

8803
(17,42)

TISHULL



NEWS DIGEST

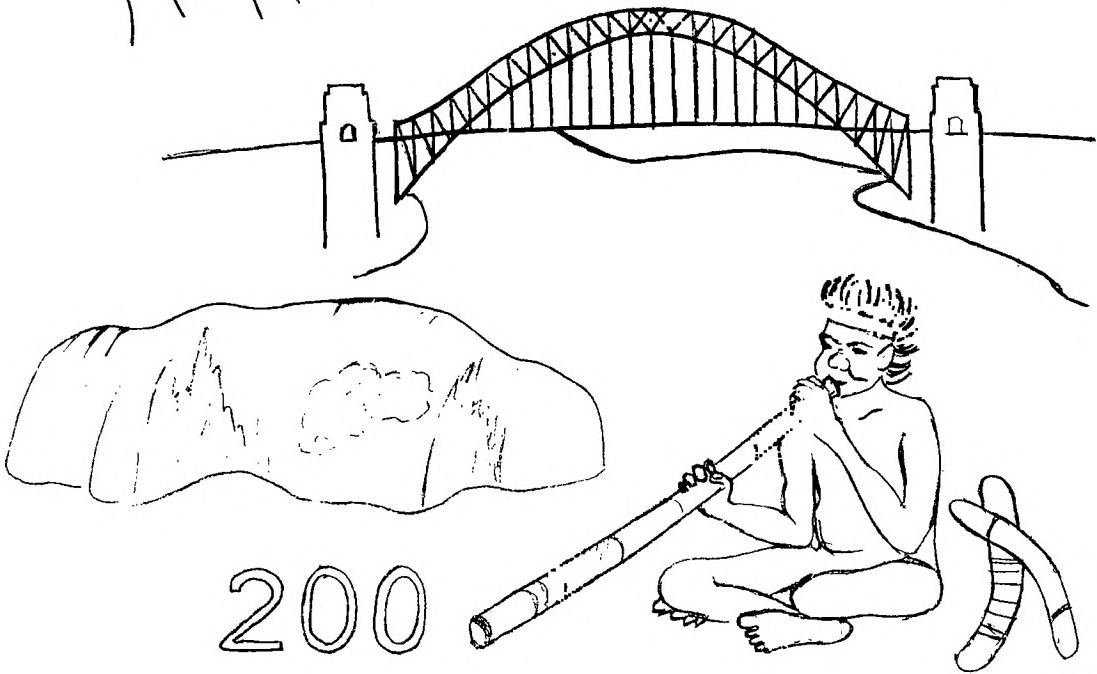
Focusing on the TI-99/4A Home Computer

Volume 7, Number 2

March 1988

Registered by Australia Post - Publication No. NBH5933

Happy Birthday Oz



200

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

\$ 2



TISHUG NEWS DIGEST

TiSHUG (Australia) Ltd

TiSHUG News Digest

TiSHUG News Digest ISSN 0819-1984

March 1988

All correspondence to:

PO Box 214
REDFERN NSW 2016
AUSTRALIA

DIRECTORS:

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

SUB-COMMITTEES:

Publication:
Geoff Trott ...Tel.(042) 29 6629
BBS sysop:
Ross Mudie ...Tel.(02) 456 2122
Merchandising:
Bob Bunbury ...Tel.(02) 601 8521
Printed Library:
Warren Welham .Tel.(043) 92 4000
Software Library:
Terry Phillips Tel.(02) 797 6313

REGIONAL GROUP CONTACTS:

Glebe:
Mike Slattery .Tel.(02) 692 0559
Central Coast:
Russell Welham Tel.(043) 92 4000
Liverpool:
Arto HeinoTel.(02) 603 8956
Illawarra:
Bob Montgomery Tel.(042) 84 2980
Carlingford:
Chris Buttner .Tel.(02) 871 7753
Sutherland:
Gary Wilson ...Tel.(02) 543 1314
Northern Suburbs:
Dennis Norman .Tel.(02) 452 3920
Coffs Harbour:
Keir WellsTel.(066) 55 1487

MEMBERSHIP AND SUBSCRIPTIONS:

Joining Fee\$ 8.00
Annual Family Dues\$25.00
Dues Overseas Airmail ..US\$30.00
 or UK£22.00
 or AUS\$50.00
Publications Library\$ 5.00
Texpac BBS\$ 5.00
BBS Membership:
Other TI User Group
 Members ..\$10.00
Public Access\$25.00

TiSHUG Sydney Meeting

The next meeting will be at 2pm on
5th March at Woodstock Community
Centre, Church Street, Burwood.
The following meeting will be on
9th April, a week late because of
Easter.



Title	Description	Author	Page #
AT multifunction card	Hardware review	Geoff Trott	6
Challenge	General interest		2
Co-ordinator's report	General interest	Chris Buttner	2
Communicators	BBS information	Ross Mudie	9
Fortran	General interest	D.N. Harris	8
From the bulletin board	Mail to all		9
Games information	General interest	Robert Brown and Stephen Judge	7
Genealogy	General interest	Dick Beery	2
Implant 4 assembly files in XB	Software hints	Ross Mudie	15
Multiplan application	The making of an index	Geoff Trott	21
Multiplan tips and techniques	Software hints	Steve Zimmerman	20
Program to type in	Snoopy Kong	Tristano	11
Program to type in	TE2 words	W.M. Johnson	13
Program to type in	Gemini printer options		13
Publications library report	Club library	Warren Welham	4
Putting it all together #4	Software hints	Jim Peterson	14
RAMcard tip	1Mbyte Horizon RAMdisk	Ross Mudie	23
Regional group news	General interest		16
Secretary's notebook	Club news	Terry Phillips	3
Speech for Milton Bradley games	Software hints	Ross Mudie	10
TiSHUG shop	For sale	Cyril Bohlsen	3
TiSHUG software column	Club software	Terry Phillips	4
Techo time	Real time clock	John Paine	5
They're off	General interest	Geoff Trott	1
Tutorial	Microsoft Multiplan	Tom Kennedy	17
XBASIC Assembly SAVE utility	Software hints	Todd Kaplan	10

They're off

by Geoff Trott

Well we have another change of Editor for our News Digest or TND for short. I am not sure what you, dear reader (is there really only one?), want to read about in your TND, so I will tell you what I plan to give you to read and will wait for you to get busy on your word processor to contact me and say that you agree or disagree with me. In other words, I am planning to show my personal preferences in the hope that it will make you pleased or angry enough to tell me your preferences. Other ways to try and get you to respond have not worked so let me try this one.

This column then will be a regular feature, and will contain my ramblings on various topics, including local events and news, and news from overseas. We get quite a few of the overseas magazines but there is always a problem finding the time and the people to type in interesting articles. I hope to give a brief resume of some interesting articles that have come to my notice, and perhaps one of you (more than one?) will want to find out more and will volunteer to type in the article.

I shall try to stay out of politics, religion and rumour, but this only leaves football and sex..... Sometimes I will not be able to stop myself from commenting, but it will only be to try and generate some correspondence on a subject of interest to all.

continued on page 22

Co-ordinator's Report

by Chris Buttner

The new Directors of the club elected at the ACM held at the Burwood RSL club on 6 February are: Chris Buttner (Chairman/Co-Ordinator) Terry Phillips (Secretary) Percy Harrison (Treasurer) Russell Welham, and Cyril Bohlsen.

The publication of the magazine will be handled by Geoff Trott while Bob Bunbury takes over as our shop-keeper. To the new Directors, sub-committee members and regional group co-ordinators I extend a very warm welcome and look forward to working with you throughout the coming year.

The handover from the old to the new directors should be completed in the next two weeks after which we should be off and running with a full head of steam.

Judges have been appointed to select the winning software for our competition. If you are an entrant, thank you for participating and helping to keep the club vibrant and alive.

Over the past few months we have had a number of new members join the club. With this in mind, I believe we are once again in the situation where it is worthwhile to run a new series of BASIC/Extended BASIC Tutorials for those with little or no programming experience. There are two broad options. The first is to keep these tutorials 'short and sweet' and the other to run them as part of our major tutorial day efforts. If you feel you need help and guidance in learning basic programming, please let us know so we can hopefully tailor these sessions to suit you best. Incidentally, if you are an old member and wish to refresh, you will also be most welcome. Such sessions can be as simple or complicated as you wish.

This year I would like to set our monthly activities well in advance so that those of you who are country members have a reasonable chance to attend if one of the topics takes your fancy. We are always open to suggestions, so make your voice heard by speaking to one of the directors at a meeting or writing to the Secretary.

At the ACM, John Paine previewed his latest project, a real time clock for the TI99/4A. This piece of hardware is very compact and will fit inside a speech synthesizer box with room to spare. For those of you who unfortunately were not able to see it, I can assure you it is very easy to use once installed. John has provided short programs which allow you to control and access the clock. I understand John has written an article about the project for those of you who want to reactivate your soldering irons.

Challenge

There is a challenge going on, 12th MARCH 88, at LAZERDOME. The challenge is that the members of SCI-FI BBS claim that they are better at the game than any other BBS about. They want to know if any other BBS has the guts to turn up and try to prove them wrong!! All you have to do, to try, is get to LAZERDOME on St Hillier's Rd at Auburn and be there by 2pm, in the carpark near my blue Honda Civic and you are in it. To play the game you need rubber soled shoes (not thongs - as you do a bit of running) and \$5 per game plus another \$6 (once only membership fee for year) if you have not played before. The game is quite good and involves being inside a MINI-CITY, with strobe lighting effects, sound effects, using light guns to shoot other players, to disable them and also to try and disable their Control tower. If you think you are a good shot and can beat SCI-FI BBS'ers, be there!!

Members of DOWN UNDER BBS are forming 2 teams to challenge all others, possibly members of TEXPAC might do the same.

I know some people say that they are not fit enough, but I am living proof that you can be fat and still enjoy the game.

Genealogy

by Dick Beery, Central Ohio 99ers

In earlier articles on this subject, I have indicated that the genealogy programs for the TI, at least the ones of which I am aware, failed to fill the basic needs of the genealogist or even the beginner family tracer.

Ken Barber's Pedigree Chart, while it does a good job of accepting worthwhile data and prints an excellent 4 generation chart, does not provide in any manner for the output of Family Group sheets, blank or otherwise. I very much like the fact that his program holds 127 files and chains easily from one file to the next.

The Cattin's Genealogy Workshop, retailing at between forty and fifty dollars, and currently being advertised makes wonderful Family Group Sheets, but the user needs to type the data in manually, and no provision is made for 4 generation or any other type of ancestor charts. I also find the instructions difficult to follow.

These two rise above the others I have commented on in earlier articles in quality, but still leave the user wanting more.

Recently I purchased Walt Davies' program GENE-III. I find it to be easy to work with and dependable. It gives the user a Family Group Sheet that can be run to the printer either blank or completely filled in. It also prints a 4-generation Ancestor Chart that lists the names of 15 individuals, together with their birth, marriage and death dates. The program accepts many fields of data: birth, christening, death, marriage, and burial dates and places, space for two spouses and fourteen children, together with the birth, death and marriage information for both sets, as well as the names of parents of both husband and wife. Brief notes, 28 in number, accompany each Family Group Sheet, and these add much flexibility.

One inclusion in the Cattin program, cited above, that I would like to see provided in future revisions of GENE-III, is the ability to sort, compare and print out persons having common birthdates and places, as this feature can greatly aid the researcher during visits to remotely located libraries and other data repositories.

The purchaser of GENE-III is advised to read carefully and follow closely the documentation provided with the program. Once the user has complied with this advice, he/she should find the program very user friendly.

The 4 generation chart provides the customary information: "Person No. 1 on this chart is the same as Person No. ___ on Chart ___."

Both charts provide the name of the compiler and the date the information was first entered. The 4 generation chart includes the compiler's address, so that others researching the same line can make contact.

While the charts may be printed out using tractor feed paper, the intent is for single sheets to be used. When the program has printed the Family Group Sheet, it pauses to allow the user to turn over the sheet, then a key press puts the notes on the back of the page. While I seldom use single sheets for other applications, I find this to be the most efficient way of handling data plus notes.

I can highly recommend this program as being the best in the field at this time, at least among the programs with which I am familiar. The price of ten dollars (originally fifteen) is very modest. The program can be obtained directly from the programmer, whose products are marketed under the name of GATOR SOFTWARE.

Write to : WALTER R. DAVIES,
17718 ORCHARD LANE,
SALINAS, CA, 93907.
U.S.A.

Please remember to include sufficient US funds to pay for the product and return postage.

Retyped by Ross Mudie for TEXPAC BBS and TISHUG News Digest, 10/2/88.

Secretary's Notebook

by Terry Phillips

First, I wish to thank all members who voted for me at the recent AGM. I shall do my best to ensure that all members TI needs are catered for during the forthcoming 12 months.

A BIG welcome to the following members who have joined us since my last report:

Charles Carrier - North Ryde
 Christine Baker - Two Mile Flat
 Stephen Shaw - Stockport, UK
 Joseph Khamis - Carlingford
 Ian Morrison - Baulkham Hills
 David Smith - Wagga Wagga
 Graham Hughes - Saratoga
 David Rivett - Niagara Park

Stephen Shaw, is our only current member from an overseas country. Hopefully he will be able to send us some information from time to time on what is happening in the UK TI scene.

Russell Welham recruited the last two mentioned new members up on the Central Coast. Keep up the good work Russell, and hope you recruit a few more through the year.

During the December/January period some 20 to 30 overseas news digests were received, and an eager band of volunteers is going through them extracting items of interest to appear both on the Bulletin Board and in the TND. Watch for these articles in future issues.

Here is an offer from John D Johnson of the San Diego TI UG. Any members who may be heading for sunny California are welcome to write to John in advance at PO Box 83821, San Diego, CA 92138. John assures me that he, and his fellow members would be very pleased to see what they can do to make you welcome.

John also advises that the Horizon RAM disk has been expanded to 1 Meg and that that item is becoming the standard TI RAM disk.

Also that Mike McCann in Nebraska is coming out with a PBox card with an on board Forth based Microprocessor. The going price however is a whopping \$400 (US). And, according to talk at the Chicago TI Faire, in the pipeline is a TI-IBM software interface that will transfer not only ASCII files but translate IBM Basic programs. Wait and see on this one.

It was good to see Ross Mudie rewarded at the AGM for his contribution to TISHUG over the past number of years. He certainly has been a prolific software author and I am sure a lot of us have learnt at least some rudiments of assembly from his tutorials. Congratulations Ross.

It was also good to see Jim Peterson rewarded for his efforts. Jim is a very well known TI man in the States and his series of Tips from the Tigercub have been produced in just about every new digest around. Not to forget the McGoverns, from Newcastle. There is no doubt, as many of us would attest, they have developed the finest piece of software around for this computer.

For Sale For Sale For

Original TI RS232 card.....\$140
 (with manual)
 TI Disk Control Card.....\$100
 Printer cable for PIO.....\$25
 RS232 cable.....\$20

Peter Schubert (02) 3585602

TISHUG Shop with Cyril

Welcome to another year with the SHOP and another MERCHANDISING OFFICER.

The name of our new shop keeper is BOB BUNBURY, and as outgoing shop keeper I would like to wish him a very successful year in his new position.

I would like to thank all the patient members who had to wait (without too many complaints) for their orders to be filled.

Please note that the SHOP is now run by BOB BUNBURY and his phone number is (02) 601-8521.

RAM DISK CARD PARTS :-

mostly sold out with the exception of the Battery set at.....\$ 14.00

THIRD RUN of RAM CARD P.C.Bs.

The response to a third run of RAM cards has not been very large up to now, (in fact only 2 cards remain on order at this time) so I think these will have to be filled by some people selling their second or third cards.

PRINTER BUFFER PARTS :-

- (a) P-BUFF PCB, EPROM, CRYSTAL, 8255 & Z80 chips.....\$ 56.00
- (b) 41256 memory chips (8 req'd.).....\$ 40.00
- (c) Computer sharer board & comp\$ 18.00
- (d) Printer sharer board & comp\$ 18.00
- (e) Plastic box (D/S 2508) small\$ 9.00
- (f) " " (D/S 2505) large\$ 9.50
- (g) 9 volt transformer.....\$ 6.50

NOW THE STANDARD ITEMS:-

- (a) New DS/DD 5 1/4" Disks (box).....\$ 12.00
- (b) T.I. Joystick handles.....\$.50
- (c) Peter Schubert's mini-expansion units
 DS/DD Disk controller card.....\$190.00
 Mini-PE mother board (with one of either):-
 32K mem : PIO : RS232 port)\$ 85.00
 Extra options on mother board
 32K memory.....\$ 50.00
 RS232 port.....\$ 50.00
 Finished painted box for Mini PE\$ 35.00

JUST RECEIVED FROM PETER SCHUBERT :-

HIS NEW MULTI-FUNCTION CARD FOR THE PE-BOX.

- (a) Complete card ie. DS/DD disk controller, RS232, PIO, 32k memory.....\$390.00
- (b) DS/DD disk controller card.....\$200.00

SECOND HAND ITEMS :-

- (a) Grom Ports.....\$ 12.00
- (b) Ivory Console Cases.....\$ 2.00

BOOKS :-

- (a) Back issues of SND.....\$ 1.00
- (b) Technical manual.....\$ 15.00
- (c) TI-writer manual.....\$ 15.00
- (d) Editor Assembler manual.....\$ 28.00
- (e) TI LOGO Curriculum guide....\$ 10.00
- (f) Micropendiums.....\$ 3.00
 1986-June to Dec./1987-Jan.to Dec.
 1988-January.

SOFTWARE:-

- (a) Club Software Tapes.....\$ 3.00
- (b) Club Software Disks.....\$ 5.00
- (c) Picasso Publisher V2.0
 (Arto Heino's).....\$ 20.00

NOTE :- Bring or send your original copy of Picasso to the shop for an updated copy.

- (d) VIATEL-1 (for serial port 1).....\$25.00
- (e) VIATEL-2 (for serial port 2).....\$25.00
- (f) CART WRITER.....\$25.00 (similar to Console-Writer)
- (g) DIAGNOSTICS.....\$25.00 (will test your TI gear)

NOTE:- NONE OF THESE MODULES WILL OPERATE ON THE 1983 VERSION CONSOLE WITHOUT A GROMBUSTER.

NOTE:-

The price of multiple disk software has been reduced :- (ie for single density software that won't fit on one disk).

- 2 disk set....\$ 8.00
- 3 disk set....\$10.00

When ordering disk based software could you please state what type of disk drives you have?
 (ie. SS/SD, DS/SD or DS/DD).

TISHUG Software Column by Terry Phillips

The competition has closed and the judges are now considering their verdicts. The judges, if you were not at the AGM are:
Ross Mudie, Arto Heino, Shane Ferrett, Robert Peverill and Gerry Levy.

Entries were received from:

Barry Gibbins - a number of games in Extended Basic. Titles are Cathay, Opal Miner, Poker, Tilo, Triange and Waitabit.
Alan Swales - a business program titled Account, Extended Basic.
Marcel Zaia - a disk loading utility in Extended Basic, titled Multi-Loader.
Tony Smith - a number of games that run in either Basic. Titles, Matchmaker - with a separate program to animate graphics - Snake and Stanley.
Hal Payne - Directory - a disk directory cataloguing utility in Extended Basic.
Craig Sheehan - Extended Display Package. A host of new programming commands in assembly.
Dennis Remmer - various utilities, games and a tape word processor. Most in Extended Basic.
Tony Imbruglia - Procalc a business utility.
Tony Imbruglia - The Diskette Caretaker. A utility to print disk sleeves with comments.
George Meldrum - an assembly utility titled Merge. Primarily designed for cassette users but with several options unavailable through XB's merge function.
Bill Longmuir - a Family Tree package together with 2 educational programs, Sentence Maker and Junior Maths.

That is only a very brief description of the entries received and does not do any of them justice. I do not envy the judges task in having to find a winner. Win or lose however, all members will be the winners because of the time and effort taken by these authors to submit entries. We certainly now have a good base of software to share both locally and overseas. My sincere thanks to all authors who have submitted entries.

NEW DISKS ADDED TO LIBRARY:

The above mentioned software has not yet been catalogued and added to the library, and that won't happen until judging is completed. In the meantime the following 3 new disks have been added:

DISK A168 - 598 Sectors - contains CENTIPEDE (57) DISKCOPY (15) DISKDOCTOR (26) EASY DESIGN (48) DISK HACKER (15) CHRISTMAS CARD (75) MASSCOPY (130) NETWORK (21) NIBBLER (32) QUICK COPIER (30) TURBO COPY (27) BANNER (69) BOWLING (46) PERPLEX (62)
Most programs on this disk require 32K together with Extended Basic, although some run from Option 5 of Editor Assembler. Christmas Card (XB/32K) is a neat salute to Christmas from Woodstock, Snoopy's offsider. Graphics are superb. Perplex is a very complex game that if anyone ever learns to play would surprise me. Banner prints out banners for you and has some graphic characters available - very slow though. Bowling appears to be directed to 10 pin bowlers - record keeper of some kind. The rest are mainly copying utilities except Network which keeps track of your current family fortune.

DISK A169 - Another disk full - 537 Sectors - of utilities and routines including a Banner program, calendars, gothic print, thank you notes and a host of others. Also includes a number of files associated with the Printstrip program that appeared in a issue of Micropendium a couple of months ago.

DISK A170 - Clock Routines from John Paine. This software will support the Clock Board details of building will appear in the TND. 95 Sectors.

Available at the shop at the next meeting will be the disks described above, A168 and A169, in either double sided format or as "flippies" for those with single sided disk systems.

As for a tape I will try to put what I can from the above disks onto it.

I have been thinking of a new software distribution means for some time now, and I intend discussing the following at the next Directors meeting. Basically I see the system operating like this - each member be given a catalog of available software which is held in the library. From this listing members be free to choose what they want with their orders being placed at one meeting for collection at the next or by mail. Costs would remain the same as under the present structure. There could still be say, a disk or tape of the month available from the shop, depending on software received during any particular month. New additions to the library would be notified in this column so a members catalog would always be up to date. I note that a number of groups in the States and Canada distribute software in this manner, and, from what I understand, the system works quite well. If any member has any thoughts on this method of software distribution, I would be happy to hear from them.

Publications Library Report

with Warren Welham

Welcome to the first regular report by me of the happenings in the library. Each month I will be stating new arrivals to the Publications library and anything happening with it so this year you will know a lot more about it. Firstly the list that has been in preparation is finished and will be available at the March meeting from me at the library.

New Tnd's (Snd) arrived this month	
May 1984	Oct 1984
June 1984	Dec 1984
July 1984	Oct 1985
Sep 1984	Jan/Feb 1988 (4)

New Books arrived this month		
...	Title	Author
	Memorandum and Articles of Association of TISHUG	TISHUG

New Arrivals of Overseas Publications		
Group	Publications Name	Date
Edmonton 99'er users society	99'er Online	Nov 87
Pittsburgh user's group	The Pug Peripheral	Nov 87
Northern New Jersey	-	Nov 87
Tacoma 99ers users group	-	Nov 87
Brisbane user group	Bug Bytes	Nov 87
Hunter Valley 99ers user group	Hunter Valley News	Nov 87
Central Ohio Ninety Niners inc.	Spirit of 99	Nov 87
Club Informatique Montreal	CIM 99	Dec 87
The Ottawa TI-99/4A users group	Newsletter	Nov 87
Lehigh 99'er Computer group	-	Sep 87
Lehigh 99'er Computer group	-	Oct 87
Channel 99 Hamilton	Clubline	Jun 87
Channel 99 Hamilton	Clubline	Jul 87
Channel 99 Hamilton	Clubline	Aug 87
Channel 99 Hamilton	Clubline	Sep 87
Channel 99 Hamilton	Clubline	Oct 87

Techo Time with John Paine

A Real Time Clock for the TI99/4A

I have recently been spoiled by using the John Johnson Menu program in conjunction with a RAMdisk installed in my PE box. The menu program does so much that it is hard to believe that with extra hardware in the system, it could do more.

The software includes the necessary routines to access the three known Real Time Clock peripherals that have been on the market for some time. The purpose of this note is to describe a fourth clock that is compatible with the menu program as well as being a stand alone device that can be used by members who may not have a RAMdisk.

Before starting, let us look at the existing clocks available for the TI.

1) CorComp Triple Tech Card

This is a PE box card that uses an OKI clock chip and can be easily read, altered and used under BASIC. Its major disadvantage is that it is controlled by a DSR and is mapped into memory location >4000. The OKI chip is relatively small in size but difficult to use because of the need for external latches and support chips. CorComp also released a stand alone unit that plugs into the side of the console.

2) MBP Clock Card

Once again, this is a PE box card, but uses a National clock chip. The MBP card is addressed to operate in the memory mapped block starting at address 8640. The National chip is much larger in size, but requires less support in terms of extra chips. Once again simple BASIC programs can be used to set and read the clock. Because of the memory address location of the clock chip, it is necessary to incorporate a CALL SOUND statement in any BASIC program immediately following the CALL LOAD statements used to 'poke' the values into the clock registers. The CALL SOUND statement will turn off the sound generator, otherwise the noise will drive you crazy. Like the Triple Tech Card mentioned above, the MBP card has a battery back up.

3) John Clulow Clock

I have only seen some sketchy documentation on the Clulow clock, but it seems to use the same chip as the MBP card and is mapped in almost the same area as the MBP clock. I will assume that the same comments will apply as with the MBP clock.

As stated earlier, all these clocks are built around PE box cards, so with this in mind, the following circuit and ideas are based on a stand alone system, which can be built into an existing external peripheral, such as speech synthesizer, thermal printer, stand alone RS232, stand alone disk controller, or even the console itself. The minimum requirement for operation is possession of Extended BASIC to set the time. Time can be read with BASIC and E/A or Minimem, but you will have to write your own software. I will provide a disk to any user group with some utilities and set clock programmes written in Extended BASIC.

Now down to business. The clock that I am about to describe is based on the same National clock chip used by MBP, and will be mapped into the same memory address. The prototype clock was built on a small Tandy prototype card, which cost \$1.67 and the total project including clock chip, crystal, resistors, diodes, dry cell batteries (in this case), and chips cost less than \$30.00.

I built the total package, including batteries, into the top compartment of the Speech Synthesizer, which allowed for complete stand alone performance, and offered the added advantage of portability to show other members etc.

It is not my intention to describe a wire by wire method of assembly, because of the many possible locations for mounting. Instead refer to the circuit diagram below for connection of the clock to the major bus lines. You will also note that the circuit shows different options for battery back up. The prototype initially used dry cells, but was upgraded to nickel cadmium cells, which can be recharged when power is supplied to the host peripheral. Although a lithium battery option is shown, I consider the cost to be too high for such a project.

The parts needed are:
 A suitable board for mounting e.g. Tandy Proto board
 One only MM58167 clock chip from NSD Melbourne
 One 32,768KHz crystal from NSD
 One 30pf variable capacitor Geoff Woods
 Note: The crystal and trimmer can be salvaged from an old LCD digital watch.
 Two only 74LS138 decoders
 One 74LS04 hex inverter
 Two or three 1N914/1N4148 diodes
 One 22pf capacitor
 One 1K resistor
 One 220K resistor
 One 100 ohm resistor (if using Ni-Cads)
 Batteries of your choice. I used half length AA cells from Tandy, although I later used a PC board mount Ni-Cad pack from Radio Spares (\$9.00 approx.).
 A heap of ribbon cable to connect the bus to the clock. There are 29 wires to be soldered to the expansion bus. This includes the 5 volt power and ground.

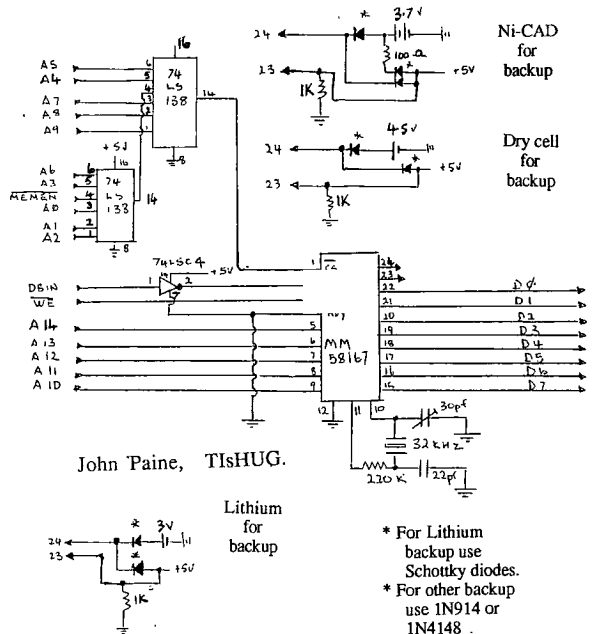
The actual wiring on the board is simple and is easily accommodated with point to point wiring. The external connections to the bus are the most tiresome.

To calibrate the clock, National Semiconductor recommends that a digital frequency meter be connected to pin 19 of the clock chip, and the trimmer be adjusted until a frequency of 500Hz is achieved. I just powered up and set the clock. After 3 weeks, the clock lost 5 seconds compared to my watch. I can live with that as it is quicker to load the SETCLOCK routine than muck around with calibration.

Load the program JUSTCLOCK from the utility disk (XB). Immediately power is connected, the clock should start and count from 00:00:00. If all is OK, then the seconds will tick upwards. Obviously after 59 seconds, the minutes will increment. If the seconds do not change, you will have to check the crystal, 220K resistor and capacitors connected to pins 10 and 11 of the clock chip. If nothing displays, then check the address and data lines.

The disk that I mentioned above will have a number of utilities and sample programmes that can be used to set the clock and also to read the clock. I have also included a couple of cataloguing programmes which also make use of the clock and will date and time stamp the printed listing.

Those of you with a RAMdisk and Menu 7.1 will see immediately the benefit of the clock, but those without should not despair as the clock is very usable. In time I will release more utilities for the clock and perhaps some of the brighter XB programmers can come up with other applications.



Real Time Clock circuit diagram

Hardware Review

A review of the AT Multifunction card

by Geoff Trott
Illawarra Regional Group, TISHUG

I have a single sided drive in my PE box and a double sided drive outside the box. With the use of menus, it is a decided advantage to have more than 360 sectors available on DSK1, and yet my normal mode of working is to have my data disk in DSK2 and the working disk with TI-Writer or Multiplan in DSK1. As the space for data gets used up quickly too, it seemed to me that the best thing to do was to get a double density controller and double the capabilities of both my drives. At this time Peter Schubert announced the PE box version of his AT disk controller, so I thought that this is just what I want. In the event I only bought the disk controller part, so my board has a lot of spare space for 32K memory, PIO and 2 RS232 interfaces. I could add these at a later date, if Peter would upgrade my ROM for me.

A visual inspection shows a professionally made board, with plated through holes and gold plated contacts. The board is also coated with a green cover for protection. There were only about 20 ICs including bus buffers, disk controller and a 32K byte EPROM. The board has no cover, so the 5 volt regulator has a large heat sink, and the far edge of the board is rather crowded to fit in the PE box and still provide for the RS232 connector, a PIO connector, and two connectors for the disk drives. All these connectors are standard TI types, so it is a little crowded and leads to the only problems with the card that I have found. This is that it does not really fit into the PE box!

The card is a bit higher than the normal cards and mine stopped the lid of my PE box closing properly until I filed a bit off the top of the board. The plug for the internal drive is too high for the card to go into the end slot, but that is not a major problem, however if the external drive connector is used then there is no way to put the lid on the PE box, as the lid sits on top of the connector. I usually have the external drive on top of my PE box which is not too heavy, but I would not like to have my TV on top of the PE box, as I have seen many do. The edge connector needs to be about one centimeter longer to avoid this interference.

Enough of this problem! With the card comes a disk with some excellent and useful software. The most useful features are the two menu programs which are easily bootable from any environment. One is obtained using Extended BASIC via the normal LOAD file, and gives 3 menus, of which the first page is the most useful, and the other ones can be customised. The second menu program is obtained from BASIC with a CALL RUN (or call run) command in immediate mode. From either menu one can load DM1000, Disk Utilities, Memory editor, TI-Writer, Formatter for TI-Writer, Mass Transfer Terminal Emulator and PRBase, a data base program. Only from the Extended BASIC menu program can the programs Clone, Create data base and Data base utilities be accessed.

All these programs seem to me to be excellent, with Disk utilities providing editing of data on disk like Disk Fixer does. The programs and documentation for them come on a double sided, double density disk, with the document files at the end of the disk. So the program files are all on the first side (except for CLONE, which has a few sectors on the second side).

As I have said, most of the programs are on the first side of the disk, which means that they should be loadable using my single sided DSK1. Of course the loaders expect all programs to be on DSK1 as well. I found some interesting things happening until I created a single sided disk. I found that some programs would load, but not run.

Perhaps I should explain that a bit further. Some of the programs try and load a file called CHARA1, or CHAR (for PRBase), which contains their character set. CHARA1 contains the latest lower case character set, with larger characters than the original TI-Writer

lower case character set. CHAR has this plus some special characters used by PRBase. For example, when Disk Utilities is loaded and runs, it then tries to load the CHARA1 file. With the double sided disk in the single sided drive, the program cannot load without error, and it return to the title screen. If Disk Utilities is loaded using the Editor Assembler module, the load terminates with an I/O error 7. Why this happens I do not know, but it would be interesting to find out.

To create a disk which ran correctly, I cloned the disk (using the Clone program), from my single sided drive to my double sided drive. This stopped with an error at the end of the first side, but left a disk with most of the programs on it. Then I used disk utilities to construct the rest of the directory information by copying sectors from the original disk to the clone (using DSK2 and swapping disks). Then DM1000 was used to delete all the document files and CLONE, which was partially on the second side. Finally, Disk Utilities was used to change the information on sector 0 to make the clone a single sided disk, and CLONE was copied back to the disk.

All this was necessary because there is a file on the disk which must be in the correct sectors to work. This is the Corcomp disk manager, which uses sectors >30 to >90, as it is loaded by a special feature of the disk controller. When the system resets, if the space bar is pressed, then this disk manager will be loaded. It can also be loaded with a CALL MGR in BASIC. One way of doing this without getting into conflict with the power-up of disk drives on power on, is to press space while also pressing FCTN(=), or pressing space quickly after a QUIT.

Using the Corcomp disk manager, when initialising a disk, the manager can be installed for you on the new disk, which is probably an easier way to do what I did, than by the method given above.

The two ways of loading programs give the following menus

Title	Ex BASIC#	BASIC#
Disk manager 1000	1	1
Disk Utilities	2	2
Memory Editor	3	3
TI-Writer	4	4
Formatter	5	5
Modem	6	6
Data Base	7	7
Create data base	8	
Clone	1 - second menu	
Data base utilities	2 - second menu	

From the Extended BASIC loader DM1000 loads with small capitals, as does memory editor and Clone. This is because these programs do not load CHARA1 when they start and so use the Extended BASIC character set. Disk utilities, TI-Writer, Modem, and the data base programs all load with lower case character set. From the BASIC loader, all load with lower case characters, except that Clone and Create data base cannot be loaded from BASIC. This follows because the menu program UTIL1 uses the lower case characters from CHARA1.

With the original double sided disk in my single sided drive the following happens. With the Extended BASIC loader, DM1000 loads with small capitals, TI-Writer, and the data base programs load as normal. Disk utilities, Modem and data base utilities return to the title screen. Clone produces a SYNTAX ERROR as part of it is not loadable. Memory editor hangs the computer with a blue screen. With the BASIC loader, DM1000, TI-Writer and Data base load as normal with lower case characters and Memory editor stops at the blue screen. Disk utilities, Formatter and Modem return to the menu, not the title screen.

Another difference between the two menu program loaders occurs when TI-Writer is selected. From the BASIC menu program, EDIT1 and EDIT2 are loaded directly and start running. The Extended BASIC menu program stops at another menu (from program WORD which loads the character set from CHARA1), which is the same as the TI-Writer module menu, namely a choice of Editor, Formatter and Utility. This menu already has the lower case letters and uses white letters on a light green background. Of course the utilities option is of no use, unless the disk is changed, as UTIL1 is already used for the BASIC menu program, and this UTIL1 does not load properly this way.

continued on page 8

Games Information

by Robert Brown and Stephen Judge

This is the first article on GAMES INFO to help people to solve games, adventures and any program that you dont understand or is giving you the hassles.

Just say that you were stuck on Return to Pirates Isle and you could not get out of bed. You could dial up TEXPAC and send some mail to username 'GAMES', and within a week you could be on your way to finishing it. Whatever your question may be, we can either solve it, or guide you in the right direction.

Now we had better get started on this month's file. This month we will be talking about test modes in games. You may not know this but quite a few of the old TI modules have the hidden test modes. This allows users to "CHEAT" in several ways, for example, change level or lives or speed depending on the capabilities of the game.

The modules we know that do this are...

Alpiner, Chisolm Trail, TI Invaders (disk tape version only), MunchMan, Munch Mobile, Moon-Mine, and TI Runner.

"How do we do this sir?", I hear you ask. Well my son, it is simple. All the above games except TI Runner and Chisolm Trail you can type '***' which is 'SHIFT-8-3-8' at the title screen. With Chisolm Trail you must wait until it gives you the prompt LEVEL (1-9) then quick as a flash type *** and you are in. With TI Runner you can skip up a level by typing FCTN-5 before you have started your present level.

If you find any more of these little surprises then tell us so we can tell everybody else.

Here we go with the solution to Hitch Hikers Guide to the Galaxy, Part 1

Here you are, mild mannered Arthur Dent, about to start the worst day of your life, although you do not know that...yet! Actually, the day is already getting off to a bad start, since you have just woken up in the dark, with a really bad headache (and it is all downhill from here).

The first thing you need to do is stand up and turn on the light. That is a little better, anyway! Or maybe not, since you are having a hard time getting coordinated. Grab the dressing gown and put it on, then look in the pocket. Ah, an analgesic! Take that, then get the screwdriver and the toothbrush, and head South to the porch (did you hear a tree fall? Rather ominous, is it not?).

Here you find something no modern home should be without: junk mail. Take the mail, and go on outside. Uh-oh! There is a very big bulldozer on its way to level your home, and there is Prosser standing by, watching it all. Are you going to take this lying down?

You bet you are! That is the only way to stop it: lie down in front of the bulldozer. No matter how close the thing gets, do not panic; it will not run you over (of course, in a short time, it really will not matter what happens to the house, but you do not know that yet). Just wait awhile until Ford Prefect shows up (read the junk mail while you wait).

Ford seems a trifle preoccupied with the sky, but he is aware enough of you to try and give you back your towel. Do not take it, or he will leave and you will be a lot worse off than you ever imagined (can things be worse than this? They sure can!).

Instead of taking the towel, ask Ford about your home. He will eventually come to his senses, and realize what is going on. When that happens, he will be able to persuade Prosser to take your place in front of the bulldozer, while the two of you head off to the pub to hoist a few.

As soon as Prosser takes your place, go South and West to the pub. Buy a cheese (?) sandwich while you wait for him to arrive (when you read the description, you will understand about the "?"). When Ford gets there, he will buy you a few beers. Drink only three of them.

Around about the time you have finished the third one, there will be a loud crash. In fact, it is the

sound of your home being demolished by the bulldozer (that will teach you to trust anyone who wears a digital watch!). Do not take that sitting down, leave the pub and return to where your house used to be. Along the way, you will see a starving dog.

While you may wonder if anything could eat that sandwich and survive, give it to the dog, who will (amazingly!) enjoy it immensely, ignoring a microscopic space fleet that whizzes past (remember that fleet). Then continue on to the ruins of your home (Ford will be right behind you).

And just about now, to put a perfect ending to a perfect day (which has just barely begun), the Vagon construction ships appear, to demolish the Earth to make way for a new Hyper-space Bypass (hmmm, maybe Ford was not kidding when he said he was from another planet, or that Earth would be destroyed in a short time).

Still, do not panic...wait until Ford drops the Sub-etha signalling device. There will not be much time after that, so pick up the device, push the green button (if you dropped the Aunt's thing, have no fear: it will turn up again later), and you will be in...the dark.

Get used to that, you will be spending a lot of time there before this adventure is over. Notice that, at first, you cannot do much. All your five senses seem to be out of order. However, if you wait, and read the descriptions very carefully, you will see that eventually, it mentions only 4 of your senses. The one that is missing is the one you can use. Keep this in mind, it will come in handy later.

Right now, your nose seems to be working again, so smell. Sniff, sniff. Ugh! Whatever it is, it sure is strong! You are also now dimly aware of a shadow, so look at it. Well, well, it turns out to be Ford Prefect! And, looking around, you find yourself in the hold of a Vagon ship. Certainly better than being on Earth (or where Earth used to be).

There is a glass case with an Atomic Vector Plotter inside, but do not bother with it yet. You have something else to do first, namely, obtain a Babel Fish. That should not be hard, right? All you need to do is push the button on the dispensing machine, and you will have one, right? Hehehehehe!

Those Babel Fish are pretty slippery characters (but, you may have found that out already for yourself!). And the cleaning robots are certainly no help, they seem to have only one mission in life; grabbing your Fish away from you. Well, we really cannot let that happen!

So, first thing to do is remove your gown and hang it on the hook. Now, wait for Ford to curl up, then get the towel and the satchel. Put the towel over the drain, and the satchel in front of the robot panel. Now comes the part that drives most people crazy: they do not know how to stop the upper-half of the room cleaning robot. But, it is so simple; just put the junk mail on top of the satchel.

Now you can push the button! Then step back and watch the Rube Goldberg shenanigans, which end with the Babel Fish stuck solidly in your ear (squish!). Bet you never thought outer space would be like this! However, now that you have the fish, you will be able to understand anyone who talks to you.

By the way, somewhere along the line, you will get a message that one of the phrases you have used was instrumental in starting a war that wiped out most of a small galaxy. There is nothing you can do about this; no matter how you try, it will come to pass. Rather unfortunate, is it not? Even more unfortunate, sooner or later, the survivors will figure out how that happened, and they will be looking for revenge...but more about that charming prospect later.

Right now, press the switch on the case. This will tell you what the code word is that will open the case so you can snatch the plotter. Make careful note of what word is required; it is chosen randomly each time. Too bad you have to listen to some pretty rotten poetry to get the word.

Speaking of poetry, in a short while, you and Ford will be hustled into the Captain's quarters, and strapped into Poetry Appreciation Chairs (worse things could happen, but right now, you probably cannot think of any).

continued from page 6

After the Vogon Captain has tortured you with the first verse, grit your teeth and enjoy the poetry. He will then, to your dismay, read you the next verse.

While you could easily live without hearing it, in this case, you do need to listen so you know the word to type in. Fortunately, after the second verse, you do not have to enjoy the poetry. Unfortunately, since you survived both verses, the Captain is going to have you and Ford shoved out the airlock (you have now found something worse than appreciating Vogon poetry).

While Ford tries to talk the guard out of spacing the two of you, type in the word from the poem. You must put quotes around the word, or it will not go through. Then get the plotter when the case opens. Now just wait awhile, and you and Ford will soon be in the airlock, with very little time left.

In fact, time has just run out, and there you are in the depths of space. Lucky for you, the Guide explained how to survive all of 30 seconds out there! Well, perhaps not so lucky, since, considering the vastness of space, it is quite improbable that another ship will come by to pick you up before the 30 seconds run out. So naturally, 29 seconds later, the Heart of Gold (the HOG) comes past and picks you up.

Well that is about it for this month. If you have any questions on the above file please direct them to username GAMES on the BBS.

Thank You

C HOCKY AND T ICTACMAN

FORTRAN by D. N. Harris

I have written a bit about C statements in FORTRAN, and if there is some reader interest, the next thing would be to write an article on DO loops, which are identical to FOR NEXT loops in BASIC so far as their function is concerned. FORTRAN is a moderately hard language to learn and one tries to show people things they can do, rather than throw a lot of syntactic rules at them straight away, in case one hears the sad refrain "it would be an excellent thing to do IF ONLY I HAD THE TIME".

For that reason I began with the C statement. There are no rules except that the first character must be a C, and no line numbers. That is, the first column, the left hand edge, before a margin or anything else, must be a C. Getting a bit further in:

```
.....
|12345
space 1 is for C when used,
space 234 and if need be 1 and 5 as well can be for
line numbers.
Space 6 is for Continuation if you want to make a line
longer than 80 columns.
```

In most FORTRANs that leaves 74 columns to fill with commands, and that can be plenty if one nests commands such as DO loops. The only statements which need a line number are FORMAT statements, statements which you want to GO TO and the ending of DO-loops. If I write a subset of FORTRAN it will be 32 columns, with 26 to fill with commands. The emulation will have lots of string processing to go through, but will be a grammatically correct FORTRAN programming exercise, and will enable the user to save short programmes as files, so that the first one will load the compiler from tape or disk, then read the FORTRAN programmes into it. Then of course, run, edit, or save them. Emulators, or user written compilers in BASIC which have the same function as FORTRAN, COBOL, or PASCAL seem to be avoided by Texas Instruments users, and it is high time we had a bash. Fully-featured compilers written in ASSEMBLER and transferred to cartridge would give us an export that could be sold to overseas user groups at \$250 a pop!

One other thing to watch, is to always make the name of any copy of this disk, whether single sided or double sided, the same as the original disk, if you want the special characters for PRBase to be loaded. This is because the program looks for a file called DSK.ATDC/M.CHAR which must be on a disk named ATDC/M on any drive.

I do not want to review the actual programs at this time, but as you can see I have made good use of most of them already, and the selection of programs is very good. I can see that this disk is going to spend a lot of time in my drive. I like the approach used for the menus and apart from the small problems I have mentioned, I would rate the controller and the software package which comes with it as excellent. I hope that some of the problems can be tidied up so that it will be even better.

continued from page 21

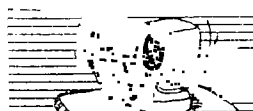
This is done twice to get the first half of the titles from both of the other files. Then the data can be sorted again, but this time all columns are sorted in reverse order to that which they are going to be printed. For example, for the title sort, first the page number is sorted, then the author, followed by the description and finally the title.

Printing from Multiplan has one problem. It will only print 80 characters on each line, so all the data must be arranged to fit in there. Another constraint is that it is only possible to change the width of a column to a maximum of 32 characters, which is the maximum that can be displayed on the screen. This can be overcome with continuous format. I decided to use the following column widths; 32 columns for the title, 23 for the description, 17 for the author and 8 for the page number. These are set using the Format command, by entering F followed by W with the cursor on the column of interest.

To print, press P to get the print menu and then press O to set the options of the area to be printed. Then go to the margins setup and set the top and left margins to 0, and the print width to 80. The print length and page length are set to give the output all on one 'page'. Then print file is chosen and a file name entered as the whole index will be merged and tidied up in TI-Writer. The file is produced after a wait and then the second half of the index is processed similarly.

The print files are about 54 sectors long and are both loaded into TI-Writer editor and some editing done, such as titles at page breaks, and removing the 'V7.' from all the page numbers. This all is reasonably simple and the final files are 117 sectors long. They are printed directly from the editor using 10 characters to the inch to spread the 80 characters out further. With the author index, the contents of the fifth column are moved into the first column where necessary, and multiple authors are given their own entries.

Checking the data for errors is aided by the sorting process. First the files are sorted into page number order, and the entries for each month checked with the actual magazine to locate any missed entries. Then the other columns when sorted allow the consistency of the use of capitals and spelling to be checked, and that names are consistent. After all the effort I was quite pleased with the result and so I used the same method to generate the index for the February issue as well which then will become the start of the next year's index. I started to enter this same one issue index into PRBase, a data base program which came with my new AT disk controller card, and found that although quick and relatively easy to use, it had some annoying quirks too. At this stage I am still convinced that Multiplan is as good as anything for this job, but PRBase may do it just as well if I was as familiar with it as I am with Multiplan.



Speech Problem in disk based Milton Bradley games

by Ross Mudie

A problem has been observed in several Milton Bradley games when they are loaded from disk into the CPU RAM. The problem is that often the program fails to recognise that the Speech Synthesizer is attached to the computer and as a result there is no speech to accompany the game. The problem is quite intermittent and has been observed in the games Big Foot, Meteor Belt, Sewer Mania and Super Fly.

All the programs use the same program flow in the area of testing for the Speech Synthesizer. The method is early in the program to write a byte of >10 to SPCHWT at >9400 then BL to a delay with a DATA of >0001 to set the length of the delay. The byte at SPCHRD >9000 is then MOVBed to >8360 and without any further delay the byte at >8360 is compared with a value of >AA to see if the speech synthesizer responded to the >10 at >9400. The problem is that according to the timing considerations for the Speech Synthesizer on pages 349 to 350 of the Editor Assembler manual, a 12 microsecond delay is required after a Speech Read, during which time the 8 bit bus can NOT be used. A sample of the disassembled code from Meteor Belt follows. All the programs seen so far use the same basic construction.

```

MOVb @>A825,@>9400 The byte at A825=>10
BL @>A9A2 Delay Sub routine
DATA >0001 No. of loops in delay subr
MOVb @>9000,@>8360 >AA if Speech is connected
** Insufficient delay here!

CB @>8360,@>A83A The byte at A83A=>AA
JNE $+>0008

```

The method employed to overcome the problem when implanting these programs in Extended BASIC, is to do the test in Extended BASIC and then pass the value into a safe memory location. The assembly code is modified at the time that the implanting is done using DEBUG. The code variation is shown below.

```
MOVb @>9000,@>8360
```

becomes

```
MOVb @>27FE,@>8360
```

and the following line is added to the Extended BASIC header to perform the test...

```
115 CALL PEEK(-28672,A):: IF A=96 THEN CALL
LOAD(10238,170)ELSE CALL LOAD (10238,0)! Speech Patch
>AOA2 was >9000, now >27FE (R.M. 11/1/88)
```

The value of hex 27FE, ie >27FE, converts to 10238 in decimal as required by Extended BASIC. I chose the location originally because the last part of the code loaded from >2800 up in 4 part programs. Any SAFE location may of course be used and >27FE may not always be safe with other programs.

If it is desired to fix the problem the proper way, then it would be necessary to replace the inadequate code with a B or a BL to a spare area of memory, probably in a part which is less than the maximum file size, and provide the SPCHRD with the appropriate delay following on the 16 bit bus BEFORE attempting to use the 8 bit bus again.

I love it when I think that I actually understand something in the Editor Assembler manual!!!

Please let me know if this information is actually found to be useful by someone else.



"I'm afraid you've made a mistake ladies, the bingo hall is next door."

Extended BASIC Assembly SAVE Utility

by Todd Kaplan, U.S.A.

* Memory map:
*
* RECALL is run by loading (ISR hook) >83C4 with >E400
* CALL INIT :: CALL LOAD(-31804,223,0)
*
* 83C4 - Address of Interrupt Service Routine hook
* DEEO - New bottom of XBASIC program
* DFOO - RECALL program + 8K buffer
*
* FFE6 - Old First Free Address in high memory
* FFE7 - Old bottom of XBASIC storage
*

```

DEF SAVE give program start of save program
BOTTOM EQU >DEEO bottom of XBASIC
AORG >DFOO get start of program space
* Start of "RECALL" program and top of program space
RECALL
LIMI 0 disable interrupts
CLR @>83C4 clear ISR hook
MOV R11,@SAVRTN save return address
LWPI MYREG get workspace
LI R0,BUFFER get buffer for program
LI R1,>2000 get top of memory to load
MLOOPS LI R2,>1000 get 4K words to move (8K bytes)
MLOOP1
MOV *R0+,*R1+ move a word of data
DEC R2 decrement counter
JNE MLOOP1 loop until done

```

```

* Clear ISR hook (Interrupt Service Routine)
* Return To Extended BASIC
SB @>837C,@>837C clear GPL status byte
LWPI >83E0 get GPL workspace
MOV @SAVRTN,R11 restore return address
RT return to calling program
* Start of "SAVE" program SAVE
LIMI 0 clear interrupts
MOV R11,@SAVRTN save return address
LWPI MYREG get workspace
* set up new bottom for XBASIC
LI R0,BOTTOM
MOV R0,@>8330 write start of line number table
MOV R0,@>8332 write end of line number table
DEC R0 back one byte for free memory pointer
MOV R0,@>8386 first free address in high memory.
* set up to move memory
LI R0,>2000 get starting address of low memory
LI R1,BUFFER get address of 8K CPU buffer
JMP MLOOPS jump to move loop
* Workspace and data buffers.
MYREG BSS 32 workspace
SAVRTN BSS 2 return address
BUFFER BSS >2000 8K buffer for low memory storage
END

```

* End of program *
Application Notes:

For this utility to be of any use, the Extended BASIC loader program must be kept short. This routine takes up about 9K of memory.

Instructions:
In Extended BASIC, type in the following:

```

> NEW
> CALL INIT
> CALL LOAD("DSK1.XB/ALSAVE") !load in SAVE program to
high memory
> CALL LOAD("DSKx.xxxxxxxx") !load in your A/L program
> CALL LINK("SAVE") !save A/L program to high memory
buffer and move XB pointers.
> 1 CALL INIT::CALL LOAD(-31804,223,0) !branch to
RECALL program
> 2 RUN "DSK1.PROGRAM"
> 3 END
> SAVE DSK1.LOAD

```

* The LOAD program plus your A/L programs are now saved. The above program may also be MERGED into XB after "SAVE" is run.


```

1 CALL CLEAR
2 CALL SCREEN(2)
3 CALL COLOR(5,16,1,6,16,1,7
,16,1,8,16,1)
4 DISPLAY AT(5,10):"SNOOPY K
ONG"
5 DISPLAY AT(14,10):"BY TRIS
TANO"
6 DISPLAY AT(24,6):"DEDICATO
A MIA MADRE"
7 FOR T=1 TO 1500 :: NEXT T
8 RANDOMIZE
9 SNOOPY=3
10 LUC=4
11 VEL=20
12 CALL CLEAR
13 COLUMN=125 :: CHIAVI=0
14 RESTORE 136
15 CALL MAGNFY(3)
16 CALL SCREEN(2)
17 CALL HCHAR(1,3,34,LUC)
18 CALL CHAR(33,"18242418182
42418")
19 CALL CHAR(43,"00187E7E181
81800")
20 CALL CHAR(34,"001824247E6
66E3C")
21 CALL CHAR(40,"FF003C7E7E3
C00FF")
22 CALL CHAR(92,"03071D2F3F1
F0202074FFF4F0F07021F80C0E0F
0B0000010F8100080A0E02020")
23 CALL CHAR(96,"03071D2F3F1
F42E2477F0F2F2F3F210780C0E0F
0B0000000E020702000000080")
24 CALL CHAR(100,"0103070F0D
0000081F08000105070404C0E0B8
F4FCF84040E0F2FFF2F0E040F8")
25 CALL CHAR(104,"0103070F0D
00000007040E0400000001C0E0B8
F4FCF84247E2FEF0F4F4FC84E0")
26 CALL CHAR(108,"03071D2F3F
1F0242FF470F8F8FFE800080C0E0
F0B00000000000000000000000")
27 CALL CHAR(112,"03071D2F3F
1F42FF47070F0F8F8FFE8080C0E0
F0B00000000000000000000000")
28 CALL CHAR(116,"0103070F0D
0000000000000000000000C0E0B8
F4FCF84042FE2F0F1F17F0100")
29 CALL CHAR(120,"0103070F0D
0000000000000000000000C0E0B8
F4FCF842FFE2E0F0F0F1F17F01")
30 CALL CHAR(124,"030F1F3733
010143FF430383C6FF4000C0F0F8
ECC8080C2FFC2C0C163FF0200")
31 CALL CHAR(128,"030F1F3733
0141FF4303030383C6FF4C  )F8
ECC80822FFC2C0C0C16:  )"
32 CALL CHAR(132,"0000000000
00030707030001010F0001000000
000008080F0F08080C0C080E0")
33 CALL CHAR(36,"3C7F7F7F7F7
F3F7CFEFFFFFFF7E000008000C
0E0800000000000008080000000")
34 CALL CHAR(44,"00007C794CC
1207CFOE0F0400000E7F70000000
0008000000000000000000080")
35 CALL CHAR(136,"0F0F0F0F07
070301000000000000000F0F0F0
F0E0E0C08000000000000000000")
36 CALL CHAR(60,"0000000000
1030707070300000000000000000
000800C0C0C0800000000000000")
37 CALL CHAR(64,"01030F1F3F7
F7F6F450101010101050280C0F0F
8FCFEFEF622000000000000000")
38 CALL CHAR(68,"00001C3E7F7
F7F7F3F1F0F07030100000000387
CFEFEFEFEFCF8F0E0C08000000")
39 DISPLAY AT(7,5):RPT$("(",
21)
40 CALL HCHAR(1,25,43,SNOOPY
)

```

```

41 DISPLAY AT(24,4):RPT$("("
,23)
42 FOR T=7 TO 27 STEP 2 :: C
ALL VCHAR(8,T,33,13):: NEXT
T
43 CALL COLOR(1,6,1,2,8,5,5,
16,1,6,16,1,7,16,1)
44 CALL SPRITE(#20,44,5,33,1
05,#21,36,10,33,105,#22,132,
12,33,125)
45 CALL SPRITE(#1,100,16,169
,44)
46 CALL SPRITE(#2,136,11,INT
((80-150+1)*RND)+150,COLUMN)
47 FOR T=3 TO 6 :: CALL SPRI
TE(#T,60,5,INT((60-130+1)*RND
)+130,10,0,INT((-VEL-VEL+1)
*RND)+VEL):: NEXT T
48 CALL POSITION(#1,Y,X)
49 CALL JOYST(2,X1,Y1)
50 CALL KEY(2,K,S):: IF S=-1
THEN 73
51 IF X1>0 THEN GOSUB 53 ELS
E IF X1<0 THEN GOSUB 63
52 GOTO 48
53 IF X>210 THEN RETURN ELSE
CALL MOTION(#1,0,5)
54 CAR=100
55 CALL SOUND(10,-5,15)
56 CALL PATTERN(#1,100)
57 CALL SOUND(10,-8,4)
58 CALL PATTERN(#1,104)
59 CALL SOUND(10,-5,9)
60 CALL PATTERN(#1,100)
61 CALL MOTION(#1,0,0)
62 RETURN
63 IF X<40 THEN RETURN ELSE
CALL MOTION(#1,0,-5)
64 CAR=92
65 CALL SOUND(10,-5,15)
66 CALL PATTERN(#1,92)
67 CALL SOUND(10,-8,4)
68 CALL PATTERN(#1,96)
69 CALL SOUND(10,-5,9)
70 CALL PATTERN(#1,92)
71 CALL MOTION(#1,0,0)
72 RETURN
73 CALL JOYST(2,X1,Y1)
74 IF X<64 THEN 51 ELSE IF X
>200 THEN 51
75 IF X<58 AND X1<0 THEN 51
ELSE IF X>190 AND X1>0 THEN
51
76 IF X1>0 THEN X=5 ELSE IF
X1<0 THEN X=-5 ELSE X=0
77 CALL SOUND(50,900,3)
78 CALL MOTION(#1,-8,X)
79 FOR T=1 TO 200 :: NEXT T
80 CALL MOTION(#1,0,0)
81 CALL SOUND(50,300,3)
82 CALL PATTERN(#1,108)
83 CALL POSITION(#1,Y,X)
84 X2=INT(X/16)
85 IF CAR=100 THEN CALL LOCA
TE(#1,Y,X2*16+21)
86 IF CAR=92 THEN CALL LOCAT
E(#1,Y,X2*16+5)
87 CAR=108
88 MOV=0
89 CALL POSITION(#1,Y,X)
90 CALL MOTION(#1,INT((3-7+1)
*RND)+7),0,INT((-VEL-VEL+1)*
RND)+VEL)
91 FOR T=3 TO 6 :: CALL COIN
C(#1,#T,10,D):: IF D=-1 THEN
137
92 NEXT T
93 CALL POSITION(#2,Y2,X2)::
IF Y2<55 THEN GOSUB 128
94 IF Y>153 THEN 125
95 CALL JOYST(2,X1,Y1)
96 IF X1>0 THEN GOSUB 98 ELS
E IF X1<0 THEN GOSUB 106 ELS
E IF Y1<0 THEN GOSUB 113
97 GOTO 89

```

```

98 IF X>200 THEN RETURN ELSE
IF CAR=108 THEN CAR1=124 EL
SE IF CAR=124 THEN CAR1=
116 ELSE IF CAR=116 THEN CAR
1=108
99 IF CAR=116 THEN MOV=16 EL
SE MOV=0
100 CALL SOUND(10,-5,15)
101 CALL LOCATE(#1,Y,X+MOV):
CALL PATTERN(#1,CAR1)
102 CALL SOUND(10,-8,0)
103 CAR=CAR1
104 GOTO 89
105 RETURN
106 IF X<40 THEN RETURN ELSE
IF CAR=108 THEN CAR1=116 EL
SE IF CAR=116 THEN CAR1=
124 ELSE IF CAR=124 THEN CAR
1=108
107 IF CAR=108 THEN MOV=16 E
LSE MOV=0
108 CALL SOUND(10,-5,15)
109 CALL LOCATE(#1,Y,X-MOV):
CALL PATTERN(#1,CAR1)
110 CALL SOUND(10,-8,0)
111 CAR=CAR1
112 RETURN
113 IF CAR=124 THEN MOV=5 EL
SE MOV=2
114 IF Y<60 AND Y1>0 THEN RE
TURN
115 IF X<COLUMN AND CAR=108
OR X>COLUMN AND CAR=116 THEN
117
116 CALL COINC(#1,#2,8,D)::
IF D=-1 AND Y1>0 THEN CALL M
OTION(#1,-Y1*MOV,0,#2,-Y1*MO
V,0)
117 CALL MOTION(#1,-Y1*MOV,0
)
118 CALL SOUND(10,-5,9)
119 CALL PATTERN(#1,CAR+4)
120 CALL SOUND(10,-5,9)
121 CALL PATTERN(#1,CAR)
122 CALL SOUND(10,-5,9)
123 CALL MOTION(#1,0,0,#2,0,
0)
124 RETURN
125 CALL LOCATE(#1,169,X)
126 CALL PATTERN(#1,100)
127 GOTO 48
128 CALL SOUND(100,-3,0)
129 CHIAVI=CHIAVI+1
130 CALL HCHAR(1,34,LUC-CHIAV
I,32)
131 IF CHIAVI=LUC AND LUC=10
THEN 311
132 IF CHIAVI=LUC THEN 149
133 READ COLUMN
134 CALL LOCATE(#2,INT((80-1
50+1)*RND)+150,COLUMN)
135 RETURN
136 DATA 45,157,109,61,141,7
7,173,125,189,93
137 CALL DELSPRITE(#3,#4,#5,
#6)
138 CALL COLOR(#1,10)
139 FOR T=4 TO 30 :: CALL SO
UND(100,-1,T):: NEXT T
140 CALL MOTION(#1,20,0)
141 CALL POSITION(#1,Y,X)::
IF Y>154 THEN 142 ELSE 141
142 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,171,X)
143 FOR T=1 TO 30 :: CALL SO
UND(80,-7,T):: NEXT T
144 SNOOPY=SNOOPY-1
145 CALL HCHAR(1,25+SNOOPY,3
2)
146 IF SNOOPY=0 THEN 288
147 FOR T=1 TO 200 :: NEXT T
148 GOTO 39
149 CALL DELSPRITE(#2.#3.#4,
#5,#6)
150 CALL HCHAR(1,3,32)

```



```

151 CALL SPRITE(#2,64,14,150
,220)
152 FOR T=900 TO 300 STEP -3
00 :: CALL SOUND(100,T,3)::
NEXT T
153 CALL CHAR(136,"FF01011F3
161C1C3C77F3F1F9F88483FFE000
08083C3E7E7FEFCF8F8F02020FC"
)
154 CALL CHAR(60,"FF01010F18
103030383F3F1F0F24140CFE0000
E03010181838F8F8F0E0485060")
155 CALL MOTION(#1,20,0)
156 CALL POSITION(#1,Y,X)::
IF Y>154 THEN 157 ELSE 156
157 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,169,X):: CALL P
ATTEN(#1,100)
158 CALL MOTION(#1,0,13)
159 CALL POSITION(#1,Y,X)::
IF X>200 THEN 165
160 CALL PATTERN(#1,104)
161 CALL SOUND(10,-5,15)
162 CALL PATTERN(#1,100)
163 CALL SOUND(10,-8,4)
164 GOTO 159
165 CALL MOTION(#1,0,0)
166 CALL LOCATE(#1,158,228):
: CALL PATTERN(#1,112)
167 CALL MOTION(#2,-10,0,#1,
-10,0)
168 CALL POSITION(#2,Y,X)::
IF Y<30 THEN 169 ELSE 168
169 CALL MOTION(#1,-3,-10,#2
,-3,-10)
170 CALL POSITION(#2,Y,X)::
IF X<180 THEN 171 ELSE 170
171 CALL MOTION(#2,0,0,#1,0,
0)
172 CALL PATTERN(#1,96):: CA
LL LOCATE(#1,33,X)
173 CALL SOUND(10,400,4)
174 CALL MOTION(#2,-5,-8,#1,
0,-10)
175 FOR D=1 TO 9
176 CALL PATTERN(#1,96)
177 CALL SOUND(10,-5,4)
178 CALL PATTERN(#1,92)
179 CALL SOUND(10,-8,4)
180 NEXT D
181 CALL MOTION(#1,0,0)
182 CALL SPRITE(#3,68,10,15,
130)
183 FOR T=300 TO 900 STEP 30
0 :: CALL SOUND(100,T,3):: N
EXT T
184 CALL SOUND(100,500,3)::
CALL SOUND(200,600,4,800,3)
185 CALL DELSPRITE(#2)
186 CALL CHAR(72,"0000000000
0100000000000000000100003E
9E3283043E0F070F020000E7EF")
187 CALL CHAR(76,"0001000307
00000000000101010000003CFEFO
FOFEFEFC1C7EFEFEFFFE70000")
188 CALL CHAR(80,"0000000200
0600070706101818180000000F8
78C80C013C3C1800000000070F")
189 CALL CHAR(84,"0007030F1F
0303303C3F131F1F1F0000F0F8CO
COFOFOFOFOFFFF3F8FCFE1E00")
190 CALL CHAR(88,"0000000309
0318000303010000000307000000
E0E0203040F0E0C00303038180")
191 CALL CHAR(140,"00031F0F3
F7F0F0F070F0F0F0F07000000COE
OCOC0E0E0C0C0F8FCFCFC7C0000"
)
192 CALL DELSPRITE(#3)
193 CALL POSITION(#1,Y,X)
194 CALL PATTERN(#1,124):: C
ALL LOCATE(#1,Y,133,#22,Y-15
,133)
195 CALL MOTION(#1,15,0,#22,
15,0)

```

```

196 CALL POSITION(#1,Y,X)::
IF Y>153 THEN 202
197 CALL PATTERN(#1,128)
198 CALL SOUND(100,-8,1)
199 CALL PATTERN(#1,124)
200 CALL SOUND(100,-4,2)
201 GOTO 196
202 CALL MOTION(#1,0,0,#22,0
,0):: CALL LOCATE(#1,169,
,22,169,122):: CALL PATI
(#1,100)
203 CALL MOTION(#1,0,10,#22,
0,10)
204 CALL SPRITE(#2,136,5,1,1
00,3,-4)
205 CALL MOTION(#20,0,-4,#21
,0,-4)
206 FOR T=1 TO 6
207 CALL PATTERN(#20,72,#21,
76,#1,104)
208 CALL SOUND(100,-8,3)
209 CALL PATTERN(#20,80,#21,
84,#1,100)
210 CALL SOUND(100,-3,6)
211 CALL PATTERN(#20,72,#21,
76,#1,104)
212 CALL SOUND(100,-4,6)
213 CALL PATTERN(#20,88,#21,
140,#1,100)
214 CALL SOUND(100,-2,6)
215 NEXT T
216 CALL MOTION(#2,0,0)
217 CALL DELSPRITE(#20,#21,#
1,#22)
218 FOR T=400 TO 1200 STEP 4
00 :: CALL SOUND(100,T,1)::
NEXT T
219 FOR T=1 TO 200 :: NEXT T
220 FOR T=30 TO 1 STEP -1 ::
CALL SOUND(10,-6,T):: CALL
SOUND(10,-7,T):: NEXT T
221 CALL MOTION(#2,-3,-4)
222 FOR A=1 TO 6 :: FOR T=1
TO 8 :: CALL SOUND(10,-6,3):
: CALL SOUND(10,-7,3):: NEXT
T :: NEXT A
223 CALL DELSPRITE(#2)
224 CALL CLEAR
225 CALL CHAR(60,"7F000001C1
C3E7E77F3F1F1F0F04043FFF8080
F88C8683C3E3FEFCF8F91112FC")
226 CALL CHAR(64,"FF01010F18
103030383F3F1F0F24140CFE0000
E03010181838F8F8F0E0485060")
227 CALL HCHAR(24,1,40,32)
228 CALL SPRITE(#1,60,5,1,9,
5,5,#2,100,16,169,30,0,5,#3,
132,12,169,10,0,5)
229 FOR T=1 TO 31
230 CALL PATTERN(#2,104)
231 CALL SOUND(50,-6,2)
232 CALL SOUND(50,-7,1)
233 CALL PATTERN(#2,100)
234 CALL SOUND(50,-6,2)
235 CALL SOUND(50,-7,1)
236 NEXT T
237 CALL MOTION(#1,0,0,#3,0,
0)
238 FOR T=1 TO 10
239 CALL PATTERN(#1,64,#2,10
4)
240 CALL SOUND(10,-6,2)
241 CALL SOUND(10,-7,1)
242 CALL PATTERN(#1,64,#2,10
0)
243 CALL SOUND(10,-6,2)
244 CALL SOUND(10,-7,1)
245 NEXT T
246 CALL MOTION(#1,-6,-3,#3,
-6,-3,#2,0,0)
247 CALL PATTERN(#2,92)
248 FOR T=1 TO 30
249 CALL SOUND(50,-6,2)
250 CALL SOUND(50,-7,1)
251 NEXT T

```

```

252 CALL PATTERN(#1,136)
253 CALL MOTION(#1,-2,-5,#3,
-2,-5)
254 FOR T=1 TO 80
255 CALL SOUND(20,-6,0)
256 CALL SOUND(20,-7,1)
257 NEXT T
258 CALL DELSPRITE(#1,#3)
259 FOR T=1 TO 300 :: NEXT T
260 CALL SOUND(250,196,0)
261 CALL SOUND(250,175,1)
262 CALL SOUND(350,196,0)
263 FOR T=1 TO 150 :: NEXT T
264 CALL SOUND(200,196,0)
265 CALL SOUND(200,175,0)
266 CALL SOUND(200,165,0)
267 CALL SOUND(200,147,0)
268 CALL SOUND(200,131,0)
269 CALL SOUND(200,147,0)
270 CALL SOUND(200,165,0)
271 CALL SOUND(300,131,0)
272 FOR T=1 TO 200 :: NEXT T
273 CALL SOUND(250,196,0)
274 CALL SOUND(150,175,1)
275 CALL SOUND(350,196,0)
276 FOR T=1 TO 150 :: NEXT T
277 CALL SOUND(200,147,0)
278 CALL SOUND(200,165,0)
279 CALL SOUND(300,131,0)
280 FOR T=1 TO 300 :: NEXT T
281 LUC=LUC+2
282 CHIAVI=0
283 CALL CLEAR :: CALL DELSP
RITE(ALL)
284 RESTORE 136
285 COLOMN=125 :: CHIAVI=0
286 VEL=VEL+10
287 GOTO 12
288 REM
289 FOR T=1 TO 200 :: NEXT T
290 CALL SOUND(300,139,1)
291 CALL SOUND(400,185,1)
292 CALL SOUND(300,185,30)
293 CALL SOUND(300,139,1)
294 CALL SOUND(400,185,1)
295 CALL SOUND(200,185,30)
296 CALL SOUND(300,139,1)
297 CALL SOUND(300,185,1)
298 CALL SOUND(300,233,1)
299 CALL SOUND(300,185,1)
300 CALL SOUND(300,233,1)
301 CALL SOUND(400,185,1)
302 CALL SOUND(200,185,30)
303 CALL SOUND(300,139,1)
304 CALL SOUND(400,185,1)
305 FOR T=1 TO 400 :: NEXT T
306 CALL DELSPRITE(ALL):: CA
LL CLEAR
307 CALL CHARSET
308 CALL COLOR(5,16,1,6,16,1
,7,16,1)
309 DISPLAY AT(12,12):"GAME
OVER"
310 CALL KEY(0,K,S):: IF S=#
1 THEN RUN ELSE 310
311 REM
312 CALL DELSPRITE(#2,#3,#4,
#5,#6)
313 CALL COLOR(1,9,1,2,10,7)
314 CALL MOTION(#1,20,0)
315 CALL POSITION(#1,Y,X)::
IF Y>154 THEN 316 ELSE 315
316 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,169,X):: CALL P
ATTEN(#1,100)
317 CALL MOTION(#1,0,13)
318 CALL POSITION(#1,Y,X)::
IF X>195 THEN 324
319 CALL PATTERN(#1,104)
320 CALL SOUND(10,-5,15)
321 CALL PATTERN(#1,100)
322 CALL SOUND(10,-8,4)
323 GOTO 318
324 CALL MOTION(#1,0,0)
325 CALL MOTION(#22,0,13)

```

continued on page 13

```

100 REM CHANNEL 99 HAMILTON
USERS GROUP
110 REM WORDS
120 REM A5-TOO31-JEU
130 REM *****
140 REM * BY W.M. JOHNSON *
150 REM *****
160 DIM BIG$(26)
170 OPEN #1:"SPEECH",OUTPUT
180 CALL CLEAR
190 CALL SCREEN(8)
200 DATA DOG,CAT,HORSE,MAN,C
AR,MOON,BOAT,COW
210 FOR I=1 TO 8
220 READ A$
230 TEST$(I)=A$
240 NEXT I
250 B$=" "
260 FOR I=1 TO 8
270 PRINT TEST$(I);B$;
280 NEXT I
290 CALL CHAR(97,"O")
300 CALL CHAR(98,"FFFFFFFFFFFF")
310 CALL CHAR(99,"FFFEFCF8FO
EOCO8O")
320 CALL CHAR(100,"O10307OF1
F37FFF")
330 CALL CHAR(101,"FF7F3F1FO
F070301")
340 CALL CHAR(102,"8OCOEOF0F
8FCFEFF")
350 DATA aadbfadcaefabaaaba
bbbbbabaaababaaab,abbbfaba
ababbbcbabaaefabaaababbbbc
360 DATA aadbfadcaefabaaaa
baaaafabcaebca,abbbfaabaa
efabaaababaaababaaadcabbbca
370 DATA abbbbabaaaaabbbbaa
baaaabaaaaabbbb,abbbbabaa
aaabbbbaabaaaaabaaaaabaaaa
380 DATA aadbfadcaefabaaaa
baabbaefadcaebca,abaaababaa
ababbbbabaaababaaababaaab
390 DATA aabbbbaabaaaaabaaa
aabaaaaabaaaabbbba,abbbbbaab
aaaaabaaaaabaaababaaaeabca
400 DATA abaadcabadcaabdcmaa
bbfaaabaeafaabaeaf,abaaaaabaa
aaabaaaaabaaaaabaaaaabbbb
410 DATA abfadbabebcbabaaaba
baaababaaababaaab,abaaababfa
ababefababaeafbabaababaaab
420 DATA aadbfadcaefabaaaba
baaabaeafadcaebca,abbbfaba
ababbbcbabaaaaabaaaaabaaaa
430 DATA aadbfadcaefabaaaba
baaabaeafabcaebcb,abbbfaba
ababbbcbabaeafaabaaefabaaae
440 DATA abbbfabaabaaaeabbbfa
aaaaabafaaabaebbb,abbbbbaab
aaaaabaaaaabaaaaabaaaaabaa
450 DATA abaaababaaababaaaba
baaababaaabaebbb,abaaabaa
baababaaababaaabaeafdcaaecca
460 DATA abaaababaaababaaaba
babababbbfbaeaecca,abaaabae
dcaebcaaadbfadcaefabaaab
470 DATA abaaabaeafadcaebcaa
aabaaaaabaaaaabaa,abbbbbaaa
dcaaadcaaadcaaadcaaadcaabbbb
480 FOR C=9 TO 10
490 CALL COLOR(C,7,1)
500 NEXT C
510 FOR I=1 TO 26
520 READ C$
530 BIG$(I)=C$
540 NEXT I
550 REM *****
560 REM * ACCEPT INPUT *
570 REM *****
580 CALL KEY(3,K,S)
590 IF (S=0)+(S=-1)THEN 580
600 IF K=13 THEN 880
610 IF (K<65)+(K>90)THEN 580

```

```

620 COUNT=COUNT+1
630 WORD$(COUNT)=CHR$(K)
640 SPEAK$=SPEAK$&WORD$(COUN
T)
650 REM *****
660 REM * DRAW LETTERS *
670 REM *****
680 P=K-64
690 FOR ROW=1 TO 6
700 FOR COL=COUNT*6-5 TO COUN
T*6
710 X=X+1
720 CALL HCHAR(ROW,COL,ASC(S
EG$(BIG$(P),X,1)))
730 NEXT COL
740 NEXT ROW
750 X=0
760 PRINT #1:WORD$(COUNT)
770 REM *****
*****
780 REM * TEST FOR COMPLETE
WORD *
790 REM *****
*****
800 FOR I=1 TO 8
810 IF SPEAK$=TEST$(I)THEN 8
80
820 NEXT I
830 IF COUNT>4 THEN 880
840 GOTO 580
850 REM *****
860 REM * SAY WORD *
870 REM *****
880 PRINT #1:"YOUR<LETTERS<A
RE"
890 FOR I=1 TO COUNT
900 PRINT #1:WORD$(I)
910 NEXT I
920 PRINT #1:"YOUR<WORD<IS"
930 PRINT #1:SPEAK$
940 PRINT #1:"DID<YOU<GET<TH
AT"
950 PRINT #1:SPEAK$
960 PRINT #1:"NOW<TRY<ANOTHE
R<WORD"
970 SPEAK$=""
980 REM *****
990 REM * ERASE SCREEN *
1000 REM *****
1010 CALL HCHAR(1,1,32,320)
1020 COUNT=0
1030 GOTO 580

```

```

100 OPEN #1:"PIO"
110 CALL CLEAR
120 PRINT "PRESS": : "1 FOR
EMPHASIZED": : "2 FOR DOUBLE
STRIKE": : "3 FOR DOUBLE WIDT
H": : "4 FOR CONDENSED": : "5
FOR ITALICS": : "6 TO SKIP PE
RFORATIONS": : "7 FOR NORMAL"
130 PRINT : "8 FOR ELITE": : "
[ENTER] TO CONTINUE": :
140 CALL KEY(O,K,S)
150 IF S=0 THEN 140
160 IF K=13 THEN 430
170 IF K<49 THEN 140
180 IF K>56 THEN 140
190 IF K=49 THEN 270
200 IF K=50 THEN 290
210 IF K=51 THEN 310
220 IF K=52 THEN 330
230 IF K=53 THEN 350
240 IF K=54 THEN 370
250 IF K=55 THEN 410
260 IF K=56 THEN 390
270 I' 'I' #1: . . $(27);("E")
280 GOTO 140
290 PRINT #1:CHR$(27);("G")
300 GOTO 140
310 PRINT #1:CHR$(27);("W");
CHR$(1)
320 GOTO 140
330 PRINT #1:CHR$(27);("B");
CHR$(3)
340 GOTO 140
350 PRINT #1:CHR$(27);("4")
360 GOTO 140
370 PRINT #1:CHR$(27);("N");
CHR$(50)
380 GOTO 140
390 PRINT #1:CHR$(27);("B");
CHR$(2)
400 GOTO 140
410 PRINT #1:CHR$(27);("@")
420 GOTO 110
430 PRINT "ENTER [CH] TO CHA
NGE PRINTER": : "FORMAT AND P
RINT FONTS": : : :
440 INPUT "TEXT? ":A$
450 IF A$="CH" THEN 410
460 IF A$="ch" THEN 410
470 PRINT #1:A$:
480 GOTO 430

```

continued from page 12

```

326 CALL POSITION(#22,Y,X)::
IF X>210 THEN 327 ELSE 326
327 REM
328 CALL MOTION(#22,20,0)
329 CALL POSITION(#22,Y,X)::
IF Y>160 THEN 330 ELSE 329
330 CALL MOTION(#22,0,0):: C
ALL LOCATE(#22,169,X)
331 CALL MOTION(#1,0,17,#22,
0,17)
332 FOR T=1 TO 4
333 CALL PATTERN(#1,104)
334 CALL SOUND(10,-5,15)
335 CALL PATTERN(#1,100)
336 CALL SOUND(10,-8,4)
337 NEXT T
338 CALL DELSPRITE(#1,#22)
339 CALL DELSPRITE(ALL):: CA
LL CLEAR
340 CALL SPRITE(#1,132,12,90
,105,#2,92,16,90,145,#3,68,1
0,70,125)
341 FOR T=1 TO 1000 :: NEXT
T
342 CALL DELSPRITE(ALL):: CA
LL CLEAR
343 GOTO 8

```

continued from page 14

```

100 DISPLAY ERASE ALL AT(3,5
):"THE COST OF CREDIT" ! by
Jim Peterson
110 S,T,X=0 :: DISPLAY AT(8,
1):"AMOUNT OF PURCHASE?" ::
ACCEPT AT(8,21):A :: B,T=A
: DISPLAY AT(10,1):"CREDIT C
ARD INTEREST RATE?" :: ACCEP
T AT(11,1):R
120 DISPLAY AT(13,1):"SAVING
S ACCOUNT INT. RATE?" :: ACC
EPT AT(14,1):SR
130 X=X+1 :: I=B*R/100/12 ::
B=B+I :: T=T+I :: P=B/10 ::
B=B-P :: S=S+P+S*R/100/12
:: IF S<A THEN 130
140 D$="$$"&STR$(INT((T+A-S-A
+.5)*100)/100)
150 DISPLAY AT(17,1):"If you
had saved the amount":of y
our minimum 10% of the":bal
ance credit card payment":e
ach month for";X;"months,"
160 DISPLAY AT(21,1):"and us
ed it to pay cash, you":wou
ld have saved ";D$ :: GOTO 1
10

```

Putting it all together Part 4

by Jim Peterson, Tigercub Software

The hardest part of learning to program is not in learning what the various commands do; it is in learning how to put them together to do what you want them to do!

Key in these mini programs and run them to see what they do. Then read the explanation of each line and see how it does what it does.

The first program is one in which the computer uses logic to play a game against you. This one also demonstrates the use of DATA.

Line 100 clears the screen, turns it white, displays the title and instructions. Note that several lines of screen text can be programmed in one DISPLAY statement, separated by the colon print separators. The last character of the line must not extend beyond the quotation mark above, if it is to fit on one line of the screen.

Line 140 calls a subprogram which we will describe later, and line 150 holds the text on the screen until any key is pressed.

Line 170 DATA contains values to be used by the computer in playing the game, 180 contains the frequencies to play a salute if you win, and 190 has the frequencies to mourn your losses.

In line 210, P is the number which was picked by the other player, which has been made invisible and cannot be picked. C is the cumulative count of numbers picked. These would be 0 by default for the first game but must be reset here because program execution returns here to start a new game.

The screen is cleared again, the sprites are set to magnification 2 (single character double sized), and the J loop places 5 sprites numbered 1 to 5 on the screen, coloured dark blue (5), with ASCII codes 49 to 53 (1 to 5), the first at dot row 10 and spaced 30 dotrows apart.

Line 220 uses the subprogram for a display, accepts input validated for a single character Y or M, then displays a blank to erase the line.

In line 230, if the computer is to go first it automatically picks 4, changes sprite #4 to colour 1 (invisible), sets P (number chosen) and C (cumulative count) to 4, and displays count.

Otherwise, execution falls through to the next line where the player's input is requested and accepted, validated as between and 1 and 5, and rejected if the same as the previous pick.

In line 250, if P is not 0 (i.e., if it is not the first move of the game), the sprite of the number picked by the computer is restored to the dark blue colour. The sprite of the number picked is made invisible, its value assigned to P, and the count is incremented and displayed. If the player has reached 37 he has won, if he has gone over 37 he has lost, otherwise the computer goes next.

Line 170 DATA contains the numbers which the computer will try to reach, in sequence, in order to win. The line is restored and the data is read, one value at a time. If the count has already gone past that value, but it is not yet 37, the next value is read. Otherwise, the number to be selected by the computer, B, is X-C, which will bring it to the desired number X.

In line 290, the "I'm thinking" is just for show and the delay loop keeps the response from being too fast. If B is more than 5, it cannot be picked, of course. In this case, the optimum number is halfway between C and X because the opponent will not be able to reuse it to reach X. But this can only be selected if B can be evenly divided by 2, i.e. if it is an even number, which is determined by $B/2=INT(B/2)$.

Otherwise, in line 330, if B is more than 5 (it will not be if successfully divided in the previous line) or if it is the same as the one previously selected, the strategy is to select the lowest number possible, which is 1 or, if 1 was just used, the -1 truth value of $B=P$ will increase it to 2.

Lines 320-330 are similar to line 260; if the game is not yet over, execution returns to 240.

Execution jumps to 340 when the player has won. Line 180 DATA is restored, read in and used to play the five notes, which are increased by 1.03 in the second voice to give the arcade effect. If the player loses, line 360 restores 190 and reads values which are placed in the 3rd voice at a silent volume but sounded through the -4 noise in the 4th voice.

The subprogram in line 370, called for all the displays, alternately displays a blank and the string T\$ at row R, column C for a fast blinking effect.

The Credit program uses some very simple equations. In line 110, the variables are reset to 0 because the program returns here after each calculation. The amount (A) is also at this time the balance to be paid (B) and the total to be paid.

X is the counter of months. I is the interest for one month (rate/100 to get a decimal, /12 to get the monthly rate). New balance is the balance plus the month's interest. Total (T) to be paid is the previous total plus this interest. The payment for the month is 1/10 of the balance, and the new balance is the balance minus the payment. Savings (S) is previous savings plus the payment plus the month's interest on the savings. If the savings do not yet equal the amount, go back for another month; else, potential savings are difference between actual cost and total paid plus any difference (in final month) between S and A.

```

100 CALL CLEAR :: CALL SCREE
N(16):: DISPLAY AT(3,8):"THE
  *37' GAME" !by Jim Peterson
110 DISPLAY AT(5,1):" We wil
  l take turns picking":a num
  ber from 1 to 5, but":"not t
  he number that was just":"pi
  cked."
120 DISPLAY AT(10,1):" The n
  umbers we pick will be":"add
  ed to the total count."
130 DISPLAY AT(13,1):" Whoev
  er reaches 37 is the":"winne
  r, but if you go over":"37 y
  ou lose."
140 CALL SHOW(20,1,"Press an
  y key to start")
150 CALL KEY(0,K,S):: IF S=0
  THEN 150
160 DATA 4,11,17,24,30,37
170 DATA 262,330,392,523,523
180 DATA 1047,784,659,523,52
  3
190 C,P=0 :: CALL CLEAR :: C
  ALL MAGNIFY(2):: R=10 :: FOR
  J=1 TO 5 :: CALL SPRITE(#J,
  48+J,5,R,10):: R=R+30 :: NEX
  T J
200 CALL SHOW(24,1,"(Y)ou or
  (M)e first?"):: ACCEPT AT(2
  4,22)VALIDATE("YM")SIZE(1):Q
  $ :: DISPLAY AT(24,1):""
210 IF Q$="Y" THEN CALL SHOW
  (22,8,"I pick 4"):: CALL COL
  OR(#4,1):: P=4 :: C=4 :: CAL
  L SHOW(3,10,"COUNT=4")
220 CALL SHOW(20,8,"Pick you
  r number"):: ACCEPT AT(20,2
  6)VALIDATE("12345"):N :: IF N
  =P THEN 220
230 IF P>0 THEN CALL COLOR(#
  P,5)
240 CALL COLOR(#N,1):: P=N :
  : C=C+N :: CALL SHOW(3,10,"C
  OUNT= "&STR$(C)):: IF C=37 T
  HEN 320 ELSE IF C>37 THEN 34
  0
250 RESTORE 160
260 READ X :: IF C<X THEN B=
  X-C ELSE IF X<37 THEN 260
270 CALL SHOW(22,8,"I'm thin
  king..."):: FOR Y=1 TO 700 :
  : NEXT Y
280 IF B>5 AND B/2=INT(B/2)T
  HEN B=B/2
290 IF B>5 OR B=P THEN B=1-(
  P=1)
300 CALL SHOW(22,8,"I pick "
  &STR$(B)):: CALL COLOR(#P,5)
  :: CALL COLOR(#B,1):: P=B ::
  C=C+B :: CALL SHOW(3,10,"CO
  UNT= "&STR$(C))
310 IF C=37 THEN 340 ELSE IF
  C>37 THEN 320 ELSE 220
320 RESTORE 170 :: FOR J=1 T
  O 5 :: READ F :: CALL SOUND(
  100,F,5,F*1.03,5):: NEXT J :
  : CALL SHOW(12,8,"YOU WIN!")
330 CALL SHOW(15,8,"Play aga
  in? (Y/N)"):: ACCEPT AT(15,2
  6)VALIDATE("YN"):Q$ :: IF Q$
  ="N" THEN STOP ELSE 190
340 RESTORE 180 :: FOR J=1 T
  O 5 :: READ F :: CALL SOUND(
  300,30000,30,30000,30,F,30,-
  4,5):: NEXT J :: CALL SHOW(1
  2,8,"YOU LOSE!"):: GOTO 330
350 SUB SHOW(R,C,T$):: FOR J
  =1 TO 10 :: DISPLAY AT(R,C):
  " " :: DISPLAY AT(R,C):T$ ::
  NEXT J :: SUBEND

```

continued on page 13

Implanting Assembly in Extended BASIC for up to 4 program files

by Ross Mudie

This article is a follow on to the technique described by George Meldrum in the November 1987 TND magazine, which shows how to implant single file programs. This is NOT a beginner's article.

The program chosen is for the example is Sewer Mania. This program has an additional problem of intermittently failing to recognise the Speech Synthesizer however this problem is NOT addressed in this article.

Sewer Mania is in 4 Memory Image files with headers as follows:

SEWERMANIA1	SEWERMANIA2	SEWERMANIA3	SEWERMANIA4
FFFF2000A000	FFFF2000BFFA	FFFF1A00DF4	00000E002800
Code length			
1FFA	1FFA	19FA	ODFA
Ultimate Load address			
A000 to BFF9	BFFA to DFF3	DF4 to F9EE	2800 to 35FA
Implanted start address			
DEF6	DFEE	E5EE	F1EE
X/B line numbers ie 8330 and 8332			
DEF5	DFED	E5ED	F1ED

Other memory allocations to be considered.

Assembly Loader . . . to FFE7 (not fully using the space).

Loader Work Space 2600 to 261F

Character moving buffer 3600 to 390F

X/B INIT routines 2000 to 24F3

Space for relocation of code if required.

3910 to 3FFF (Not needed for Sewer Mania, but included to assist with facing the problem in other programs.)

Part 1 has the assembly file commencing at FEFO thus the implanted assembly must reside between DEF6 and FEFO.

Parts 2, 3 and 4 have no assembly, thus the top for these is FFE8 rather than FEFO which was used for part 1. The actual top is one byte less than this, because these addresses are the start of the next code area. Hence the actual tops are FFEF and FFE7. However use FEFO and FEF8 to get your maths correct.

The Extended BASIC lines follow for the headers of the four parts. It is only necessary to tell X/B that it has no extension memory once with the CALL LOAD(-31868,0,0) in the first part. The 4 X/B headers can be saved as merge files and merged in as necessary. These are saved out of X/B using SAVE DSK1.M1, MERGE etc. I use M1 for part 1, M2, M3, and M4 for parts 2, 3 and 4.

```
100 REM SAVE DSK1.SEWRMAN1XB
110 CALL INIT
120 ON ERROR 190
130 DISPLAY ERASE ALL:"LOADING PART 2"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
150 DISPLAY :RPT$(" ",28);"* IGNORE REWIND TAPE
MESSAGE";RPT$(" ",28)
160 CALL LOAD(-31868,0,0):: RUN "CSI"
170 DISPLAY : "DSK1.SEWRMAN2XB FILE"
180 CALL LOAD(-31868,0,0):: RUN "DSK1.SEWRMAN2XB"
190 CALL LOAD(-31868,255,231)
```

```
100 REM SAVE DSK1.SEWRMAN2XB
110 CALL LOAD(8196,254,240)
120 ON ERROR 190
125 CALL LINK("SEWER2")
130 DISPLAY ERASE ALL:"LOADING PART 3"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
150 DISPLAY :RPT$(" ",28);"* IGNORE REWIND TAPE
MESSAGE";RPT$(" ",28)
160 RUN "CSI"
170 DISPLAY : "DSK1.SEWRMAN3XB FILE"
180 RUN "DSK1.SEWRMAN3XB"
190 CALL LOAD(-31868,255,231)
```

```
100 REM SAVE DSK1.SEWRMAN3XB
110 CALL LOAD(8196,254,248)
120 ON ERROR 190
125 CALL LINK("SEWER3")
130 DISPLAY ERASE ALL:"LOADING PART 4"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
```

```
150 DISPLAY :RPT$(" ",28);"* IGNORE REWIND TAPE
MESSAGE";RPT$(" ",28)
```

```
160 RUN "CSI"
170 DISPLAY : "DSK1.SEWRMAN4XB FILE"
180 RUN "DSK1.SEWRMAN4XB"
190 CALL LOAD(-31868,255,231)
```

```
100 REM SAVE DSK1.SEWRMAN4XB
110 CALL LOAD(8196,255,0)
120 CALL LINK("SEWER4")
```

Order of doing implanting.

1. Load DEBUG.
2. OLD the part 1 memory image file into memory.(I/O Error 50 must be received)
3. CALL LINK("DEBUG")
4. Inspect VDP memory to find size etc eg, M0975V,097F
5. Move the code as necessary, eg, N097BV,DEF6,1FFA
6. Change the X/B line number pointers, eg, M8330 <enter> DEF5 <space> DEF5 <enter>
7. Quit out of Debug, eg, Q <enter>.
8. Merge X/B header, eg, MERGE DSK1.M1
9. Load the already assembled object file containing the corect addresses, eg, CALL LOAD("DSK1.M0")
10. Save the implanted program file, eg, SAVE DSK1.SEWRMAN1XB
11. Go to step 2 for the next part or end.

It is also possible to save the X/B header with the assembly game program but without the loader. Later when the loader has been edited to provide all the required addresses and assembled, follow the following order.

1. CALL INIT
2. OLD DSK1.SEWRMAN1XB
3. CALL LOAD("DSK1.M0")
4. SAVE DSK1.SEWRMAN1XB

The order of 1 and 2 may be reversed. 3 must follow 1 and 2 so that the code from the object file will overwrite the memory space.

How to tell if the loader is too big.

The assembly loader program has some free space between its end and the end of the space allocated. If you need to add extra code to relocate code etc you may end up past FFE7. If this happens then Extended BASIC will overwrite your code and your loader code will fail to operate. The way to check where your code will reside is to get a LIST file at assembly. If the printer is on PIO then at the LIST prompt type in PIO, noting that the period is required after the device name. The options required will be RL for Registers prefixed by R and L to produce a List file. Since the AORG signifies absolute code, then column 2 of the list file will show the addresses in memory and column 3 will contain the hex machine code. The last address in your code must not exceed FFE6. If your code is too big then it must be made more compact or the AORG must be moved down. If this occurs then the implanted code must be moved down and the LINK addresses changed in the X/B to match the new assembly addresses.

What to do if the program overwrites the loader.

In the example program the part 3 file did not overwrite the area in which the loader was placed. The file at DFF4 is only 19FA bytes long and ends at F9EE. Larger files can overwrite the area FEFO up and destroy the loader. When this happens it is necessary to move the remaining part of the loader before this happens into free space, then branch into the start of the relocated code. Care must also be taken to avoid placing the program over your own work space or using buffer space for the character definition moves which overwrites the program.

To relocate, here is an example of suitable code:

```
LI R2,MOVFIN Address after end of code being
moved
LI R0,MOVST Address of start of code being moved
S R0,R2 Subtract to find size
LI R1,>3910 Where to put it, after chars buffer
space
```


Regional Group News

```

MOLOOP MOV *RO+,*R1+ Move code 2 bytes at a time
                (These 3 lines could be
DEC R2          Count down (replaced by BL @LOOP1
JNE MOLOOP     Finished yet? (in the following source
B @>3910        Branch to code

```

```

MOVST          Start of code being relocated
*              . . End of code being relocated
MOVFIN
*

```

* LOADER SOURCE CODE.

* S=MS O=MO

```

ENDVDP EQU >8370 pointer to end of vram buffer
VMBR EQU >202C vram multiple byte read
VMBW EQU >2024 Vram multiple byte write
VWTR EQU >2030 VDP write to register
AORG >FEF0 i.e. >FFE8 minus the length of this
*routine = >FEF0 with a bit of space to spare for
*extra code. The following entry points are just like
*the X/B DEF table. The first 6 bytes are the link
*name and the next 2 bytes are the entry point
*address. The CALL LINK(8196,254,240) etc in the X/B
*program places the address of the name to link to in
*the normal pointer to the start of the DEF table.
*When the CALL LINK("SEWER2") etc occurs, it is found
*at the first test.

```

```

TEXT 'SEWER2' link name (at >FEF0 - 254,240)

```

```

DATA PART2

```

```

TEXT 'SEWER3' link name (at >FEF8 - 254,248)

```

```

DATA PART3

```

```

TEXT 'SEWER4' link name (at >FF00 - 255,0)

```

```

DATA PART4

```

```

LOOP1 MOV *RO+,*R1+ Subroutine to transfer code
DECT R2
JNE LOOP1
RT

```

```

PART2 MOV R11,@SAVRTN Save return address

```

```

LWPI >2600 Load my workspace

```

```

LI RO,>DEF6 Start of implanted code in IXB

```

```

LI R1,>A000 destination address in CPU RAM

```

```

LI R2,>1FFA length of code to move

```

```

BL @LOOP1 Do the move

```

```

MOV @ENDVDP,RO Store end of VDP address in RO

```

```

AI RO,->1FF9 1 less than length of code find
                start of 2XB code in VDP

```

```

LI R1,>BFFA destination ram address

```

```

LI R2,>1FFA length of code

```

```

BLWP @VMBR transfer to ram

```

```

JMP RETURN

```

```

PART3 MOV R11,@SAVRTN

```

```

LWPI >2600 my workspace

```

```

MOV @ENDVDP,RO Store end of VDP address in RO

```

```

AI RO,->19F9 1 less than length of code find
                start of 3XB code in VDP

```

```

LI R1,>DFF4 destination ram address

```

```

LI R2,>19FA length of code

```

```

BLWP @VMBR transfer to ram

```

```

RETURN CLR RO Standard return to Extended BASIC

```

```

MOV B @>837C

```

```

LWPI >83E0

```

```

MOV @SAVRTN,R11

```

```

B *R11

```

```

PART4 LWPI >2600 my workspace

```

```

MOV @ENDVDP,RO Store end of VDP address in RO

```

```

AI RO,->0DF9 1 less than length of code find
                start of 4XB code in VDP

```

```

LI R1,>2800 destination ram address

```

```

LI R2,>0DFA length of code

```

```

BLWP @VMBR transfer to ram

```

```

LI RO,>03F0 from

```

```

LI R1,>3600 to BUFFER

```

```

LI R2,>0310 how many

```

```

BLWP @VMBR read character definitions

```

```

LI RO,>08F0 back to

```

```

BLWP @VMBW relocate to E/A position

```

```

LI RO,>030E

```

```

BLWP @VWTR E/A default color table

```

```

LI RO,>0401

```

```

BLWP @VWTR E/A default pattern table

```

```

LI RO,>07F5

```

```

BLWP @VWTR E/A default screen color

```

```

B @>A000 execute PROGRAM

```

```

SAVRTN BSS 2 When assembled this MUST be at or
                below >FFE6

```

```

END

```

The information presented here is obtained from the BBS. Some Regional groups are not advising of their meetings, which makes the maintenance of this file very difficult.

Meeting summary.

Carlingford	16/3/88	???
Central Coast	12/3/88	Toukley
Glebe	10/3/88	Glebe
Illawarra	21/3/88	Keiraville
Liverpool	11/3/88	???
Sutherland	18/3/88	???

TiSHUG in Sydney

Regular meetings are normally at 2pm on the first Saturday the month, except January, at the Woodstock Community Centre, Church Street, Burwood.

BANANA COAST Regional Group

(Coffs Harbour area)

For information on meetings of the Banana Coast group, contact Keir Wells at 9 Tamarind Drive, Bellingen, telephone (066) 55 1487.

CARLINGFORD Regional Group.

Regular meetings are usually on the third Wednesday of each month at 7.30pm.

Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02) 871-7753, for more information.

CENTRAL COAST Regional Group.

Meetings are normally held on the second Saturday of each month at 6.30pm at

Toukley Tennis Club hall,

Header St, Toukley.

Contact Russell Welham (043) 92 4000

GLEBE Regional Group.

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

The activities at these regional meetings are rather informal and include looking at new hardware, hardware repairs, looking at new software and having a general chat.

ILLAWARRA Regional Group.

Regular meetings are normally on the third Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre.

Contact Bob Montgomery on (042) 28 6463 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TiSHUG Sydney meeting at 7.30pm.

Contact Arto Heino (02) 603 8956 for more information.

NORTHERN SUBURBS Regional Group.

Regular meetings are on the third or fourth Thursday of each month.

Contact Dennis Norman on (02) 452 3920 or Dick Warburton on (02) 918 8132 for further information.

SUTHERLAND Regional Group.

Meetings are held on the third Friday of each month.

Group co-ordinator is Peter Young, telephone (02) 528 8775.

BBS Contact is Gary Wilson, user name VK2YCW on this BBS.

All TiSHUG Regional groups are invited to submit items for this department. Send details as system mail to SYSOP.

Tutorial Tutorial

Microsoft Multiplan^(tm) An Electronic Spreadsheet

by Tom Kennedy, U.S.A.

ELECTRONIC SPREADSHEETS... CELLULAR ANALYSIS...
FORMULAS... CELLULAR REFERENCING... WORKSHEETS...
ABSOLUTE REFERENCING...

These are buzzwords of a form of Data Processing that on the surface appears to be incomprehensible to all but accountants and the bridge crew of the Star-Ship Enterprise. As Word Processing is to writing a letter, Data Processing is to using a multiplication table.

Many people have a hard time using spreadsheets, because working with data in this format is similar to learning a new language. But once you learn to use the commands, and the procedure of working with data in a two dimensional row/column format instead of a one dimensional equation, you will find many ways in which Multiplan will allow you to "crunch numbers" faster and easier than using a calculator and notebook. More than that, Multiplan is flexible enough to be used anytime you want to display, or use, numbers or words in a row/column format. In fact, you could even adapt Multiplan to use it as a Word Processor!

What is a spreadsheet? In business, you often hear reference to "our books". The "books" that the businessmen, and you and I, keep are a pen and paper record showing the Debits and Credits of various bills paid and assets gained, plotted against a scale of time. Each intersection of row and column contains an entry for a value. The last column and/or last row contain a summation of all previous columns or rows. In an electronic spreadsheet, you recreate the printed form with the addition that each "cell" (a row/column intersection) can also contain a mathematical equation, or "formula", that automatically acts upon pre-defined cells and displays the result accordingly. If any value in any cell is changed, the formula instantly updates displayed results. This self maintenance ability is what pays off in using an electronic spreadsheet, such as Multiplan.

To operate Multiplan software on the TI, you must have 32K memory expansion, and at least one disk drive. An RS232 card and a printer are also handy, but not mandatory because unlike word processing, where the end result is a printed piece of paper, the end result with a spreadsheet is useful data, which may be used many ways. Most worksheets are well over 80 characters in width, (up to 2016!) and this requires a cut and paste job, so a wide carriage printer is preferable.

To load Multiplan, you insert the cartridge and program disk, select Multiplan from the menu, and press <ENTER> to load. Before pressing <ENTER> you can select one of eight screen color combinations by pressing the space bar.

The first thing you will see is a grid across the top and left side of the screen. These numbers are the row and column locations in the top left, or "HOME" position. There are 255 possible rows and 63 possible columns, with the screen framing a small section. Each "CELL" is referred to by its row/column location, such as: R1C1, R10C22, etc. In R1C1, the Home position, there is a solid rectangle, as large as the width of the cell. This is your cursor, or "CELL POINTER", which is where any entry will appear. Below the cell grid is the COMMAND LINE, which shows the primary commands you will use. There are a number of sub-commands related to each of these, but you must type the first letter of the primary command first, or place the cursor over the command and hit <ENTER>. Below the command line is the MESSAGE LINE, which prompts you for further information when needed. In the bottom left corner is the current cell pointer location, and to the right of that is the contents of the current cell. Lastly, in the bottom right corner is the available memory space remaining.

Appendix A is a list of the twenty commands shown in the Command Line, with the forty sub-commands that apply to each.

- ALPHA The first command given before entering text into the current cell location. All alpha-nums can be used, but numbers will be treated as text, and cannot then be used as values for calculations.
 - BLANK Used to erase the contents of a specified cell or range of cells. Blanked cells retain their location and format.
 - COPY Allows you to duplicate any cell or cells in any number, including both cell format and content.
 - DELETE Completely erases a row or column.
 - EDIT Allows you to edit the contents of a cell, or the formula in that cell, without re-entering the data. Requires careful use of the EDIT keys.
 - FORMAT Defines all of the various parameters of cell width, content, and display of data.
 - GOTO Moves your cell pointer immediately to any cell, by giving the row/column or pre-defined name. Also used to move from one window to another.
 - HELP Calls up a detailed HelpFile (from disk) that covers the whole Multiplan software, including a command summary.
 - INSERT Inserts a blank row or column, formatted to DEFAULT settings.
 - LOCK Protects the cell, or formula, from accidental overwrite.
 - MOVE Moves a cell, or group of cells, to specified row/column, deleting the original.
 - NAME Allows you to give a name to a cell or group of cells to aid in future references to that cell. "Total" or "Sales" are examples.
 - OPTIONS Covers global options such as RECALC, MUTE, and ITERATION. The most important of which is cancelling RECALC, to avoid waiting for each entry to recalculate the entire worksheet.
 - PRINT Used to print the worksheet. Before printing, you must first define the extent of the field to be printed with MARGINS and OPTIONS, then select PRINTER to start output. PRINT FILE outputs to disk instead of the printer to be included into a Word Processor file, or other cases where you need the worksheet stored in ASCII format.
 - QUIT Self explanatory, provides a "safe" exit.
 - SORT Sorts entries in cells in a specified column, in either ascending or descending order.
 - TRANSFER Includes six sub-commands which are used: to LOAD, SAVE, RENAME, or CLEAR an active worksheet. Also, to DELETE a file from a disk, and an OPTIONS command to define disk name and format.
 - VALUE Used to enter a numerical value or formula into a cell. This must be used for numbers that will be used in calculations.
 - WINDOW A window is used to overlay one or more portions of a worksheet with another. As an example, to hold the titles of columns fixed while the data scrolls underneath. The sub-commands define how the windows are opened, closed, or linked together. A border can be defined to offset it from the worksheet.
 - EXTERNAL Allows related worksheets on disk to be linked as a source of data for the active sheet. Any range of cells can be drawn up for reuse. Multiple worksheets can be linked relative to each other so as to work together.
- *****
Appendix B is a list of the Key Functions used in the TI version of Multiplan. Most functions have two optional keystrokes choices. This is to allow you flexibility in accessing them.

CELL POINTER CONTROL KEYS:
- F-E
 - F-X Typical cursor keys, scroll the sheet up,
 - F-S down, left, right. As in BASIC.
 - F-D
 - C-E
 - C-X Page scroll. Similar to cursor scroll,
 - C-S except moves in one screen width or height
 - C-D blocks.

- C-6 (C-W) Moves cursor to the next window as defined with the WINDOW command
- C-3 (C-F) Jumps to the next unlocked, unblank cell to the right, skipping over cells protected with the LOCK command.
- C-1 (C-Q) Jumps to the "HOME" position, row 1/column 1, which is the view seen when first starting up Multiplan.
- F-1 (C-Z) Opposite of C-1 ("HOME"), except stops at the lower right corner boundary of the area you are working on.

ACTION KEYS:

- <SPACE> In Command mode, (when command choices are displayed) skips through the commands, highlighting each with the cursor. Hitting <ENTER> selects the highlighted command. In the command menu, forward spaces through each response field.
- C-H (F-9) (When not editing) Backspaces through the response field, opposite of <SPACE>, to make selections of options.
- C-A (C-I, C-2) TAB key. Tabs over a response field to the next selection.
- C-C (C--) Cancels current operation. A failsafe "escape" to abort command selection.
- F-4 (When printing file) Aborts printing operation.
- <ENTER> Activates a menu selection or command.
- F-7 (F-I) Activates and displays the helpfile, which must be on the default disk drive (see TRANSFER OPTIONS)
- F-8 Recalculates the entire worksheet when the RECALC feature has been cancelled with the OPTIONS command.
- <-, +, 0-9> Invokes the VALUE option of data entry, as opposed to ALPHA, for text entry.

EDIT KEYS:

- C-H (F-9) Backspaces through data entry for editing
- F-0 (C-Y) Single-character delete.
- C-4 (C-L) Skips to the next character right (like "F-D" in BASIC)
- F-4 (C-K) Skips to the next character left (like "F-S" in BASIC)
- C-5 (C-P) Skips to the next word right.
- F-5 (C-O) Skips to the next word left.
- C-7 Changes all relative references to cell locations (e.g. R+1C+2) to absolute references. (e.g. R2C5)

Upon selecting a command, a command menu appears with a number of response fields shown. In each response field is a proposed response, which is the default selection unless you change it. To use a command, type its key letter and fill in the response fields. To move through the fields, use the tab key until the cursor is highlighting the correct area. Type in your response, and either tab to the next field or hit <ENTER> to activate the command.

When the necessary response is a row/column cell reference, there are two ways to respond: Absolute and Relative. Absolute referencing is a numerical definition of the cell location, such as R5C10 (the intersection of row 5 and column 10). A group of cells, a RANGE, is called by giving the boundary intersections separated by a colon (:), such as: R2C1:R4C10 (or R2:4C1:10) defines a 3-by-10 cell grid consisting of columns 1 through 10 on rows 2 through 4.

Relative referencing involves identifying the desired cell by displacement from another cell, usually the one the cell pointer is currently on. As an example, if you are on row 5, column 10, (R5C10) and you wish to refer to a cell two rows up and three columns over, (R3C13) you could type in R-2C+3 or use the cursor keys to move the cursor over R3C13.

The relative address will automatically update as you move. When the cursor is in place, hit <ENTER> (or tab to the next field) and the reference is defined.

So far, I have covered what you see on the screen and the response to the various commands; what the commands and key functions are; and how to fill in response fields where needed. Before going on to building a worksheet, you will need to know how to save what you are working on, and how to load it back in. Besides the fact that you will want to take a break occasionally, it is nice to be able to experiment with the commands, "messing up" the worksheet, and then

loading the "clean" version back in to continue.

The LOAD and SAVE commands are under the command TRANSFER (a lousy name). Hit "T" and the menu will be displayed. Since the first option is LOAD, hitting <ENTER> now will prompt for a filename. To select SAVE, (or another option) hit the first letter and <ENTER>, or tab through to the desired item and hit <ENTER>. When loading or saving, you will be prompted for a filename the first time, which will become the default response.

Multiplan also incorporates an extensive helpfile contained on disk. When the command line is displayed, you select HELP with either the Help action key (<AID> or "?") or by typing "H". The worksheet will be replaced by the beginning of the helpfile. If a command has been selected, hitting the help key will display the specific section of the helpfile that pertains to the command you are using. The help menu allows you to RESUME (return to command menu), START at the top of the helpfile, or move to NEXT or PREVIOUS page of information.

The first step to creating a worksheet is to decide how many rows and columns you will need, and how the data will be displayed. It is best to sketch this out on paper to get a feel for how it will look. Also, you will need to decide what formulas will have to be created that use the data contained in the worksheet. Lastly, you will probably want to change the format of many of the cells, usually by rows or columns. Most often, the formatting required is for display purposes. Cell width, alignment of the data within the cells, etc.

Now that you know how everything will look, begin by formatting the cells. Upon start up, the cells are set with a number of defaults. You may want to change the widths of some columns, to between 3 and 32 columns, to show all of the entry for the cells. If the data in a cell is too large to fit the width of the cell, it will be truncated to fit, unless it is a numerical entry, where it will be replaced by a string of "#s".

FORMAT CELLS is used to set cell alignment and display format. A cell can be aligned to either center text for columnar headers, etc., or to align data displayed in tables. For instance, a table of dollar values could be shown with a "\$" in front and decimal points aligned.

The display formats are used to show how the data appears in a cell. CONTinuous allows the text in a cell to run over the right boundary to the next cell. If all cells are made continuous, you have a word processor type format. EXP displays numbers in scientific notation. Fixed Point rounds off decimals to a defined number. GENERAL is as you see when starting up, values displayed as entered. INTEger rounds off all numbers to integers. "\$" (Dollar) adds a dollar sign to numbers and rounds to two decimals. "*" replaces the number with an equivalent number of asterisks, to use like a bar graph. "%" displays the number in percent form. Lastly, the "-" just leaves the setting at the previous option.

Now that the cell formats are defined, it is time to start entering data. Begin by labelling your rows and columns, as necessary. To enter data, either text or values, move the cursor to the desired cell and hit either "A" or "V", depending on the type. The command line will disappear and you will be prompted for either text or value. Type in your entry and hit either <ENTER> to return to the command line, or use the appropriate FCTN-ARROW key to move to the next cell. With the FCTN key, when you land on the next cell, you are prompted only for text/value entry. In this case, you do not hit A or V to declare type, but when you begin entering data, Multiplan decides what style the data is, and responds accordingly. The only disadvantage is that there is a slight delay between the first character of your entry and the remainder, so if you type in, for instance, the word "TOTALS" too quickly, all you will see in the cell is "TTOLS". After a bit of use, a "stutter" habit is developed in how you enter data, so this becomes less apparent. When entering data, if an error is made, do not use the FCTN-S key to backspace for correction (as programmers are used to), the backspace key is CTRL-H (as telecommunication folks are used to). continued on p 19

If, after creating part of a worksheet, you need to add or delete rows or columns, three commands apply. DELETE completely removes any number of rows or columns. BLANK just removes the data in the cells, the row/columns remain and retain their formats. INSERT creates a new row or column set to the default settings.

Formulas are used to perform a mathematical computation upon the data in a cell or group of cells. One example is in a sales order form, where you have a column of data that is totalled at the bottom, multiplied by a tax percentage, and the tax added to the result. The cell in which the sub-total is to appear would contain a formula describing a sum of the data in the columns, expressed as either a chain addition problem, (R3C5+R4C5+...+R10C5) or using the SUM() function and a range of cells (SUM(R3C5:R10C5)). The formulas can become quite complex, depending on the work performed. Appendix C contains a list of the mathematical functions that can be used in building formulas.

Formulas can also consist of names of cells as the operand, as in "SUBTOTAL * .079", to calculate the entry for a cell named TAX. Names are assigned with the name command. Names can be any continuous string of alphanumeric characters, but must begin with a letter. Simply place the cursor over the cell to name and press N. Type in the desired name to the response field, and TAB to the next field. The current cell will be shown as the proposed response. If a range of cells is desired, hit the FCTN key, at the cell response, to move the cursor from the current location to the end point, then hit <ENTER>. In this manner, a whole row or column can be named. Names can also be used in the GOTO command to aid in moving quickly to a location. "GOTO TOTALS" for example.

Windows allow you to view more than one area of your worksheet at one time. You can split a row or column of titles to form a window over the data, so that as the cursor is moved throughout the worksheet, the headers remain in place to see what data is shown. Also, separate worksheets can be developed in one and divided into windows so all can be seen at once. After selecting the window command, four options are shown. SPLIT opens the windows, either horizontally, vertically, or at preset titles. LINKing two or more windows scrolls them together as you move through the worksheet. BORDER is used to put a border of any character surrounding the windows, to make them easier to read. A window is cancelled with the CLOSE option.

Once you have finally created the worksheet, and all the data has been entered, what do you do with it? In a sense, the end product is the worksheet, because you may refer to it constantly as new data is applied, and a printed copy might become outdated quickly. After all, that is part of the reason you are working on an Electronic Spreadsheet in the first place, the instant and easy update of information.

In some cases though, a printout is desired, either in the form of a disk file that can be incorporated into a document on a word processor, or a hard-copy printout for reference. The printer command has four options used in printing the worksheet. FILE prints the worksheet to disk in display variable 80 format, which can be loaded into a word processor. Before printing a hard copy, you must first set margins and print options. The MARGINS option sets the limits of rows and columns in the printout, along with indentations and paginations. OPTIONS defines the portion of the worksheet to be printed, using a range of cells. The set-up field contains the device name of your printer. The last two fields let you print the formulas "hidden" in cells, and whether or not to print the row/column numbers. After margins and options are defined, select the PRINTER option to begin the print out. If the width of the worksheet exceeds the width of your printer carriage, the left half will be printed entirely, then the right half below that, so the two can be cut and pasted together.

In some cases, you may be working on a number of worksheets that are related to each other, such as in a business with SALES/PAYROLL/INVENTORY spreadsheets. These separate files can be linked together so data can be drawn from, as an example, the INVENTORY file to be

used in the SALES worksheet and information from SALES could be used in PAYROLL.

The eEXTERNAL command, (press "X" at command line) is used to COPY data from an inactive sheet into the active one. You are prompted for the filename of the source sheet, the name (or R/C reference) of the source cell, the destination cell of the data, and the LINK option. If LINK is selected, then the two sheets will become linked so that when the destination sheet is loaded, the source sheet will automatically be used to supply data where needed. The LIST option displays the names of all sheets supporting the active sheet. The USE option allows you to switch which inactive sheets will support the active sheet, so long as they are in the same format. As an example, the SALES sheet would call upon different INVENTORY sheets for each month, all created in the same format, with different data.

Multiplan is one of the most powerful tools to be used on any computer. Its versatility allows it to be used in many different applications, such as word processing, record keeping, budget/accounting, etc. Any application that requires storing data in a tabular format. The instant update of information and the advanced mathematics capability can be used in a variety of ways.

Versatility is the main attraction of the many spreadsheet programs used on various machines, and in fact, Multiplan can even use files stored in VISICALC(tm) format. VISICALC, one of the "first" major spreadsheets, is similar to Multiplan in many ways: the screen display; cursor positioning; error correction; and entering data and formulas. The referencing of cells is more detailed with Multiplan, including the ability to name cells for ease of use. It has been shown that Multiplan can be easier to pick up and use for the person not familiar with spreadsheets, although once the concepts are mastered, the usage is similar in all. With a familiar knowledge of a program like Multiplan, you could do away with a word processor, a database manager, or even a pocket calculator, although each has its own specific advantages.

I have tried to cover the basics of getting started in working with spreadsheets, but I have still only scratched the surface of the wealth of information within the manual supplied with Multiplan. A walk through in the first half provides a very good introduction, and the second half documents each command and function in detail. There are also a number of good books available on Multiplan, and the software is the same on nearly every machine.

APPENDIX A COMMANDS

ALPHA	NAME
BLANK	OPTIONS
COPY	PRINT
COPY DOWN	PRINT FILE
COPY FROM	PRINT MARGINS
COPY RIGHT	PRINT OPTIONS
DELETE	PRINT PRINTER
DELETE COLUMN	QUIT
DELETE ROW	SORT
EDIT	TRANSFER
FORMAT	TRANSFER CLEAR
FORMAT CELLS	TRANSFER DELETE
FORMAT DEFAULT	TRANSFER LOAD
FORMAT DEFAULT CELLS	TRANSFER OPTIONS
FORMAT DEFAULT WIDTH	TRANSFER RENAME
FORMAT OPTIONS	TRANSFER SAVE
FORMAT WIDTH	VALUE
GOTO	WINDOW
GOTO NAME	WINDOW BORDER
GOTO ROW-COL	WINDOW CLOSE
GOTO WINDOW	WINDOW LINK
HELP	WINDOW SPLIT
INSERT	WINDOW SPLIT HORIZONTAL
INSERT COLUMN	WINDOW SPLIT TITLES
INSERT ROW	WINDOW SPLIT VERTICAL
LOCK	eEXTERNAL
LOCK CELLS	eEXTERNAL COPY
LOCK FORMULAS	eEXTERNAL LIST
MOVE	eEXTERNAL USE
MOVE COLUMN	
MOVE ROW	

 APPENDIX B KEY FUNCTIONS

- FCTN-E
- FCTN-X CURSOR
- FCTN-S SCROLL
- FCTN-D
- CTRL-E
- CTRL-X PAGE
- CTRL-S SCROLL
- CTRL-D
- CTRL-W (CTRL-6) NEXT WINDOW
- CTRL-3 (CTRL-F) NEXT UNLOCKED CELL
- CTRL-1 (CTRL-Q) HOME
- CTRL-Z (FCTN-1) LOWER RIGHT
- <SPACE> MENU RESPONSE TAB
- FCTN-9 (CTRL-H) BACKSPACE
- CTRL-A (CTRL-2, TAB CTRL-I)
- CTRL-C (CTRL-) CANCEL
- FCTN-4 PRINT CANCEL
- <ENTER> ENTER RESPONSE
- FCTN-I (FCTN-7) HELP
- FCTN-8 RECALC
- , +, 0-9 VALUE RESPONSE
- FCTN-9 (CTRL-H) BACKSPACE
- FCTN-0 (CTRL-Y) CHARACTER DELETE
- CTRL-4 (CTRL-L) CHARACTER FORWARD
- FCTN-4 (CTRL-K) CHARACTER BACK
- CTRL-5 (CTRL-P) WORD FORWARD
- FCTN-5 (CTRL-O) WORD BACK
- CTRL-7 REFERENCE

 APPENDIX C MATHEMATICAL FUNCTIONS

- | | |
|------------------------|--------------------------|
| ABS absolute value | MAX maximum value |
| AND logical and | MID select from text |
| ATAN inverse tangent | MIN minimum value |
| AVERAGE average | MOD modular division |
| COLUMN present column | NA not available |
| COS cosine | NOT logical inverse |
| COUNT number of values | NPV net present value |
| DOLLAR conv. to dollar | OR logical or |
| EXP power of e | PI 3.1415926... |
| FALSE logical false | REPT repeat text |
| FIXED fixed decimal | ROUND round value |
| IF decision operator | ROW present row |
| INDEX select cell | SIGN sign of number |
| INT integer function | SIN sine |
| ISERROR error check | SQRT square root |
| ISNA not available chk | STDEV standard deviation |
| LEN length of text | SUM sum of values |
| LN natural logarithm | TAN tangent |
| LOG10 log to base 10 | TRUE logical true value |
| LOOKUP search for val. | VALUE text to number |
- *****

Multiplan Tips and Techniques

by Steve Zimmerman, U.S.A.

Many thanks to those of you who left questions for me on the TI SIG and SCCG boards! I will try to answer some of them here this month.

One of the most useful features of Multiplan (and one that I miss on my Tandy 200 portable!) is WINDOWS! Windows are a BIG THING now, just read the magazine reviews and ads for programs such as Framework and Symphony. Windows enable you to see two parts of the screen, in different areas of the worksheet, at the same time. Sounds simple enough, but what good is it? Well, let us say that you have a worksheet with about 60 rows of labels in Column 1, and that you enter data for each day in each month in columns 2 to 32. Since your basic column width shows only 4 columns, you can see the labels only while entering the first 3 days of data. If you make the columns narrower, of course, you can see more, if the numbers are not too long! If the numbers are too long, and you make the columns narrow, you just see "#####", an error symbol which means that you have too many numbers to display in that column width. To make it easier to enter data in the correct columns past column 4, use windows.

To do this, position your cursor (the cell pointer) in RIC2 (one column to the right of your labels), and hit the W key (for Windows). On the next menu, you want to key S (for Split). On the next menu, key V for vertical, and Multiplan will respond with "at column 2", Linked yes no, with the NO highlighted. Use the Tab (Ctrl 2) to move the command cursor from the 2 in "at column" to the Linked field, key Y to link the windows, and then key Enter. With practice, this command sequence becomes W S V Tab Y Enter. You now have a window, linked vertically. As you move the cell pointer down in the active window (the one you just created), the labels move along. As you move the cell pointer to the right, columns will disappear on the left side of the second window, but the labels will remain in view.

To 'uncreate' this window, key W (for Window) C (for Close); Multiplan will respond with #2, key Enter, and the window closes.

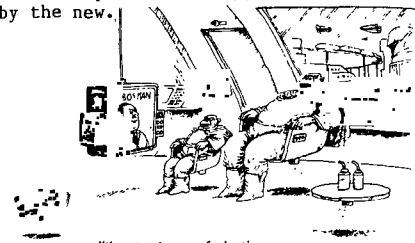
You can do the same thing with horizontal windows, if you have labels across the top and you want to see them as you move up and down. One thing you cannot do, however, is to link windows which intersect in the same cells. RIC1 cannot be in 2 linked windows, one for RI (horizontal) and one for CI (vertical).

You can move the cell pointer between windows by using CTRL(6) (change window), which makes a different window the active window. CTRL(6) again will take you back to the original active window.

I hope that this 'raises the shades' for those of you with windowing questions! To cover some other questions, I do not know of any way to load the recalculate routines when you start out in Multiplan. This will, of course, cause those with only one drive to have to swap disks, starting with the Multiplan disk, then your data disk, and then back to the Multiplan disk to load the recalculate routines. The program will also have to access the system disk to set up Names, to do external copies, to show you your disk directory, and to load the routines for logical operators. It appears that these are handled by some type of overlay, which is loaded from the system disk only when needed. I do not know for sure, but it is possible that one or more of these routines uses the same (or overlapping) addresses in memory. Oh yes, one more thing which requires access to the Multiplan system disk is the Help file. This can be quite handy, though! To get Help, just move your command line cursor to the command you need Help with, and key FCTN(?). To get Help with Help, move the command cursor to Help and hit Enter.

One more handy hint on recalculation; to recalculate a single cell, move your cell pointer to that cell, key E (for Edit), DON'T CHANGE ANYTHING, and hit Enter. This will cause the contents of THAT CELL ONLY to be recalculated. Of course, if you left recal turned on...

Other handy hints—to Un-Name Names, enter the Name, and delete the reference in the "to refer to" field. This deletes the name by making it refer to nothing. To delete an external reference or link, the same procedure is used. You enter X (for external), C (for Copy), from sheet (sheetname), Name (enter the name), to: (delete anything in this field), Linked (yes), Enter. This redefines the external copy link to refer to nothing, and the value placed in the cell by the external Copy command will disappear, and the cell will now be unlocked. To change the target of an external copy, use the same procedure, but specify the new cell or cells in the "to:" field. Since each cell or range on the supporting sheet can have only one target (on the dependent sheet), the old link will be replaced by the new.



It's nature's way of adapting, son. Now that computers do all our brainwork we don't need such big heads.

Multiplan Application

The making of an Index

by Geoff Trotter

As I was thinking about the TND and the problems of what to put in the first issue of a new year, it seemed that an index of the previous year's effort would be very useful. This would add to the one produced last year by Brian Graham and provide a way of making it an ongoing affair. Done in this way, year by year, it would not be too daunting a task. That was the start of many hours of work.

I am not a data base person, but I have used Multiplan a bit, and I find that it caters for my data base needs. Personal Record Keeping always seems too inflexible to me, and my brief attempt to do something with Data Base Manager did not get too far. I have done quite a few things like student records with Multiplan, and although slow at times, it does a good job within its limitations, and is relatively easy to use. What is that I hear you say? Well at least the menu of things you can do at any time is always in front of you. I can not help it if you do not understand what all those choices are. Perhaps I can help however, if I explain how I used Multiplan to create the index for 1987, and then obtained the two indexes sorted by Title and by Author.

First, consider the problem. What I want to do is to go through each magazine in turn and type up its index. That is I want to type in the Title of an article, the Author and the volume number - page number of the article. I decided to put in a description as well, as some regular features could do with a description qualifying the title. For example, Link-it #14 is not a very complete description, or letter to the Editor, and so on. So there are 4 things to be filled in for each article. Hold on folks, sometimes there is no Author as such shown, but the magazine or organisation is given. This is catered for with a fifth entry. The nice thing about Multiplan is that all these decisions could be made after starting to enter the data as the need arises.

Multiplan is a spreadsheet program, which means it holds the data in rows and columns. It has a maximum of 255 rows and 63 columns, so I used 5 columns to hold the Title, Description, Author, Page number and Source of an article. Then each article was on a row, and Multiplan allows the rows to be sorted, based on the contents of one column. All the data for each article could be entered in any order, and then sorted according to the contents of one of the columns. It is also easy to copy cells from one to another, or to groups of cells, which can speed up the entry of numbers of similar items. In fact the index of one month can be the basis of the next month's index. Anyway enough of these drivelling generalizations, on with the actual facts of this case.

Put in the Multiplan cartridge and get its loading message on the screen. At this point you can press space twice to change the screen colours. I like black on green, so I stop there, but you can cycle around the combinations and stop at one you like. When happy, press <ENTER> and Multiplan loads from a disk called TIMP in whichever drive you have put it. After a short wait, a display of 18 rows and 4 columns appears on the screen with a selection menu beneath it. There is a reverse video bar in the top left corner of the spreadsheet and also over the first item in the menu 'alpha'. The bar on the spreadsheet can be moved about with the arrow keys, FCTN giving one cell motion, and CTRL giving screen motion. The menu bar can be moved with the space bar or tab (CTRL(A) or CTRL(2)). Pressing <ENTER> causes the selected menu item to be active. It is quicker to press the key of the first letter of the menu item.

Example: The first thing I do after loading Multiplan is to press O for options menu. Then press N to set recalculate to no and press <ENTER>. Then I press T for transfer and O for transfer options. Then I press CTRL(A) to step over the first item in that menu and type in DSK2 in the setup field as I have two disk drives and store my data on disk 2. Then I am ready to start to enter data or read in some data from a file already on disk. Let us assume I am starting the entry of data at the start of the index.

I am only going to enter text or alpha data as the

page number will be of the form V6.4.16 for Volume 6 (1987) number 4 (May) page 16. A number cannot have two .s or any other delimiter so all data will be text. Sometimes you have to be careful as Multiplan is waiting for either text or numbers and makes its decision on which it is by whether the first character is a number or not. This is another reason for using a V as the first character of the page number. First you will notice a message "enter a command" when the menu is present. To start then, press A and the menu disappears and the prompt ALPHA appears. The first thing to enter is the heading of the first column in row 1 column 1. This will be Title, but if the first character is a space then it will stay at the top of the column when the whole column is sorted. So type in 'Title'. If <ENTER> is then pressed we return to the menu and what was typed in goes into the box. However if right arrow is pressed (FCTN(D)), the highlight box moves one cell to the right and the prompt ALPHA/VALUE appears so that it is ready for the next item of data. Type in 'Description' but be careful to pause after the first character to give it time to decide that it is ALPHA you want. You only have to pause after the first character when the ALPHA/VALUE prompt shows. You can avoid this by pressing <ENTER> twice to select ALPHA, but this is slower anyway. Using the arrow keys to go across and back and down the spreadsheet, slowly the data is entered. To take a break, or just for safety's sake, save the data periodically. From the menu type T followed by S for transfer save, and then type in the filename, without the DSKn which has already been setup with the transfer option command. The file name will be already there if it has been loaded or saved during this session. Just press <ENTER> then and wait. If the file has been already used for Multiplan then you will have to confirm that you want to destroy the existing data. However, if the file exists but is not a Multiplan data file, it will be destroyed without any confirmation.

One of the most annoying things about data entry in Multiplan is to get used to making corrections with FCTN(O), FCTN(9), FCTN(4) and CTRL(4). The arrow keys are used for other things, so they cannot be used for moving the cursor through the data. It takes some getting used to and then causes uncertainty when using TI-Writer for example.

Data entry is quicker if all the cells are using the defaults, which means all the data does not appear on the screen, but we can fix that after we have everything in there. About the end of Number 6 there was only 17% memory left and 175 rows entered. Time to start another file, but how are we to sort over two files? Things are starting to get interesting at last.

So after all the data is entered there are two files each of which almost fills Multiplan's buffer, one with 175 rows and the other with 157 rows. Each file could be sorted on Titles and on Authors, although since I entered authors with first names first, they were sorted by first name. Also with the move commands of Multiplan, it is easy to change the order of the columns and rows. To get the whole index in alphabetical order of the titles the following steps are followed.

First sort the two files separately. It takes about 2 minutes to sort each column with the amount of data present. Then a halfway point must be chosen so that the first half of the two files can fit into Multiplan so they can be merged and sorted. The idea is to end up with two files containing the two halves of the sorted index of the whole year's index. Once the division point is found, say on the word Program, then all titles which start with a word before Program will be in the first file and those after and including Program will be in the second file. To get data from one file to another requires the use of eXternal copy command. To use this, the parts you wish to move to another file have to be named. Press N on the menu selection, when you will be asked to type in a name. Enter a suitable name and press CTRL(A) and then define the area in rows and columns which you want to access. While you are in each file, define a name for both halves of the data. Then enter a new file and press X to get to the eXternal copy menu. Here you must enter the file name, CTRL(A), the area name, CTRL(A), the cell where you want the area to start, CTRL(A) and N for no to linking.

continued on page 8

TISHUG NEWS DIGEST

continued from page 1

The event of the month was the AGM. Hopefully there will be a report from the directors about the AGM, but I left it much happier than from the one last year. It still seems that I have trouble expressing myself so that all of you can understand what I am worrying about. That is probably why I have been sentenced to a year as the Editor, to get some practice. What I am worried about, is the falling membership of TISHUG. Of course it has not fallen yet because everyone is paid up to April, but it worries me that the secretary believes that numbers will fall. Every member will receive at least 2 more TNDs, before they make up their mind whether to rejoin or not. For members outside the Sydney area TND may be the only thing they get out of their membership, and so it may be what determines whether they rejoin or not. We of the Illawarra Regional Group, who have had responsibility for the TND since March 1987, think that the changes to the format of the TND have made it a better publication and better value for money. It certainly has been cheaper to produce, about half price, and I hope you all agree with the Co-ordinator that the quality has risen at the same time. I think Bob Montgomery and crew deserved an honourable mention at the AGM, but all they managed was a standing ovation (well almost anyway).

We have plans to raise the quality of the publication (did you notice the headings and Wade Bowmer's article?), but this will require the expenditure of a bit of money (yes, I will send in a budget Chris). We would also like to raise the quality of the contents and this is where you can help by offering your time to type in articles of interest and sending letters to the editor to say what interests you. If you do not try to do something, then you should not blame us if TND does not match your needs. At least continue your membership and give us a chance by writing to us with your reasons for considering giving up. It cannot be the fault of the 99/4A as it goes from strength to strength, and has such a variety of abilities that it can keep so many interested for so long in unravelling its mysteries. In fact it was a very well and cleverly designed computer that it stands up well even today against the currently available home computers. The fascinating thing is that it is a world wide phenomenon, this interest in an orphan.

Speaking of the AGM, did you notice that our current assets increased by 5K over the year? If that keeps up we should have 50K in five years.

Back to my original point I was trying to make: is the reason people leave TISHUG only because they sell out, or are they also concerned about value for their money?

Some of the articles from magazines received recently include the following. From Chug-a-lug (Canberra) September 1987 issue there is an article on adding an 8K RAM in the DSR address space currently used by device DSRs. This article is written by Jason Winter.

Anyone with a Corcomp RS232/PIO card should check the September 1987 issue of Northern NJ users group (New Jersey). In the December issue there is help on loading Funnelweb onto RAMdisk and an article on a Multiplan application. January 1988 contains a few articles by Jim Peterson about debugging and programs that write programs.

From the August 1986 issue of the Lenigh 99er computer group (Pennsylvania), there is an article on using buffers for beginners at assembler, a good review of TI99-opoly (cheers for Ross), and an article, reprinted from Lima 99/4 users (Ohio), on the various types of disk files which can contain a program, and how it can be loaded. There is also a DM1000 revision record up to version 3.5. The November 1987 issue has an article on file processing, and a reprint of a Jim Peterson article from TISHUG! The January 1988 issue has some reviews of software for the Geneve and Creative Filing System of Mark Beck, as well as a one line disk catalog program.

The October 1987 copy of the Ottawa group has an interesting article on prescanning in Extended BASIC. The September issue of the same group has an article on word processing on the Geneve. It also has a tutorial on Extended BASIC, which looks at various ways to clear

the screen. It gives a routine for a screen dump in Extended BASIC using the Corcomp disk controller, which should work with Peter Schubert's AT controller. There is a keyboard bus table, which shows the character codes returned and which wires in the cable are joined by particular key presses. There is an article on sprites and graphics using Forth. This gives some bug fixes: screen 58 line 10 should read
VDPMDE @ 4 < IF SMTN 80 0 VFILL 300 ' SATR ! ENDIF
Screens 53, 54 and 55 line 1 should end with SETVDPZ.
Screen 59 line 9 change the OOFF to OOFFE. Finally the manual needs a change in chapter 6 page 10 first line should read [HEX 3800 ' SATR !]. There is also an article on editing and debugging in BASIC. In December there is a Geneve update, a review of A-MAZ-ING, Expansion port interfacing part 2, Extended BASIC tutorial and BASIC tutorial. The Extended BASIC and BASIC tutorials continue in January, and there is a review of The Printer's Apprentice, a desktop publishing program written in Forth by Mike and Cynthia McCann. It looks quite good.

The Tacoma users group newsletter (Washington) has a very good Data base programs review in August. This looks at 8 different data base programs and compares them all. A good read if you are just about to start using one. The September issue has an article about using the ImageWise Video digitizer with the TI99/4A. This should help anyone who wants to get pictures into their computer. In December there are articles on removing the alpha lock problem with joysticks, and the program to change the screen colour "permanently". There is also a review of The Printer's Apprentice versus Fontwriter II. In January there is the correct expression for the colour changes and an article by Chris Bobbitt on the IBM PC as a failure.

In the West JAX 99er news (Florida) December 1986, there are articles on PILOT 99, debugging LOGO programs and simple GPL routines for assembler programmers.

The Northwest Ohio 99er News is the newsletter for the OH-MI-TI and New Horizons groups. The October 1986 issue contains an article on hidden characters, and a table of disk drive specifications for 21 manufacturers. It also has a list of error codes which could be quite useful. The December 1986 issue has a reprint from TND of an article by D.N. Harris on setting the printer. There is an article on installing the speech synthesizer inside the console and articles on printer control codes using TI-Writer.

December 1987 issue of Spirit of 99 from Central Ohio has some 'growls' from Jim Peterson about piracy and BBS's. It also has a TI-Writer tutorial session which continues in the January issue. There is also an application for Multiplan, a joystick adapter to build, and a review of a 'Gramulator', a Gram Kracker equivalent. If you may be interested in such a device write to:

Mark Van Coppenole
52 Audobon Road
Haverhill, MA 01830
(617) 372-0336

It is supposed to be cheaper than the Kracker, and yet have all its features.

R/D Computing from Haliburton (Ontario, Canada) has some interesting reading in its January 1987 issue on the Geneve and other commercial products.

Rocky Mountain 99ers (Colorado) December 1987 issue has a constitution, computer comparison chart, and how to remove the background noise in the TV by adjusting the modulator. There is also the table of error codes.

TI-SIG comes from San Diego (California), and in the November 1987 issue has a c99 floating point demo program. There is mention of Peter Schubert's AT Mini expansion system but the contact is at the old TISHUG postal address. The December issue is mainly on testing hardware, while the January issue adds to and corrects the c99 floating point demo program of the November issue. There is some advantage in getting these late! It also has a fix for DM1000's habit of putting 16 sectors/track on the header of an initialized disk, although it formats 18 sectors/track. In the first sector of the MGRI file at byte 216 (or there about) you should see (in hex):

10 00 02 D0 00 5A

Change the 10 to 12 and write this back to the disk.

continued on page 23

The Edmonton 99er Computer users' society (Alberta, Canada) has an interesting chart showing how you can tell who made a floppy disk in their December 1987 issue. In the January issue there are articles on a LOAD interrupt switch and a RESET switch. I am not sure I can recommend the approach used.

Bug Bytes, the newsletter of TIBug of Brisbane December 1987 issue contains information about the TI-99/4A expo they are running on 21st May. I will mention more details later, but they suggest that everyone post a copy of their advertisement where ever they are allowed to in an attempt to reach owners of TI99/4As who are not in touch with users groups. Attention, directors, that seems like a good idea, does it not? Apart from that there is an article on algorithm design and other articles copied from MICROpendium, TND and other places.

The Hunter Valley 99'ers News of December 1987 contains a wealth of information. There is a column called in the news which gives details on local and international news. Briefly summarized there is mention of Funnelweb V4.0, arrival of masses of technical data, MAXIMEM with 56K of RAM and 16K of ROM from Canada, 64K cartridges from Databiotics, Pro Type cartridge, The Printer's Apprentice software, OMEGA terminal emulator program, Grand RAM (advertised in BUG Bytes) and a new version of Mass Transfer. I hope that whets your appetite! There is a column of software tips for XB and another about timing XB code. There are some hints for Hitchhiker's guide to the galaxy with a map of the Heart of Gold. Printing graphics using the formatter is the subject of another article and there is also a review of Funnelweb V4.0. Those interested in Forth should read the Forth column, which talks about the new NOVIX Forth processor, and a symposium on it in Sydney in May. The article has a program for menu selection with reverse video. A hardware article shows memory expansion using a card made for an ETI project (eti687), and there is another showing how to add a reset switch. All in all an excellent magazine for information.

December 1987 issue of ATICC from Adelaide is rather thick but contains mainly articles copied from MICROpendium and TND. There is an article on recovering a floppy which has had something spilled on it, another on sprites and a third on installing 2 slimline disk drives in a PEB.

TIMES from Melbourne came on fanfold paper. Perhaps the photocopier was not working, but at least everything must have been typed in or come off a disk. They have been working on a light pen and there is a construction article and some software to use it. There is a table of CALL PEEK and CALL LOAD addresses. For those into Lotto, there is a program to help your choice, but no guarantees are provided.

Finally there is MICROpendium. In the November issue I noticed a letter which gave some fixes for the OLD to work with Minimemory and the line by line assembler. The c99 column talks about loops and arrays and the Forth column has Quicksort for strings. There is a review of a music preprocessor which sounds great (both the review and the music apparently). There is a hint for using the EDGE character in Extended BASIC, and a hint on how to recalculate just one cell in Multiplan (edit the cell; seen that one somewhere else). There is notification of a bug in c99 as filenames are truncated to 6 characters in an #include. How to install a RESET switch is also presented, along with a program for cataloguing a disk for only specified categories of files. In the December issue, there is an article on DEFS in BASIC, functions and strings in c99 and conversion to D/V80 from Forth. There is an interesting article on how to get Music Maker files from cassette to disk. You need a RAM disk but at least it can be done. At last TISHUG gets a mention, but as TISHUG, in the user group additions. There is a label printing program and instructions for putting your Extended BASIC cartridge into the console (with a switch). As you can see it took a bit of catching up with the mail accumulated over Christmas.

I received two letters from Shane Anderson this month. The first was about a contact he has made with GARY TAYLOR, President of the Pittsburgh User's Group and Sysop of the Harvester BBS. He is interested in an

exchange of Newsletters and software, so if you want to write to him the address is 1694 Dorman Avenue, Pittsburgh, PA 15216. The second letter was in praise of the February issue of TND. Thank you Shane from all of us here for your thoughtfulness.

I would like to encourage everyone to start writing letters to me, about any topic that you are interested in. I will either raise your point in this column, or print your letter in full (corrected if required). You can either use pen and paper, or do it electronically with cassette, disk or BBS. You can use a word processor or just REM (!) statements in BASIC. We have the technology to handle them all! The part we cannot do for you is to write it for you. The address to use is the registered address of TISHUG for postal mail, EDITOR for BBS users, and the telephone for quick replies. If you send something and it does not get a response, give me a call.

This month we are presenting a number of articles on Multiplan. I hope you find them interesting and useful. Once you know how to use a program like Multiplan, all sorts of uses appear, like the one I saw when thinking of constructing last year's index.

We had some problems at the printers last month which led to the mailing being a bit late. This month we are a bit late getting to the printer, but hopefully the mailing will be a bit earlier than last month. I had a breakdown of my ancient vehicle on the vital weekend when we were to do the layout, and had to spend 6 hours changing over the rear end, as well as 7 hours driving to Bathurst and back to pick up a child. Murphy was certainly about that weekend!

RAMcard Tip

Horizon RAMdisk up to 1 Megabyte!!!

from Ross Mudie 16/2/88.

Details have just been received which show how a Horizon Ram Disk can be set up with 1Meg of memory. The same PCB is used but the 32K HUG chips, 62256LP12 are used in place of the previous RAM chips.

The RAM disk size may be built up as follows:

Quantity of 62256LP12 for size:-

3 for 96k
6 for 192k
12 for 384k
16 for 512k
32 for 1MEG

The memory test program for the 1Meg RAMdisk has not yet been received, but is expected soon. Full details will be available from the TISHUG library.



"The headmaster's very keen on new technology, but he still believes in the good old fashioned virtues of outdoor sport."