000000000000000000000000000000
0000000000000000000000000
```

continued on page 20