(042
8902

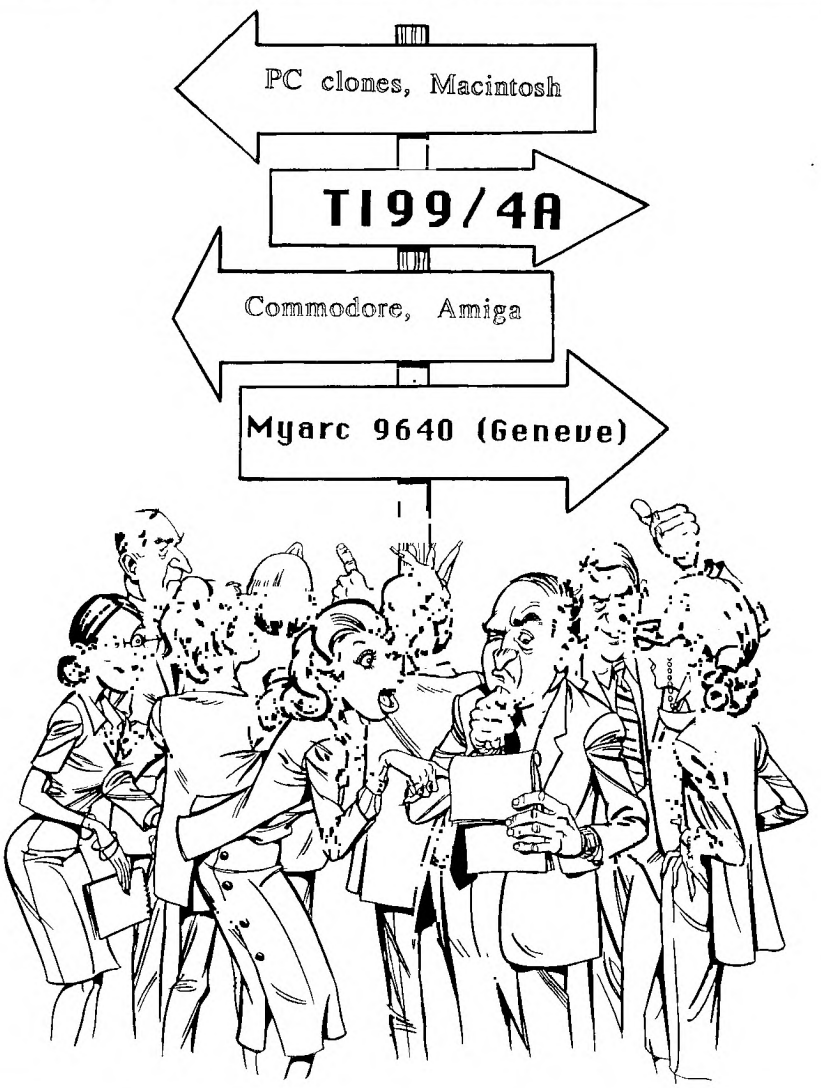
TISHUG

NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 8, Number 2 February, 1989

Registered by Australia Post - Publication No. NBH5933



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

TIsHUG (Australia) Ltd.

TIsHUG News Digest

February 1989

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board

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

Sub-committees

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

Regional Group Contacts

Carlingford
Chris Buttner (02) 871 7753
Central Coast
Russell Welham (043) 92 4000
Coffs Harbour
Kevin Cox (066) 53 2649
Glebe
Mike Slattery (02) 692 0559
Illawarra
Bob Montgomery (042) 28 6463
Liverpool
Larry Saunders (02) 644 7377
Northern Suburbs
Dennis Norman (02) 452 3920
Sutherland
Peter Young (02) 528 8775

Membership and Subscriptions

Joining fee	$5.00
Annual Family Dues	$25.00
Overseas Airmail Dues	AUS$50.00 or £22.00 or US$45.00
Publications Library	$5.00
Texpac BBS	$5.00

TIsHUG Sydney Meeting

The next meeting will be the AGM at 1 pm on 4th of February at Burwood RSL Club Limited, 96 Shaftsbury Road, Burwood.

Printed by
The University of Wollongong
Printery

TIsHUG News Digest ISSN 0819-1984

Index

They're off

by Geoff Trott

I hope you have all had a happy Christmas and look forward to a prosperous New Year. It has certainly been nice for me to have a bit more time before the next deadline and, as this is my last batch of words for a while, I may make more sense than usual (I hope). It was very pleasing to receive so many compliments at the Christmas Party from so many. However, I am not going to be here for the first part of the year so someone else must take over. I have also not been working on my own but have been aided and abetted by Rolf Schreiber. In fact Rolf has been doing the layout for two years, first with Bob Montgomery and then with me. If you like the layout of the articles then Rolf is to blame for that, with his skill in deciding how everything will fit together. He also coped with a serious illness earlier this year and timed it so that he did not miss an issue. The way we have done the magazine requires two people to do it; there are not enough hours in the nights to allow one to do it alone to the standard we have tried to keep up. So although I seem to have received all the praise, Rolf deserves at least half of it.

continued on page 42

Secretary's Notebook

by Terry Phillips

Well another year has come and gone and I would personally like to wish all members a very happy Christmas and all the best for the year ahead.

The December meeting was well attended and most members seemed to be enjoying themselves. Thankfully the weather was fine and we could all gather round the barbeque for some social chat.

The software imported from Inscebot proved tremendously popular with all items selling out on the day. A further order has been placed and new stocks should be available soon. Members who purchased version 1.02 of TI-Base and did not register with me, should let me know of their purchase quickly so that an updated version 2.0 can be provided to them as soon as received.

Due to the popularity of the Inscebot range, I would be pleased to receive feedback on what other software members would be prepared to purchase if import arrangements can be made. Let me know please.

A big welcome to new member Peter White from Regentville. Peter is on this BBS, username KEN. Peter is a milkman by trade and would like to hear from others working in this industry. He is currently working on a program to assist him with his work. Any "milkos" out there please get in touch with Peter.

Welcome back also to Dr David Murphy of St Ives who has been overseas for 12 months. David is also on this BBS and can be contacted through username ANYTIME.

Here are some excerpts from an item written by Tom Arnold which appears in the December edition of TI Focus, the newsletter of the Hamilton, Ontario User Group concerning the recently held Chigaco Faire.

"My wife and 2 daughters set off for Chigaco on Friday very early in the morning. The trip was uneventful, taking about 10 hours. Having reserved in advance we booked into the Holiday Inn, where the Faire was being held. Moving to the hotel lobby I met a host of people considered leaders in the the TI99/4A world, included were Paul Charlton, Peter Hoddie, Warren Agee, Charles Earl, Chris Bobbitt, Tom Freeman, Jack Riley, Jim Horn, and Bud Mills.

"The highlight presentation of the day saw at least 200 people crowded into a room for Chris Bobbitt, Charles Earl and Ruth O'Neill to give a demonstration of PRESS. It was not ready for sale as Charles had been ill and not able to complete it. There were several copies of the manual there. Written by Ruth, the manual is quite impressive, certainly the nicest manual for a TI99/4A piece of software since TI left the market. It is enclosed in a 3 ring ring plasticised binder which in turn is kept in a plasticised holder or box. I would bet that 20% of the cost of the package is in the manual. Chris and Charles gave a demonstration of a beta version of PRESS. Many features did not work but I must say that it was impressive. You will be able to see on screen your text as it will be printed out. No more Formatter or formatting commands. Up to 256 columns, you may set multiple columns in your text. For example you could have 3 columns in the upper half of your page and then switch to four columns on the lower half. This will be the biggest boon to newsletter editors since glue! The program operates using pull down menus, I saw as many as 4 on screen at once. The program was in colour, many of the text indicators (bold for example) are shown as a different colour. There is a configure menu which will allow you to set all the parameters, it is compatible with all printers because you can tell it your printer codes. Many questions were asked and all answers were positive. I do not think Charles has missed anything. The program will even translate your formatter codes when you load a TI-Writer file. So if you have put the formatter into italics then when the document is put on the PRESS screen the text will be in italics. I find this amazing. Cost of PRESS is US$59.95.

"After this exciting demonstration I headed for the main hall and circled the room to see what was offering. Asgard Software was across from the entrance and had a huge crowd around their booth. They were taking orders for PRESS, selling OLIVER'S TWIST, an adventure, DINOSAURS, Artist pictures, CALENDAR MAKER 99, BEYOND VIDEO CHESS, QUICK RUN, an Extended BASIC fast loader, and many other fine software programs.

"B&D Computer Supplies had an excellent assortment of paper at very low prices as well as an assortment of computer related items.

"Bud Mills was there with his Horizon RAM Disks and P-Gram cards. Another Horizon type card was the Chicago RAMdisk, their booth was busy all day. Rave 99 also had a RAMdisk as well as their keyboards and synthesizer cards.

"C&G Drives had some very good deals on disk drives. The original TI99/4A drives were selling for $30, $60 with box and power supply. Half height drives were $45. They had one very interesting item, a mini hard drive, for $225.

"Competition Computer had most of the original TI99/4A software and hardware, most repackaged used equipment. Prices were not checked.

"Another busy booth was Genial Computerware manned by Peter Hoddie and Paul Charlton. They had First Base, Hypercopy (Geneve), Picture Transfer, Triad, Browse, Macflix and Graphic Expander.

"Great Lakes Software was there but the person running the booth was not very familiar with the products. Their much publicised new release was not there. Prices were excellent, the were selling Joy Paint, Certificate Maker and Joy Paint Pal all for $10 a piece.

"Myarc had a booth demonstrating their Hard Drive Controller. It looks and sounds impressive. Too bad it cost so much. This booth did not draw the crowds I expected it to. I guess the novelty of the Geneve has worn off. There were several Geneves for sale late in the day at several booths but all the Hard Drive Controllers were sold out very early.

"I found a surprise booth, a company I had never heard of. This was a software company called Comprodine, run by Roger Merritt, 1949 Evergreen Avenue, Fullerton, CA 92635. He had several very impressive printing utilities for sale. I bought JIFFY FLYER which creates really neat flyers quickly. I also bought FORM SHOP which allows you to create boxes, graphs, lines etc. in your DV80 files created in TI-Writer. This comes with its own editor by can be used with TI-Writer. They also sell PRINT-IT, JIFFY CARD, and PICTURE-IT, all of which I wished I could have bought.

"Hunter Electronics was selling all the Texaments products including TI-Base. I am sure we will see articles on the merits and comparison of TI-Base and First Base. They are quite different. Hunter had lots of hardware, Myarc RAM Cards seemed plentiful.

"Other booths included Ottawa Users, Fox Valley Users, Toronto Users, MICROpendium, National Myarc Users, Milwaukee Users, Will County Users, Cadd Electronics, Data Systems, Tigercub Software, Triton Products, St Louis Users, Chicago Users, Genie and more that I cannot remember. I spend from 11am to 6pm in the hall and really did not get to look at all there was to buy. Of course I ran out of money!"

My ratings for the show:
Location - Excellent
Demonstrations - Very Good
Hotel - Excellent
Exhibitors - Excellent
Organizers - Fantastic"

Well, that is what Tom thought of the Faire. I also received a bit of news from Jim Peterson on the same subject. Here is what he had to say.

"You asked for a report on the Chicago Faire. Since I was tied down to my own table and did not find time to make the rounds even once, I cannot tell you much. The Faire this year was held in one of the banquet rooms of a large motel, which made it much more convenient, and the room was also much less crowded than before. I did not hear any figures on attendance but it was probably not much less than last year. Most of the "big names" of the TI99/4A world were in attendance, although I did miss the boy wonder, Mike Dodd. Most of the usual vendors were there, but neither TAPE nor RYTE DATA, so the German products were not for sale - they do not seem to have found much of a market in this country.

"The big new product was expected to be PRESS, a very sophisticated word processor from Asgard. It was demonstrated, but was not yet quite ready for sale. The biggest new hardware product was the Myarc Hard and Floppy Disk Controller, but there was also much interest in the P-Gram Card from Bud Mills Services, a gram emulator with optional real time clock. The Gramulator, successor to Craig Miller's Gramkracker, was also being sold. According to the Lima UG newsletter, another little noticed but very important product was the Dijit AVPC card which connects to the monitor and, with the right software, gives an 80 Column text display with the same height resolution and colour spectrum as the Geneve."

News in the November issue of Melbourne Times advised that the Melbourne group have decided to run a world wide competition for the TI99/4A, software and hardware during 1989, with the winner to be announced in December. The rules as printed are:

1. The competition shall be named the TIMES TROPHY.
2. It will be open to all TI99/4A owners who are members of a recognised user group.
3. Each user group will be allowed one entry only, and it will be their responsibility to choose their entry.
4. The entry can be a program or a piece of hardware. It must be for the TI99/4A. If it works on the Geneve all well and good.
5. There is no entry fee. The trophies will be valued at about $100 each. The value on the prize is in the satisfaction of winning.
6. The competition is open to freeware, fairware, commercial ware programs or tutorials, it is also open to hardware modifications to the TI99/4A computer.
7. Programs must be submitted on disk so that they are readable in TI-Writer or its equivalent. Hardware must be submitted by detailed drawings and instructions for building the hardware, it must also be accompanied by details of what it set out to achieve.
8. Anything published before January the first 1989, is disqualified from entry.

Address for entry is Melbourne TI User Group Inc., 88 Main Street, Blackburn Vic. 3130.

No doubt, there will be more news on this competition early during 1989.

Letter to the Editor

In the TIsHUG News Digest we have probably the most appealing and informative publication offered to a TI99/4A user being printed today. It has extracts from many overseas TI99/4A clubs' newsletters, notably from USA and the UK. It caters for the novice through to the most advanced enthusiast.

However getting newsworthy items from members is part of a Directors responsibility. How? By aura of the authority of office, delegation and example. One Director's field should be to solicit news items via his fellow directors, their regional leaders and in turn their publicity officers and on down the line. At present we have a team turning out a worthy Digest but much more can be done by more people being involved.

The standard of this publication is worthy of additional funds to further its progress. Lapsed members may be influenced to return if made aware of the present quality of items being presented. Why not send them a copy?

Disk Library:

Here is an avenue that has terrific potential for expansion. Not limited to those who have a disk drive system but available to regional group leaders to print out programs for cassette users, download to tape or extract selected files at will! Group meetings are enhanced by the thought of participation in "copy days". Selectivity is the crux of programs on disk. We as members cover a wide variety of interests.

Publication Library:

Postal charges are hard to bear when $5 or more is spent on a pig in a poke. Each regional group should be allotted the task of screening and reporting on the worthwhile items of interest covering tape types, disk basic, Extended BASIC, assembly etc.

At present our group has started to dissect some publications but more will have to be done as keen and able members are found. Exchange between groups fosters good will among us all.

Bulletin Board:

We have a fine informative bulletin format that rivals its competitors for interest. More audience participation is anticipated next year. Country members would appreciate a reduction in "costs of running" by upgrading to 1200 baud and the ability to accept all modems, would be a help. Perhaps waiving the fee would help the "long distance calling" sufferers.

Congratulations to the retiring Directors for their results in 1988 and may their successors rest assured that they will have a lively membership in support of their acceptance as our leaders.

Regards to all and Good Directing.

John W Ryan (A country member.)

TIsHUG Shop with Stephen Carr

Hello Again. Now that the Christmas New Year period is over we can all now get back into the swing of using our computers and all the goodies that we have received from our loved ones. Enough of that and on to more important things like what is for sale at the moment.

The special for this month is 2 boxes of disks plus a storage box for $35. We still have Peter Shubert's ROS on an EPROM with installation instructions and set-up disk for $25.

Hopefully we should have the latest issue of MICROpendium Magazine plus some more of the software from the US as well, plus all of the regular items for sale. As the February Meeting will be at Burwood RSL Club, the shop will operate after the conclusion of the meeting.

A special thank you to all who supported the shop over the last 6 months, with a special thank you to my sister Jenny and Gerry for their help behind the counter.

I hope to seek a second term as Shopkeeper and hope that I have tried to do the best that I can for you, the members of the club.

In closing I would like to wish all members a safe and enjoyable 1989.

Your's in computing, Steven Carr.

TI-HUG Software Column by Terry Phillips

Here is a rundown on new disks added to the software library over the past few weeks.

Disk A288 - Some dumped Milliken educational modules. Titles are Percents, Equations and Maths Courseware. Also included on the disk is Logo Fun which will not work. If a speech synthesizer is connected it may have to be unplugged to get some of this software to run correctly. 349 sectors and requires Extended BASIC and 32K expansion.

Disk A289 - RLE pictures of movie stars and others. Contents are Flynn, Fonda, Gable, Garbo, Green Hornet, Madonna, Monroe, Mouseland, Neuman, Pickford and Sappho. RLE loader required. 352 Sectors.

Disk A290 - RLE pictures of various scenes etc. Contents Birthday Card, Eye Picture, Geisha, JMB Quote, Prehistoric scene, Rural scene, Taj Mahal and Train. Disk also contains an Extended BASIC imbedded version of the RLE loader. 398 Sectors.

Disk A291 - More dumped modules containing educational software. Titles are Bowling Math, Fractions and Frog Jump. Entertainment for the younger ones while they learn simple maths. Extended BASIC and 32K Expansion required. 317 Sectors.

Disk A292 - More of the same. This time we have Addition and Subtraction 3, Multiplication 2 and Star Maze. Extended BASIC and 32K Expansion required. 360 sectors.

Disk A293 - 339 sectors of text files of various information. Much of this will be placed on the bulletin board.

Disk A294 - An interesting disk which contains all the files needed to access the foreign language versions available through TI-Writer. You must have the TI-Writer module to use this disk. It is understood that files created using this version of TI-Writer cannot be read by the normal English version of Funnelweb. 251 sectors.

Disk A295 - Contains a banner program, a massive picture creating utility and a different train watchers program. Also contains the classic Bert and Ernie speech demonstration. Extended BASIC required with 32K for some of the programs. 334 sectors.

Disk A296 - There are a host of small home utility programs on this disk. Some of the titles include Addresses, Debts, Freezer, Gradebook, plus more. Most run in Extended BASIC . 357 sectors.

Disk A297 - The disk contains 4 programs only. The latest version of Graphic Designer, Print an ID and Print a Tag, as well as an English version of that different space invaders game. You know the one, I do not need to tell you! 121 sectors and Extended BASIC required for all programs.

Disk A298 - Rhapsody in Blue. Very heavy going music but quite well programmed. Extended BASIC with 32K expansion. 287 sectors.

Disk A299 - A disk containing various small utilities and a couple of games. OK but nothing outstanding. Extended BASIC required. 351 sectors.

Disk A300 - Tips from the Tigercub #29 through to #42. Great information from the master programmer, Jim Peterson. You will need a printer to get anything out of these as hard copy is essential. Most have appeared in back issues of the TND. 714 sectors.

Disk A301 - Ross Mudie has been hard at it converting of lot of memory image files to run out of Extended BASIC with the imbedded code. Titles on this disk are Artillerie, Astroblitz, Berzerk, Cannonball Run, Circus, Compu-Car, Connect 4, Desert Flight, ET at Sea, Angler Dangler, Flyer, Fraggles, Frogger, Henhouse, Mars Escape, Math Catcher, Mission X, Space Station Pheta, Space Wars, St Nick, Startrap, TI Mazogz and Topper. Extended BASIC and 32K expansion is required but all can be loaded from cassette provided there is some form of 32K expansion. 1129 sectors.

As you can see from this list, software continues to arrive, and while some of it may not be new or real wizbang stuff, it undoubtably would hold interest for some of the members.

There will be no shop facilities at the next meeting (February AGM) so obviously there will not be any disks released. If you want any of the above in the interim please let me know and I will be happy to send out the disks to you. If you want any other disks from the library, or a hard copy, disk copy of the library contents, let me know this also. o

continued from page 8

multiple character GCHAR command: GETSTR. The position of the new window in WNWSPEC$ is obtained in line 11130 and is moved to the beginning of WNWSPEC$ enclosed by square brackets in line 11140.

Information on the windows size and first array element is retrieved on 11150 and the window displayed on line 11160. The boundaries of the window are modified in the last line so as not to include the border.

The final two subprograms are used to lock and unlocked a window. This is simply done by replacing the character that represents whether the window is locked or not with the appropriate character.

That ends a rather quick tour of the routines for interactive windows. You may wish to improve their speed by storing the attributes in numeric arrays (calling windows by number rather than by name). You may also wish to add other subprograms such as hide window, delete window, move window, etc. to the list.

The only reference from this article in the XDP Utility Guide for this month is GETSTR (page 23). To understand this months listing, refer to the Extended Basic manual as well as the XDP Utility Guide and ask what each part of the program is doing as well as if there is a better way of doing it. Pay particular attention to the string manipulations (eg line 11140) as it appears many programmers are rusty on these.

For those who do not have a copy of the XDP disk, either make an arrangement with Terry for a copy or come to the Northern Suburbs Regional Group meeting (see back cover) where a copy can be made.

Next month: I will tie up some loose ends on programming with XDP and show some handy peeks and loads for use with the package. o

continued from page 27

If you need an answer and do not get a reply, please write again.

Oh yes, My address is: Barry Boone, Box 1233, Sand Springs, OK 74063.

Postscript for Myarc 9640 Users

If you are currently using version 3.0 or 3.01, these are very unreliable on Geneve. v3.02 has been "fixed" to be compatible with MDOS. My apologies for any problems you may have encountered, please understand that A) I do not have a Geneve, and B) if the Geneve were truly TI99/4A compatible, v3.0 and v3.01 would have worked. It is a bit difficult to debug around Geneve's quirks without actually owning one. o

The Communicators

Special Interest Group for users of the TEXPAC BBS
by Ross Mudie

The New Year of 1989 is here and the Bi-centennial is over, so here is hoping that 1989 will be a little less hectic than 1988. I want to thank the people who contributed to the BBS during 1988. Without their help the BBS could not have contained all the news and programs, some old and some new, that were available. If anyone wants repeats of any item that has been on the BBS since I started looking after the BBS in September 1986 then just ask as I have the old news and program disks available. I will need help again with material for the BBS, if you can contribute then please offer, do not wait to be asked.

The usage in the third quarter of 1988 dropped off to an alarmingly low level, but picked up again towards the end of the year. There were 3113 valid calls from 102 user names (including sub-editors and regional groups, with the BBS on line for 1229 hours, compared to 112 users, 5053 calls and 1583 hours in 1987. There are currently 89 financial BBS members.

During 1988 the BBS suffered several failures due to power mains interruptions. It is hoped that time will be available to set up a battery back-up system for the BBS. A small un-interruptable power supply (UPS) was tried back in 1987 but it too was found to be lacking in long term reliability. The BBS console had a TMS9901 chip fail in August and one particularly savage electrical storm managed to switch the BBS modem from answer to originate mode. (That was the same storm which a newspaper photographer photographed a massive lightning strike on Centrepoint tower). Restoration after such failures is usually next day because of the unattended operation of the BBS.

The TI disk controller in the BBS was changed to a Peter Schubert AT disk controller which provides double sided double density (DSDD) disk operation. Due to the number of programs possible on the system being 40, the programs are now changed on a 2 monthly cycle to allow users time to download. The time limit for calls into the BBS is now 60 minutes and users are asked to allow 30 minutes between successive calls in case someone else wants to use the system. The BBS has only one line thus only one person can be on at one time.

During December I received some rather critical comments from one BBS member with regard to the lack of interactiveness of the BBS compared to Viatel. It was suggested that the BBS should have a HELP LINE, a Risque Comment or joke page and a controversial page where stirrers can incite a riot (in the brain at least). My response to this is "GREAT"!! Is anyone prepared to run these areas? During 1988 I put in over 1000 hours for TIsHUG. The sub-editor facilities in the BBS provide a NON-SYSOP method for files to be placed in the BBS news files area and any user can be given sub-editor access, it is not limited to a privileged few. Contact SYSOP on the BBS for more information and sub-editor access, (this will also allow me to spend more time on the computerised model train set project).

The program upload/download facility was provided in the BBS after requests for the facility from members. There was a rush to use the system when it was first provided which exceeded the capabilities of the system, then usage fell away to almost nothing. The Program upload/download area can take up to 28 programs. The programs are saved on the mail disk which is a DSDD disk. The available space on the disk is dependent on the amount of mail in the system but it is usual for over 1000 sectors to be available for programs. The user who uploads a program can also delete a program.

Access to the BBS is only available to BBS members, to gain access contact the Secretary of TIsHUG or the SYSOP at the address or phone numbers on page 1 of this zine.

From the Bulletin Board

MAIL TO : ALL
MAIL FROM : SARA

* ACT NOW *** SALE *** FREE GIFT *

Micro-Educational Pty. Ltd., Unit 8, 235 Darby Street, Cooks Hill, NSW 2300 (Newcastle)

5.25" DSDD DISKS 85 cents, no limit.

3.5" DSDD DISKS $2, no limit.

These disks are normally $1 and $2.99 each. Offer closes Friday 23rd Dec 88.

FREE FREE FREE

DD100 Disk storage box with every 100 x 5.25" disks sold.

How to order:-
Phone (008)025 229, Local (049)26 4122
Mail: PO Box 160, The Junction, 2291
Call in at above Newcastle
FAX (049)26 2194, Viatel node 3#

All country members should be on their mailing list as goodies are there for asking.

(I have just received my free box and it leaves my YA-100BL and Chandler's special in the proverbial. Hence my sharing this offer with fellow members)

Kevin Cox put me on their list some months ago, little did I know I would be up for $85 plus $4 postage and handling so soon.

The disks are made in the USA, Wabash by name and have a lifetime Warranty plus 100% error free, Pillowed jacket liners. Meets ANSI, ECMA, IBM, ISO, all industry standards. 48 tracks/inch 35/40 tracks per side.

This is to certify that I have no connection with this company other than as a satisfied customer. I ordered by phone (plastic money) Thursday 1500 and postman delivered to the door Monday at 1400 hours, at Mullaway north of Coffs Harbour!

John Peschar is the manager and he tells me that they have 47,000 on their mailing list. Are you getting your share?

Merry Christmas to you all from my wife (SARA) and me.

Sincerely John Ryan. 881219.

MAIL TO : ALL
MAIL FROM : DOBELL

I would like to start a special interest group next year around TI-Artist. I need the services of a resident expert on TI-Artist who would make himself available over a period of time to show us how best to use this great program. Time and places etc., can be arranged to suit the person. I can be contacted via the BBS or at home Phone 319-2577 Les.

MAIL TO : ALL
MAIL FROM : SMITTY

Received this add today from USA. I thought you might be interested

Technical Application Product Engineering (TAPE)
1439 Solano Place,
Ontario CALIFORNIA, CA 91764
Phone ..714 989-9906
Super Closing Down Sale
Extended BASIC 11+... US$23.95
994a Intern..... US$6.95
Mechatronics 80 column card... US$89
Mechatronics power supply US$10
Mechatronics EPROM programmer with software and power supply.... US$59.95
Mechatronics TI mouse/SW/ power supply. US$59.95
Quick disk for CC-40;TI-74;TI-95
Stock on hand only..
Add $(US)2.50 for US s/h.

Suggest add fair bit more for Air-Mail to Australia.

Looks good those prices do not you think. I do not know how long the offer lasts but looks good.

Regards Peter.

MAIL TO : ALL
MAIL FROM : DOBELL

I have once again a limited supply of A4 printer paper 2000/2500 sheets per carton tractor feed plain white $10 each. 699 5137 after 5pm DOBELL

For Sale

MAIL TO : ALL
MAIL FROM : GAMES

Hi everybody, I have a TI99/4A console for sale complete with both standard books, and has an adapter for both UHF and VHF television. It also comes with a tape recorder and the necessary cables. A joystick can be added easily with an adapter to fit Atari, Commodore etc. joysticks.

Together in the pack are 10 software packs. They are: Microsurgeon (cannot be converted) Video Games I, Mind Challengers, TI Invaders, Hunt the Wumpus, Parsec (old favourite), Touch Typing Tutor (rare and necessary) and 2 tapes: Ant Wars, Beginners Basic Tutor.

This is for sale not by a traitor getting out and moving to a bigger computer, this is an unused spare computer that has never been extensively used so is still in near perfect condition.

It would make an excellent beginners computer or with a small simple adaption can be boosted to 32K.

Get to the price you moron, I hear you ask. All this pack costs is.....$300

Please mail here on Texpac for : GAMES

MAIL TO : ALL
MAIL FROM : CHOCKY

==========
FOR SALE FOR SALE FOR SALE FOR SALE
==========

I have the following modules available.

(1) The Attack - shooting game from TI
(2) Fathom - explore the skies and the sea to find treasures and save the girl, also from TI.
(3) Mash - The game of the show. ie save men from the fields and do operations on them. From 20th Fox.
(4) Video Games I - 3 games in one. ie pinball, hunting etc., from TI.
(5) Maths Games IV - This is a baseball game, including maths ie to run you need to get an answer right etc.
(6) Parsec - The first and the best shooting game written for the TI99/4A, from TI.
(7) Amazing - You are the mouse, find the cheese and avoid the cat(s) at the same time. Excellent Game. From TI.

If you are interested in any of these modules please leave a message for me (CHOCKY) at this BBS or ring (02)86 4602

Please make an offer.

MAIL TO : ALL
MAIL FROM : PETESAKE

Goodies for sale.

Viatel + 300 modem card for PE box. Very reliable communications and can be used to send files, receive files at 1200 baud. Also has auto answer, auto disconnect. Run your own bbs, etc. Asking $250, only one.

Disk drive full size DSDD guaranteed good worker, only $80. Also have standalone power supply for drive or console (+5, +12, -5 volt) $25. Contact Peter on 358 5602 or via BBS.

MAIL TO : ALL
MAIL FROM : PAINEY

**** FOR SALE ****

I have a number of high power switch mode power supplies available which have regulated outputs of 8 volts dc at 10+ amps, +15 volts at 2.5 amps and -15 volts at 2.0 amps. These supplies are ideal as an alternate power source for the TI Peripheral expansion box, Mini PE system and with the addition of the appropriate 5v and 12 v regulators, can be used for external disk drives.
The supplies can run on both 240 vac or 110 vac and just need a power cord to be connected.
All supplies have been tested and are in good working order.
Price..... a steal at $20.00 each.... Originally sold at $250.00 Please contact John Paine
Home phone 6256318 (after 7.00pm , please)

MAIL TO : ALL
MAIL FROM : NERK

It hurts to have to say it, but I wish to sell my TI99/4A system which has been a good friend and useful ally, not to mention the camaraderie and knowledge membership of TIsHUG has brought my way.

System comprises: console with widget; expansion box with 32K memory, RS232 card, disk controller, single disk drive, power supply module and vhf modulator about 12 cartridges which include disk manager, Extended BASIC, Typing, various maths tutors. Software: TI-Writer, Multiplan, Paint and Print plus all manuals, plus disks with games, joysticks and adapter. Also a couple of good books about using the TI99/4A. A Cedata 300/300 1200/75 baud modem including cable and Fasterm, Mass Transfer software is included. Viatel software and manual. Please contact me on 633-9102 if you wish to make me an offer. Thanks.....NERK

MAIL TO : ALL
MAIL FROM : GAMES

At this moment Games Inc. would like to offer a incredible range of TIsHUG tapes for sale. It includes over 30 tapes.

It is a devoted member's collection of tapes way back from the end of '84 until the beginning of this year. Every tape of 85 and 86 are here and all but 2 of 1987, and a few of late 84. These tapes must be sold together. The cost for the 42 tapes is only $60 . These tapes brand new would cost well over $100.

Also for sale is a Tunnels Of Doom cartridge plus 2 tapes full of different scenarios so this is not a cartridge that one can get bored of. The price for the cartridge and 2 cassettes is only $45.

Please forward your questions about these goods to GAMES. Thankyou.

PS: Games would like to point out that these are virtually collectors items that will be desired among many.

MAIL TO : ALL
MAIL FROM : PETESAKE

I have some PE BOX CARDS I wish to sell. This is a good opportunity to upgrade your system.

#1: TripleTEC Card (USA) has a 64K printer buffer, real time clock and socket to mount Speech Synthesizer inside PE box. $150.
#2: 1200/75/300 Modem Card for PE box with Viatel, Auto-answer, Auto-disconnect. A very reliable modem. $250
#3: Horizon RAMdisk card with either 400K and EPROM ROS, or 512K and Horizon ROS (over 2000 sectors!). A very reliable RAMdisk, uses 32K chips, asking $460.
#4: TI RS232 card. Works well. $140
#5: A matching pair of National 360K slim-line disk drives (new). The best drive money can buy. Work reliably at 3ms H.S. $350 the pair.
#6: The above drives can be bought mounted in a highly customised PE box. It has 2x240V sockets at rear for monitor and printer switched from main switch, power outlets for extra disk drive and for powering console from PE box. High capacity switch mode power supply/transformer to run all this, and quieter fan. $650 with two disk drives.
#7: Special TI Console for above $100.

I am selling these items reluctantly because I am now using a MINI-PE system with twin 3.5 inch drives and 400K RAMdisk plus Clock.

If you are interested in any of the above items, or a special price on the lot give me a ring (Peter) on (02)358 5602. My address is PO Box 28, Kings Cross, 2011.

I now have brand-new 3 1/2 inch disk drives which I have mounted into the same size diecast box as the Mini-PE System. These are complete with power supply and interface cable and with the A.T. disk controller will store 180K (720 sectors) in SSDD format on standard 3 1/2 diskettes and can use the fastest Head-step setting of 3mS. Power consumption is only 300mA at 12 volt so these drives can be used on a portable system. Also included is a switch to select as drive number 1 or 3. Price of the complete disk drive box is $130 while they last.

Extended Display Package part 4

by Craig Sheehan

Having discussed the principles behind XDP's windows in the December TND we can now discuss a more concrete application. The program shown below contains a set of subprograms that enables interactive windows to be used. As an example, they have been used in a simple editor program, listed below. For the benefit of members with RSI, I will upload the program onto the BBS during February, for those members with RSI. Whether you down load it from the BBS, or type it in yourself (good luck!), do not forget to load the XDP package before running the program.

```
1 ! ######################
2 ! #   XDP MINI EDITOR   #
3 ! #   For Jan/Feb TND   #
4 ! #  By Craig Sheehan.  #
5 ! #                      #
6 ! #   XB, XDP must be   #
7 ! #    first loaded.     #
8 ! # Ver1.00    29/12/88 #
9 ! ######################
100 DIM WNW$(65),TEXT$(50):: MAXELM=65 :: NXTELM=0 ::
    MAXLINES=50 :: WNWSPEC$=""
110 CALL LINK("XDP",40):: CALL LINK("SCREEN",16,6)
120 RESTORE 1000
130 READ N$ :: IF N$="#" THEN 140 ELSE READ T,B,L,R ::
    CALL NEWWINDOW(WNWSPEC$,MAXELM,NXTELM,WNW$(),N$,
    T,B,L,R):: GOTO 130
140 CALL LINK("CHAR",232,"0000007C7C7C7C78000000FCFCFC
    FC00000000F8F8F8F878787878787878787878")
150 CALL LINK("CHAR",236,"78F8F8F8F800000000FCFCFCFC00
    0000787C7C7C7C0000007878787878787878")
160 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"LOAD"):: CALL
    LINK("DISPLY",2,2,"Load file name?.CR DSK1.")
170 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"SAVE"):: CALL
    LINK("DISPLY",2,2,"Save file name?.CR DSK1.")
180 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"PRINT"):: CALL
    LINK("DISPLY",2,2,"Device name:.CR PIO")
190 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"MENU"):: CALL
    LINK("DISPLY",2,2,"MINI EDITOR.CR 1.  Edit.CR 2.
    Load.CR 3.  Save.CR 4.  Print")
200 CALL LINK("DISPLY",7,3,"5.  Purge.CR 6.  Exit.CR(2)
    Option? _")
210 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"SURE"):: CALL
    LINK("DISPLY",2,2,"Are you sure? (Y/N)"):: CALL
    DISPLAYWNW(WNWSPEC$,WNW$(),"EDITOR")
220 CALL LOCK(WNWSPEC$,"LOAD"):: CALL
    LOCK(WNWSPEC$,"SAVE"):: CALL LOCK(WNWSPEC$,"MENU")
230 ROW,COLUMN,FIRSTL=1
500 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"MENU")
510 CALL LINK("ACCEPT",10,11,".BE.SI(-1).VL(123456)",
    OPT)
520 ON OPT+1 GOTO 510,530,540,550,560,570,600
530 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"EDITOR"):: CALL
    EDIT(TEXT$(),MAXLINES,ROW,COLUMN,FIRSTL):: GOTO 500
540 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"LOAD"):: CALL
    LOAD(TEXT$(),MAXLINES,WNWSPEC$,WNW$()):: GOTO 230
550 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"SAVE"):: CALL
    SAVE(TEXT$(),MAXLINES):: GOTO 500
560 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"PRINT"):: CALL
    PRINT(TEXT$(),MAXLINES):: GOTO 500
570 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"SURE"):: CALL
    YN(2,22,A$):: IF A$="N" THEN 500
580 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"EDITOR"):: CALL
    LINK("CLRW")
590 FOR X=1 TO MAXLINES :: TEXT$(X)="" :: NEXT X ::
    GOTO 230
600 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"SURE"):: CALL
    YN(2,22,A$):: IF A$="N" THEN 500 ELSE CALL
    LINK("NORMAL")
1000 DATA "EDITOR",1,24,1,40
1010 DATA "LOAD",6,11,8,26
1020 DATA "SAVE",6,11,8,26
1030 DATA "PRINT",6,12,8,31
1040 DATA "MENU",3,15,5,19
1050 DATA "SURE",14,18,8,32,"#"

10000 SUB EDIT(TEXT$(),MAXL,ROW,COLUMN,FIRSTL)
10010 AID=1 :: DOWNARROW=10 :: UPARROW=11 :: ENTER=13
    :: BACK=15
10020 CALL LINK("ACCEPT",ROW,1,
    ".SI(-38).CP(.VA).EK(1,10,11,13,15,.VA)",
    TEXT$(ROW+FIRSTL-1), COLUMN,EKP) 10030 IF EKP=BACK
    THEN SUBEXIT
10040 IF EKP=ENTER THEN COLUMN=1 :: EKP=DOWNARROW
10050 IF EKP=UPARROW THEN ROW=ROW-1 :: IF ROW=0 THEN
    ROW=1 :: IF FIRSTL>1 THEN CALL SHIFTTEXT(TEXT$(),
    FIRSTL,1,-1)
10060 IF EKP=DOWNARROW THEN ROW=ROW+1 :: IF ROW=23 THEN
    ROW=22 :: IF FIRSTL<MAXL-22 THEN CALL
    SHIFTTEXT(TEXT$(),FIRSTL,22,1)
10070 IF EKP=AID THEN COLUMN=COLUMN+5 :: IF COLUMN>38
    THEN COLUMN=38
10080 GOTO 10020 :: SUBEND

10090 SUB SHIFTTEXT(TEXT$(),FL,ROW,DIR):: FL=FL+DIR ::
    CALL LINK("SCROLL",-DIR,0):: CALL LINK("DISPLY",
    ROW, 1,TEXT$(FL+ROW-1)):: SUBEND

10100 SUB LOAD(TEXT$(),MAXL,WNWSPEC$,WNW$()):: CALL
    LINK("ACCEPT",3,2,".SI(-15).K3.CP(6).BE",FILE$)
10110 OPEN #1:FILE$,INPUT,DISPLAY,VARIABLE 80 :: LINE=1
10120 IF EOF(1)=0 AND LINE<=MAXL THEN LINPUT #1:I$ ::
    TEXT$(LINE)=SEG$(I$,1,38):: LINE=LINE+1 :: GOTO
    10120
10130 CLOSE #1 :: FOR LINE=LINE TO MAXL ::
    TEXT$(LINE)="" :: NEXT LINE
10140 CALL DISPLAYWNW(WNWSPEC$,WNW$(),"EDITOR"):: CALL
    LINK("CLRW"):: FOR R=1 TO 22 :: CALL LINK("DISPLY",
    R,1,TEXT$(R)):: NEXT R :: SUBEND

10200 SUB SAVE(TEXT$(),MAXL):: CALL LINK("ACCEPT",3,2,
    ".SI(-15).K3.CP(6).BE",FILE$):: LASTLINE=MAXL
10210 IF TEXT$(LASTLINE)="" AND LASTLINE>0 THEN
    LASTLINE=LASTLINE-1 :: GOTO 10210
10220 IF LASTLINE=0 THEN SUBEXIT
10230 OPEN #1:FILE$,OUTPUT,DISPLAY ,VARIABLE 80 :: FOR
    L=1 TO LASTLINE :: PRINT #1:TEXT$(L):: NEXT L ::
    CLOSE #1 :: SUBEND

10300 SUB PRINT(TEXT$(),MAXL):: CALL LINK("WINDOW",7,
    11,10,29):: CALL LINK("ACCEPT",3,1,
    ".SI(-40).K3.BE", FILE$):: LASTLINE=MAXL
10310 IF TEXT$(LASTLINE)="" AND LASTLINE>0 THEN
    LASTLINE=LASTLINE-1 :: GOTO 10310
10320 OPEN #1:FILE$,OUTPUT :: FOR L=1 TO LASTLINE ::
    PRINT #1:TEXT$(L):: NEXT L :: CLOSE #1 :: SUBEND

10900 SUB YN(R,C,X$):: CALL LINK("DISPLY",R,C,"N")::
    X$=""
10910 CALL LINK("DISPLY",R,C,"N")
10920 CALL LINK("ACCEPT",R,C,".SI(-1).VL(YN).K3.BE",
    X$) :: IF X$<>"Y" AND X$<>"N" THEN 10910
10930 SUBEND

11000 SUB NEWWINDOW(WNWSPEC$,MAXELM,NXTELM,WNW$(),
    NAME$,TR,BR,LC,RC):: DEF VLE$(X)=CHR$(X+48)
11010 CALL WINDOWPOS(WNWSPEC$,NAME$,P)
11020 IF (P<>0)OR(LEN(WNWSPEC$)+LEN(NAME$)+9>255)OR
    (MAXELM<NXTELM+BR-TR+1)THEN SUBEXIT
11030 WNWSPEC$=WNWSPEC$\"{"\NAME$\"}"\ VLE$((NXTELM AND
    992)/32)\VLE$(NXTELM AND
    31)\VLE$(TR)\VLE$(BR)\VLE$(LC)\VLE$(RC)\"0"
11040 FOR AE=NXTELM+1 TO NXTELM+BR-TR-1 ::
    WNW$(AE)=CHR$(239)\RPT$(" ",RC-LC-1)\CHR$(235)::
    NEXT AE
11050 WNW$(NXTELM)=CHR$(232)\RPT$(CHR$(233),RC-LC-1)\
    CHR$(234):: WNW$(AE)=CHR$(238)\
    RPT$(CHR$(237),RC-LC-1)\CHR$(236)
11060 NXTELM=AE+1 :: SUBEND

11100 SUB DISPLAYWNW(WNWSPEC$,WNW$(),NAME$):: IF
    ASC(WNWSPEC$)=123 THEN 11130 ELSE
    OLDW$=SEG$(WNWSPEC$,2,POS(WNWSPEC$,"]",1)-2)
11110 CALL WINDOWDATA(WNWSPEC$,OLDW$,FA,TR,BR,LC,RC,
    LOCKED):: IF LOCKED THEN 11130 ELSE CALL
    LINK("WINDOW",TR,BR,LC,RC)
11120 FOR R=1 TO BR-TR+1 :: CALL
    LINK("GETSTR",R,1,WNW$(FA+R-1),RC-LC+1):: NEXT R
11130 CALL WINDOWPOS(WNWSPEC$,NAME$,P):: IF P=0 THEN
    SUBEXIT
11140 WNWSPEC$="["\NAME$\"]"\SEG$(WNWSPEC$,
    P+LEN(NAME$)+2,7)\ SEG$(WNWSPEC$,1,P-1)\
    SEG$(WNWSPEC$, P+LEN(NAME$)+9,
    LEN(WNWSPEC$)-P-LEN(NAME$)-8)
```

```
11150 CALL WINDOWDATA(WNWSPEC$,NAME$,FA,TR,BR,LC,RC,
    LOCKED):: CALL LINK("WINDOW",TR,BR,LC,RC)
11160 FOR R=1 TO BR-TR+1 :: CALL
    LINK("DISPLY",R,1,WNW$(FA+R-1)):: NEXT R
11170 CALL LINK("WINDOW",TR+1,BR-1,LC+1,RC-1):: SUBEND

11200 SUB LOCK(WNWSPEC$,NAME$):: CALL
    WINDOWPOS(WNWSPEC$,NAME$,P)
11210 WNWSPEC$=SEG$(WNWSPEC$,1,P+LEN(NAME$)+7)\"/"\
    SEG$(WNWSPEC$, P+LEN(NAME$)+9,
    LEN(WNWSPEC$)-P-LEN(NAME$)-8)
11220 SUBEND

11250 SUB UNLOCK(WNWSPEC$,NAME$):: CALL
    WINDOWPOS(WNWSPEC$,NAME$,P)
11260 WNWSPEC$=SEG$(WNWSPEC$,1,P+LEN(NAME$)+7)\"0"\
    SEG$(WNWSPEC$, P+LEN(NAME$)+9,
    LEN(WNWSPEC$)-P-LEN(NAME$)-8)
11270 SUBEND

11900 SUB WINDOWPOS(WNWSPEC$,NAME$,P)::
    P=POS(WNWSPEC$,"["\NAME$\"]",1)+
    POS(WNWSPEC$,"{"\NAME$\"}",1):: SUBEND

11950 SUB WINDOWDATA(WNWSPEC$,NAME$,FA,TR,BR,LC,
    RC,LOCKED):: DEF
    VLE(X)=ASC(SEG$(WNWSPEC$,X+DP,1))-48
11960 CALL WINDOWPOS(WNWSPEC$,NAME$,P)::
    DP=P+LEN(NAME$)+1 :: FA=32*VLE(1)+VLE(2)::
    TR=VLE(3):: BR=VLE(4): : LC=VLE(5):: RC=VLE(6)
11970 LOCKED=VLE(7):: SUBEND
```

The program can be broken down into four basic sections. Lines 100 to 230 (with data from 1000 to 1050) initialise the program, lines 500 to 600 contain the menu and the calls to the utility routines. Lines 10000 to 10930 hold to editor's utility routines whilst lines 11000 to 11970 hold the subprograms for the windows. It is this final block of the program that we will concentrate on.

The menu offers six options: edit, load, save, print, purge and exit. The subroutines for the first four of these is on lines 10000 to 10930. The use of the accept command on line 510 is rather interesting. If no number is entered (ie text other than a legal number), then the accept statement returns zero into the variable OPT. If OPT returns zero, then the ON .. GOTO on line 520 return control to 510 so as to obtain a proper menu option. The EDIT subprogram on lines 10000 to 10090 is similar to that presented in the November TND, except that it has been modified for 22 lines and 38 columns. To return to the menu from the editor, simply press BACK (FCTN[9]).

There are several global variables in this program. Of these WNWSPEC$, WNW$( ), MAXELM and NXTELM are concerned with the windows. The other global variables are concerned with the editor section of the program. The text within each window is stored in the array WNW$( ). The text in each window needs to be stored in case one window appears on top of another. One element corresponds to a row of the window and include the characters that form the border.

MAXELM holds the maximum number of array elements that are available in WNW$( ) for the window subprogram to use. In this case, since WNW$( ) is dimensioned with 65 elements (line 100), MAXELM is also set to 65. NXTELM tells the subprograms which is the next available array element.

Initially it should be set to 0 or 1 for option bases 0 and 1 respectively. In the above program, no option base statement appears, so option base 0 is assumed and accordingly NXTELM is set equal to zero. Once initialised, MAXELM and NXTELM should not be changed outside the window subprograms.

All of the information necessary to characterise each window is encoded within WNWSPEC$. The information is encoded within a number of fields in the order: window name, first array element, top row, bottom row, left column, right column and whether the window is locked or not.

The windows name is enclosed in either square ([ ]) or curly ({ }) brackets depending on whether the window is currently on the screen or not respectively. The rest of the information is stored in the seven characters that follow the name. A value is represented in the string by adding 48 to the value and converting to the corresponding ASCII character.

First array element refers to the element in WNW$( ) that holds the text in that window. It is stored as two characters as a base 32 number. Top row, bottom row, left column and right column are next to appear in that order and are coded as single characters. Finally the locked character contains either the "0" (value 0) or "/" (value -1) for an unlocked or locked window respectively. Locked windows do not have their contents in WNW$( ) altered when a new window is selected.

As an example of this encoding system, consider two windows called "WIN1" and "WIN2" that have 14, 18, 4, 19 and 2, 16, 20, 28 as their window boundaries. The first window is locked whilst the second one is not. The first window is initially not on the screen and so is enclosed in curly brackets. The next array element available (NXTELM) is 53, say, which as a base 32 number is 1 | 32 + 21. The window boundaries are 14, 18, 4, 19 and the window is locked. The results in the set of seven values 1, 21, 14, 18, 4, 19 and -1 which results in the string "{WIN1}1E>B4C/" using the encoding system above. The new value for NXTELM would become 58 and so the other window would be encoded as "{WIN2}1J2@DLO".

Having discussed the encoding system used by WNWSPEC$, we can now look at the subprograms that implement the windows. Two routines are provided that are used by all of the other subprograms. "WINDOWPOS" finds the position within WNWSPEC$ at which first character of the information describing the window NAME$, and returns that value in P. For our earlier example where the WNWSPEC$ would equal "{WIN1}1E>B4C/{WIN2}1J2@DLO", the statement CALL WINDOWPOS(WNWSPEC$, "WIN2", P) would result in P being set to 14.

The second common subprogram returns the attributes of the window called NAME$ in the various variables in the parameter list. Notice that this subprogram calls WINDOWPOS. The statement CALL WINDOWPOS(WNWSPEC$, "WIN1", FA, TR, BR, LC, RC, LOCKED) would return FA=53, TR=14, BR=18, LC=4, RC=19 and LOCKED=-1.

Before displaying a window, its attributes must first be defined. This is done with the NEWWINDOW subprogram on lines 11000 to 11060. The main job of this subprogram is to encode the information for the window's attributes into the string WNWSPEC$. Lines 11010 and 11020 make some basic checks as to whether this new window can be accepted. All that is checked is if a window with that name has already been defined, whether WNW$( ) has enough elements left to add this new window and whether or not WNWSPEC$ has enough characters for the new window to be added. If any of these conditions are not satisfied, then the program is returned to with no action taken. However, no check is made on the values for window boundaries. Line 11030 adds the new window to the end of WNWSPEC$. Lines 11040 and 11050 initialise the array elements of the window to spaces, with the border characters added. Finally, the value of the next array element (NXTELM) is updated.

Windows are brought up onto the screen using the DISPLAYWNW subprogram (lines 11100 to 11170). Line 11100 check if any windows are currently displayed on the screen. If not, there is no need to read the contents of the last window. If there is a window already on the screen, the name of the "top" one is placed in OLDW$ and its attributes obtained in 11110. If the window is locked, the code for updating the contents is again bypassed. Line 11120 reads in the contents of the last window from the screen using the

continued on page 4

Jenny's Younger Set

Dear Jenny,
Reviewing past TNDs, I came across a program I had sent in which went like so:-

```
100 REM GUESSING GAME
110 N=3
120 INPUT "WHAT IS THE MYSTERY NUMBER? ":A
130 IF A=N THEN 160
140 PRINT "WRONG"
150 GOTO 220
160 PRINT "RIGHT! ANOTHER GO?"
170 PRINT "ANSWER YES OR NO"
180 INPUT B$
190 IF B$="YES" THEN 100
200 IF B$="NO" THEN 220
210 GOTO 170
220 END
```

Now some of my improvements and additions to this program may not have been correct. Firstly, here is how the program works.
The number is set to 3. It asks you for the secret number. If you are right it transfers control to 160. If you are wrong it continues via 140, 150, 220.
To change the mystery number you need to put a different number in line 110. For example, suppose you wanted 2:

```
110 N=2
```
or whatever you want.

If you want a completely secret mystery number, put this in 110:

```
110 N=INT(RND*(whatever you want the maximum number to
    be, say 11))
```

Therefore, for the above example it would be:

```
110 N=INT(11*RND)
```

But say you get it right and you want to change the mystery number to another secret number. You would put:

```
105 RANDOMIZE
```

But say you get it wrong but you want to try again for another secret mystery number but do not want it changed if you get it right, you put:

```
99 RANDOMIZE
```

This would come in handy if you wanted to have another go, even if you got it wrong, change line 220 to this:

```
220 INPUT "ANOTHER GO? ":C$
230 IF C$<>"YES" THEN 250 240 GOTO 100
250 END
```

In this case do not put in line 105 or you will have a changing secret number every time you guess.
Happy programming and a happy New Year, Vincent Maker

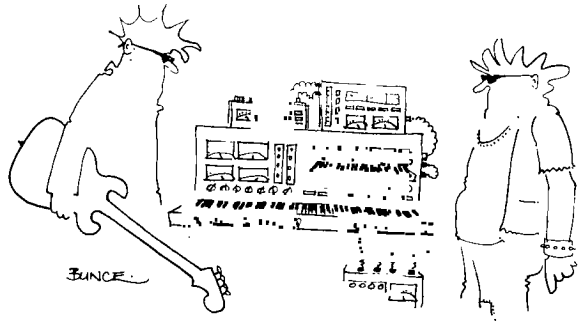
Thank you Vincent, although I am not sure that I understand all that you are suggesting here. I think that it is a great idea to go back and read the TNDs again. There has been so much information in them over the last few years that you cannot take it all in the first time. Perhaps that is why it does not come out in January to give everyone a chance to read them again. I find that I need to read some articles at least 3 times to understand them. Happy New Year to all my younger set readers from me too!

Dear Jenny,
Here is a program that I have written. To run the program, you need the TEII plugged in and you select BASIC. The idea of the program, is to be able to say any word with the speech synthesizer (** the speech synthesizer is needed to run this program).

```
100 OPEN #1:"SPEECH",OUTPUT
110 CALL CLEAR
120 INPUT "PHRASE -> ":A$
130 PRINT #1:A$
140 PRINT : :"R(EPEAT N(EW E(ND"
150 INPUT B$
160 IF B$="R" THEN 130
165 IF B$="E" THEN 180
170 GOTO 110
180 CLOSE #1
181 END
```

I hope you like the program. I would also like to mention that Stephen Judd and myself (Robert Brown) write a monthly column in the TND. It is called GAMES INFORMATION. I would like to think that hundreds of people read this, but I do not think they do. Our column is not only for the younger people of TIsHUG, but for everyone. Adults play games and we also included things about adventures and new information on games and adventures from the States.
Did you know about a karate game for the TI994/A, well GAMES are doing a review on it in the future. And do not forget, GAMES is happy to hear from any reader on questions or to get advice, but they also like a hello or thanks each now and again.

The TI99/4A way
Robert Brown and Stephen Judd



'It's decided to go solo. . .'

Extended BASIC Colours

Adapted from MICROpendium by Lou Amadio

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

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

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

continued on page 41

Disk Hacker V1.0

by Will McGovern, Hunter Valley

BRIEF PROGRAM DESCRIPTION

Disk Hacker part one is the only program of its type available to the TI99/4A community. Disk Hacker is an extremely powerful disk utility which allows the user to examine the actual format of each track of a disk. Have you ever wanted to get an inside look at the latest protection schemes? Well, Disk Hacker part one allows you to do this, and much more.

Equipment required to run Disk Hacker Files required

32K Expansion	Editor/Assembler or	DISKHACKER
TI Disk controller	Extended BASIC or	DISKHACKXB
At least one disk drive	MiniMemory	HACKRDOC/1,2

LOADING DISK HACKER

Loading from EDITOR/ASSEMBLER:-

o Select Editor/Assembler from main selection screen.

o Select LOAD AND RUN from E/A selection screen.

o Type DSKn.DISKHACKER. (n = drive number to load from.)

o Disk Hacker will autostart.

Loading from MiniMemory:-

o Select MINI MEMORY from main selection screen.

o Select LOAD AND RUN from MiniMemory selection screen.

o Type DSKn.DISKHACKER. (n = drive number to load from.)

o Disk Hacker will autostart.

Loading from EXTENDED BASIC:-

o Select Extended Basic from main selection screen.

o Type in: RUN "DSKn.DISKHACKXB"

o Disk Hacker will autostart.

OPERATING DISK HACKER

When the Disk Hacker title screen appears, it will stay displayed for a few seconds and then the main operating screen will come up. There are a few areas of this screen which you must become familiar with before you start to use the program. These areas are as follows:-

o The operation window is the second window from the top of the screen which contains the Disk Analyser Vn 1.0 message. The function of this window is to tell you what the program is doing at any particular time. If the 'Disk Analyser Vn 1.0' is being displayed, then the program is waiting for your input. Whenever the program halts all you have to do read the display in this window to find out what is going on. Errors are also displayed in this window. Therefore, if disk activity stops for no reason, read the display in the window to find what has gone wrong.

o The large space in the middle of the screen, between the two small windows, is where the program displays its prompts for the information you have to supply for disk analysis, such as number of sides and drive number to use.

o The final area to be explained is the very bottom line of the screen just below the lowest window. This line explains which function keys are active at any particular time. The only time that the function keys are not displayed is when you are in analysis mode. This will all be explained further on.

Answering the Prompts

Now that you understand the basic functions of the important areas of the initial screen we can proceed to the next stage, answering the prompts that the program gives you.

You probably noticed that when the title screen vanished and the initial screen appeared that the question 'Drive number [1-3]:' appeared in the prompt area. This is the first of the five prompts you will have to answer before you can begin the analysis. The information asked of you may seem complicated when you first start but you'll eventually be able to understand and answer the prompts quite easily. The next section will be devoted to explaining each of the prompts and showing you how to answer them.

o Drive number [1-3]

This is the simplest of the questions you have to answer. All that you have to do is enter the number of the drive you wish to use. This value must be between 1 and 3 because 3 is the maximum number of drives that the TI controller can handle.

EXAMPLE: Drive number [1-3]: 1 <enter> selects drive one as the drive you wish to work on.

o [S]ingle or [D]ouble sided

This prompt simply asks you whether the drive you have selected is a single or double sided drive. (ie. Whether the drive has a second head to read the second side of the disk. NOTE: If you have to flip the disk over (FLIPPY) to use the second side of the disk then the drive is still only single sided.) You answer this prompt with 'S' for single sided or a 'D' for double sided. If you choose single sided then the program will not be able to analyse the tracks on the second side of the disk. If double sided is selected then it makes reading the second side of the disk possible.

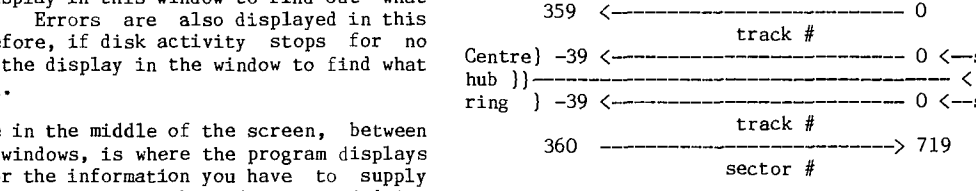
EXAMPLE: [S]ingle or [D]ouble sided: S <enter> selects single sided operation.

[S]ingle or [D]ouble sided: D <enter> selects double sided operation.

o To fully understand the next two prompts you must first understand how the tracks on a floppy disk are arranged. Here is a brief explanation of this structure.

On a normal TI99/4A compatible disk you have 40 tracks on each side of the disk. On each side the tracks are numbered from 0 -> 39 inclusive, with track 0 being furthest from the hub ring in the centre and track 39 being closest to the centre. The diagram shows how the tracks are arranged on a normal double sided disk. For single sided drives the format is exactly the same except that the drive is not capable of reading or writing from or to the second side.

Diagram 1. Normal floppy disk format.



o Starting track

The process of entering track numbers in Disk Hacker is very simple. For single sided operation you

enter exactly what you would expect, a number between 0 and 39. Numbers outside these limits will be rejected. For double sided operation however, to make it easy to enter track numbers for the second side, a new system is used. The tracks on the second side, instead of being numbered 0 to 39 and clashing with the side 1 track numbers, they have been renumbered as tracks 40 to 79. Therefore, the track numbers on side 1 are from 0 to 39 and the track numbers on side 2 are from 40 to 79. (ie. Track 40 is really track 0 on side 2). So, the limits for the entry in double sided mode are 0 to 79.

EXAMPLE: Starting track: 10 <enter> selects the starting track as track 10 on side ONE.

Starting track: 50 <enter> selects the starting track as track 10 on side TWO.

o Stopping track

Exactly the same system applies for entering the stopping track number as for the starting track number. (ie) Single sided 0 -> 39, Double sided 0 -> 79.

o The next prompt is is a very advanced, but complicated, feature of Disk Hacker. Before you just answer 'Y' or 'N' I will try to explain in easy terms the meanings of the two aspects of sectors involved in this prompt, CRC's and Data Marks.

A standard, non-protected track on a TI99/4A floppy disk contains 9 sectors each of 256 bytes length. On the track there is space between each sector which contains a special series of information bytes called an address or HEADER. Addresses will fully explained further on in this documentation. Somehow, the disk controller must be able to locate where the end of this gap is, which is where the sector starts. To find the start of the sector the disk controller looks for a special 'mark' written on the disk when it was formatted. This 'mark' tells the controller that it is at the end of the inter-sector gap and that the read/write head is positioned at the start of the sector data. This 'mark' is called a DATA ADDRESS MARK and can be in two different forms, normal or deleted. The TI controller uses only normal data address marks but some copy protection techniques use sectors with the deleted data address marks, which cannot be created using normal disk access methods. Therefore, to check if the disk is an original all they have to do is check to see if the deleted marks are there. DISK HACKER WILL TELL YOU WHICH SECTORS ON A TRACK HAVE DELETED DATA MARKS AND WHICH HAVE NORMAL DATA MARKS.

CRC is short for CYCLIC REDUNDANCY CODE and it a very common method of checking for errors in a data stream. A CRC can be thought of as just a complicated checksum. Each sector has two 2 byte CRC error check codes, one for the sector address and one for the sector data. If a CRC for the sector is incorrect then it becomes a 'bad' sector, gives an error code and cannot be read using the TI-DSR. DISK HACKER WILL CHECK THE HEADER AND THE DATA CRCs AND TELL YOU WHICH OF THE SECTORS HAVE CRC ERRORS.

o Check marks and CRC's [Y/N]

This prompt asks whether or not you wish to check for deleted data marks and data CRC errors. Header CRC's are checked automatically and marked even if you answer 'N' to the prompt. If you select 'Y' then the analysis will take approximately twice as long as it would without the checking, but the information gained can be invaluable.

ANALYSING THE DISK

Now that you have given the program all the information it needs to know you can start the disk analysis procedure. After you have pressed enter for the last prompt the message 'PRESS SPACEBAR TO BEGIN ANALYSIS' will appear in the operation window. When you press the spacebar the selected drive will start up and the operation window will start to display the current track and side number. All the information you entered in the information area will stay displayed so you can remember what responses you made to the prompts. The track count will begin at the start track you entered and stop when it reaches the specified stop track.

NOTE: If you feel that you have made an incorrect entry in the information area do not worry. DISK ACTIVITY MAY BE TERMINATED AT ANY TIME BY PRESSING FCTN[4] (BREAK). HOLD THE BREAK KEY DOWN UNTIL THE PROGRAM CONFIRMS THAT THE COMMAND HAS BEEN TERMINATED.

When the track count reaches the specified stop track the program will inform you that the analysis has been completed and will then go into the next stage of operation, your viewing of the information.

VIEWING THE INFORMATION

Now that the analysis has been completed and there is no disk activity you go into the second stage. The second stage is a system designed to allow you to view all the information obtained from the analysis in an easy to understand, easy to read format. All the information is displayed in a columnar arrangement with each row representing one sector from the track and one track taking one full screen.

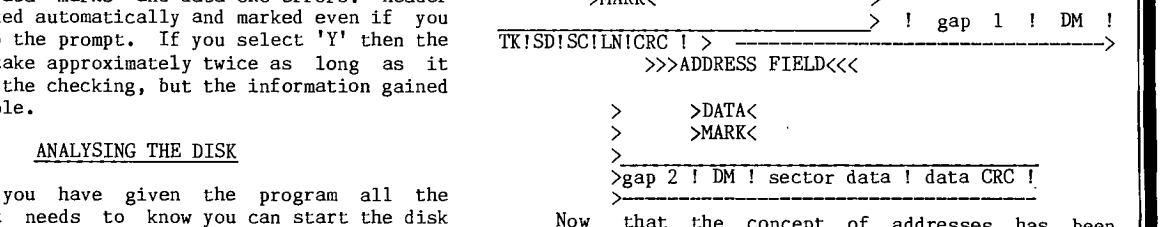
If we start at the top of the screen and work down explaining as we go then you will achieve the best understanding of what is actually being displayed.

o At the very top of the display window we have the TRUE physical information for the track. First, the drive number that the track was analysed from is displayed, then the physical side of the disk on which the track is located, and the physical number of the track the information was read from.

o The next row of information displayed is the actual number of sectors found on the track. This saves you from the annoyance of having to count the number of rows displayed in the display area.

Before we go on to the next set of information you should understand the concept of addresses. When a track is formatted the sectors are arranged so as to be uniformly distributed about the track. To do this, especially on the longer outside tracks of the disk, a space must be left between each sector to give the disk controller time to get ready for reading the next sector. Preceding each sector in this inter-sector gap is an ID address field or address for short. This address consists of six bytes containing information about the proceeding sector, this being track number, side number, sector number, sector length and CRC bytes in respective order. The CRC bytes are a precaution so we can tell if the address read has not been read correctly. The disk controller must be able to locate this address in the inter-sector gap, so, when the track is formatted a special 'mark', like the data address mark, is placed directly before the address. When the controller locates this mark in the gap it knows that the following six bytes are the ID address field bytes.

DIAGRAM : Structure of one ninth of a normal TI disk track; (ie) only one of the nine sectors is shown with associated gaps.



Now that the concept of addresses has been explained we can go on to explain the display between the two lines across the window. All these symbols are abbreviations of parts of the ID address field and terms used to describe certain aspects of sectors. Here is an expanded list of what all these symbols stand for :-

o TRK stands for TRACK NUMBER. This track number is the number obtained from the address field. This does not necessarily have to contain the same number as the physical track number displayed at the top of the screen but on normal TI disks it is always the same, otherwise it is unreadable by the standard disk DSR.

o SID stands for SIDE NUMBER. The side number is obtained from the address field and if it is 0 then the sector is saying it is on side 1. If it is 1 it says it is on side 2. As with the track number the address side number does not have to correspond to the physical side number that the track is on but any normal TI disk has the correct value.

o SEC stands for SECTOR NUMBER. On any TI-DSR readable disk the sector numbers from the address field for the 9 sectors range from 0 to 8 inclusive. (ie. Every track has a set of 9 sectors, each with a unique number between 0 and 8). However, the sector number can vary from 0 to 255 but no normal TI track contains sector numbers above 8. The column of sector numbers is the sector interlace.

o LGTH This stands for LENGTH of the sector. Any normal TI-DSR compatible disk contains sectors of length 256 bytes. However, it is possible to create sectors of other lengths. Disk Hacker will display the length of every sector it finds. If Disk Hacker runs across a sector of indeterminate length it will display 'inv', for invalid, in place of the length.

o CRC This column contains the 2 header CRC bytes from the ID address field. The CRC bytes, if correct, indicate a valid header, if incorrect then the header contains an error. Disk Hacker checks this header CRC and tells you if it is invalid. If the header CRC is incorrect a small arrow appears to the left of the CRC value for the sector.

These next two columns only display any information if you responded to the 'Check marks and CRC's [Y/N]' with a 'Y', as they refer to sector data and not header data.

o M stands for DATA MARK. The information displayed in this column tells you whether the data mark for any particular sector on the track is a DELETED data mark. If a deleted data mark is detected for a sector then a 'D' is displayed in the column on that sector's row. If the data mark is normal then nothing is displayed in the column, making it easy to spot the sectors with DELETED DATA MARKS.

o C stands for DATA CRC. If the data in the sector does not produce the correct CRC then there is a DATA CRC error. If there is a data CRC error for a particular sector then an 'E', for error, is displayed in this column. If there is no error then nothing is displayed, making it easy to spot DATA CRC errors.

NOTE : There are two special cases which affect the display in this column. They are these :-

o If 'inv' is displayed in the length column (invalid length) then no data is displayed in the M and C columns because it is impossible to check for this information.

o If no data address mark can be found after the address (ie. no sector data, fake ID field), then 'RNF' is displayed across the M and C columns.

PAGING THROUGH THE INFORMATION

Because only one track is displayed per screen, if you analysed more than one track then you must 'page' through the track information. Because all the analysis has already been done the disk does not have to turn on again unless you wish to analyse another disk. To page through the information for each track use the FCTN[E] and FCTN[X] keys. FCTN[X] pages forward through the tracks and FCTN[E] pages back through the tracks. Note that as you page through the tracks the information at the top of the screen is also updated. The track number at the top is handy if you want to compare the address track numbers in the data area to the real, physical track number.

The physical track number at the top does not go past 39, even if you specified double sided track numbers above 39, because the address track numbers start at zero again when they start on the second side (see diagram in first file). So, when the physical track counter exceeds track 39 on side 1 it starts again from track 0 on side 2 so you can easily compare the address track numbers with the physical track number without having to subtract 40 all the time.

DECIMAL / HEXADECIMAL TOGGLING

While examining the information you may find it more convenient to look at the information in hexadecimal rather than decimal. If you press FCTN[6] you will toggle between DECIMAL and HEX. The header CRC bytes are always displayed in HEX because they are of no use in decimal.

FUNCTION KEYS FOR DISK HACKER

Press at any time : CTRL[=] ---> QUIT DISK HACKER (not while disk on)
FCTN[9] ---> GO TO DISK HACKER TITLE SCREEN
FCTN[8] ---> GO TO DISK HACKER INITIAL SCREEN

Cursor function keys : FCTN[1] ---> DELETE CHARACTER
FCTN[2] ---> INSERT SPACE
FCTN[3] ---> ERASE LINE
FCTN[S] ---> MOVE OR BACK ONE CHARACTER
FCTN[D] ---> MOVE CURSOR FORWARD ONE CHARACTER

During disk activity : FCTN[4] ---> TERMINATE DISK ACTIVITY

Remember, this is only Disk Hacker Part one. Part two will dig even deeper into tracks. o

Fairware Information

Program name : Disk Hacker, Part one
Disk Controller : TI Controller only.
Version number : Vn 1.0
Author : Will McGovern
Address : 215 Grinsell St.,
Kotara, NSW 2289, Australia

If you find this program useful please send a contribution to the above address. Personal cheques from outside Australia present problems. Cash is usually safe in first class air mail.

Note: If there is a large enough demand I will rewrite these programs to run with the Myarc and CorComp disk controllers. Do not rename the Disk Hacker files or the program will not load.

NOTE: o This program is released as FAIRWARE, so, if use is made of it a contribution to the author would be greatly appreciated.

o This program is not capable of destroying data contained on floppy disks unless it is used incorrectly.

o No responsibility can be taken by the author for the loss of data due to misuse of this program.

o

Funnelweb Configuration Instructions

ROM, Orange County, CA, USA

(1) Instructions

A. These instructions are presented as a service to the TI99/4A community by UGOC, the user group of Orange County (California). They supplement the Funnelweb versions documentation. This release supports versions 4.10 (May 30 1988) 4.11 (July 4 1988) AND 4.12 (August 12 1988).
Comments and suggestions should be sent to:-

UGOC c/o
Jim Swedlow
7301 Kirby Way
Stanton, CA 90680

B. Print and read the Funnelweb documentation. Using the formatter, print all files that start with "FWDOC" and "-READ-ME".

C. These instructions assume that you will operate Funnelweb from drive 1. If you use another drive you will need to modify them accordingly.
They also assume a basic system and normal use. If you have special features or uses you should be able to modify them to meet your needs.

D. If you need to press a key or keys, the "<>" signs will be used (e.g. "press <ENTER>"). When two keys must be pressed together it will be shown this way: "press <CTRL[C]>", which means hold the <CTRL> key down with one finger and then press <C> with the other. Release both fingers simultaneously.

E. In boxes where you input information (a file name, etc.), indicate that you are done by pressing <ENTER>. When this is necessary it will be shown this way:- "enter the correct file name".

F. In configure menus you may normally choose options by pressing the first letter. For example, the top menu has three choices:

sysinfo
quit
install

press <S> for sysinfo , <Q> to quit and <I> to install. Such choices will be shown as "press <Quit>" or "press <Q> to quit". A menu layout of all configure menus follows these instructions.

G. <CTRL[C]> is used to move from the current menu to the previous menu. You may use <BACK> or <FCTN[9]> instead of <CTRL[C]>. A number of times these instructions ask you to press <CTRL[C]> to return to a previous menu. Sometimes it will be necessary to press <CTRL[C]> more than once! <CTRL[C]> also now works in DM1000 instead of <BACK>.

H. In many places in the configuration program pressing <?> or <FCTN[1]> windows in help screens. These screens are filled with useful information. you should check them frequently the first time you configure Funnelweb.

I. Also available in most places in the configuration program is the quick directory. Press <FCTN[7]> to access the disk directory function.

J. Abbreviations :

Fwb	Funnelweb
XB	Extended BASIC
EA	Editor Assembler
TI-Wr	TI-Writer

(2) Starting Up.

A. Make a working Fwb disk. Use DM1000 to copy the files you will be using to a freshly initialized disk. A list of files and some suggested arrangements follow these instructions. Configure only your working copy. Keep an unmodified master copy "just in case".

B. Load Fwb with your working Fwb disk in drive 1. If you are using XB, Fwb will autoboot when you choose Extended BASIC. If you are using the EA module, choose "5. RUN PROGRAM FILE" and then press <ENTER>.

C. If you loaded from XB, choose configure from the XB menu. If you loaded from EA choose user list from the EA menu. You can load it from the loaders function. Choose option 2. The file name is DSK1.CF

(3) Top Menu

A. Press any key to get past the opening screen.

B. Press <s> to modify the system information (sysinfo).

(4) Sysinfo Menu.

A. Press <L> to load the system configuration file. Enter the file name (DSK1.SYCON). You can have multiple configuration files with any names you choose. To implement one you must activate the configuration program, load the configuration file and then install it (see step 12).

B. Press <E> to edit the system configuration.

(5) Loading

A. Press (L) to access the loading menu.

B. Boot tracking should be on if you are loading from a disk drive. RAMdisk users should turn off as Fwb cannot track booting in RAMdisks. Press (B) to toggle boot tracking on or off.

C. The number following "TI-Writer side" is the drive number where the files that support the choices on the TI-Writer menu will be found. This number is used if boot tracking is off or if it fails. The same applies to "Edit/Assembler side". Normally both are 1. To change them, press (T) of (E) and then enter the drive number.

D. "Working drive" is the drive number of the drive in which you will put your data disk. If you have a PIO drive system, this will normally be 2. Press (W) to change the working drive.

E. If "UL immediate" is on, the user list will be the first menu you will see when you load through EA. Unless you want to access the UL immediately you will normally want these off. Press (U) to toggle this on and off. Turning this on allows you to load Fwb from EA and have a menu that you designed completely appear first. This feature was added with version 4.11

F. When all values are correct, press <CTRL[C]> to return to the main edit menu.

(6) Devices

A. Press <D> to access the devices menu.

B. Press <E> for the editor printer. If you have a parallel printer, enter "PIO". If you have a serial printer, make sure that all the switches are correct. (RS232.BA=1200.DA=8)

C. Press <F> for formatter printer. If you have a parallel printer, enter "PIO.LF". If you have a serial printer, make sure that all the switches are correct and that the printer name includes ".LF".

D. The object, work and program file names are defaults for various Fwb functions. If you enter a name, include the drive designator (for example, DSK1.loadfile). If you leave the field blank, the drive number you designated as the "working drive" in the loading menu will appear. For example, if you designated the working drive as 2, it will be "DSK2.". If you want the default to be a different drive than the working drive, you can enter "DSKn.". Any file you tag in the quick directory will over ride these default names.

E. Press <O> to update the file. This is the default file name when an object (D/F 80) EA file is required. For example, when loading an object file.

F. Press (W) to update the work file name. If you put a name here, it will be a default when you use loadfile in the editor.

G. Press <P> to update the program file. This is the default when loading EA program files (TI PGM, GPL PGM, E/A PGM etc.).

H. When all values are correct, press <CTRL[C]> to return to the main edit menu.

(7) Colours.

A. Press <C> to access the colours menu.

B. The cursor will be on the first colour choice. This is the screen colours that will appear when Fwb boots. The other colours are those that cycle when you press <CTRL[3]> in the editor or 0 (zero) on most screens that invite selection by number. You have several choices from this point:

<E>dit: modify the current colour.
<N>ext: move to the next colour on the list.
<B>ack: move to the previous colour on the list.
<X>chg: exchange two sets of colours.
<R>edo: restore the colours to what they were when you started.
<V>iew: the current colour.

C. When all colours are correct, press <CTRL[C]> to return to the main edit menu.

(8) Menu

A. Press <M> to edit the choices on the main TI-Writer and EA menus.

B. Press <T> to edit the choices on the TI-Writer menu and <E> for the Editor Assembler menu. You can change items 4 through to 7 on both menus.

C. For each line on the menu you have these options: <E>dit <N>ext <B>ack and <R>edo. These are the same as in colours (see step 7b.) If you <E>dit a menu line you will be required to enter the following information:

i) Name: this is the name that will appear on the menu. This can be anything you want up to ten (10) characters.
ii) File name: this is entered just to the right of the menu name and must be only two (2) characters long. This is the file name that will be loaded when you opt for that menu choice. You cannot enter "DSKn." as Fwb gets the drive number from boot tracking or the loading information (see step 5b and 5c)
iii) Type: here you tell Fwb the file type. Move the light bar between options by pressing <N>ext and <B>ack.

You have the following choices:

TIW pgm: emulates option 3 from the TI-Writer menu. The file must be in E/A format.
GPL pgm: this is used for most programs loaded by the E/A "run program file " option.
E/A pgm: this is for "run program files" that need E/A utilities. In general use "E/A pgm" if "GPL pgm" does not work.
Script: this allows you to write a script for loading a series of object (D/F 80) E/A files. See FWDOC/UTIL for details.
LD/RUN: Use this for most "load and run" object (D/F 80) E/A files.
Low mem: loads object (D/F 80) files into low memory. See FWDOC/UTIL for details.

E. When you are satisfied with the TI-Writer and E/A menus, press <CTRL[C]> to return to the main menu.

(9) XB List.

A. Press <X> to modify the menu that appears after you load Fwb from XB.
The first three choices ("TI-WRITER", "EDIT/ASM", and "XB return") cannot be modified.

B. You now have four choices:
<E>dit: use this option to edit the current entries (that is, the XB list in the SYSCON file).
<F>etch list: use this option to obtain the menu in the load program. Normally this is the same as that in the SYSCON file.
<M>ake reserve: if you <F>etch a list, press <M> to make it the reserve list. Then if you later press <R>edo this is the list that will be restored.
<X>chg buffers: restores the previous "reserved" list.

C. During initial configuration, you should press <E>dit.

D. <E>dit will show you the choices in the XB menu. You have the normal keys active for moving from item to item. <E>dit <N>ext <B>ack and <R>edo. See step 7E for definitions.

E. If you choose to <E>dit an entry, you will be required to enter the following information:

i) Name: this is the name that will appear on the menu. this can be anything you want up to 10 characters.
ii) Boot tracking on/off: keep boot tracking on if the disk with the file will be in the drive from which you loaded Fwb. Turn it off if it will be in another drive. boot tracking does not work with RAMdisks.
iii) Secondary no/yes: if the drive for this file is different than the boot drive and if you answer yes to this question Fwb will look for the user file in the drive specified in step 5C.
iv) Reminder no/yes: if this is yes, Fwb will remind you to insert the disk with the file. If it is no, Fwb will read the file immediately. make this yes if the disk with the file will not be in the named drive when you invoke this menu choice.
v) Filename: the name of the file that Fwb should load. Include "DSKn." in the file name.
vi) Type: here you tell Fwb the file type. Move between options with <N>ext and <B>ack. You cannot leave this field as a blank. You have the choice listed in step 8D(3) and XB PGRM. This is a standard XB program.
vii) XB retn: this returns you to the XB "ready" screen. It works like the new command.

F. When you are satisfied with the XB menus, press <CTRL[C]> to return to the main edit menu. ◆

continued from page 26

These last few words are written November 26th, just before the December 1st text deadline for this issue, and lots to print out still!

Just as I thought the supply of new software was declining, along comes the postman with a small package with three innocent looking disks inside, containing (in archived format) a bit more than 3000 sectors of software catalogue. Lots that we have, some duplications, some we have no interest in, and one or two (ha!) items that we have been looking for for years and years, so watch out for some nice new goodies in the coming quarter!

One disk library user had a problem using HBMPRINT, a utility to print the data from Home Budget Manager in various formats. It would only load with Editor Assembler. Now, this program was in Forth, which if you recall Versions 3.x of Funnelweb, had to be flagged K=60 in the Funnelweb load menu. That menu is no longer available, and it would appear that Funnelweb is no longer able to produce that particular load environment.

No problem. Some time ago, the peculiar original TI Forth Load requirement was removed, and an amended format produced which would load as a standard Load and Run file. You can find it on the MiniMemory version of Forth in the disk library. All I had to do was swap the load routine and now HBMPRINT loads fine with Funnelweb.

Hands up anyone who remembers the Adam computer! It was on sale in this country for about a week I think! It was a Z80 based micro, like the ZX80 and ZX81.

continued on page 16

TI Bits, Numbers 6, 7, 10, 11

by Jim Swedlow, ROM, CA, USA

FORMATTING DISK TEXT FILES

This month we will explore further into using TI-Writer and disk files as output. Two simple utility programs accompany this article.

First, a bit about what the Text formatter does. If you include the command ".FI;AD", the formatter will right justify your text (so both the right and left columns are straight lines). When you save a file to disk from the Editor, however, you have a "ragged right" (or not right justified) file. If you want right justification on disk (and to use the other features of the formatter), all you do is specify a disk file name as the Print device name in the formatter.

There is a small hitch. Each and every line in the disk file will end in a line feed <CHR$(10)>. Then if you print that file without adding ".LF" to the printer name, your text will be double spaced. It will even be stranger if you use underlining and bold face.

The reason is that the formatter expects to output to a printer. Since line feed and carriage return are about the only two universal printer command codes, the folks who wrote TI-Writer had to come up with a way to do bold face and underline using only those two commands.

Here is what they did. Most printers will advance the print one line when they receive a line feed and return the print head to the left column when fed a carriage return.

To underline a word: print the line, execute a carriage return (so that the print head goes back to the beginning of the same line) and print underline characters (FCTN[U]) under the word to be underlined. Then send line feed and a carriage return and start the next line. Bold face is similar except that TI-Writer prints the bold face word four times. You add ".LF" to the printer name in the formatter so that TI-Writer can control when line feeds are sent. All of this is fine for a printer but not for a disk file.

If you are going to save your formatted text to disk, first do not use either bold face or underline (if you want them use your printer commands rather than the formatter commands, eg ESC G rather than @). After you have run it through the formatter, you must load the formatted file into the Editor and then save it back to disk. Why? Well, if a line has 80 characters, the formatter will add an LF to the end making it 81 characters long. Then when a BASIC program attempts to read that line, it will lock your system up. By loading and saving through the editor, all lines are trimmed if they are over 80 characters long.

Use Print File to save the file so that the Editor will not add the tabs.

Then use the program LF Stripper (elsewhere in this issue) to strip the line feeds from the ends of the lines.

(If the text has no control codes that you wish to keep, you can strip ALL control codes, including LF and CR by using PF then C DSK1.FILENAME - that is a C in front of the disk filename, when renaming. This will remove all control codes and is much faster than using LF Stripper!)

```
100 ! LF Stripper
110 ! BY JIM SWEDLOW
120 ! OCTOBER 22, 1986
130 !
140 CALL CLEAR :: PRINT "Line Feed Stripper": :
150 INPUT "Old File: DSK":A$ :: PRINT :: INPUT "New
    File: DSK":B$
160 PRINT :"Working"
170 OPEN #1:"DSK"&A$,INPUT :: OPEN #2:"DSK"&B$,OUTPUT
180 IF EOF(1)THEN 220 ELSE LINPUT #1:A$
190 IF A$=CHR$(10)THEN PRINT #1:" " :: GOTO 180
200 I=LEN(A$):: IF I=1 OR SEG$(A$,I,1)<>CHR$(10)THEN
    I=I+1
210 PRINT #2:SEG$(A$,1,I-1):: GOTO 180
220 CLOSE #1 :: CLOSE #2 :: STOP
```

IF THEN IN Extended BASIC

A number of the Extended BASIC columns discussed alternatives to IF THEN. Here is another. Suppose that A$ depends on the value of I. You might use this code:

```
IF I=1 THEN A$="FRED" ELSE A$="PAUL"
```

A simpler way is to use the SEG$ function:

```
A$=SEG$("PAULFRED",1-4*(I=1),4)
```

Will this work if the two variables have different lengths? Yes. Remember that SEG$ does not produce an error if the length of the new string (the last number) is longer than the source string. If our two names are "PAUL" and "SAM", this works:

```
A$=SEG$("PAULSAM",1-4*(I=1),4)
```

ERROR TRAPPING AFTER RUN

The following will NOT work:

```
100 ON ERROR 200
110 RUN "DSK1.TEST"
200 PRINT "TEST PROGRAM NOT FOUND"
210 PRINT "INSERT DISK IN DRIVE ONE"
220 PRINT "AND PRESS ANY KEY"
230 CALL KEY(0,K,S) :: IF S<1 THEN 230
240 RETURN 100
```

If you run this program without a program called TEST in drive one, lines 200 through 220 will print their message and then your program will fail in line 230 with 'SYNTAX ERROR IN 230'.

Why? As near as I can tell, when your TI99/4A executes the RUN part of RUN "DSK1.TEST", it clears memory including the variable table (think of it as 'un-pre-scanning' the program). The program and the line number table, however, remain.

When your TI99/4A tries to execute line 230, it looks for the variable table to find out where the values for S and K are stored. Finding none, the CPU decides that an error condition exists and ends the program.

What to do? One way is to add a disk directory reading routine to find out if the desired program is on the disk. This will significantly increase the time it takes to load the program, however.

Another way is to use RUN. This will recreate the variable table. Since you can specify the line number where program execution starts, you can control what happens. This works:

```
100 ON ERROR 200
110 RUN "DSK1.TEST"
200 RUN 210
210 PRINT "TEST PROGRAM NOT FOUND"
220 PRINT "INSERT DISK IN DRIVE ONE"
230 PRINT "AND PRESS ANY KEY"
240 CALL KEY(0,K,S) :: IF S<1 THEN 240
250 GOTO 100
```

What about pre-scan? Even though you specified that program execution started at line 210, the entire program is pre-scanned. You must be careful, however, to return to a point in the program that will assure that all necessary variable initialization is completed.

IF THEN Again

IF THEN is the most versatile command in Extended BASIC.

Often you will use a variable as a flag. You set it to one value at the beginning of a program and then change it if something happens. Then, using IF THEN, you can tell if that 'something' happened and branch your program's operation accordingly.

When doing this, remember two things: First, during prescan, all numeric variables are set to zero. Second, the IF THEN test considers zero as false and all other values as true.

The variable that you are using as a flag, then, does not need to be given an initial value.

Further, any change to a non zero value would make it look like a true statement in an IF THEN test. If A is your flag you could do this:

```
IF A THEN 220 ELSE 300
```

If the 'something' had not happened then A would still be zero, the IF test would be false and control would pass to line 300. If the 'something' had happened, A would not be zero, the IF test would be true and execution would branch to line 220.

This should help you simplify your IF THEN testing.

SPEED

You always hear how slow BASIC and Extended BASIC are when compared to assembly language. Alas, it is true. There are ways, however, to make your program run a bit faster. Consider this code:

```
10 FOR I=1 TO 200
20 B=3*I
30 IF B>999 THEN 50
40 NEXT I
50 END
```

When you RUN this, your TI99/4A notes that there are two variables and assigns memory space for each. Each time line 20 executes, the memory location for I is found and the value obtained and multiplied by three. Then the location of B is identified and this value stored there (the order of these steps may be a bit different but each step must happen).

Now look at this:

```
10 FOR I=1 TO 200
20 IF 3*I>999 THEN 40
30 NEXT I
40 END
```

Now there is only one variable and the number of steps is reduced. The result: running time drops from 3.80 seconds to 3.01 seconds and memory drops from 92 bytes to 69.

In this example time dropped 21%. If there had been a PRINT command in the loop, running time would be 10 times higher and the time drop under 5%. The reason is that it takes a lot of time to print on your screen.

PRINT SEPARATORS

There are any number of ways to display text on your screen. In this and the next two items, I will cover some ideas that might help you. All of these will also work in BASIC except where noted.

Suppose that you want to display menu choices at screen lines 5, 7, 9 and 11. You could use four DISPLAY commands:

```
DISPLAY AT(5,1):"FIRST LINE"
DISPLAY AT(7,1):"SECOND LINE"
DISPLAY AT(9,1):"THIRD LINE"
DISPLAY AT(11,1):"FOURTH LINE"
```

Another way to do this is to combine your DISPLAYs into one command:

```
DISPLAY AT(5,1):"FIRST LINE": : "SECOND LINE": :"THIRD
    LINE": : "FOURTH LINE"
```

Note that the print separator between lines is : :.

I have seen (and used) :"":. The double quote was inserted to tell the computer to display a blank line. This is not necessary, : : works just as well and uses less memory. In Extended BASIC you must type a space between the colons or your TI99/4A will think that you are inputting a line separator.

This also works with PRINT statements in both BASIC and Extended BASIC.

==========

PROMPTS FOR INPUT

You have probably used a prompt with the INPUT command like:

```
INPUT "INPUT NAME":N$
```

The limitations here are frustrating. You cannot use a string variable, if you want to leave blank line then you must use a lot of spaces, etc, etc.

Sometimes I use a PRINT statement for the prompt and then follow with INPUT. By accident I discovered that INPUT respects any trailing print separator on the preceding PRINT command:

```
PRINT A$;TAB(20);
INPUT B$
```

The INPUT ? will appear at the 20th column.

PRINTING STRINGS

If you are printing two strings, you would normally use a semi-colon <;> between them. When you do this, your TI99/4A looks at the current location on the screen and the length of the next print item. If it is longer than the space left on the current line, the string starts at the beginning of the next line.

This can result in annoying blanks in your right hand margin. There is another way:

```
PRINT A$B$
```

Regardless of the length of A$, B$ will start in the next empty column. This may or may not give you the result you wanted, but it does give you options.

==========

Letter to the Editor

It will not be long before we celebrate our 8th year of active support for the TI99/4A Home computer, and as I look back, I am very proud of what has been achieved in this time. We are still being supported by a club of dedicated members, who's task it is to ensure that our computer lives on. This, in itself, some years after the computer was taken off the retail arena, has been a massive job, yet the challenge has been met and through you, the TI99/4A lives on. Give yourselves a pat on the back, you deserve it. I am so proud to have been involved, and continue my contact, with each of you. We, as a family of TI99/4A users, have been through a lot in the past years, and have become an example for other computer clubs to follow. Keep up the great work.

Seasons Greetings from Shane Andersen
(A very proud member)

continued from page 14

It appears that some surplus Adam PCBs duly found their way into some TI PEBs over in the states, providing a 2nd (Z80) processor! I shall try to find out a little more, it seems to have been a well kept secret. Maybe we can do something with ZX80's!

Some RAMBLES text can be found on GEnie, references 2676 and 2677!

Time to stop and print out! Best wishes, Happy New Year! and do write and tell me what you want me to write about!

Deadline by Infocom

Solution by Ian of TIBBS

This is a complete walk through solutions to DEADLINE and INFIDEL. If you do not want to know certain details of these games then the answer is simple.

*** DO NOT READ THIS ARTICLE ***

I have no time for those who criticise the publication of adventure clues/solutions/maps etc. You have been warned! This, being an Infocom adventure, is only one way to complete this adventure, there may be, and probably are, many other ways to reach a solution. There are also lots of nice details, fun items, cut from this solution to make it reasonably brief, I urge you to discover them for yourself. They are what make adventuring the fun/challenge that it is. Enough said!

Directional movements are in brackets, e.g. [N,N,E].

From the front path of the Robner's estate, go [N] to the front door of the house. Type OPEN DOOR and go [N] into the Robner's house. From inside the door, go [N,E] and type CLIMB STAIRS twice (or you can just go [U,U]) to get to the second level of the estate. From there, go [W,W,W,W,N] to the library, where you will start the first of a series of Sherlock Holmes type activities.

In DEADLINE, you need to establish the motive and method for the murder beyond all reasonable doubt before you can arrest the guilty person/s. If you do not have an airtight case the jury will acquit the defendant/s. It is here in the library where we go about establishing the method by which poor Mr. Robner was done in. First off, type EXAMINE RUG (or just LOOK RUG). You will find some mud spots, which is your first clue. Now, GET THE CUP, PAD, CALENDAR AND PENCIL. Now RUB PENCIL ON PAD, and then TURN PAGE OF CALENDAR. Aha! Perhaps a clue as to the motive? Let us see if we can substantiate the method a little more. What about that mud on the rug, very interesting! Type OPEN BALCONY DOOR and go [N] onto the balcony. Check out the railing by typing EXAMINE RAILING and you will see some scratches, lending credence to the theory that perhaps the murderer climbed up the balcony from the ground below where he (or she) got mud on their shoes. Let us have a look below and check for some indication that the murderer was indeed below the balcony. To leave the balcony, go [S,S,E,E,E,E,D,D,W,S]. Type OPEN DOOR and go [S] back to the front door.

Now go [E,E,SE] to the shed, where you will see a ladder. Type EXAMINE LADDER. Hmmm! This ladder-and-balcony theory is looking good! Let us see if we can prove the ladder was below the balcony. This will have to wait a while, though, because it is getting late in the morning and we have to do some more checking in the house before the reading of the will takes place. Besides that, we need to talk to Mr. McNabb and he does not seem to be in the mood right now. Go back to the house by heading [N,S,N] and head back upstairs with [N,N,E,U,U]. Let us see what else we can find upstairs.

Go [S,S] into Dunbar's bathroom. Type OPEN CABINET and EXAMINE LOBLO. Hmmm. Now we go back downstairs and see if we can find Mr. McNabb to see if he knows anything about a ladder under the balcony. Go [N,N,D,D,W,S,S]. Then take a break for a while. Type WAIT UNTIL 11.30. Now for Mr. McNabb. Let us try the garden path first with [E,NE,E,W]. If McNabb is not around, just wait for a while or snoop around the area and he will soon show up.

DEADLINE is very unpredictable about the comings and goings of some characters, but also very particular about certain times for certain events. Best to question Mr McNabb between 11.30 and noon, for the best results. Once you spot McNabb, go to him and say HEY MCNABB followed by WHAT IS WRONG. He will tell you about some holes he found in his garden, so naturally you say SHOW ME THE HOLES. He will take off and you FOLLOW HIM. When he stops, type EXAMINE HOLES. Eureka! The ladder was here and the depth of the holes proves somebody climbed it up to the balcony! To make sure we cover every angle, type EXAMINE GROUND and DIG AROUND HOLES. I wonder what this could be about? To find out, type ANALYZE FRAGMENT FOR LOBLO. Keep an eye on the time, at noon you must be in the living room for the reading of the will. So go [N,SW,SE,E] to the house and [N,N,W] into the living room.

Now, just WAIT for the will to be read. After the will is read, you decide to see if you can roust some of those present into giving you some clues as to the guilty party and perhaps the motive for the crime. Let us start with George. Type SHOW GEORGE THE CALENDAR. He will get very nervous and start heading out of the room. Type FOLLOW HIM until he finally goes to his room. He will keep telling you to leave him alone, but just keep at him until he enters his room. At this point, you decide to see if George knows more than he is telling. You are not going to get anything from him here, so let us go to the balcony and wait to see if he does anything. Go [W,N,N] to the balcony.

Now type WAIT 10 MINUTES. Voila! Here he comes! Wait until he goes behind the bookshelf and then type WAIT 4 MINUTES to give him time to really get his hands into the cookie jar. When your 4 minutes are up, go [S], EXAMINE BOOKSHELF, PRESS BUTTON and go [E]. Caught him redhanded! Type GET WILL, LOOK SAFE, GET PAPERS and READ PAPERS. Things are beginning to hot up! Let us see if we can substantiate some of this stuff. Go back to the living room with [W,S,E,E,E,E,D,D,W,W,].

You should find a cosy gathering of characters. Type HEY BAXTER, WHAT ABOUT FOCUS. You know he is lying so you SHOW PAPERS TO BAXTER. Ah, that is better! Now for some clever psycho-detective work. Type SHOW LAB REPORT TO DUNBAR and SHOW LAB REPORT TO BAXTER. Whip around and ACCUSE DUNBAR. She will be a bit nervous (shame huh?) Perhaps we should go off and wait to see what develops. Go [E] to leave the room and WAIT FOR DUNBAR. Just as we suspected! When she passes you, type FOLLOW HER. Once outside the house she should drop a ticket. It is vital, so make sure you do it right. Type GET TICKET and READ TICKET. This will tip the scales of justice. Type SHOW TICKET TO DUNBAR. You know you have got her on the run, so head off to the shed to wait and see what happens. It is [E,E,SE] to the shed.

Now, WAIT FOR BAXTER. When they both show up, SHOW TICKET TO BAXTER then the moment you have been waiting for, ARREST BAXTER AND DUNBAR. You did not believe them for a minute, did you?

Due to the dynamic nature of DEADLINE, there are several possible endings, none better than this one, that I have found to date. The secret in getting the conviction of both Baxter and Dunbar is to get them separately and confront them with the ticket stub, to discount each others alibi. Otherwise you will get a hung jury. If you manage this particular possible outcome, you will be given the choice of seeing the Author's version of the events. Congratulations, you have completed DEADLINE! o

Infidel by Infocom

Solution by Ian of TIBBS

You are lying in your cot, trying to shake off the effects of a drug given by your absconding workers. Since time is of the essence, just get up, leave the tent, and make your way due South to the Work Tent. Along the way, pick up the matches near the fire pit. The tin foil, which is in the empty cigarette pack, can be safely ignored. While you are doing this a plane will appear overhead. It has the navigation box you have been waiting for. You have to dig where the "X" is. In the Work Tent is a knapsack that contains a rope and a canteen. Open the sack and get the canteen, then get the sack, which will automatically go over your shoulders. Return to the fire pit then go due West to the Supply Tent.

Take both the axe and the shovel, step outside and walk North and West to a river bank. Open the canteen and fill it with water. Head East and you will be outside your tent again. Enter it and break the lock on the trunk. Get and drop the lock, then open the trunk. Inside is some food, a map and an inspection sticker. Get the food and the map. Inside the map is a stone cube, which you will soon need to enter the pyramid. The map you can drop, and the sticker you do not need though you might want to read it before moving on. Now, leave the tent and Go East to the North Path. Follow the path South until you come to the South Path. From here, go East twice. You are now at the spot where the pyramid is buried.

Start digging until you find the top of the pyramid with the square hole in it. When that appears, put the cube in the hole and the door to the pyramid will open. Drop the shovel, since you will not be needing it anymore (also the box, if you have it with you). Go down into the pyramid. You stand in the Chamber of Ra, near an altar. Drop your sack and get the rope. Tie the rope to the altar, and then throw the rope North. You will be climbing down that way soon. In the meantime, get the torch and the jar. Open the jar, which has oil inside, and dip the torch in it. Light a match, then light the torch. Close the jar and put that and the matches in the sack. Now get the sack and you are ready to explore. (Note; somewhere along the way you will get hungry. When that happens, just eat the beef and drink a little water.). Now, climb down the rope and you will be in the Circular Room.

Here you see a golden cluster, a statue and four doorways with counter balanced doors. If you attempt to go down any of the passage-ways , you will find that a door will descend and prevent you from going more than about halfway along the corridor. Here I might add that things are weighted heavily against you, almost everywhere there are traps to be sprung by unsuspecting grave-robbers (like you!). Be careful in your movements, so make frequent Game Saves.

So, what you need to do is to find a way of keeping the doors up while you explore the passages. Fortunately, there is an easy, if tedious, way to do this. Roll the statue towards one of the passage-ways (for example; "Roll statue NW"). The first time the statue will fall and the head will break off (ho hum!). Get the head, then roll the statue into a passageway, make sure you drop the head with the statue as well. Now you can go to the opposite passageway and pick up the treasure that is there. You will have to roll the statue (and do not forget the head!) into each of the four passages in turn so that you can get all four of the jewelled clusters. To continue with the example, so you know exactly what has to be done, after rolling the statue into the NW passage go back to the SE passage and you will be able to proceed to the room that has the opal cluster of Neith. As you get each cluster, drop it off in the Circular Room. When you have all five clusters, drop the sack and put all the clusters in it. The gold one is just a Treasure, but the other four will have a very important purpose later. For now, climb back up the rope into the Chamber of Ra.

OK, now we are going to visit the Barge Room and its environs. Head along South to the landing and continue down to the Narrow Hall. From there, go NE and then NW and you are in the Barge Chamber. You will be visiting the barge itself in a short while; for the moment, go West, then North, then East and you will be behind the Barge where a hallway starts. Go all the way North along this hallway to the Inner Chamber. Ignore the corpse, which is wearing a jewelled ring. The ring is not treasure in the game (no points for getting it) and is in fact a deadly trap, so it is best not to touch it. From the Inner Chamber go West to the Golden Room and then South to the Golden Alcove. Pick up the Gold Chalice, then return to the Inner Chamber, then East to the Silver Room, South to the Silver Alcove. Pick up the silver chalice and return to the barge. Once in the Barge Chamber, go back to the front of the boat. Enter it by going North.

You are in the middle of the barge and there is a mast here. Go East into the Aft Cabin, Down into the hold, then West into the West End of the hold. Here the mast ends in a slot. If you look in the slot you will see a piece of wood (called a "shim") wedged in the slot to hold the mast in place. Get the shim and drop it. Return to the deck and get the beam. If you like, you can go West to enter the Fore Cabin to read the little scroll of hieroglyphics, although that is not necessary. Go back to the Chamber of Ra. From there go East into the Cube Room, then West, then South. You will be in front of a panel with some bricks in it. Remove and drop the First, Third and Fifth bricks. When you remove the Fifth one a secret passageway to the East will open. Go through it to the Turning Passage, then down to the bottom of the stairs. Use our pick to dig the plaster away from the door, then go West through the Narrow Passage until you come to the room with the two niches. Here is where the fun starts.

Put the beam in the niches, then stand on it. Dig away the plaster with your axe. As you do so the floor will fall away! Since you are standing on the beam, you are safe. OK , now open the door and go West into the antechamber. Since there are still a few other sneaky things to avoid, get the beam then go South. This is another antechamber with a door in the west wall. Of course, there is also a trap here for the unwary so put the beam in the door. Now you can open the door safely. Go West into the Slab Room. (Note; somewhere along the line here your torch will start to sputter. When that happens, get the jar and the matches from your pack. Light a match then turn off the torch and dip it in the oil. Your match will go out, but you will then have time to light another one so you can light the torch. After this you can leave the jar and matches because you are near the end of the game and will not need them anymore).

Here is a slab with holes in each of its four corners. In fact, the placement of the holes is very reminiscent of the Circular Room. So, drop the pack and take out the four jewelled clusters (the gold one you can leave in there). Now, put the clusters in the Slab as follows: Diamond in the First Hole; Ruby in the Second Hole; Emerald in the Third hole; Opal in the Fourth Hole. As each one is inserted into its hole there will be a click from the slab. Once the last cluster is in place you can raise the slab. Inside you will find a golden spatula and a book. Get the book. The spatula can be left in the slab, unless you want to read the hieroglyphics in the book (you need the spatula to turn the pages). In any case, you have what you came for, so pick up the pack and go East back to the antechamber and get the beam. Now go North twice to the antechamber with the timbered door. As you may have guessed, this door is also trapped. Put the beam under the timber, then break the seal on the door. (That beam certainly comes in handy!). Now, open the door and go North into the Burial Chamber then East into the Treasury where the scales are.

You must balance the scales to be exactly even with the table top in order to safely take the scarab. So, drop the sack and get the two chalices. Put the Gold Chalice on either of the scales. Fill the silver chalice with water and put it on the other scale. Now you can get the scarab. Return to the Burial Chamber. Place the book in the large recess and the scarab in the small recess. There will be a click from the statues that hold the sarcophagus cover in place. Now, turn the statues in the following order: Neith, Selkis, Isis, Nephthys.

At this point you have your perfect score. If you continue the game by opening the sarcophagus you will die; there is no way around that. You can also go back to the surface through the stairs in the Chamber of Ra, but all you can do then is wander around in the desert. Of course, you can quit the game here and receive your Master Adventurer rating! Congratulations! You have finished INFIDEL.

Games Info

by Robert Brown and Stephen Judd

Welcome to yet another GAMES INFO. We would like to think that many people read these articles, but it is hard to tell, because we rarely receive any feedback. If you wish to ask for a review of anything, please contact us and we will be happy to do a review on it.

This month we will continue our review on SPAD XIII (mark II) and also do a review of Legends 1.1.

In the coming months we will be doing reviews on karate, Leather Goddess of Phobas and many other exciting new stuff.

Here is SPAD.....

The horizon is a dark green line, with white line of clouds always above and parallel to it. The sun is in the south eastern sky. It is a yellow burst of lines.

On the ground there are:
your plane's shadow;
trees, dark green;
your airfield, dark green parallel lines with one black hangar;
enemy airfield, 2 hangars and 1 parked plane;
trenches - jagged dark yellow lines, running north-south;
Seine River - Dark blue, running north-south;
Eiffel Tower - black multi-lined tower
French villages - green multiple boxes with church steeple at centre.
destroyed villages - as above, but black;
hills - polygonal green shapes.

The skies are infested with:
German 2 seaters - black; defended by observer machine gun;
observation balloons - black and cigar shapes;
your bombs - black, destroy German hangars.

(a) Take off
Press Key 1, to get the instrumental view and apply full throttle. When your airspeed equals 100 knots, pull back on the stick, until the screen moves then let go. Depress the throttle a few notches. Do not attempt any turns until you have reached 100 feet.

(b) Climbing
Look out your front view and notice that the dark green line is below your gunsight. This means you are climbing. If you raise your nose to much with the elevator, you airspeed will drop dramatically and you will stall.

(c) Descending and Landing
Use the ailerons to move the airfield to the centre of the view, being careful not to let the plane bank too far right or left. The rudder can be used for fine adjustment. As the airfield comes up, lower the throttle even more to keep the approaching near end in your gunsight. Make sure that your wings are level. Pull back on the stick at 50 feet once to keep your airspeed less than 100 knots and your nose up. It is not necessary to land precisely parallel at the airstrip since it is all level ground. You have landed when your altitude is zero and your nose rises abruptly. You must taxi to an area near the hangar and let your speed drop to zero. When you have done this the plane will be serviced, fuel, bombs, and ammunition replenished, and the plane will move to the starting point, facing due north.

ACROBATICS

(a) Turns and Extreme turns.
At full throttle you can turn 40 degrees (right or left stick * 5) without losing altitude, but turns of 45 degrees or more require you to pull back on the stick to stay level.

(b) Climbing and Stalling
The best way to climb is just to put the throttle up. The conditions for a stall are met when your air speed goes below 40 knots. When you stall your plane it will lose its lift catastrophically and will plunge down and sideways. Depending on the conditions, this plunge may or may not involve into a wild downward spin. The best way to handle these out of control situations is to lower the throttle first and to turn counter to the spin.

(c) Barrel roll
A barrel roll is when you use your ailerons to roll the aircraft through 360 degrees. To do a barrel roll, top your throttle, put your nose down with the cross-hairs well below the horizon, until your speed reaches 150 knots. Let your nose come up until the cloud line is below your guns. Push the "A" key for a left barrel roll and the "F" for a right barrel roll and do not let up until you are level again.

(d) Loops
The simple loop is also very easy. Get to top throttle, push the stick forward until you reach 150 knots, and press the "V" key until you have gone all the way around.

You are a pilot of a brand new SPAD XIII, assigned to the 94th squadron of the air service of the American forces. Your air field located on the eastern outskirts of Paris is a small strip of land with one hangar. The plane is fully fueled and ready to go, carrying 5 bombs and 1000 rounds of ammunition for your machine guns (enough for 100 bursts).

Enemies:
The German airfield is recognised by its 2 hangars and a plane on the ground, it
(1) German 2 seater
(2) Observation balloons. There are 10 of these between 5000 and 1500 feet, forming a straight line just behind the German trenches. Each one is protected by a flak battery.
(3) The enemy airfield, located due east behind the line of the balloons, is also protected by flak. Your principle mission is to bomb the two hangars there.

Damage:
Either the flak or the machine guns of the observation planes can damage the Spad. The least severe damage will be the destruction of the rudder control. The next level of damage is the destruction of the elevator. (With these 2 destructions, it is still possible to get home.) If your engine is hit it will splutter and lose power. The worst damage you can suffer is to the ailerons. When the control line is severed, the ailerons will fall into a hard left bank. This is almost uncontrollable, although pressing the extreme right key "F" and pulling the stick back the the right "C" can level you. When more than one of these damages occur it is very unlikely that the plane can land safely.

Repairs:
Landing in your hangar refuels and re-arms, repairs damage you have done, and lets the Germans repair their hangars if you have bombed them.

End of the game:
To end the game successfully you must
1) Land undamaged anywhere and press the "K" key,
2) Land damaged on French territory, or
3) Land damaged on German territory and escape capture.

continued on page 24

```
10 REM ********************
20 REM ***                  ***
30 REM *** BRAIN TEASER ***
40 REM ***                  ***
50 REM ********************
60 GOSUB 110
70 GOSUB 770
80 GOSUB 1000
90 GOSUB 2410
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 730
150 PRINT "IN THIS GAME YOU
ARE GIVEN A3x3 GAME BOARD WI
TH AN X IN THE CENTER.  THE
BOARD WILL RESEMBLE THIS:"
160 PRINT
170 C=14
180 PRINT TAB(C);"---"
190 PRINT TAB(C);"-X-"
200 PRINT TAB(C);"---"
210 PRINT
220 PRINT "THE TRICK IS TO M
OVE SO THATTHE GAME BOARD LO
OKS LIKE.  THIS:"
230 PRINT
240 PRINT "    XXX      AND
    ---       X-X   NOT LIK
E   ---       XXX     THIS:
    ---"
250 FOR I=1 TO 4
260 PRINT
270 NEXT I
280 GOSUB 710
290 GOSUB 730
300 PRINT "YOU MAY ONLY MOVE
 TO AN     OCCUPIED SPACE (A
 SPACE WITHAN X ON IT.)"
310 PRINT
320 PRINT "WHEN YOU MOVE, CE
RTAIN      SQUARES WILL REVE
RSE THEIR  CONDITION (CHANGE
 FROM AN X"
330 PRINT "TO A BLANK, OR VI
CE VERSA.)"
340 FOR I=1 TO 12
350 PRINT
360 NEXT I
370 GOSUB 710
380 GOSUB 730
390 PRINT "IF YOU MOVE TO A
CORNER,    ALL OF THE ADJACE
NT SQUARES REVERSE."
400 PRINT
410 PRINT "IF YOU MOVE TO TH
E MIDDLE OFA SIDE, ALL OF TH
E SQUARES  ON THAT SIDE WILL
 REVERSE."
420 PRINT
430 PRINT "IF YOU CHOOSE THE
 CENTER    SQUARE, THAT BOX
AND THE    FOUR MIDDLE BOXES
 WILL ALL  REVERSE."
440 FOR I=1 TO 8
450 PRINT
460 NEXT I
470 GOSUB 710
480 GOSUB 730
490 PRINT "HERE IS A QUICK R
EVIEW OF   THE VARIOUS MOVES
, AND THE  RESULTING REVERSA
LS."
500 PRINT
510 PRINT "    M*-
    ---       **-      AND
    ---       ---
    *M*"
520 PRINT
530 PRINT "    --*
    -*-       --M      AND
    *M*       --*
    -*-"
540 PRINT
550 PRINT "THE M DENOTES THE
 MOVE, AND THE ASTERISKS DEN
OTE WHICH  SQUARES WILL REVE
RSE."
560 FOR I=1 TO 5
570 PRINT
580 NEXT I
590 GOSUB 710
600 GOSUB 730
610 PRINT "THE SQUARE THAT Y
OU MOVE TO WILL ALSO BE REVE
RSED."
620 PRINT
630 PRINT "THE BOARD IS DESI
GNATED BY  POSITION:"
640 PRINT
650 PRINT TAB(C);"123"
660 PRINT TAB(C);"456"
670 PRINT TAB(C);"789"
680 FOR I=1 TO 11
690 PRINT
700 NEXT I
710 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
720 RETURN
730 CALL CLEAR
740 PRINT TAB(5);"*** BRAIN
TEASER ***"
750 PRINT
760 RETURN
770 REM
780 REM ***SETUP***
790 REM
800 CALL CLEAR
810 DIM GB(9)
820 CALL COLOR(9,5,5)
830 PRINT "ENTER THE NUMBER
OF THE     SQUARE TO WHICH Y
OU WANT TO MOVE (1-9)."
840 REM  DRAW GAME BOARD
850 FOR I=1 TO 7
860 CALL HCHAR(6+I,14,96,7)
870 NEXT I
880 FOR I=2 TO 6 STEP 2
890 FOR J=2 TO 6 STEP 2
900 CALL HCHAR(6+I,13+J,32)
910 NEXT J
920 NEXT I
930 CALL HCHAR(10,17,88)
940 FOR I=1 TO 9
950 GB(I)=32
960 NEXT I
970 GB(5)=88
980 RV=56
990 RETURN
1000 REM
1010 REM ***PLAY***
1020 REM
1030 CALL KEY(0,N,T)
1040 IF T=0 THEN 1030
1050 N=N-48
1060 IF (N<1)+(N>9)THEN 2210
1070 IF GB(N)<>88 THEN 2210
1080 ON N GOTO 1110,1240,133
0,1460,1550,1630,1720,1850,1
940
1090 IF N=5 THEN 1570
1100 IF N<>1 THEN 1140
1110 GB(1)=32
1120 IF GB(2)=32 THEN 1140
1130 RV=-56
1140 GB(2)=GB(2)+RV
1150 RV=56
1160 IF GB(4)=32 THEN 1180
1170 RV=-56
1180 GB(4)=GB(4)+RV
1190 RV=56
1200 IF GB(5)=32 THEN 1220
1210 RV=-56
1220 GB(5)=GB(5)+RV
1230 GOTO 2060
1240 GB(2)=32
1250 IF GB(1)=32 THEN 1270
1260 RV=-56
1270 GB(1)=GB(1)+RV
1280 RV=56
1290 IF GB(3)=32 THEN 1310
1300 RV=-56
1310 GB(3)=GB(3)+RV
1320 GOTO 2060
1330 GB(3)=32
1340 IF GB(2)=32 THEN 1360
1350 RV=-56
1360 GB(2)=GB(2)+RV
1370 RV=56
1380 IF GB(5)=32 THEN 1400
1390 RV=-56
1400 GB(5)=GB(5)+RV
1410 RV=56
1420 IF GB(6)=32 THEN 1440
1430 RV=-56
1440 GB(6)=GB(6)+RV
1450 GOTO 2060
1460 GB(4)=32
1470 IF GB(1)=32 THEN 1490
1480 RV=-56
1490 GB(1)=GB(1)+RV
1500 RV=56
1510 IF GB(7)=32 THEN 1530
1520 RV=-56
1530 GB(7)=GB(7)+RV
1540 GOTO 2060
1550 GB(5)=32
1560 FOR I=2 TO 8 STEP 2
1570 IF GB(I)=32 THEN 1590
1580 RV=-56
1590 GB(I)=GB(I)+RV
1600 RV=56
1610 NEXT I
1620 GOTO 2060
1630 GB(6)=32
1640 IF GB(3)=32 THEN 1660
1650 RV=-56
1660 GB(3)=GB(3)+RV
1670 RV=56
1680 IF GB(9)=32 THEN 1700
1690 RV=-56
1700 GB(9)=GB(9)+RV
1710 GOTO 2060
1720 GB(7)=32
1730 IF GB(4)=32 THEN 1750
1740 RV=-56
1750 GB(4)=GB(4)+RV
1760 RV=56
1770 IF GB(5)=32 THEN 1790
1780 RV=-56
1790 GB(5)=GB(5)+RV
1800 RV=56
1810 IF GB(8)=32 THEN 1830
1820 RV=-56
1830 GB(8)=GB(8)+RV
1840 GOTO 2060
1850 GB(8)=32
1860 IF GB(7)=32 THEN 1880
1870 RV=-56
1880 GB(7)=GB(7)+RV
1890 RV=56
1900 IF GB(9)=32 THEN 1920
1910 RV=-56
1920 GB(9)=GB(9)+RV
1930 GOTO 2060
1940 GB(9)=32
1950 IF GB(5)=32 THEN 1970
1960 RV=-56
1970 GB(5)=GB(5)+RV
1980 RV=56
1990 IF GB(6)=32 THEN 2010
2000 RV=-56
2010 GB(6)=GB(6)+RV
2020 RV=56
2030 IF GB(8)=32 THEN 2050
2040 RV=-56
2050 GB(8)=GB(8)+RV
2060 A=0
2070 FOR I=1 TO 3
2080 FOR J=1 TO 3
2090 A=A+1
2100 CALL HCHAR(6+2*I,13+2*J
,GB(A))
2110 NEXT J
```

continued on page 23

```
100 REM  ********************
110 REM  *COUNTY FAIR DERBY*
120 REM  ********************
130 REM 99'ER VERSION 1.6.1-
48
140 CALL CLEAR
150 CALL COLOR(2,2,14)
160 FOR I=3 TO 8
170 CALL COLOR(I,2,12)
180 NEXT I
190 CALL HCHAR(24,2,42,29)
200 PRINT
210 PRINT TAB(8);"COUNTY FAI
R DERBY": :
220 PRINT TAB(8);"A FIVE HOR
SE RACE": :
230 PRINT TAB(4);"YOU CAN BE
T FOUR WAYS:": :
240 PRINT "<1>WIN        PAYS
 4 TO 1": :
250 PRINT "<2>PLACE      PAYS
 3 TO 2": :
260 PRINT "<3>SHOW       PAYS
 2 TO 3": :
270 PRINT "<4>PARLAY     PAYS
 15 TO 1": :
280 CALL HCHAR(24,9,42,14)
290 PRINT : :
300 PRINT "PARLAY<PICK 1ST;
 AND 2ND;>": :
310 PRINT "EACH PLAYER IS GI
VEN $200": :
320 CALL HCHAR(24,2,42,29)
330 CALL VCHAR(1,2,42,24)
340 CALL VCHAR(1,30,42,24)
350 RESTORE 4370
360 READ DU,NO
370 IF DU=0 THEN 400
380 CALL SOUND(300*DU,NO,5)
390 GOTO 360
400 PRINT "PRESS ANY KEY"
410 CALL KEY(0,KEY,STAT)
420 IF STAT=0 THEN 410
430 CALL CLEAR
440 PRINT TAB(6);"*** HANG O
N***": : : :
450 PRINT TAB(7);"GOTTA GET
THE": : :
460 PRINT TAB(11);"HORSES":
: : : : :
470 DIM H$(50)
480 H$(1)="0000000040201OOF"
490 H$(2)="000008080F1F30F0"
500 H$(3)="0F0F102040000000"
510 H$(4)="F0F0080402000000"
520 H$(5)="0000000000000O7F"
530 H$(6)="00000000601E3EF0"
540 H$(7)="0F0F080402000000"
550 H$(8)="F0F0102040000000"
560 H$(9)="0000000103070101"
570 H$(10)="00000F1F30000007
"
580 H$(11)="00001F3F31000003
"
590 H$(12)="0000000001030600
"
600 H$(13)="0000070706060707
"
610 H$(14)="0000808080808080
"
620 H$(15)="0000C0E0606060E0
"
630 H$(16)="0000C0E0F07070E0
"
640 H$(17)="000060E0E0606060
"
650 H$(18)="0000F0F00000C0F0
"
660 H$(19)="0101010107070000
"
670 H$(20)="2F1830303F3F0000
"
680 H$(21)="030000313F1F0000
"
690 H$(22)="0F0F0000000000O0
"
700 H$(23)="0000060702010000
"
710 H$(24)="80808080E0E00000
"
720 H$(25)="80000000E0E00000
"
730 H$(26)="E07070F0E0C00000
"
740 H$(27)="F1F1606060600000
"
750 H$(28)="71111111F0E00000
"
760 H$(29)="0000303030303131
"
770 H$(30)="00000C0C0C0C8C8C
"
780 H$(31)="00003C3C18181818
"
790 H$(32)="0000303030C3E3733
"
800 H$(33)="00000C0C0C0C0C8C
"
810 H$(34)="00003F3F3030303F
"
820 H$(35)="0000FCFC0C0000FC
"
830 H$(36)="1F1F1F1E1C1C0000
"
840 H$(37)="F8F8F8F838380000
"
850 H$(38)="181818183C3C0000
"
860 H$(39)="3130303030300000
"
870 H$(40)="CCEC7C3C1C0C0000
"
880 H$(41)="3F0000303F3F0000
"
890 H$(42)="FC0C0C0CFCFC0000
"
900 D=120
910 K=1
920 FOR D=D TO D+7
930 CALL CHAR(D,H$(K))
940 K=K+1
950 NEXT D
960 IF D>152 THEN 980
970 GOTO 910
980 CALL CLEAR
990 CALL COLOR(11,15,6)
1000 CALL COLOR(12,14,11)
1010 CALL COLOR(13,13,11)
1020 CALL COLOR(14,2,11)
1030 CALL COLOR(15,7,11)
1040 CALL COLOR(16,5,11)
1050 CALL COLOR(2,2,12)
1060 CALL CLEAR
1070 X=1
1080 CALL CLEAR
1090 PRINT "TYPE PLAYER'S NA
ME ?": : :
1100 PRINT "AFTER THE LAST P
LAYERS NAME": :
1110 PRINT "HAS BBEEN ENTERE
D TYPE LAST": : : :
1120 INPUT "NAME ?":NAME$(X)
1130 IF NAME$(X)="LAST" THEN
 1570
1140 IF X>=9 THEN 1160
1150 GOTO 1190
1160 PRINT "EIGHT IS THE MAX
.NUMOF PLAYERS"
1170 PRINT "TYPE LAST TO CON
TINUE"
1180 GOTO 1120
1190 TOT(X)=200
1200 CALL CLEAR
1210 GOSUB 1230
1220 GOTO 1080
1230 PRINT :"O.K. ":NAME$(X)
;" PICK A HORSE ?": :
1240 INPUT "HORSE ? ":HO(X)
1250 IF HO(X)>5 THEN 1270
1260 GOTO 1310
1270 GOSUB 1290
1280 GOTO 1240
1290 PRINT :"NUM. TOO BIG TR
Y AGAIN": :
1300 RETURN
1310 PRINT :"WHAT KIND OF BE
T ?<1 TO 4>": :
1320 PRINT "<1>=  WIN": :
1330 PRINT "<2>=  PLACE": :
1340 PRINT "<3>=  SHOW": :
1350 PRINT "<4>=  PARLAY": :
1360 INPUT "KIND ? ":KI(X)
1370 IF KI(X)>4 THEN 1400
1380 IF KI(X)=4 THEN 1500
1390 GOTO 1420
1400 GOSUB 1290
1410 GOTO 1360
1420 PRINT :"HOW MUCH DO YOU
 SET ?<$1 TO $200>": :
1430 INPUT "BET ? ":BET(X)
1440 IF BET(X)>200 THEN 1460
1450 GOTO 1480
1460 GOSUB 1290
1470 GOTO 1430
1480 X=X+1
1490 RETURN
1500 PRINT :"YOU PICKED NO."
;HO(X);"  TO WIN": :
1510 PRINT "WHICH HORSE TO P
LACE ?": :
1520 INPUT "PLACE ?":PA2(X)
1530 IF PA2(X)>5 THEN 1550
1540 GOTO 1420
1550 GOSUB 1290
1560 GOTO 1520
1570 CALL CLEAR
1580 PRINT "PRESS S TO START
"
1590 CALL COLOR(2,11,11)
1600 FOR X=1 TO 22
1610 PRINT
1620 NEXT X
1630 CALL CHAR(110,"81C36618
1866C381")
1640 CALL HCHAR(9,1,119,30)
1650 CALL HCHAR(20,1,119,30)
1660 X=10
1670 Y=2
1680 FOR A=1 TO 10
1690 CALL HCHAR(X,Y,42,29)
1700 X=X+1
1710 NEXT A
1720 RESTORE 4350.
1730 READ DU,NO
1740 IF DU=0 THEN 1770
1750 CALL SOUND(200*DU,NO,5)
1760 GOTO 1730
1770 CALL HCHAR(10,2,49)
1780 CALL HCHAR(12,2,50)
1790 CALL HCHAR(14,2,51)
1800 CALL HCHAR(16,2,52)
1810 CALL HCHAR(18,2,53)
1820 Z=1
1830 D=120
1840 R=10
1850 V=3
1860 GOSUB 2490
1870 D=128
1880 R=12
1890 V=3
1900 GOSUB 2490
1910 D=136
1920 R=14
1930 V=3
1940 GOSUB 2490
1950 D=144
1960 R=16
1970 V=3
1980 GOSUB 2490
1990 D=152
2000 R=18
2010 V=3
```

```
2020 GOSUB 2490
2030 Z=0
2040 A=10
2050 B=4
2060 I=16
2070 J=4
2080 E=12
2090 F=4
2100 O=18
2110 P=4
2120 G=14
2130 H=4
2140 K=0
2150 S=0
2160 CALL KEY(0,KEY,STATUS)
2170 IF STATUS=0 THEN 2160
2180 IF KEY=83 THEN 2200
2190 GOTO 2160
2200 RANDOMIZE
2210 N=INT(5*RND)+1
2220 ON N GOTO 2230,2280,233
0,2380,2430
2230 R=A
2240 V=B
2250 IF B=0 THEN 2200
2260 D=120
2270 GOTO 2470
2280 R=E
2290 V=F
2300 IF F=0 THEN 2200
2310 D=128
2320 GOTO 2470
2330 R=G
2340 V=H
2350 IF H=0 THEN 2200
2360 D=136
2370 GOTO 2470
2380 R=I
2390 V=J
2400 IF J=0 THEN 2200
2410 D=144
2420 GOTO 2470
2430 R=O
2440 V=P
2450 D=152
2460 IF P=0 THEN 2200
2470 CALL HCHAR(R,V-1,42)
2480 CALL HCHAR(R+1,V-1,42)
2490 CALL HCHAR(R,V,D)
2500 CALL HCHAR(R,V+1,D+1)
2510 CALL HCHAR(R+1,V,D+2)
2520 CALL HCHAR(R+1,V+1,D+3)
2530 CALL SOUND(5,700,2)
2540 IF Z=0 THEN 2560
2550 RETURN
2560 IF Q=1 THEN 2610
2570 Q=1
2580 V=V+1
2590 D=D+4
2600 GOTO 2470
2610 D=D-4
2620 Q=0
2630 IF V>28 THEN 2840
2640 V=V+1
2650 IF D=120 THEN 2720
2660 IF D=128 THEN 2750
2670 IF D=144 THEN 2780
2680 IF D=152 THEN 2810
2690 G=R
2700 H=V
2710 GOTO 2200
2720 A=R
2730 B=V
2740 GOTO 2200
2750 E=R
2760 F=V
2770 GOTO 2200
2780 I=R
2790 J=V
2800 GOTO 2200
2810 O=R
2820 P=V
2830 GOTO 2200
2840 D=(D-112)/8
2850 IF S<>0 THEN 3000
2860 S=D
2870 ON S GOSUB 2890,2910,29
30,2950,2970
2880 GOTO 2990
2890 CALL COLOR(9,2,14)
2900 RETURN
2910 CALL COLOR(9,15,13)
2920 RETURN
2930 CALL COLOR(9,15,2)
2940 RETURN
2950 CALL COLOR(9,2,7)
2960 RETURN
2970 CALL COLOR(9,2,5)
2980 RETURN
2990 ON S GOTO 3030,3050,307
0,3090,3110
3000 IF K<>0 THEN 3130
3010 K=D
3020 ON K GOTO 3030,3050,307
0,3090,3110
3030 B=0
3040 GOTO 2200
3050 F=0
3060 GOTO 2200
3070 H=0
3080 GOTO 2200
3090 J=0
3100 GOTO 2200
3110 P=0
3120 GOTO 2200
3130 R=22
3140 V=10
3150 X=S+8
3160 FOR Y=1 TO 4
3170 CALL CHAR((95+Y),H$(X))
3180 X=X+5
3190 NEXT Y
3200 FOR Y=1 TO 2
3210 CALL HCHAR(R,V,95+Y)
3220 V=V+1
3230 NEXT Y
3240 V=V-2
3250 R=R+1
3260 FOR Y=3 TO 4
3270 CALL HCHAR(R,V,95+Y)
3280 V=V+1
3290 NEXT Y
3300 CALL COLOR(10,15,6)
3310 CALL COLOR(11,15,6)
3320 R=22
3330 V=13
3340 Q=1
3350 Y=1
3360 FOR Y=Y TO Y+6
3370 CALL CHAR(103+Y,H$(28+Y
))
3380 CALL HCHAR(R,V,103+Y)
3390 V=V+1
3400 NEXT Y
3410 IF Q=0 THEN 3470
3420 R=23
3430 V=13
3440 Q=0
3450 Y=8
3460 GOTO 3360
3470 PRINT :TAB(7);K;"PLACES
"
3480 PRINT :TAB(7);D;"SHOWS"
: :
3490 PRINT "PRESS ANY KEY"
3500 CALL KEY(0,KEY,STATUS)
3510 IF STATUS=0 THEN 3500
3520 CALL COLOR(2,2,12)
3530 CALL CLEAR
3540 X=1
3550 IF NAME$(X)="LAST" THEN
 4130
3560 ON KI(X)GOTO 3570,3640,
3720,3810
3570 IF HO(X)=S THEN 3600
3580 GOSUB 3970
3590 GOTO 3880
3600 BET(X)=BET(X)*4
3610 BET(X)=INT(BET(X)*100+.
5)/100
3620 GOSUB 4090
3630 GOTO 3880
3640 IF HO(X)=S THEN 3680
3650 IF HO(X)=K THEN 3680
3660 GOSUB 3970
3670 GOTO 3880
3680 BET(X)=BET(X)*3/2
3690 BET(X)=INT(BET(X)*100+.
5)/100
3700 GOSUB 4090
3710 GOTO 3880
3720 IF HO(X)=S THEN 3770
3730 IF HO(X)=K THEN 3770
3740 IF HO(X)=D THEN 3770
3750 GOSUB 3970
3760 GOTO 3880
3770 BET(X)=BET(X)*2/3
3780 BET(X)=INT(BET(X)*100+.
5)/100
3790 GOSUB 4090
3800 GOTO 3880
3810 IF HO(X)<>S THEN 3830
3820 IF PA2(X)=K THEN 3850
3830 GOSUB 3970
3840 GOTO 3880
3850 BET(X)=BET(X)*15
3860 BET(X)=INT(BET(X)*100+.
5)/100
3870 GOSUB 4090
3880 X=X+1
3890 IF X>5 THEN 3550
3900 IF X>4 THEN 3920
3910 GOTO 3550
3920 GOTO 4130
3930 CALL CLEAR
3940 GOTO 3550
3950 IF X<=8 THEN 3550
3960 GOTO 3930
3970 IF TOT(X)<BET(X)THEN 40
20
3980 PRINT "SO SORRY ";NAME$
(X);"YOU LOSE $";BET(X)
3990 TOT(X)=TOT(X)-BET(X)
4000 PRINT "YOU NOW HAVE $";
TOT(X): :
4010 RETURN
4020 TTOT(X)-TOT(X)*-1
4030 PRINT "HEY*";NAME$(X);"
 YOU LOSE AGAIN *": :
4040 TOT(X)=TOT(X)-BET(X)
4050 TTOT(X)=TOT(X)*-1
4060 PRINT "YOU OWE THE TRAC
E $";TTOT(X): :
4070 PRINT "WE HOPE YOUR CRE
DIT IS GOOD": :
4080 RETURN
4090 TOT(X)=TOT(X)+BET(X)
4100 PRINT "BREAT*";NAME$(X)
;" YOU WIN $";BET(X)
4110 PRINT "YOU NOW HAVE $";
(TOT9X): :
4120 RETURN
4130 PRINT "PRESS ANY KEY"
4140 CALL KEY(0,KEY,STATUS)
4150 IF STATUS=0 THEN 4140
4160 IF NAME$(X)<>"LAST" THE
N 3930
4170 CALL CLEAR
4180 L(K)=L(K)+1
4190 U(D)=U(D)+1
4200 W(S)=W(S)+1
4210 PRINT TAB(8);"PAST RECO
RDS": : :
4220 PRINT "NO:1 ";W(1);"WIN
";L(1);"PLACE";U(1);"SHOW"
4230 PRINT :"NO:2 ";W(2);"WI
N";L(2);"PLACE";U(2);"SHOW"
4240 PRINT :"NO:3 ";W(3);"WI
N";L(3);"PLACE";U(4);"SHOW"
4250 PRINT :"NO:4 ";W(4);"WI
N";L(4);"PLACE";U(4);"SHOW"
4260 PRINT :"NO:5 ";W(5);"WI
N";L(5);"PLACE";U(5);"SHOW"
4270 PRINT : :"PRESS ENTER"
4280 CALL KEY(0,KEY,STATUS)
```

continued on page 23

```
10 REM ********************
20 REM ***                  ***
30 REM *** ARTIST BOARD ***
40 REM ***                  ***
50 REM ********************
60 GOSUB 100
70 GOSUB 490
80 GOSUB 650
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 450
140 PRINT "BY USING ARTIST B
OARD AND   YOUR CREATIVE TAL
ENT, YOU   CAN CREATE PICTUR
ES IN LOW-"
150 PRINT "RESOLUTION GRAPHI
CS."
160 PRINT
170 PRINT "THE 4 ARROW KEYS
DETERMINE  WHERE THE COLOR(S
) WILL     OCCUR.  YOU HAVE
THE TOOLS"
180 PRINT "TO CREATE SOME VE
RY NICE    DRAWINGS."
190 PRINT
200 PRINT "CHECK YOUR MANUAL
 TO SEE    WHICH COLOR IS RE
PRESENTED  BY WHICH NUMBER (
1-16)."
210 FOR I=1 TO 6
220 PRINT
230 NEXT I
240 GOSUB 430
250 GOSUB 450
260 PRINT "FOR PROGRAMMING R
EASONS,    ONLY THE FOLLOWIN
G COLORS   WILL BE AVAILABLE
; NOT 1 OR 2, BUT 3-16."
270 PRINT
280 PRINT "TO CHANGE COLORS,
 TYPE C    FOLLOWED BY THE N
UMBER OF   THE NEW COLOR."
290 FOR I=1 TO 12
300 PRINT
310 NEXT I
320 GOSUB 430
330 GOSUB 450
340 PRINT "TYPE M TO MOVE, Q
 TO QUIT,  AND Z TO ERASE."
350 PRINT
360 PRINT "IF YOU PRESS ONE
OF THE FOURARROW KEYS BEFORE
 TYPING M, THEN THE PREVIOUS
LY USED"
370 PRINT "COLOR WILL BE DRA
WN AT EACH SPOT YOU MOVE OVE
R."
380 PRINT
390 PRINT "THE BRUSH WILL BE
GIN IN THE MIDDLE OF THE SCR
EEN."
400 FOR I=1 TO 9
410 PRINT
420 NEXT I
430 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
440 RETURN
450 CALL CLEAR
460 PRINT TAB(5);"*** ARTIST
 BOARD ***"
470 PRINT
480 RETURN
490 REM
500 REM ***SETUP***
510 REM
520 CALL CLEAR
530 FOR I=3 TO 16
540 CALL COLOR(I,I,I)
550 NEXT I
560 X1=12
570 Y1=16
580 CO=70
590 CALL HCHAR(12,16,70)
600 CALL HCHAR(1,2,66,30)
610 CALL HCHAR(24,2,66,30)
620 CALL VCHAR(1,2,66,24)
630 CALL VCHAR(1,32,66,24)
640 RETURN
650 REM
660 REM ***PLAY***
670 REM
680 GOSUB 1310
690 CALL KEY(0,N,T)
700 IF T=0 THEN 680
710 IF N=81 THEN 1340
720 IF N=77 THEN 800
730 IF N=90 THEN 1010
740 IF N=69 THEN 1030
750 IF N=83 THEN 1060
760 IF N=68 THEN 1090
770 IF N=88 THEN 1120
780 IF N=67 THEN 1150
790 GOTO 680
800 CALL KEY(0,N,T)
810 GOSUB 1310
820 IF T=0 THEN 800
830 CALL SOUND(100,500,3)
840 IF N=68 THEN 890
850 IF N=69 THEN 920
860 IF N=83 THEN 950
870 IF N=88 THEN 980
880 GOTO 710
890 IF Y1=31 THEN 800
900 Y1=Y1+1
910 GOTO 800
920 IF X1=2 THEN 800
930 X1=X1-1
940 GOTO 800
950 IF Y1=3 THEN 800
960 Y1=Y1-1
970 GOTO 800
980 IF X1=23 THEN 800
990 X1=X1+1
1000 GOTO 800
1010 CO=32
1020 GOTO 680
1030 IF X1=2 THEN 680
1040 X1=X1-1
1050 GOTO 1280
1060 IF Y1=3 THEN 680
1070 Y1=Y1-1
1080 GOTO 1280
1090 IF Y1=31 THEN 680
1100 Y1=Y1+1
1110 GOTO 1280
1120 IF X1=23 THEN 680
1130 X1=X1+1
1140 GOTO 1280
1150 CALL KEY(0,N1,T)
1160 IF T=0 THEN 1150
1170 IF T=-1 THEN 1150
1180 N1=N1-48
1190 IF (N1<1)+(N1=2)+(N1>9)
THEN 1150
1200 IF N1>1 THEN 1260
1210 N1=10
1220 CALL KEY(0,N2,T)
1230 IF T<>1 THEN 1220
1240 N2=N2-48
1250 IF (N2<1)+(N2>6)THEN 12
20
1260 CO=(N1+N2)*8+24
1270 N2=0
1280 CALL HCHAR(X1,Y1,CO)
1290 CALL SOUND(100,500,3)
1300 GOTO 680
1310 CALL GCHAR(X1,Y1,X2)
1320 CALL HCHAR(X1,Y1,42)
1330 CALL HCHAR(X1,Y1,X2)
1340 RETURN
```

continued from page 20

```
2120 NEXT I
2130 RV=56
2140 FOR I=1 TO 9
2150 IF GB(I)=32 THEN 2170
2160 GT=GT+1
2170 NEXT I
2180 IF (GB(5)=32)*(GT=8)THE
N 2370
2190 GT=0
2200 GOTO 2300
2210 FOR I=1 TO 9
2220 CALL HCHAR(3,12+I,ASC(S
EG$("TRY AGAIN",I,1)))
2230 NEXT I
2240 FOR I=1 TO 100
2250 NEXT I
2260 FOR I=1 TO 9
2270 CALL HCHAR(3,12+I,32)
2280 NEXT I
2290 GOTO 1030
2300 FOR I=1 TO 9
2310 IF GB(I)=32 THEN 2330
2320 GT=99
2330 NEXT I
2340 IF GT=0 THEN 2400
2350 GT=0
2360 GOTO 1030
2370 W=5
2380 FOR I=1 TO 400
2390 NEXT I
2400 RETURN
2410 REM
2420 REM ***END***
2430 REM
2440 IF W<>5 THEN 2540
2450 CALL CLEAR
2460 A$="CONGRATULATIONS!!!"
2470 FOR I=1 TO LEN(A$)
2480 CALL HCHAR(12,I+7,ASC(S
EG$(A$,I,1)))
2490 NEXT I
2500 FOR I=1 TO 200
2510 NEXT I
2520 PRINT "GOOD JOB!"
2530 PRINT "THIS GAME IS NOT
 EASY."
2540 RETURN
```

continued from page 22

```
4290 IF STATUS=0 THEN 4280
4300 CALL CLEAR
4310 X=1
4320 IF NAME$(X)="LAST" THEN
 1570
4330 GOSUB 1230
4340 GOTO 4320
4350 DATA 1,523,1,523,1,523,
1,440,1,440,1,440,1,349,1,44
0,1,349,2,256
4360 DATA 1,349,1,440,1,523,
1,523,1,523,1,440,1,440,1,44
0,1,256,1,256,1,330,2,239,0,
0
4370 DATA 1,392,1,392,1,392,
1,330,1,392,1,440,1,392,2,33
0,2,294,1,330,2,294
4380 DATA 1,392,1,392,1,392,
1,330,1,392,1,440,1,392,2,33
0,2,294,1,330,1,294,2,256,0,
0
```

continued from page 19

You only see the final score screen if you end the game successfully. At this time the number of enemy planes, observation balloons and enemy hangars will be tallied.

Extra sneaky buttons:

The move key "M" will extract those who frowned on the 10 to 15 minute trip to get the Spad over the German lines for a dogfight. You can use the key to see different areas of the countryside such as the Eiffel Tower, the German airfield and the French airfield.

Now here is Legends...

Legends is set on a fairly large island with different types of terrain. Actually its a lot like Ultima, a game on other inferior computers that some of you traitors out there may know of. Some terrains are grasslands (mildly dangerous), forests (dangerous), swamps (very dangerous), mountains (impassable), and water. Scattered over the island are Inns (good for resting players), teleporters and ruins. Your quest team consists of four players; a fighter, a ranger, a wizard and a cleric. The game comes with a team assembled and ready to go, but I prefer to build my own. Your choice of players will affect the rest of the game.

The quest in Legends is to close the Western Portal through which the Ashtar Creel has loosed all the dreadful monsters which you will have to deal with. To do this, you will need certain items which are hidden somewhere on the island or in the dungeon. Before you go too far from the city of spells, or they will not live long. To train your players at the adventurer's guild, you will need to accumulate experience, and gold to pay for the training. Experience is gained by fighting monsters (and winning). Gold is won in the training your players are extremely important in battle. With them you can inflict damage on your opponents, heal your players, protect yourself, and a number of other useful things.

After I have trained my players, I found that I could then hit three times each go. I felt bad when I walked into a forest and met the monsters that could also hit three times each. Balancing this is the spell "SLOW". Once you have trained enough, you can cast a "slow" spell at the monsters you are fighting, so that they can only hit once each. Unfortunately, they can also cast a "slow" spell at you. The only way to reverse this is by casting a "dispel magic" spell, which unfortunately, your wizard does not learn until level 2. Trying to stay alive at level 1 is a bit of a challenge.

What I like most about Legends is the sly sense of humour evident in the game e.g. Death Turnips! These are very dangerous monsters that live in the forest.

If you are interested in this program it is available from Asgard software in the US for US$27.95. If you would like to see it first, somebody in the club would probably have it.

If you have any queries please feel free to leave mail to GAMES on the BBS or to write to:

Games Information,
C/- Robert Brown
141 Beecroft Road,
Beecroft NSW, 2119

PS: Remember no commas or underlining. ◦

Second thoughts on TI Planner

(aka TI Calc) from DataBiotics

from Stephen Shaw, England

Following on from my earlier fairly good review, more serious use of the module (I did say I was definitely using it!) has revealed two further bugs which users must know about.

In putting together a long spreadsheet, a small period of slight panic was caused when the module filled the disk (any size disk!) and reported that it could not save the spread.

Try loading, no problem. Loaded, well, imperfectly, somewhat corrupted.

Many hours later, the precise nature of the problems comes to light, and now I am once more a very happy and satisfied user of this easy to use and reasonably fast spread sheet module.

First bug:

Filling the storage medium! (Whatever its size).

This is caused by placing anything on line W. Actually, if you only place something in cell W01 and leave the rest of line W blank, you merely have a corrupted spread! Anywhere else on line W and your storage medium will be filled and you will be told the save failed.

In practice you may be able to load your data back in, delete all contents in line W, and carry on.

So: lesson 1: Never use Line W.

(It may be unrelated, but Control W is used to change the size of an empty spread sheet. Pure coincidence. Is it?)

Second bug:

When writing a long BASIC program, you save it from time to time to protect yourself against lockouts! (After the first lock out you do anyway).

If you do this with TI Planner, you could find yourself with a hopelessly corrupted spreadsheet unless defensive action is taken.

When saving to disk with 32K attached, the user may select from a spreadsheet measuring 50x50, 40x63, or 35 x 72.

If you start with a 50x50 spread, the first time you save it, it is saved properly as a 50x50 spread. The next save will be 40x63, and the next will be 35x72. (Your screen will keep on telling you you have a 50x50 spread, it is only when you reload you find a different sheet size and corrupted formulae and possibly lost data.) If you start with a sheet measuring 35 x 72, the first save will be a correct 35 x 72, the second save will be 50x50 and the third save will be 40x63.

Examining the disk files, it is apparent that the first word is a flag which signals the sheet measurements. Apparently as you SAVE a sheet, an internal flag is incremented, and from then on your spreadsheet thinks it is a different size, unless you reset the (register?) by reloading the sheet.

So: lesson 2: everytime you save to disk or cassette, if you wish to do any more work on the sheet, immediately reload it!

These two defensive actions are now taken here, and no additional problems have been found. The spreadsheet works like a charm.

Further examining the files created by the module seems to show that every time you save, the data is saved three times, once in each format, with the current actual format first. Upon reloading however, only the first format is read. This is especially noticeable when using cassette, as the save may occupy a count of 15 on your tape indicator, but the load will terminate on a count of five!

Tape users may wish to note that 102 cells save to tape (in the UK) in about four minutes.

Tape users without prior experience of using tape DATA files should also be aware that cassette data can be accepted by the computer in a corrupted form if the tape playback volume is slightly off, unlike the usual BASIC programs, which will not corrupt without an error message. ◦

Rambles

by Stephen Shaw, England

Greetings fellow TI99/4A users. This issue of Rambles is going to be a short one, as there is less than a month between the text for this issue being required, and the text for the last one having been submitted. Plus the early weeks were spent in preparing for the Alternative Micro Show.

I have had very little mail in the last month (well, alright, my reader has totally failed to write this time!) leaving little scope for inspiration, so I shall be submitting some program listings to keep your fingers active.

Your questions are always welcome, especially from unexpanded owners and the address is:
10 Alstone Road, STOCKPORT, Cheshire, ENGLAND, SK4 5AH.
If a direct reply is required, please send an SAE (2 IRC if you are abroad).

I am sending a separate report on the Aston Show. Good to see so many of you (sorry the rest of you could not make it!). I prepared a host of goodies for the show, disks and tapes, and brought most of them home with me. Ah well. As a result, if anyone would like a look at a sample copy of MICROpendium, they can have a copy of the September 1988 issue from me for just 2 Pounds including postage. This is the issue with an article by some English TI99/4A owner whose name rings a bell or two.
And if you would then like to subscribe to MICROpendium, as all good TI99/4A owners should, you may either send your $$$ to them direct or if you prefer, you can send them to me. Drop me a line for the current amount of Pounds, the exchange rate is doing silly things again right now!

Turbo Pasc 99 mentioned in the last issue is available from:
L.L. CONNER ENTERPRISE:
1521 Ferry Street, LAFAYETTE, IN, USA, 47904.
Telephone: (317) 742 8146
Visa (Barclaycard) and MasterCard (Access) welcome.
The cost for a single copy to the UK by insured airmail is US$70.55. A direct mail offer to all our disk drive owning members resulted in just one enquiry, so we cannot pass on the user group discount offered to us.

Indeed, the last issue of TI*MES had several user offers, all of which have been very badly taken up, and will not therefore be repeated. Our members (you!) are not showing the collective interest to justify asking for discounts. It is a bit bad when we cannot even raise five orders! There was one exception, and that was TI-Base, which is surely going to be the (THE) TI99/4A database, as TI Artist is THE graphic program, and we were able to send two orders to Texaments for this.

If any member wants any TI99/4A product and cannot locate it, I can probably obtain a location for you, or if you prefer not to order direct from the USA, I am happy to order any software on your behalf, including any modules. Such orders are prepaid and taken at your risk! The minimum for a single module by import is likely to be ten pounds. If you desperately seek anything, do let me know, and do remember that SAE please!

I mentioned earlier this year a disk containing a VAT ACCOUNTING MODULE, an unreleased TI99/4A module, and indicated it had not been heard of. Correction. This module was listed as "forthcoming" in the first issue of TIHOME (circulation 14) and was then not heard of again!

At the Aston show, one member fought shy of obtaining a copy of Ray Kazmers Christmas Demo (free remember!) because he had it. Oh no he did not; he had only the initial 1987 issue, which left little Woodstock crying for his present. The 1988 release has Woodstock retrieve the parcel and open it! If you do not yet have this first class example of pure Extended BASIC programming, superb use of graphics, you can still send for it. Just a disk and return postage is all that is required. Sorry the 1988 version is far too long for cassette.

Query: can I load a CHARA1 disk file into an Extended BASIC program? Answer: of course, dig out issue 18 of TI*MES!

TI*MES contributions: if you cannot meet the requirements set out inside the front cover of the last issue (and I suspect many of our members cannot) this is not an excuse not to write anything! Write to me, Stephen Shaw, and I will transcribe as required (this may take a little longer than submitting camera ready text of course!). Handwritten A5 sheets of paper are fine, plain or ruled, anything legible!

Attention all you unexpanded owners. How about module reviews? Games tips? BASIC and Extended BASIC tips and queries? All owners, what programs do you use? Write in now, and keep writing in!

Members wants: there were some very finely priced sales offers in the last issue, did you see them? However as we only publish once every three months, that can mean a seller waiting a long time to find a buyer! If you are interested in second hand equipment, especially Expansion Boxes and disk drives, which we cannot afford to stockpile, please would you drop me a line and indicate: what you would like, how much you wish to pay as a maximum (please be reasonable) and if possible a telephone number. Then we can try to put you together with any sellers we hear of. I will pass details of your wants onto other relevant committee members, so you can report all your wants together to me. The equipment you want does become available (or we may be able to obtain new at a reasonable price) but we do need to know what it is you are looking for. This service will also assist sellers to find a buyer more quickly!

Did you type in the Paint program on page 15 of the last issue? If you do not have speech, just substitute DISPLAY AT(2,1):nnn" instead of CALL SAY("nnnn"). I would personally delete the "useless" bit anyway; I keep my computer suitably subservient! The original program was a masterpiece of compact programming, and Jim's amendments make it even nicer. Not typed it in? Get that issue out and try it now! It really is a good program!

No guidance on where to take the Enhanced (PRK) BASIC, so until I have at least a couple of letters to support the time it takes to put that lot together, no more. If you want an article I desperately need to know about it! Research time can be quite long, and a total silence afterwards is not encouraging...

Nobody has come forward to ask for the Lisp disk from the library so I guess I will forever remain ignorant of it. And despite lengthy write ups in TI*MES (and a lot of people taking c99 from the disk library) only one disk owner has asked for Turbo Pasc 99, a remarkably easy and friendly language. Is there anyone out there still programming? Why not submit your work during the last couple of years so we can see what you are doing! Do not be shy, you can teach us and maybe we can help you!

We presently have around 200 members, who as far as we can tell do not all have every TI99/4A product ever produced, so if you have any software or equipment not in use, why not offer it to your fellow members, or even to the Group! Drop the relevant librarian or myself a line indicating what you have and what you want. None of us have a lot of money these days, so please be reasonable on pricing! The group is not well funded and cannot afford to buy your equipment, but I am happy to stockpile it and I can at least repay you the postage cost!

Machine Code Rambles:

In an Extended BASIC program is it possible for the program to sense that the alpha-lock key is depressed or not without the user having to press a key? Yes, if you also have 32K RAM.

Here is the source code for a very short machine code routine:

```
       DEF    ALPHA
ALPHA  MOV    R12,@>FFFC save old R12
       CLR    R12        9901 CRU Base = 0
       SBZ    21     Signal alphalock key line
       TB     7     Check alphalock other side
       JNE    STATE      jump if state = on
       SETO   @>FFFE     state=off
       JMP    JUMPA      As off skip next line
STATE  CLR    @>FFFE     state=on
JUMPA  SBO    21       stop sending to alphalock key!
       MOV    @>FFFC,R12 restore old R12
* standard Extended BASIC return now
       SB     @>837C,@>837C clear error for BASIC
       B      @>0070 return to calling program
       END    ALPHA
```

To use this code, assemble it to a non-compacted object code, say DSK1.K/OB. In Extended BASIC, start your program with:

```
10 ! K by M Gikow, Andover, MA
11 ! August 1988.  Use with K/OB
12 CALL INIT :: CALL CLEAR :: CALL LOAD("DSK1.K/OB")
your program here
```

Now, an example of sensing the state of the alphalock:

```
100 GOSUB 4000
110 PRINT A
120 GOTO 100
4000 CALL LINK("ALPHA")
4001 CALL PEEK(-1,A)
4002 RETURN
5000 END
```

A word of further explanation may help, thanks to Mike Gikow for this: the keyboard is connected to the computer by a device (TMS9901) that uses the Communications Register Unit (CRU) which is accessed by CRU instructions such as SBZ, SBO and TB (Set Bit Zero, Set Bit One, and Test Bit).

The Technical Data Manual shows connections to the keyboard such that the alpha lock key is connected between the output for CRU bit 21 and the input for CRU bit 7.

When the switch is closed, the output can be read by the input.

This program tests the condition of the switch and writes a value of -1 or 0 to memory locations -1 and -2. The CLR instruction produces the 0, and SETO produces -1.

The CALL LINK statement in Extended BASIC then tests memory location -1 (=>FFFE) for a value of 0 or -1. Note that the PEEK statement will set A=255 rather than -1! So you will see either a 0 or 255.

==========

One heart felt cry, from more than one member, involves the printing of text files using TI-Writer. TI-Writer normally send a carriage return and line feed at the end of each line, and life is much easier if you switch the internal dip switches in your printer so that your printer does not itself add an automatic carriage return and line feed at the end of each line! If both the printer and TI-Writer add a line feed, you end up with double spacing, whether you want it or not.

The TI RS232 card contains a number of software switches, and if you use RS232 as your printer name you will certainly know all about those! However PIO users generally have little call to use them and so remain unaware!

You can instead of using just PIO, name your printer as PIO.CR or PIO.LF. If you use the Formatter, you may need to use PIO.LF even if PIO on its own is OK when printing from the Editor.

If you ever use a Graphics program, you will need to use PIO.CR to prevent an automatic carriage return every 80 characters, something that can make your graphics look a little untidy.

In case of difficulty with line feeds, go through all the possible printer names and use the one that is best, be it PIO.CR or PIO.LF or possibly even PIO.CR.LF It makes life easier if you can switch the auto line feed off at the printer. Consult your printer manual.

Another problem is that several members use printers which are not 100% compatible with the Epson control codes, and when printing text which has these embedded, you may meet problems, such as a printer freeze. Again, TI-Writer has been written to take care of this problem. You can instruct TI-Writer to print the text file but to strip out the control codes, that is to print only the "printable" characters ASCII 32 to 126. To do this, instead of using the output device name of PIO you use C PIO, that is, a capital C followed by a space followed by the normal printer name.

And as a reminder, if you want TI-Writer to save in DF80 format instead of DV80, select PF then type F DSK1.FILENAME. Use the print file command instead of the usual save file, and add an F and a space in front of the output device name. You remember of course that TI-Writer can load both DF80 and DV80 files! In fact it will even handle DF and DV files longer than 80, but only load the first 80 characters of each record. TI-Writer is an unusually well written program!

And remarks regarding TI-Writer also hold true for Funnelweb!

Reminder to PRbase users about the bugs in that database:

1. Output Device Name:
   When you initialise a disk (any disk) the sectors are filled with >E5's. When you select CREATE using PRbase, the default output device name appears, if you have nominated one previously. Otherwise you will see an apparently blank input field. Not so. What you cannot see is an input field full of >E5's, so if you type "PIO", the program will then try to use a device called PIO followed by several CHR$(229)'s and tell you you do not have any such device!
   This is mainly a problem when using PIO, as RS232 users usually make use of the cards software switches and put a full stop in there someplace. The card is so programmed that the excess >E5s are then ignored.
   PIO users should type PIO and then fill the input field with blank spaces by holding space bar down, or first use ERASE (FCTN[3]) before entering the printer name.

2. Bill Warren is aware that some users with TI disk controllers have been unable to format a double sided disk using the PR Base utility. Their solution is simple, use another disk manager to format your disks! (I suspect that Bill's problem may have been with the stand alone disk controllers which are for single sided use only? SS)

3. Mis-Alignment of columns in TAB reports:
   This time · to >00's.! When you type the header, do not use [D] to move to the right. It will leave >00's behind. You must type spaces with the space bar. Bill does not fill the screen with spaces, he used nulls !

Note: Bill is not working on updates to PRbase. Version 2.0 is his last. However Mike Dodd has written a version 2.1 which makes normal use of sectors 0 and 1, but is incompatible with Vn 2.0 data files. Both versions are available from the disk library.

continued on page 14

Archiver III Version 3.02

by Barry Boone

Notes on improvements from V2.4

At long last, v3.0 is finally out. Version 3.0 was completely re-written from scratch, since v2.4 was simply too large and complex to try and add other features without running out of memory. This is the primary reason for the long delay between v2.4 and v3.0, and also the reason that v2.4 did not have many of the options and improvements v3.0 now has.

The greatest improvement to v3.0 is that the Pack/Compress and Unpack/Decompress steps have been consolidated into one step, which means you may now simultaneously pack and compress a group of files, decompress and unpack a group of files, or even extract selected files from a compressed archive! The program is still 100% compatible with v2.4, and still allows the option of creating uncompressed archives. It will also automatically detect whether an archive is compressed or decompressed and react appropriately.

Finally, compression has been optimized, so that you may expect to get slightly higher compression yields over v2.4, which will vary according to the types of files you are compressing. For example, Arc v2.4 compressed 187 sectors of text down to 85 sectors, while v3.0 compressed it down to 78 sectors. The secret to this improvement is that v3.0 takes into account that the disk files, in general, waste a certain amount of space on each sector, which usually contains meaningless garbage that gets compressed along with the rest of the file. Version 3.0 fills this unused space with zeros, resulting in improved compression according to the amount of unused space in the files. Variable and certain Fixed files will show the best improvement, while program files, in general, will show little or no improvement.

A View text file option has been added in addition to the other file management options in version 3.0, and the input menu system is much more friendly. FCTN[9] (BACK) will still take you back to the main menu, and the main menu now contains all options available; the Utility menu was done away with. You may now press FCTN[8] (REDO) at any point while entering input, and you will be able to either re-enter the answers to the prompts, or accept those you previously typed by entering over them; the program now uses defaults everywhere. Also, the Copy Files command now places the first filename you enter as the default for the second, no longer forcing you to type the same filename twice.

Finally, by popular demand, the Catalog disk and Arcfile commands now allow you to dump catalogs to a printer, and also informs you of how many files were listed and how many sectors the files occupy, so that you can easily determine how many sectors are needed to Unpack/Decompress an archive.

A number of other changes have been made here and there as well, and some very minor bugs in v2.4 are no longer present. The internal sector buffer has been increased to 50 sectors, so that single disk users should not have to do as much disk swapping.

Simple Instructions for new users of Archiver III

Archiving of files is simply the process of taking a group of files and combining them into one single file, which can occupy much less space than the original file (compression from 30-70 percent is common, depending on the type of files you are dealing with). The advantage of archival is that you can store a group of related files together, and extract one or all of them at a later time; saving you considerable disk space, and allowing easy organization of your disk library, especially those programs you wish to keep, but don't use much, and also your backups of programs.

Modem users get an added incentive in that sending compressed archives over the phone can easily be 50% faster, (therefore 50% cheaper) than sending the original files.

To avoid confusion: compressed archives are stored in Internal/Fixed 128 format, and uncompressed archives are Display/Fixed 128. When uncompressed archives are created, they are automatically write protected, since most terminal programs (modem programs) require that D/F 128 files be write protected either to work, or to transfer at an acceptable speed.

When you select "1) archive files" from the main menu, you will be asked for pertinent information about the files you are about to archive. When you see the "Compress? (Y/N)" prompt, you are being asked whether you wish to apply data compression to the files you are archiving. In most cases the answer will be yes, however there are a very few kinds of files which cannot be compressed (attempting to compress them can actually make them larger! More on this later) so the option to not compress files is available.

If you choose the same disk drive for both input and output, you will be asked if you want to "Swap Disks". Answering Yes will cause you to be prompted to insert Source and Output disks at the appropriate times. Answering No will perform all operations on the same disk.

Extracting files is also quite simple. You may choose to remove any or all files from an archive by giving the appropriate instructions to Arc v3.0. The option "Catalog Arcfile" allows you to examine or print out the contents of an archive without having to extract it, and tells how many sectors are needed for extraction. The remaining options are self-explanatory, 'fill in the blank' commands.

The Load FW command is a loader for the Funnelweb program, and the filename it looks for is UTIL1, although if you are an experienced disk hacker, you may wish to alter it. I have reserved enough room there for a 10 character filename.

Files that are sufficiently random in nature may not compress. These are very rare, and are almost always already compressed in some way. Known offenders are GIF pictures and Infocom games. Both of these kind of files are already compressed, in fact, GIF pictures use the same compression technique Archiver III uses! Which brings me to the next point: do not bother trying to compress a file twice; it will just get larger the second time. Oh yes, Speech Synthesizer data tends to be unsqueezable as well, as it is itself squeezed using what is known as LPC code.

Miscellaneous

Unlike Archiver v2.4, I have about 2K to play with in Archiver III, room enough to add a few more things, so I am going to leave what will be added up to you: persons sending donations for Archiver III should make suggestions if there is anything you would like to see added to v3.0. Assuming there is sufficient response to Archiver III to make it worthwhile, I will add the most frequently asked for routines to Archiver III, assuming they will fit in the remaining memory.

Basically, unless response to Archiver increases considerably, there will probably not be any future releases of Archiver from me (or anything else for that matter). I am a full time college student, and anything you send would help. Even a note saying how much you like the program, but that you cannot afford to send anything would at least reassure me that this work is appreciated! I would like to take this opportunity to thank those of you who contributed and supported the Archiver II series, especially the 5 User Groups who took group donations for the program. Without your support and suggestions, Archiver III could have never have been created. If you write me with questions or expect a reply, please allow a month or two, as the letters go to my parents, and I generally do not see the letters until I return home on break.

continued on page 4

AT Music Card

by Peter Schubert

Specifications

The AT Music Card uses 4 sound chips similar to the one found in our TI99/4A Console to provide 16 programmable sound channels. Twelve of these are tone generators, and the other 4 are noise generators. Each channel is accessed by address decoding in the same way that the Console produces sounds, only the addresses are different. The 16 channels are grouped into two lots of 8 channels combined to output into 2 RCA connectors which are intended to connect to a stereo Hi-fi.

The board plugs into the side of the Console and has provision for other peripherals to plug into the other side similar to the Speech Synthesizer. The PE expansion Box also plugs onto it. Those who have a Mini-PE System can also piggy-back the board onto it.

Forti Compatibility

This card is designed to be software compatible with the Forti Music System and any software written for this commercial product that was once produced in the US. A review of the software was printed in the March 87 TND by Arto Heino. A disk of demonstration software is available to show the capability of the card. A future article will cover this in more detail.

Parts List (see circuit diagram)

U1	74LS245	IC
U2	74LS161	IC
U3	74LS04	IC
U4	74LS27	IC
U5	74LS20	IC
U6	74LS32	IC
U7,U8, U9,U10	76494	IC
XL	3.5795Mhz	crystal
5	10MF 16V tag	capacitor
2	0.1MF mono	capacitor
1	0.01MF ceramic	capacitor
2	1K2 1/4W	resistor
4	330R 1/4W	resistor
2	PCB mount RCA	connector
1	PC board	AT Music board

Parts kit; available from TIsHUG Shop or from me for $65 contains all above parts including professionally made PC board with plated through holes. Due to a shortage of sound chips only 16 kits will be made available. If you can obtain the 4 sound chips yourself a few PCBs will be available separately from me.

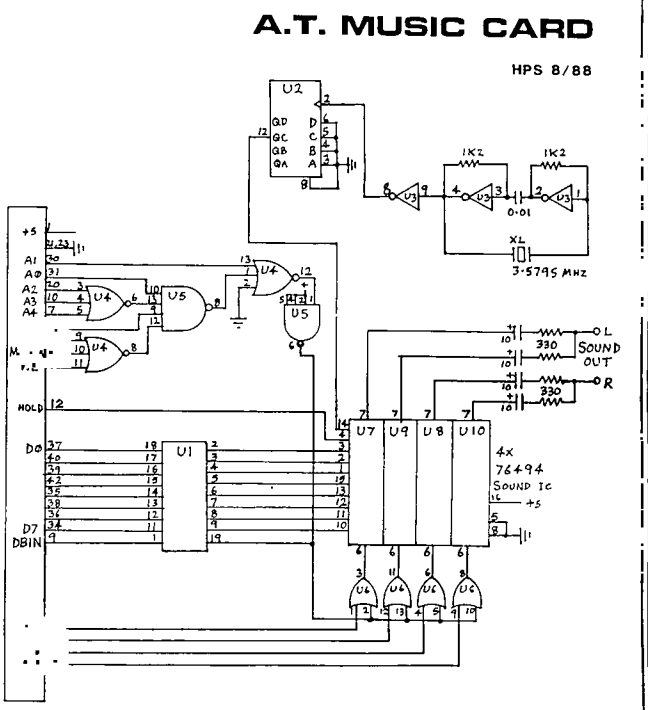
Optional Parts - (see me)

44 way connector (female)
20 pin strip connector for mini-PE (2 off)
Diecast box 190x120x50mm
Length of shielded audio cable to stereo Hi-fi with RCA plugs at both ends.

Circuit Description

The music card is address decoded as a peripheral by logic gates U4 and U5 to provide a board enable at pin 6 of U5. This turns on the Data Bus Buffer chip U1 and enables access to the four sound chips U7-10. Address lines A11 to A14 select which sound chip(s) is enabled via U6. An explanation of how the sound chip operates is too lengthy to go into here so I refer you to articles elsewhere that cover this. The TI GPL manual is one source, also the TMS9919 Sound Generator Chip Specification booklet. Pin 7 of the sound chip is the audio output and since most of us do not have a true Quadraphonic sound system I have combined the output of 2 sound chips for Stereo systems. A headphone adapter could be used for convenience.

U2 is used to divide the crystal frequency down by 4 to provide the sound clock signal on pin 14. During the sound access cycle the console CPU is halted by a low level from pin 4 of the sound chip to the HOLD line on the bus (pin 12). When the sound chips are not busy the hold line must be high.



Assembly of Board kit

The use of 74LS chips (TTL) allows normal handling of components, although care should be taken to avoid excess static charge. Now refer to parts layout diagram.

The component side of the PCB is the side with the lettering AT Music Card. All components mount on this side. First solder on the 44-way connector. If you have a Mini-PE System you do not need this, instead you must mount 2 strip connectors (20 pin). They insert from under the board with the first pin towards the front. This will leave a vacant hole at the other end.

With the connectors on you should now connect up the system and make sure you still get the TI99/4A Title Screen and there are no problems loading a program. If the system will not power-up or function properly then recheck the board for solder shorts between connector pins or board faults. Once past this stage you can now start to insert ICs.

All ICs are inserted the same way round with pin 1 near the round dot on the PCB. Refer to the parts layout diagram. Insert U1, U6, U4, then U2, U5, U3. To avoid over heating chips solder only one side of pins on each chip then start again on other side. You may wish to use sockets for the sound chips, otherwise solder these in next, again pausing to avoid excess heat.

Now insert the resistors and capacitors. The Tag capacitors must be inserted correct way with + lead to the + on the PCB. If these are bent flat to board as shown the label will face you. Insert the Crystal laying it flat on board, then mount the 2 RCA audio connectors, and that finishes the assembly.

Connect up the system again and you should be able to load the disk of demonstration Software. Connect the audio to your AUX input of the Hi-fi (or headphone adapter). To load the demonstration disk use E/A option 3 with file name DSK1.FORTI

continued on page 30

Tips from the Tigercub #53

by Jim Peterson, Tigercub Software, USA

Copyright 1988
Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 120 original programs in BASIC and Extended BASIC, available on cassette or disk, now reduced to just $1 each, plus $1.50 per order for cassette or disk and postage and packing. Minimum order of $10. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette. Descriptive catalogs, while they last, $1 which is deductable from your first order.

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

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

NUTS & BOLTS disks

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended BASIC. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS (No. 1) has 100 subprograms, a tutorial on using them, and 5 pages of documentation. NUTS & BOLTS No. 2 has 108 subprograms, 10 pages of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pages of documentation. Now just $15 each, postpaid.

Tips from the Tigercub

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

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

Tigercub Care Disks #1, #2, #3 and #4. Full disks of text files (printer required).

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

Someone wanted a program to teach how to make change.

```
100 CALL CLEAR
110 DEF I$(X)="$"& SEG$(STR$(X),1,POS(STR$(X),".",1)+2)
120 CALL COLOR(1,2,8,2,2,8,3,2,8,4,2,8,5,2,8,6,2,8,
    7,2,8,8,2,8,9,2,8,10,2,8,11,2,8,12,2,8)
130 CALL SCREEN(5):: D$=RPT$(" ",112)!programmed by Jim
    Peterson Apr. 1988 for the
140 GOTO 180
150 CALL KEY :: CALL CLEAR :: CALL SOUND
160 A,P,C,G,T,F,O,Q,D,N,S,TT,X,B$,QQ,K,M$,J
170 !@P-
180 DISPLAY AT(2,8)ERASE ALL:"MAKING CHANGE":;:"     by
    Jim Peterson" :: RANDOMIZE :: CALL KEY(3,K,S)
190 DISPLAY AT(6,1):"Do you want to":;:" (1) Input
    amounts":" (2) Use random amounts" :: ACCEPT
    AT(6,16)SIZE(1)VALIDATE("12")BEEP:A :: CALL CLEAR
200 IF A=2 THEN 240 :: DISPLAY AT(2,1):"Price of item?
    $":;:;:;: :: ACCEPT AT(2,17)VALIDATE(NUMERIC)BEEP:P
    :: IF P<0 THEN 200
210 DISPLAY AT(4,1):"Amount offered by customer?":"$"
    :: ACCEPT AT(5,2)VALIDATE(NUMERIC)BEEP:C
220 IF C<>INT(C)AND P-INT(P)<>C-INT(C)THEN DISPLAY
    AT(23,1):"Even dollars please!" :: GOTO 210
230 IF C<P THEN DISPLAY AT(23,1):"Not enough!" :: GOTO
    210 ELSE IF C=P THEN DISPLAY AT(23,1):"No change
    needed!" :: GOTO 200 ELSE GOTO 260
240 RANDOMIZE :: P=20*RND+.01 :: P=INT(P0)/100 ::
    DISPLAY AT(2,1):"Price of item " :: DISPLAY
    AT(2,16):I$(P+.001)
250 C=5*INT(4*RND+1) :: IF C<=P OR C=15 THEN 250 ::
    DISPLAY AT(4,1):"Customer offers $" :: DISPLAY
    AT(4,18):STR$(C)
260 G=C :: C=C-P :: T=INT(C/10):: C=C-10*T ::
    F=INT(C/5) :: C=C-5*F :: O=INT(C):: C=C-O ::
    Q=INT(C/.25):: C=C-Q*.25 :: D=INT(C/.1):: C=C-D*.1
    :: N=INT(C/.05)
270 C=C-N*.05 :: X=100*C :: TT=0
280 DISPLAY AT(7,1):"OK, start adding change from the
    price until you reach the customer's amount." ::
    TT=P+.001
290 IF X=0 THEN 330 :: GOSUB 640 :: IF B$<>"P" THEN
    M$="No, give pennies to reach  "&I$(TT+X*.01):: 
    GOSUB 670
300 DISPLAY AT(11,1):"How many pennies?" :: ACCEPT
    AT(11 ,19)VALIDATE(NUMERIC)BEEP:QQ
310 IF QQ=X THEN 320 ELSE IF TT+QQ*.01>G THEN GOSUB 680
    :: GOTO 300 ELSE GOSUB 690 :: GOTO 300
320 DISPLAY AT(7,1):"You have reached
    "&I$(TT+X*.01):;:;:;: :: TT=TT+X*.01
330 IF N=0 THEN 380 :: GOSUB 640
340 IF B$<>"N" THEN M$="No, if the cents portion is .10
    or .20 or .35 or :: GOSUB 670
350 DISPLAY AT(11,1):"How many nickels?" :: ACCEPT
    AT(11,19)VALIDATE(NUMERIC)BEEP:QQ
360 IF QQ=N THEN 370 ELSE IF TT+QQ*.05>G THEN GOSUB 680
    :: GOTO 350 ELSE GOSUB 690 :: GOTO 350
370 DISPLAY AT(7,1):"You have reached
    "&I$(TT+N*.05):;:;:;: :: TT=TT+N*.05
380 IF D=0 THEN 430 :: GOSUB 640
390 IF B$<>"D" THEN M$="No, add dimes to reach .25 or
    .50 or .75 or .00" :: GOSUB 670
400 DISPLAY AT(11,1):"How many dimes?" :: ACCEPT
    AT(11,19)VALIDATE(NUMERIC)BEEP:QQ
410 IF QQ=D THEN 420 ELSE IF TT+QQ*.1>G THEN GOSUB 680
    :: GOTO 400 ELSE GOSUB 690 :: GOTO 400
420 DISPLAY AT(7,1):"You have reached
    "&I$(TT+D*.1):;:;:;: :: TT=TT+D*.1
430 IF Q=0 THEN 480 :: GOSUB 640
440 IF B$<>"Q" THEN M$="No, add quarters to reach an
    even dollar." :: GOSUB 670
450 DISPLAY AT(11,1):"How many quarters?" :: ACCEPT
    AT(11,20)VALIDATE(NUMERIC)BEEP:QQ
460 IF QQ=Q THEN 470 ELSE IF TT+QQ*.25>G THEN GOSUB 680
    :: GOTO 450 ELSE GOSUB 690 :: GOTO 450
470 DISPLAY AT(7,1):"You have reached
    "&I$(TT+Q*.25):;:;:;: :: TT=TT+Q*.25
480 IF O=0 THEN 530 :: GOSUB 640
490 IF B$<>"O" THEN M$="No, add dollars to reach a
    multiple of five dollars." :: GOSUB 670
500 DISPLAY AT(11,1):"How many dollars?" :: ACCEPT
    AT(11,19)VALIDATE(NUMERIC)BEEP:QQ
510 IF QQ=0 THEN 520 ELSE IF TT+QQ>G THEN GOSUB 680 ::
    GOTO 500 ELSE GOSUB 690 :: GO TO 500
520 DISPLAY AT(7,1):"You have reached "&I$(TT+0):;:;:;:
    :: TT=TT+0
530 IF F=0 THEN 580 :: GOSUB 640
540 IF B$<>"F" THEN M$="No, add a five dollar bill  to
    reach a multiple of ten." :: GOSUB 670
550 DISPLAY AT(11,1):"How many fives?" :: ACCEPT
    AT(11,17)VALIDATE(NUMERIC)BEEP: QQ
560 IF QQ=F THEN 570 ELSE IF TT+QQ*5>G THEN GOSUB 680
    :: GOTO 550 ELSE GOSUB 690 :: GOTO 550
```

```
570 DISPLAY AT(7,1):"You have reached "&I$(TT+F*5)::
    TT=TT+5*F
580 IF T=0 THEN 620 :: GOSUB 640
590 IF B$<>"T" THEN M$="No, add ten dollar bills to
    reach"&I$(G):: GOSUB 670
600 DISPLAY AT(11,1):"How many tens?" :: ACCEPT
    AT(11,16)VALIDATE(NUMERIC)BEEP:QQ :: IF QQ=T THEN
    620
610 IF QQ>T THEN GOSUB 680 :: GOTO 600 ELSE DISPLAY
    AT(14,1)BEEP:"That's not enough!" :: GOTO 600
620 DISPLAY AT(7,1)BEEP:"You gave the correct
    change!":"":"":""
630 DISPLAY AT(14,1):"       PRESS ANY KEY" :: DISPLAY
    AT(14,1):"       press any key" :: CALL
    KEY(3,K,S):: IF S=0 THEN 630 ELSE 200
640 DISPLAY AT(11,1):"Will you now give":" (P)ennies":"
    (N)ickels":" (D)imes":" (Q)uarters":" (O)ne dollar
    bills":" (F)ive dollar bills":" (T)en dollar bills"
650 DISPLAY AT(19,1):"":"":"":"":"":""
660 ACCEPT AT(11,19)SIZE(1)VALIDATE("PNDQOFT")BEEP:B$
    :: DISPLAY AT(11,1):"":"":"":"":"":"":"":"" ::
    RETURN
670 FOR J=1 TO 5 :: DISPLAY AT(20,1):D$ :: DISPLAY
    AT(20,1)BEEP:M$ :: NEXT J :: RETURN
680 CALL SCREEN(7):: FOR J=1 TO 15 :: CALL
    SOUND(-99,110,0,-4,0):: DISPLAY AT(14,1):"" ::
    DISPLAY AT(14,1):"You gave too much change!" ::
    NEXT J :: CALL SCREEN(5):: RETURN
690 DISPLAY AT(14,1):"No, that's wrong!" :: RETURN
```

And here is a one-screen tinygram. You could convert this to speech and it would do a better job of making change than most clerks do nowadays!

```
1 !*************************
  !*      CHANGEMAKER      *
  !*       TINYGRAM        *
  !*    by Jim Peterson    *
  !*************************
100 CALL CLEAR :: S$(1)="S"
110 DEF P$(X)="$"&SEG$(STR$(X),1,LEN(STR$(X))-1)
120 DATA 10,5,1,.25,.1,.05,.01,CENT,NICKEL,DIME,
    QUARTER,DOLLAR BILL,FIVE DOLLAR BILL,TEN DOLLAR
    BILL
130 PRINT TAB(8);"CHANGEMAKER";;: :: INPUT "PRICE? ":P
    :: INPUT "AMOUNT OFFERED? ":B :: IF B<P THEN 130
140 PRINT :: C=B-P :: FOR J=1 TO 7 :: READ X(J)::
    A(J)=INT(C/X(J)):: C=C-A(J)*X(J):: NEXT J ::
    Z=P+.001
150 PRINT P$(P+.001);" OUT OF ";P$(B+.001):;: :: FOR
    J=7 TO 1 STEP -1 :: READ A$ :: Z=Z+A(J)*X(J):: IF
    A(J)>0 THEN PRINT "AND";A(J);A$&S$(-(A(J)>1))&" IS
    ";P$(Z):;:
160 NEXT J :: PRINT :: RESTORE 120 :: GOTO 130
```

The Extended BASIC Manual did a very poor job of showing us how to use USING. My thanks to Karl Romstedt for telling me how to do it with DISPLAY AT : put a semicolon directly before USING.

```
100 CALL CLEAR :: DISPLAY AT (12,5):USING "$##.##":1.23
```

Other commands can go either before the AT or after the parameters.

```
110 DISPLAY ERASE ALL BEEP AT(12,5):USING "$##.##":1.23
120 DISPLAY AT(12,5)ERASE ALL BEEP:USING "$##.##":1.23
```

However, to output to a printer, put a comma before USING:

```
120 OPEN #1:"PIO" :: PRINT #1,USING "$##.##":1.23
```

The trouble with PRINT USING "$##.##" is that it will print nothing but asterisks if the integer contains more digits than the number of # left of the decimal, and will leave blanks between the $ and the first digit if the integer contains less digits than the number of ## left of the decimal. This algorithm will correctly print dollars and cents values of any size, rounded off to the nearest cent and with the dollar sign directly before the first digit or decimal.

```
100 INPUT A :: PRINT USING "$"&
    SEG$(RPT$("#",LEN(STR$(INT(A)))),1-(INT(A)=0),255)&
    ".##":A :: GOTO 100
```

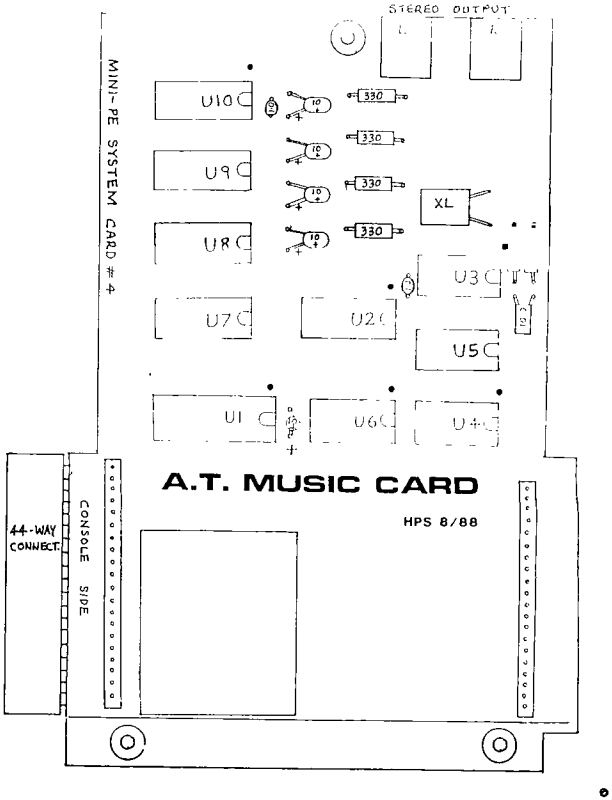
Thanks to Ed Machonis for some of these improvements to the Printall program which was published in Tips #45:

```
171 DISPLAY AT(8,12):"V.1.2" 190 DISPLAY AT(18,7):"TURN
    PRINTER ON!":;:"SET TOP OF FORM HALF INCH BELOW
    PERFS"
200 DISPLAY AT(23,8):"PRESS ANY KEY" :: DISPLAY
    AT(23,8):"press any key" :: CALL KEY(0,K,S):: IF
    S=0 THEN 200 ELSE CALL CLEAR
330 IF P=1 AND SS$<>"Y" THEN DISPLAY
    AT(12,1):"EMPHASIZED? (Y/N) Y" :: ACCEPT
    AT(12,19)VALIDATE("YN")SIZE(-1)BEEP:E$ :: IF E$="Y"
    THEN PRINT #1:CHR$(27);"E";
390 IF NC=1 THEN 410 :: AV=INT(TA/(NC-1)):: DISPLAY
    AT(12,1)ERASE ALL:"COLUMN SEPARATION?":"MINIMUM
    2":"MAXIMUM "&STR$(AV)&" AVAILABLE ":"2"
400 ACCEPT AT(15,1)VALIDATE(DIGIT)SIZE(-2)BEEP:CS :: IF
    CS<2 OR CS>AV THEN 400 ELSE S$=RPT$(" ",CS)
450 LSP=12 :: DISPLAY AT(10,1):" ":" ":"LINES PER PAGE?
    60":" ":" ":" ":" " :: ACCEPT
    AT(12,17)VALIDATE(DIGIT)SIZE(-3):LP :: IF LP<61
    THEN 490
460 !DELETE
470 !DELETE
480 LSP=72/(LP/10):: PRINT #1:CHR$(27);"A";CHR$(LSP)
510 DISPLAY AT(15,1):STR$(LP)&" lines per page":"with
    "&STR$(INT(LSP))&"/72 line spacing"
640 ! DELETE!
650 IF LEN(M$(IP))<=CW THEN 670 ::
    T$=SEG$(M$(IP),1,CW):: CALL
    SOUND(1000,110,0,-4,0):: DISPLAY AT(12,1):M$(IP);"
    OVER";CW;"CHARACTERS":"TRUNCATED TO ";T$:"OK?"
660 CALL KEY(3,K,S):: IF S=0 THEN 660 ELSE IF K<>89
    THEN STOP ELSE M$(IP)=T$
```

Memory full! Jim Peterson

continued from page 28

Once the Menu appears type ALBUM and all music will be played in sequence. You should note the stereo effect and sharp clarity of sound. Relax and enjoy the finest in computer generated sounds. There is tremendous scope for creation and composing of music with this system, and I hope some of our members can contribute in this field. I have seen a similar system on another machine at much greater cost and complexity and I assure you this card is every bit as effective.



Controlling a Small Train Set using the Wire Accessory Interface and TI99/4A

by Ross Mudie

This tutorial shows how the Wire Accessory I/O Interface is used with a TI99/4A Home Computer to control a small train set. The Wire Accessory I/O Controller can be used for a multitude of functions, not just a train set. This tutorial booklet shows all the circuits used for the Train Set Interface together with a description of the hardware. The ideas contained in the Train Set Interface can be applied to other purposes. The program to control the train set is written in assembly linked from Extended BASIC and contains some routines which may be applied to other functions.

The design is intended to use a TI99/4A 32K console dedicated to a control function. The program is assembled on a disk system using Editor Assembler then implanted using a simple linked assembly routine from the Extended BASIC immediate mode. After merging the Extended BASIC header the program can be saved normally on cassette tape.

The tutorial information is contained in a booklet photocopied for the Tutorial Day and on disk MUDIE_88/1 which contains all programs required in suitable format as described in the tutorial. The Extended BASIC program named LOAD will print the tutorial on a printer under Extended BASIC. Loose leaf circuit diagrams (which are in the tutorial book) are supplied with the original disk.

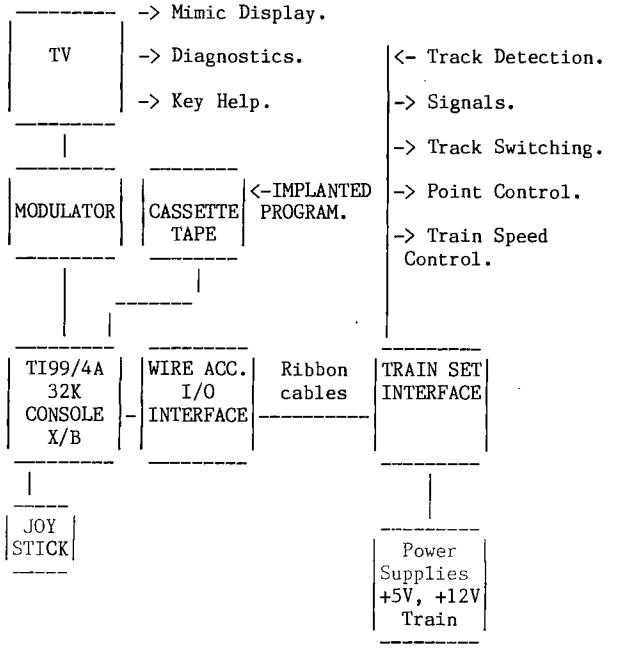


Figure 1. Train Set System Block Diagram.

Disclaimer: The author or TIsHUG will accept no responsibility for any damage to any equipment or injury to any person which may arise from the use or application of circuits or information contained herein.
Safety: Wiring of 240 volt mains circuitry must only be carried out by competent persons in accordance with the relevant standards for mains operated equipment.

Contents

1. FOREWORD.

The directors of TIsHUG asked me to construct a small train set to demonstrate how the Wire Accessory Interface could be used. The result is a very functional small train set with very short controlled track sections. This tutorial is presented "as is" with some areas capable of improvement. The very short time period, for a project of this magnitude, from the request until the tutorial day has forced me to say "good enough" in some areas, in the interest of getting the whole project running and reasonably documented.

The whole project has been prototyped on veroboard and at present there are no plans to design printed circuit boards until there is sufficient interest shown.

Train enthusiasts please note that this tutorial has been prepared from a an electronic design and computer point of view rather than that of an avid railway modeller. For the avid modeller it may present useful ideas towards computerisation of a prototypical model railway.

2. HARDWARE.

a) THE TRACK.

The inside rail is common around the whole layout. The outside rail is divided into short sections which are insulated from previous and succeeding sections by insulating track joiners.

The Track sections are controller power fed via two parallel diodes in the input of each Track Detector (TD) circuit. The power to the track sections is also fed via relay contacts in the Track Control (TC) and Signal Units.

b) TRACK DETECTION, TRACK CONTROL and SIGNAL UNIT.

This circuit comprises active components of 3 transistors, T1, T2, T3, and a relay designated SIG. This circuitry has been repeated 13 times on the train set interface which detects and controls 12 track sections, plus an extra signal lamp as a turnout indicator at one of the points. The system also allows detection of track power polarity to cater for manual track controllers instead of the computer driven track controller.

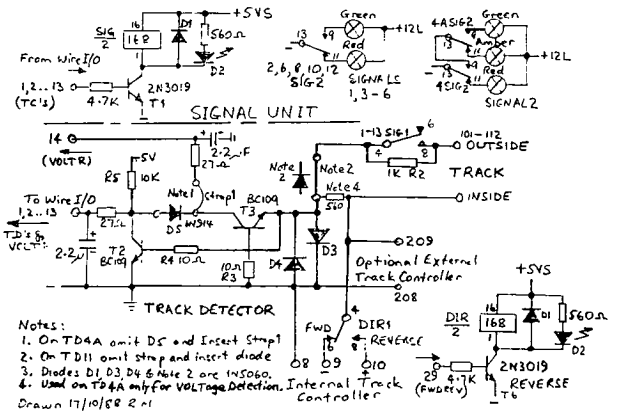


Figure 2. Track Detection, Control and Signal Unit.

FUNCTIONS:

i) To detect the current flowing into a track section due to a train engine in the section and to translate the condition to a TTL logic state.

ii) To respond to instructions issued by the program running in the computer and operate output (SIG) relays. The relays in turn control the track side signals and when operated provide a low resistance path from the track controller to the track.

OPERATION:

Whilst the computer program is not running the Watchdog relay is normal which removes the 5 volt power supply from the Signal Units. (The operation of the watchdog is covered later). Whilst the SIG relays are normal all track side signals are at RED and the power is removed from the track. When the program is running the operated watchdog relay provides 5 volt power to the SIG relay circuits via +5S.

When the computer program considers it safe to do so, it provides a logic 1 on the output wires of the I/O interface which turns transistor T1 ON, operating the appropiate SIG relay and lighting the parallel LED. (The LEDS are provided to show which relays are operated as a diagnostic aid). Diode D1 clamps the back EMF voltage pulse which occurs when any relay is released. (The back EMF is generated in the coil of the relay when it is de-energised and the collapsing magnetic field generates very high voltage spikes. If the spikes are unclamped they are capable of punching through from collector to emitter of the associated transistor when it turns off). When SIG operates it changes the associated signal from RED to AMBER or GREEN at contact unit SIG2 and SIG1 completes the circuit for track power. When a train engine enters a track section, in the forward direction, assuming that sufficient voltage is available from the track controller, then the voltage dropped across D4 is sufficient to turn T3 ON. T3 in turning on pulls the logic 1 on the TD output down to about zero volts. Diode D5 is provided to prevent the logic output from going to -0.6 volts when T3 is turned ON.

When the train is in reverse D3 conducts, turning T2 ON which provides logic 0 to the computer I/O Interface. When a SIG relay is at normal preventing sufficient current for train motor operation, the 1K ohm resistor allows sufficient current for operation of the Track Detector so the system can detect the presence of an engine in a turned off section of track.

The 2.2uF capacitor on the output of each track detector helps to reduce "flickering" caused by the pulse output track controller. The 27 ohm resistor between the 2.2uF capacitor and the collector of T2 and T3 reduces the discharge current from the capacitor into the collector of T2 or T3 when either transistor turns on.

c) REVERSE SWITCHING.

The DIR relay is operated via an identical circuit to the SIG relays using T6. When DIR is operated the train reverses due to the connection of a reverse polarity pulsed DC power source. If an external manual track controller is connected and used, providing that keys I and M and joy stick 1 are not activated then the program will use the information for forward and reverse from the VOLT detectors. Connection of an external track controller should bypass the contacts of the DIR relay.

d) THREE ASPECT SIGNAL SIG2.

Signal 2 is located adjacent to point 2 which is a turn out point. When the point is set for the main line the relay contacts provide a green signal lamp. When the point is set to turn out onto the branch line the combination of SIG contacts allow the AMBER lamp to light in place of the GREEN lamp. Signal 2 is controlled by TC4 and TC4A from the program, if Point 2 is set for Crossing then TC4A=logic 0 and TC4ASIG is released lighting the AMBER lamp instead of GREEN.

e) DIRECTION DETECTION.

The Track Detector associated with TC4A has no track sections to detect from, so it is used to detect voltage present on the track. In TD4A, transistor T3 is used to detect VOLTForward and T2 detects VOLTReverse.

The direction detection allows use of a non computer controlled track controller. Since TC4A does not connect to any track section, a 560 ohm resistor is provided as shown in figure 2 note 4 to provide a suitable input condition for the voltage polarity detector.

When the track controller is controlled by the computer program the detection of direction is derived from the program and the information from the direction voltage detectors is not used by the program.

f) POINT CONTROL.

The points have 2 ohm coils which are operated chiefly by the charge in a 4700uF capacitor on the output of the point power supply. The computer issues a pulse of logic 0 for approximately 200 milliseconds, which turns T4 ON and in turn this turns T5 ON. This simple configuration switches a 6 ampere load with an input current of only 4mA. The 5 volt supply (+5S) to the emitter of T4 is controlled by the watchdog relay which only allows operation of the points when the program is running.

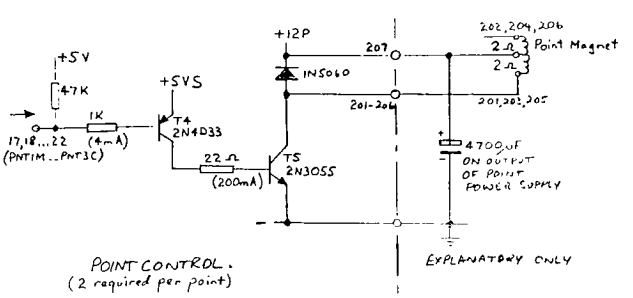


Figure 3. Point Control.

g) HARDWARE WATCHDOG.

The program issues pulses of approximately 350 milliseconds ON, 350 milliseconds OFF via the Watchdog output. This provides a positive going edge (logic 0 to logic 1 transition) once every 700 milliseconds. The edge triggers the watchdog monostable IC which is retriggerable and has a drop out time of approximately 1.2 seconds, set by the 1uF timing capacitor and 1.2 megohm timing resistor. Whilst the monostable is triggered, the logic 1 at the Q output holds the WD relay operated via T7. The 10uF capacitor, 10K ohm

resistor and diode on pin 3 of the monostable provides a power-up reset. There is a resistor and LED on the input of the watchdog which shows the continuing pulse stream from the computer interface and the LED on the 5 volt switched supply shows that the WD relay is operated.

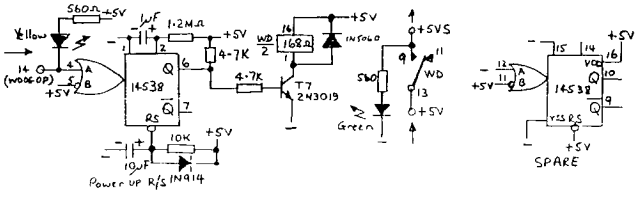


Figure 4. Hardware Watchdog.

h) 5 VOLT POWER SUPPLY.

The 5 volt power supply is used for the 5 volt circuitry on the train set interface board and the Wire Accessory I/O board which is plugged into the side of the console. The 5 volt rail connects to the Wire I/O via 2 wires in the ribbon cable which interconnects the Wire I/O and the train set interface. A 240V to 12 VAC, 1 ampere transformer is used with a full wave bridge rectifier. The LM309K fixed 5 volt regulator is best suited for this application with 0.1uF disk ceramic capacitors mounted physically and electrically close to the regulator to prevent unwanted oscillation. The diode around the LM309K from output to input prevents destruction of the regulator if the regulator input voltage should become less than the output voltage, (this feature is also used in all other regulator circuits in the train set).

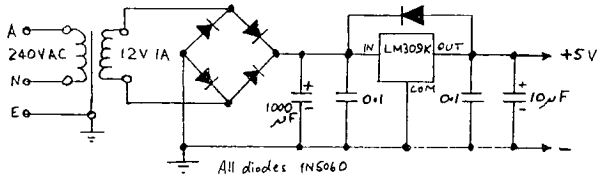


Figure 5. 5 Volt Power Supply.

i) POINTS and LAMPS POWER SUPPLY.

Both these supplies are obtained from a common 15 volt, 1 ampere transformer and bridge rectifier. The positive side of the bridge rectifier feeds via isolating diodes into the filter capacitors and regulators. This prevents the point supply from discharging the lamp supply capacitor as a point change occurs.

The point supply has a small input capacitor (1000uF) and large (4700uF) output capacitor. The purpose of this is that the 4700uF capacitor stores sufficient energy to sucessfully change the point but if a fault condition should permanently energise any point then the 4700uF will discharge quickly allowing the fault current to be limited by the 7812 to a little over 1 ampere. The 1000uF input capacitor will also allow a reasonable amount of ripple under fault conditions, which additionally reduces the power dissipation. The 7812 has a minimal sized heatsink and if the regulator should overheat under fault condition, it will thermally limit, further reducing the fault current.

Once a fault is removed the regulator cools and recovers automatically.

Under normal operating conditions the 7812 allows recharging of 4700uF point drive capacitor because the program only allows one point to change at a time and the program also provides a pause of approximately 200 milliseconds between point changes to allow the 4/00uF capacitor to recharge.

The 4700uF capacitor in the input of the 12 volt lamp supply provides sufficient reserve at the input to the lamp regulator to prevent significant dimming of the signal lamps when the points change. Early in the development of the project the signal lamps and points were run off the same supply but when a point operation occurred the signal lamps almost fully extinguished. AC operation of the signal lamps could also be used but an additional transformer output and current limiting would be required to prevent damage under fault conditions. The DC supply also opens the way for use of Light Emitting Diodes in the signal lamps.

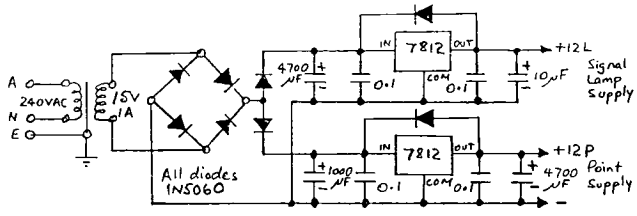


Figure 6. Points and Signal Lamp Power Supply.

j) TRAIN MOTOR CONTROL POWER SUPPLY, (Track Controller).

This controller uses a Pulse Width Modulation technique to control the train speed. The width of pulses of DC voltage is varied by changing the binary code on 4 input wires to the train controller. The binary code on the 4 wires is obtained from the Wire Accessory I/O which is driven by the computer program. Because common earthing would not allow the output of the train controller to be reversed when reversed train direction was required, two train controller outputs are provided. The forward direction is provided by a LM337K negative regulator whilst the reverse direction is provided by the LM317K positive regulator. The circuit diagram is shown in figure 7.

The 4 input wires carry weighting values of 1, 2, 4 and 8 so in theory 16 levels would be obtainable. In practice 13 levels have been obtained with 0 being no output (no pulses) and then wider output pulses for all input values from 1 up to 12 and 13 being a continuous (non-pulsed) full output state. The outputs are 12 volts when on and 0 volts when off. A problem when using the LM317/337 regulators is that when the control pin is tied down to ground the regulator has a minimum output of 1.2 volts. This is overcome by returning the control wire to -1.2 volts for the positive regulator and +1.2 volts for the negative regulator. The maximum output voltage is set by the 2370 ohm resistor between the regulator control pin and the reference voltage.

The inputs from the computer turn on or off, as required, transistors T1 to T4 in a resistive ladder. The output of the resistive ladder is buffered and fed to a comparator stage. T1 connected to the ladder at the earthy end of the ladder contributes approximately 1 unit of change by virtue that the collector is either at earth potential or +5V potential (via a 1K ohm resistor). T2 being higher up the ladder contributes 2 units of change, T3 contributes 4 units of change and T4 contributes 8 units of change. The voltage signal at the top of the ladder is buffered by T5 and T6 and feeds to the comparators IC1c and IC1d. IC1a is a triangular waveform generator and the triangular waveform at IC1 pin 2 is buffered by IC1b and fed to the "other" inputs of the comparators IC1c and IC1d. Each comparators provides an output which drives the transistors controlling the positive and negative output regulators when the triangular waveform exceeds (one way or the other) the DC control voltage. If

manual control was required with the same output stage then a potentiometer could be substituted to control the level of DC at the control input to the comparators.

RV1 adjusts the voltage source at the emitter of T8 to set the operating voltage for the input stage. With all inputs off (logic 0) the emitter of T8 (which is designated +7.5A for Adjustable) should be set initially to 7.5 volts. If an oscilloscope is available RV1 should be set so that with the track controller output is 0 volts with all logic 0 input and an output pulse of 0.25 milliseconds with an input value of 1, (Speed 1). The pulse widths obtained on the prototype are shown in table 1. The test switch which allows convenience in testing of the track controller must have all switches set to open for normal operation.

INPUT VALUE	OUTPUT ms	INPUT VALUE	OUTPUT ms
0	0	7	2.7
1	0.25	8	3.1
2	0.7	9	3.5
3	1.1	10	3.9
4	1.5	11	4.25
5	2.0	12	4.6
6	2.3	13-15	full on

Table 1 - Track Controller Output Pulse Widths.

3. TRAIN SET LAYOUT.

The train track is divided into 12 sections by means of insulation between the sections of the outside rail. The inside rail is continuous. The designation TD refers to Track detector and TC is Track Control. These designations apply in both the hardware and the software. The track sections which are shown boxed, ———— , have dual designation status as both TCx and SIGy in the software. (This becomes obvious when reading the source file).

Since the train detection system used only detects the position of the engine on the layout, then a limit must be set for the maximum length of trains if more than one is operating on the same track at a time. It is the function of the part of the program which makes the logical decisions to turn off the track power for the necessary number of sections behind the engine to prevent a collision from a following faster train. As an example if a train engine is detected in TD4 then it is necessary to turn off TC3, TC2 and TC1. This will also result in SIG1 being set RED.

See Figure 8 on page 35

4.1 SOFTWARE FUNCTION BLOCKS

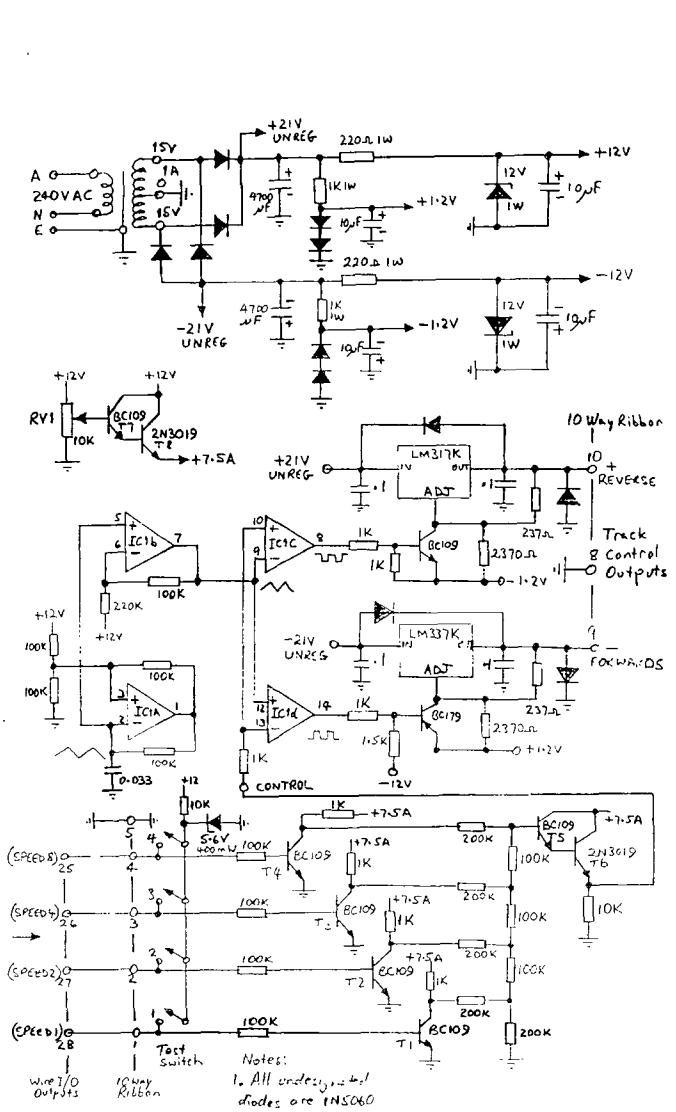
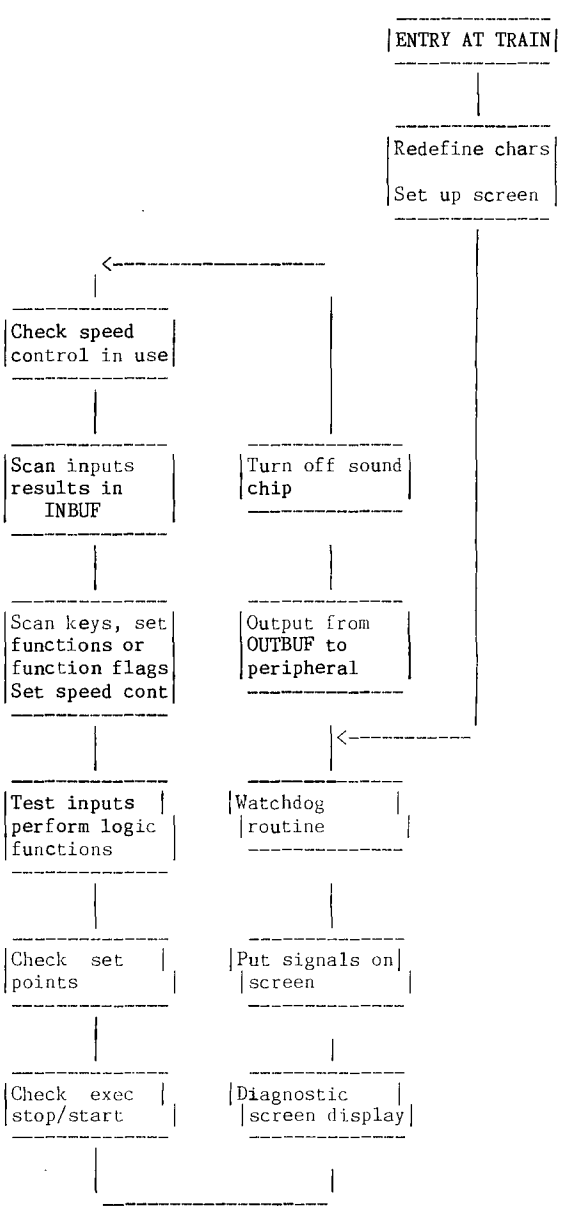
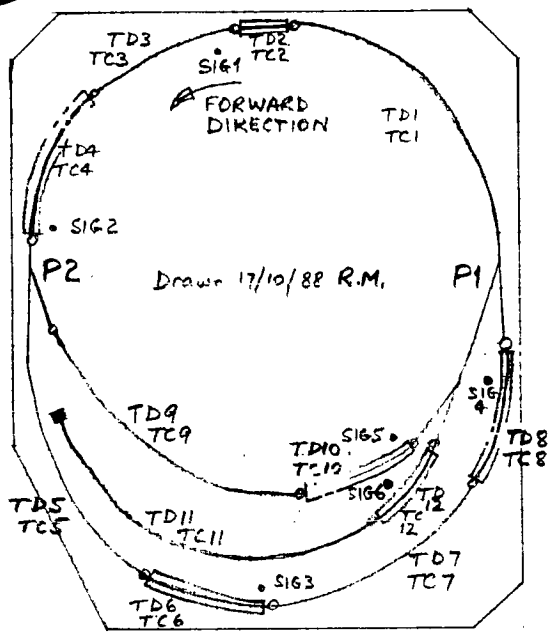


Figure 7. Train Motor Controller.





TD denotes Track Detector
TC denotes Track Control
-o- denotes end of section of control

Figure 8. Train Set Track Sectioning Layout.

4.2 DEFINITION OF OPERATING FUNCTIONS.

This section sets the major operational parameters for the control of the train set by the computer and describes some of the operating features.

a) SCREEN.

The computer screen is black background with white mimic display of track layout. Train positions are shown with amber lamps in each track section to show progress of the engine. Signals are displayed green, amber (as required) and red beside the track.

Diagnostic display, help text or version text may be optionally displayed on the right hand side of the screen whilst the track mimic display is also present and the program is normally operational.

When STOP is invoked a number or letter representative of the Track Section designations will flash on the track display and a prompt on the bottom right hand corner will prompt "STOP AT" with a RED flashing character. When a valid character is received the "menu" characters will return to normal track layout and the stopped section will flash alternately RED or section designation.

When RESTART is invoked the prompt "START AT" will display on the bottom right hand corner of the screen with a green flashing lamp.

Points show the operated positions by character redefinition and the keys that require to be pressed to change a point are shown in the mimic display of the track adjacent to the points.

Track controller speed, Forward Hold and Pause are displayed on the screen as appropriate.

The Watchdog routine flashes a green lamp on the bottom left hand side of the screen.

b) INPUTS

The program reads inputs from 4 bytes of memory from >868C to >868F and places the result in a memory space of 32 bytes designated INBUF. When a train is detected (logic 0 from the hardware Track Detectors), a value of 1 is stored in the appropriate byte of INBUF. When a logic 1 is detected from the hardware then a value of 0 results in the appropriate byte in INBUF.

If Forward Hold has been invoked then when a train detection signal is lost the program will check to see if the train has been detected in the next section. If the train has not been detected in the next section then the loss of detection signal is ignored.

If the voltage detectors indicate reverse direction then the Forward Hold operates by looking at the next section backwards.

c) KEY BOARD SCAN.

The keyboard is scanned and action taken in the following order of tests.

If FCTN[4] is detected then the program returns to the calling Extended BASIC program. Quit is ignored by the assembly program.

Once a key is pressed and the action has been taken, it is then ignored until it has been released. The exception to this is I, M and joy stick 1 which are used for train speed control.

P invokes a Pause function for all train movements, only. In pause all signals go red and the track power is removed. Points may still be changed, the speed controller adjusted and the over riding Stop/Restart conditions changed. Pause is removed by pressing P again.
D invokes or removes Diagnostic mode.
F invokes or removes Forward Hold mode.
H invokes or removes the Help screen.
S enters the Stop at a section mode, followed by a valid key as will be seen flashing on the mimic display on the screen. E can be used to Escape this mode without doing anything.
R enters the Restart mode to counter the action of a S command. R is followed by the appropriate section symbol seen flashing on the screen mimic display or E to Escape the mode without taking any action.
V invokes or removes the Version screen.
Numbers 1 to 6 change points when R or S is not invoked.
I or joy stick 1 up will increase the speed of the computer controlled track speed controller.
M or joy stick 1 down will decrease the speed of the computer controlled track speed controller.

By decreasing speed below zero the speed becomes a negative value, the DIR relay in the train set interface will operate and the speed value will increase driving the train backwards. M or joy stick 1 down will make the train go faster backwards. The I key or joy stick 1 up will decrease the speed of reverse motion and automatic reversal to forward direction will occur on crossing zero to +1 speed.

d) TESTS AND LOGICAL DECISIONS.

This routine resets all Track Control (TC) bytes to a value of 1, then goes through all the Track Detectors (TD) to see what should be set to zero. If Pause has been already invoked then all the TC bytes are set to a value of zero which makes all signals red and disconnects track power.

In each section tested if the TD is 1 (i.e. a train is detected) then there will be an AMBER lamp shown on the track section in the mimic display on the screen, else the normal track will be shown.

Tests:
TD1 at 1 makes TC8, TC10, TC12, TC7, TC9 and TC11 zero. Points 1 and 3 will be prevented from changing, (whilst train in section).
TD2 at 1 makes TC1, TC8, TC10 and TC12 zero.
TD3 at 1 makes TC2, TC1, TC8, TC10 and TC12 zero.
TD4 at 1 makes TC3, TC2 and TC1 zero.
The flag for Point 2 Crossing (P2C), if at a value of 1, makes TC4A zero. (This gives the amber crossing signal).
TD5 at 1, dependant on the setting of P2C moves the position of the train in section indicator for TC5.
TD5 at 1 makes TC4, TC3 and TC2 zero. Point 2 is prevented from changing.
TD6 at 1 makes TC5 zero and dependant on the setting of point 2 to main will make TC4 zero.

J (VOLTage Reverse Direction) in conjunction with P2M (Point 2 Main) at zero makes TC6 zero.
(Prevent reversing over wrong setting of point 2.
TD7 at 1 makes TC6 and TC5 zero and dependant on the setting of point 2 to main will make TC4 zero.
TD8 at 1 makes TC7 and TC6 zero and dependant if Point 2 is set to main makes TC5 zero.
TD9 at 1 makes TC4, TC3 and TC2 zero and prevents point 2 from being changed.
VOLTRD (VOLTage Reverse Direction) in conjunction with P2C (Point 2 Crossing) at zero makes TC9 zero.
(Prevent reversing over wrong setting of point 2.
TD10 at 1 makes TC9 zero and dependant on the setting of P2C can make TC4 zero.
TD11 at 1 only displays trains in its section when the voltage is forward in direction due to a diode in the power feed to this track section.
TD12 at 1 only displays a train in its section.
The point flags P1MF, P1CF to P3CF are tested to see if a point is about to be changed. If it is then the associated signal is set to RED.

If a train is reversing into the terminating siding and the computer controlled speed controller is in use and the speed exceeds 7 then the speed is reduced to 7 to prevent the train from colliding with the buffer in the end of the terminating siding.

e) POINTS.

When a point is to be changed a value of 1 is placed in the point flag by the key test routine. The point flags are tested and if one is found waiting to be changed the point output is set to zero in OUTBUF and the timer for the point pulse is set. Once one point is receiving a pulse no other point can be changed, however other requests for points to change will be stored if requested. The order of execution of point changing is P1M, P1C etc to P3C, i.e. 1 to 6 order as the keys are shown on the screen for the points.

After a point has changed a delay timer (PTD) prevents another point from changing until the capacitor in the points power supply has had a chance to recharge. The point pulse and the recovery delay are set at approximately 200 milliseconds.

The point routine also updates the point status flag and after a point has been changed the point character on the screen mimic display is updated.

f) TRACK STOP and RESTART.

When a train is to be stopped in a section, by pressing S the STOP routine is entered. The normal key functions are ignored and the STOP routine flashes the numerals 1 to 9 and letters A to C in the track sections to which they apply. The prompt "STOP AT" is displayed on the bottom right hand corner of the screen and a RED lamp flashes after the prompt. The normal functions of the train set continue whilst waiting for a valid key press. When a valid key is pressed the STOP information is stored in the appropriate byte in STOPBF. On each scan all the stopped sections are placed in OUTBUF for later transmission to the peripheral.

When R is pressed the same routine is invoked except only the stopped sections are flashing with the appropriate number and the prompt is "START AT" with a GREEN lamp prompt cursor. After pressing a valid key the byte in the STOPBF is reset to normal.

E is a valid Escape from the routine if it is no longer required after entry.

g) DIAGNOSTIC SCREEN DISPLAY.

The diagnostic display is entered or exited by pressing D. The routine displays the contents of values in INBUF, OUTBUF and 64 flags used in operation of the program. A value of >90 is added to all the values except LASTKY which has >60 added. The value of >90 makes a real value of 0 into >30, (the hex value for character zero), and then the extra >60 required for screen display under Extended BASIC. LASTKY only needs >60 added since it already contains the hex value from the key press. A key to the meanings of the locations in the Diagnostic mode is contained in the start of the assembly program source file.

h) SIGNALS ON SCREEN.

This group of routines examines the contents of each SIGnal flag in OUTBUF and displays RED, AMBER or GREEN, as required, on the screen mimic display.

i) WATCHDOG ROUTINE

A monostable watchdog timer is provided on the train set interface board which must be pulsed once every 1.2 seconds or it will close down the operation of the train set. The watchdog turns its byte in OUTBUF on for approximately 350 milliseconds and off for approximately 350 milliseconds. It also flashes a green lamp on the bottom left hand corner of the screen.

j) OUTPUT TO PERIPHERAL.

This routine reads 16 bytes of OUTBUF and then places the resultant word into the appropriate 2 bytes of the Wire I/O starting at >8680. The routine then does the next 16 bytes, placing the appropriate value into the next 2 bytes of the peripheral.

k) TURN OFF SOUND.

Since the Wire I/O Peripheral is mapped into the same area as the sound chip some values written to the peripheral will turn on the console sound generator. This routine routinely turns off all sound generators however some clicking will occur with the rate dependant on the composition of bytes being written to the Wire I/O peripheral and if Diagnostic or normal mode is invoked.

4.3 CATALOG of DISK MUDIE_88/1 prepared for TIsHUG Tutorial 5th November 1988 by Ross Mudie.

DSK1 - DISKNAME= MUDIE_88/1
AVAILABLE= 28 USED= 690

FILENAME	SIZE	TYPE	P	
ASSM1	33	PROGRAM		Assembler Pt1 use with E/A cartridge.
ASSM2	20	PROGRAM		Assembler Pt2 use with E/A cartridge.
DEBUG	19	PROGRAM		TI Debug in Extended BASIC implanted form.
EDIT1	25	PROGRAM		Editor for use with E/A cartridge.
IMP	2	PROGRAM		X/B Implant aid, use line 110 in immediate mode.
LOAD	4	PROGRAM		X/B program, prints tutorial text.
MERGE	2	DIS/VAR163		X/B MERGE format, X/B header for Train Set pgm.
MO	3	DIS/FIX 80		Object from MS.
MOO	3	DIS/FIX 80		Object from MOS.
MOS	3	DIS/VAR 80		Implanter source code places Object O in X/B.
MS	3	DIS/VAR 80		Relocater source code relocates implanted code.
O	77	DIS/FIX 80		Assembled object code from S1,S2,S3 or T5,T6,T7.
T	3	DIS/VAR 80		Copy directives to assemble T5,T6,T7.
T1	97	DIS/VAR 80		Tutorial pages 1-10.
T10	48	DIS/VAR 80		Tutorial pages 51-56, (51-54 in book) T11 14 DIS/VAR 80 Tutorial page 57, (55 in book).
T12	21	DIS/VAR 80		Tutorial page 58, (56 in book).
T2	12	DIS/VAR 80		Tutorial page 11.
T3	58	DIS/VAR 80		Tutorial pages 12-16.
T4	5	DIS/VAR 80		Tutorial page 17.
T5	68	DIS/VAR 80		Tutorial pages 18-26.
T6	56	DIS/VAR 80		Tutorial pages 27-37.
T7	37	DIS/VAR 80		Tutorial pages 38-44.
T8	31	DIS/VAR 80		Tutorial pages 45-47.
T9	20	DIS/VAR 80		Tutorial pages 48-50.
TBOOK	4	PROGRAM		Program to print Tutorial book (skips file T10) TRAIN 22 PROGRAM Implanted X/B program for Train Set.

6. IMPLANTING TECHNIQUE, EXTENDED BASIC PROGRAM and ASSEMBLY LOADER.

The assembly software for this project has been prepared using TI Editor, Assembler and a RAMdisk for faster saving and assembly of the files. I use the source file name of S and the object file name of O as no doubt you will have already noticed. (If using tutorial disk MUDIE_88/1 use the source file name T with an object file O and it will assemble tutorial files T5, T6 and T7).

The aim of implanting the object file in Extended BASIC is to allow easy and quick loading from cassette tape into the 32K console which is connected to the train set interface via the Wire Accessory I/O. I recommend the use of a dedicated 32K console for this function.

The format of the implanting is very similar to that described by George Meldrum in the November 1987 TIsHUG News Digest. The AORGed assembly relocater program is located between >FFCA and >FFE7. This is a fixed assignment, the source file is called MS and the object file is called MO. The train set assembly object file, which is apt to grow in size as it develops, ends at >FFC9. The start is dependent on size which is found by loading it into the computer and then reading the next free space flag. More on this shortly. Once the size is found the addresses can be calculated and placed in a MOVER program which does most of the implanting for you, the source file for this is called MOS and the object MOO.

Here is how I have done the implanting: Go to Extended BASIC and type

```
NEW
CALL INIT
CALL LOAD("DSK1.O")
CALL PEEK(8194,A,S) :: PRINT A;S
```

A;S gives the address of the next free byte in the low RAM. At this point I convert the result to hexadecimal by looking up the ASCII to hex conversion table. Results of 56 114 convert to >3872. >3872 minus >24F4, use the hex arithmetic function of TI Debug if you do not trust your ability in hex arithmetic, the answer is >137E. The address >24F4 is the start of the free Low RAM after CALL INIT has loaded the assembly link routines and you can prove this point for yourself by doing the CALL PEEK(8194,A,S) and printing the values of A and S before loading the object file.

Whilst you have the hex calculator in memory work out where the implanted code will go. >FFCA minus >137E gives >EC4C. I hope that you wrote down each value as you went!

Next load MS which is the relocater source file.

```
* Move program for top of implanted code.  Moves *
                implanted code into low memory.
* Source=MS Object=MO          AORG >FFCA (255 202)
       TEXT 'TRAIN
       DATA START
START  LI   R1,>EC4C Location of start of implanted
              code in x/b
       LI   R2,>24F4 Destination in low RAM
       LI   R3,>137E Size of code to be moved.
LOOP2  MOV  *R1+,*R2+
       DEC  R3
       JNE  LOOP2
       RT
       END
```

Put the values for start of code in the implanted form in R1 and the size in R3 if your values are different to mine. Save it as MS and assemble it as MO.

Next load the source file that does the implanting called MOS and put the same values in it.

```
* Source=MOS  Object=MOO
       DEF   MOVER
FROM   EQU  >24F4     Assembly is loaded here by CALL
              LOAD("DSK1.O")
TO     EQU  >EC4C     This is where it is implanted to
SIZE   EQU  >137E     Size of the code
MOVER  LI   R1,FROM
       LI   R2,TO
       LI   R3,SIZE
LOOP1  MOV  *R1+,*R2+
       DEC  R3
       JNE  LOOP1
       LI   R1,TO     This puts the start address of
              the implant in R1
       DEC  R1        then adjusts the value in R1 for
              the next address
       MOV  R1,>8330 useable by x/b.  The address
              values are then placed
       MOV  R1,>8332 in the pointers used by Extended
              BASIC.
       RT
       END
```

Save the amended source file as MOS and assemble it as MOO.

The following program should be saved on disk in MERGE mode (unless you like to retype the same thing over and over).

```
100 ! SAVE DSK1.TRAIN
110 CALL INIT
120 CALL LOAD(8196,255,202)
130 CALL LINK("TRAIN")
140 CALL LOAD(8194,56,114,63,248,"",16376,84,82,
    65,73,78,32,36,244)
150 CALL CLEAR
160 CALL LINK("TRAIN")
```

Save this program by typing SAVE DSK1.MERGE,MERGE What this program does is as follows:

Line 120 tells the computer that the the start of the link names table is at 255, 202, i.e. >FFCA which is the relocater entry point.

Line 130 links to "TRAIN" at >FFCA and moves the code down to the low RAM. This was done with the code from the source file MS and at completion it returns to the Extended BASIC program.

Line 140 readjusts the pointers in 8194 and 8196 to reflect the actual area used by the assembly program and the position of the DEF table for linking to. The 56, 114 which gets loaded into 8194 and 8195 should be changed to reflect the actual values for the size of your program after merging the program into memory. Other relocatable code could be loaded after line 140 and before 160 since the pointers were readjusted by line 140.

Line 150 clears the screen and line 160 links into the implanted assembly.

I use another little program stored on disk to do most of the implanting as follows.

```
100 ! SAVE DSK1.IMP
110 CALL INIT :: CALL LOAD("DSK1.O"):: CALL
    PEEK(8194,A,S):: PRINT A;S :: CALL
    LOAD("DSK1.MOO"):: CALL LINK("MOVER"):: CALL
    LOAD("DSK1.MO")
```

Once the program is in memory I call line 110 into edit mode, then press <ENTER> then <REDO> and I wipe off the line number. The following line then executes from the Extended BASIC editor buffer.

```
CALL INIT :: CALL LOAD("DSK1.O"):: CALL
    PEEK(8194,A,S):: PRINT A;S :: CALL
    LOAD("DSK1.MOO"):: CALL LINK("MOVER"):: CALL
    LOAD("DSK1.MO")
```

The values that print on the screen are from the peek at 8194-5 these should match the values previously obtained and are the x and y values.

Since the final code in MOO, (refer to MOS), readjusted the Extended BASIC memory usage pointers, Extended BASIC program lines may now be entered or merge mode can be used.

When the cursor returns type MERGE DSK1.MERGE and use the values on the screen for x and y in line 140 CALL LOAD(8194,x,y,63,248 etc. Then press <ENTER>.

Type 100 FCTN[X] <ENTER> <REDO> edit off the 100 ! and press <ENTER> and the program saves to disk.

Type SAVE CS1 to save it to tape.

If you want to save the Extended BASIC part of the program as a merge file just type SAVE DSK1.MERGE,MERGE. The implanted code is ignored and your merge file has the latest value to load at 8194-5.

This may seem a lot of trouble at first but once used to it I have found it quick, plus I know exactly what is going on!!

7. INTERCONNECTIONS

This lists the plug/socket interconnections used in the train set. There are 2 x 25 pin D connectors used to connect from the Train Set Interface to the train set, a 10 way ribbon cable connects from the Train Set Interface to the computer controlled track controller and 2 x 34 way ribbon cables between the Wire Interface I/O board and the Train set Interface.

a) The 25 pin D connectors are numbered 101 to 125 for plug 1 and 201 to 225 for plug 2.

PLUG PIN	FUNCTION	PLUG PIN	FUNCTION
101	TD1	201	POINT1 MAIN
102	TD2	202	POINT1 CROSSING
103	TD3	203	POINT2 MAIN
104	TD4	204	POINT2 CROSSING
105	TD5	205	POINT3 MAIN
106	TD6	206	POINT3 CROSSING
107	TD7	207	+12VP Point supply
108	TD8	208	External Track Controller
109	TD9	209	External Track Controller
110	TD10	210	
111	TD11	211	
112	TD12	212	
113	SIG1 GREEN	213	
114	SIG1 RED	214	
115	SIG2 GREEN	215	
116	SIG2 AMBER	216	
117	SIG2 RED	217	
118	SIG3 GREEN	218	
119	SIG3 RED	219	
120	SIG4 GREEN	220	
121	SIG4 RED	221	+5V Logic Supply
122	SIG5 GREEN	222	0 volt (Earth)
123	SIG5 RED	223	
124	SIG6 GREEN	224	
125	SIG6 RED	225	

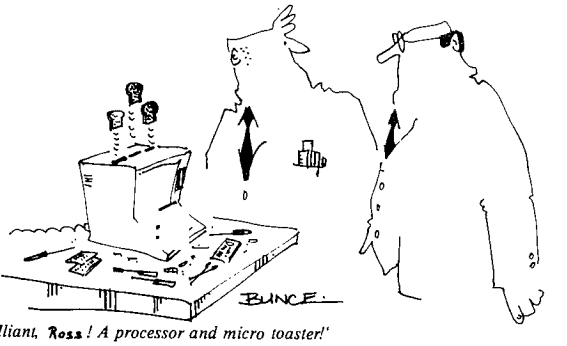
b) A 10 way ribbon cable connects between the Train Set Interface and the computer controlled Track Controller.

PIN	FUNCTION	INTERCONNECTION
1	SPEED1	W I/O out 28
2	SPEED2	W I/O out 27
3	SPEED4	W I/O out 26
4	SPEED8	W I/O out 25
5	EARTH	Train Interface earth
6		
7		
8	EARTH	Train Interface earth
9	- PULSE	DIR 6
10	+ PULSE	DIR 8

c) There are 2 x 34 way ribbon cables between the Wire I/O and the Train Set Interface. These are designated Wire I/O INPUT and OUTPUT.

WIRE I/O PIN INPUT	FUNCTION	CONNECTION IN TRAIN SET INTERFACE
1	TD1	Cct 1 T2c
2	TD2	Cct 2 T2c
3	TD3	Cct 3 T2c
4	TD4	Cct 4 T2c
5	VOLTF	Cct 5 T2c
6	TD5	Cct 6 T2c
7	TD6	Cct 7 T2c
8	TD7	Cct 8 T2c
9	TD8	Cct 9 T2c
10	TD9	Cct 10 T2c
11	TD10	Cct 11 T2c
12	TD11	Cct 12 T2c
13	TD12	Cct 13 T2c
14	VOLTR	Cct 5 T3c
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33	+5VOLT LOGIC SUPPLY	
34	EARTH	

WIRE I/O PIN OUTPUT	FUNCTION	CONNECTION IN TRAIN SET INTERFACE
1	TC1	SIGNAL UNIT 1
2	TC2 + SIG1	SIGNAL UNIT 2
3	TC3	SIGNAL UNIT 3
4	TC4 + SIG2	SIGNAL UNIT 4
5	TC4A	SIGNAL UNIT 5
6	TC5	SIGNAL UNIT 6
7	TC6 + SIG3	SIGNAL UNIT 7
8	TC7	SIGNAL UNIT 8
9	TC8 + SIG4	SIGNAL UNIT 9
10	TC9	SIGNAL UNIT 10
11	TC10 + SIG5	SIGNAL UNIT 11
12	TC11	SIGNAL UNIT 12
13	TC12 + SIG6	SIGNAL UNIT 13
14	WDOGOP	Watch Dog Mono pin 14
15		
16		
17	PNT1M	POINT CONTROL 1
18	PNT1C	POINT CONTROL 2
19	PNT2M	POINT CONTROL 3
20	PNT2C	POINT CONTROL 4
21	PNT3M	POINT CONTROL 5
22	PNT3C	POINT CONTROL 6
23		
24		
25	SPEED8	10 way ribbon 4 T4
26	SPEED4	10 way ribbon 3 T3
27	SPEED2	10 way ribbon 2 T2
28	SPEED1	10 way ribbon 1 T1
29	FWDREV	T6b
30		
31		
32		
33	+5VOLT LOGIC SUPPLY	
34	EARTH	



'Brilliant, Ross ! A processor and micro toaster!'

Press - The Next Generation

An inside look, by Chris Bobbitt

By now, if you have not seen the advertisements for Press, Asgard evidently have not done their job well enough. Even if you have, despite all that has been said and advertised about the program these last few months it is hard to condense a complete description of this remarkable program into 2 pages so they used 5 large pages. Hopefully, this article will serve a number of purposes; to clear the air about what the program is and what it is capable of, to better inform people who have already purchased the program, and for those on the fence (we will make no bones about it), to hopefully persuade you to consider purchasing the program.

This article is not a review of Press, but is instead an overview of the program, its origins, and a short history of word processors in general.

It has been said before but it is worth repeating. Press is the object of an intensive development effort that has spanned the last year. It is principally the product of 3 people: Charles Earl (who is responsible for virtually all of the programming), Ruth O'Neill (who is responsible for most of the documentation), and Chris Bobbitt. Unlike most other TI99/4A software projects, this was not a part-time effort by hobbyist, instead it was a carefully planned and executed project by relative experts in the many fields required by the scope of this project.

Charles Earl is principally known for his Telco (which is widely considered the best terminal emulator for the TI99/4A and 9640), and also co-authored Batch-It (a batch processing language for the TI99/4A and Geneve). While a relatively recent arrival to the TI99/4A scene, Charles has years of assembly programming experience on the TI99/4A and elsewhere, and is widely considered one of the best programmers for the machine.

Ruth O'Neill is an extremely capable writer best known for the documentation of Telco. Ruth has a degree in translation, but prefers to use her training in writing and editing. In any good development project, the person writing the documentation has a large role in the design and implementation of the program (if not the mechanics). If it cannot be easily explained on paper it sure will not be easy to use by the user. Ruth and Charles understand this well, and both the program and the documentation reflect it.

Finally Chris Bobbitt has managed Asgard Software since 1984, and over the years has developed skills in graphics arts and package design, documentation, and the more mundane arts of manufacturing and distribution.

Press is one of the most complicated projects Asgard has ever attempted, with over 15 types of components to plan, design and manufacture.

Press is the result of a partnership between Asgard Software, Charles Earl and Ruth O'Neill. Over the last year the concept has been defined, redefined, and refined. But, rhetorically speaking, what is the concept of Press?

Press is the first original word processor since TI-Writer was released in late 1982. The first word processors were written in Extended BASIC, and date back to within months of when the TI99/4 was originally released. The first assembly language word processor, Companion, was released in the middle of 1982. This word processor was based on the (then popular) Wang word processing system (incarnated on PCs as Multimate), which utilized a number of symbols to represent formatting commands. Later that year, TI introduced its own word processor which was based on the Wordstar word processing system (which found its inspiration in the mainframe text formatter Runoff, found on TIs own minicomputer systems).

Until Press, all TI99/4A and 9640 word processors were based on systems devised in the late 60's to mid 70's. Almost all word processing systems released since 1983 have been variations of TI-Writer (TK-Writer, BA-Writer, Funnelweb Editor, and My-Word to name some of the more common ones). This is mainly because TI made the program public domain on exiting the home computer market, and the source code came to be distributed through the usual non-official channels. Over the years TI-Writer and its standard acquired inertia because it was the first widely used TI99/4A word processor, and programmers tend to be lazy and stay away from re-inventing the wheel. Programmers interested in word processing spent most of their time correcting TI-Writer's deficiencies. Word processing users spent their time learning the eccentricities of TI-Writer, and template packages such as Plus! (by Jack Sughrue), became very popular.

However, the rest of the computer industry has not been standing still. The level of competition in both the Macintosh and MS-DOS worlds has caused a rapid evolution in the user interface and capabilities of word processors, if not the actual concept of them.

One of the first innovations in modern word processors was the combination of the editor and the formatter. Instead of making it 2 separate steps (as it is in TI-Writer and in most PC word processors prior to 1984), formatting was done at the time the text was composed. The advantage to this is pretty obvious. The amount of guesswork required to create a properly formatted document was greatly reduced. However, such an approach was not possible until (supposedly) standard memory amounts increased and processors were fast enough, and (more likely) graphics programming knowledge had reached a certain level. Combining the formatter and the editor was not possible until the concept of "what you see is what you get" (hereafter WYSIWYG) became more widespread.

Because of the need to display special type styles (such as underline and bold), and the micro-formatting sometimes necessary (where letters have to be adjusted on a pixel basis), WYSIWYG and by extension, combining the formatter and the editor required very graphics capable systems. That is why the first modern word processors got their start on the Apple Macintosh, which was dramatically more graphics oriented then the text oriented CP/M and MS-DOS systems of that time.

The Macintosh is also responsible for a more recent change in formatting capabilities which is just now becoming more common on MS-DOS machines (with the release of Word Perfect 5.0), full-column support. Early word processors occasionally offered crude or clumsy support for multiple column pages, but generally such utilities were difficult to use at best. WYSIWYG, again, opened doors to new capabilities, and the ability to create newspaper style and tabular style columns has recently become more common, and more demanded by users. Multimate, and later versions of Wordstar introduced another innovation that later came to be considered an integral part of modern word processors - a "virtual" text buffer. Virtual memory, or the act of treating disk storage as an extension of regular memory, was itself not a new concept. Text editors on mainframe systems commonly supported this feature in the early 70's. However, it did not really became very common in microcomputer word processors (and text editors) until disk drives began operating at faster speeds. The micro-computer word processors that supported it in the late 70's and early 80's often did so at a great sacrifice in speed (and hence, a dramatic increase in user frustration). In all modern word processors, the area available for a text file is limited only by the available disk space.

As word processors continued to evolve, the spelling checker came to become more then just an after thought. After integrating the editor with the formatter, it soon was obvious that considerable time was being wasted in leaving the word processor, spell checking the document, and then returning to the word processor to continue writing. The spell checking process was much faster if paragraphs could bechecked as they were composed (and hence fresh in the mind of the writer). In 1984 and 1985 spelling checkers started to become integrated within word processors until, by 1986, virtually every word processor on the market had a spelling checker built-in. Such programs as Borland's Lightning even offered spell checking on the fly, but again, at a great sacrifice in speed and memory. The real time spelling checker (which checks as you type) has not yet became the norm. That is because it is rather a nuisance for the program to constantly interrupt you when you are composing a sentence (and your thoughts). However, the integrated spelling checker, which allows you to check what you want, when you want to do it, became very popular, In 1986, the computer thesaurus also became a popular add on, but as of yet still is not commonly included in word processors.

The WYSIWYG approach to the user interface necessitated another change in they way word processors worked, but this time of a less dramatic nature; block operations. Block oriented text editors were also introduced in the 70's, but many early word processors and text editors (including TI-Writer) were line oriented. The advantages of block oriented editors over line oriented ones are more subtle, but can be profound. Block oriented editors allow you to select a block once (usually by marking the ends of it with the cursor, or a mouse), and then execute a move, copy or delete operation. While this is functionally similar to a line oriented editor, blocks are inherently easier to use. This is because block oriented systems usually indicate the selected area before a potentially destructive operation. Many functions can be easily made into a potentially destructive operation, and many functions can be easily made to operate on a block as easily as they do the whole document (the integrated spelling checker discussed earlier, and the search and replace function to name two).

Finally, unlike line operations, blocks do not necessarily have to be limited to everything on the lines selected; more sophisticated modern word processors allow you to select a range of columns within a range of lines - literally any "block" of text on the screen! Block operations of this sort make moving columns in a table a lot easier, and more natural.

Finally, many other less profound functions have become common on word processors in the last 5 years, mainly due to careful research of what people want in a word processor. Some word processors generate tables of contents, foot notes and an index, and even have outlining capability. Most include automatic and manual hyphenating, more sophisticated search and replace functions, line graphics, more capable header and footer functions, macros, and disk management functions within the program. Only the last innovation mentioned has to any degree been implemented in TI-Writer derivatives to date.

To sum up, modern word processors have a number of characteristics. They combine the editor with the formatter in a WYSIWYG manner, support a variety of columns and relatively free form page creation, feature an integrated spelling checker, offer "virtual memory" support for the text buffer, and they all have block oriented functions. They also have a plethora of lesser features that extend common functions and automate previously time consuming tasks.

Additionally, they support all the functions found in more primitive word processors (such as mail-merge). It should be obvious, at this point, that TI-Writer does not even compare to a modern word processor. Technically, it should be very difficult or impossible to support many of these features on the TI99/4A, what with the severe memory limitations, and slow bitmap graphics found in a TI99/4A.

However, this is exactly what we intended to do with Press.

Press has numerous features, including all of the major ones mentioned above as crucial components of modern word processor. Press is not "one" program, actually it is more akin to an operating system that reads in little programs for each function as they are needed by the user. Not only is the text area in "virtual memory", the program itself actually loads in parts of itself as needed, and the vast majority of the program resides on the disk at any one time. The advantage to this is greatly expanded functionality, and the ability to do things which are technically "impossible" on a TI99/4A.

Press occupies a single SS/SD disk plus utility disk, and the spell-checker dictionary uses 3 other SS/SD disks. It comes with an extensive manual that includes a highly detailed table of contents, an index, and tutorial. Tabs are included for quick reference. The program is packaged in an "IBM-style" binder with slip-case, includes a disk-page to hold the 5 disks, a quick- reference card, a TI99/4A keyboard strip, and a registration card. The documentation, and the program itself to some degree, is geared towards the user converting from TI-Writer to Press. As such, the Press user will note that some commonly used functions are accessed with the same keys, TI-Writer files can be converted (the formatting commands are even converted to Press format), and the documentation features numerous sections on the differences between TI-Writer and Press where they are relevant.

The most obvious change when the program loads is a difference in the user interface. TI-Writer has a "command line" interface, in that, there is a separate area on the screen where the user types in commands. Press does not offer a command line, instead it has a menu bar. The user uses the arrow keys to move a highlighted bar over the option of choice, and selecting the option desired, chooses the sub option needed in a similar manner from the sub menu presented. Whenever you select an option that needs more information (such as loading a document or entering a search string), a window appears on the screen and the desired information is entered within. This method is great for beginners (and considerably more natural then a command line), but soon grows old for the more experience user. Hence, Press also allows you to select each function through a direct key press, bypassing the menus entirely. Press, like all modern word processors, has different interfaces for the beginner and the advanced user. Press offers a plethora of options, the vast majority of which are selected directly and through menus. These options are arranged in the manual, and within the program, by type.

In editing, Press will allow you to move the cursor up and down by line, screen or by page (the program keeps track of actual physical pages) using familiar commands. It also allows you to window over to display up to 256 columns (which can be defined with the page format options). Directly on the screen, Press displays normal text, underlining, bold text, italics, overstriking, and super and subscript text.It also allows you to combine any of these attributes. Additionally, it supports such functions as hidden hyphens, an undo key (not to be confused with the " OOPS!" key) which reverses many commands, the block functions discussed (including moving, copying, deleting, or saving a block to a file), a range of text deletion functions (delete character, word, sentence, line, to the end of line, paragraph, a column, a page, or the document), an "OOPS!" function which allows you to recover your last 2 deletions. It has a comprehensive search and replace that is case sensitive (it will preserve upper and lower case letters), is bi-directional and supports several types of wildcards. In the editing mode Press also allows you to view and edit embedded formatting codes (which are printer independent), including ASCII characters directly in text. You will also be able to draw line graphics with the cursor (for boxes, borders and tables), as well as insert printer codes directly.

Press also has a large number of formatting commands. Press will allow you to define virtually all the characteristics of a page, including the vertical dimensions, the margins, the tab settings, indents and outdents, the pitch, justification style (left, right, center, or full), the header and footer, the page number set, columns on the page, the number of lines per page, the top and bottom margins, the lines per inch and the line spacing. While you type in the text you can change some of these options (such as the justification) in order to create text effects on the screen (such as a centred title followed by fully justified). Press also supports columns by offering you two column types, and will allow you to enter dimensions of each column manually or will automatically generate them for you.

Once a document is created it is printed, and Press offers you tremendous flexibility there as well. Utilizing your printer with Press is simply a matter of selecting your printer from a list of the available types, Press takes care of the details. Press will allow you to print the document currently being edited, print a document, save to another disk, choose from several paper feed options, set the printer device name, insert printer codes, and even print a previously selected block of text.

Press also features a full range of disk management functions. When you first start Press you must open a document, either a new one or a previously created one. Press requires a disk filename so it knows where to put your text as you compose it. (Press only keeps a portion of your document in memory at once.) After opening and creating a document, you can close it to save your file changes. A unique safety feature inherent in Press is that since it saves your document as you compose it, if your computer crashes for some reason you will rarely lose very much of your work. Press will also allow you to read a document in and merge it into the one you are currently using, save a previously marked block of text to its own file or the end of another, get a disk directory, view a text file on disk, copy a file, move it, delete, rename, protect and unprotect files. Press will also allow you to save a file as straight ASCII for transfer to another computer, or read in an ASCII or TI-Writer file. Press will interpret the formatting commands as it reads them, and presents the formatted document to you on the screen in Press format. Finally, for harddisk users, Press will allow you to change the name of the directory currently in use (the path).

As mentioned previously, Press has an integrated spelling checker. Not only will it check any previously selected block of text, it will also check a word, a page, a range of pages, the whole document, or an external document (one not being edited). The same function will also optionally count the number of words in a document for those people who write professionally or for school. While checking, if it finds a word not in its dictionary the word will be displayed within the line it was found on, and you will be given the option of skipping the word, skipping just to ordinary, (editing) the word. Press will also optionally change all occurrences of a misspelled word (so if you consistently spell something wrong Press will do most of the work for you).

Press is a series of stored keystrokes which may be recalled in the order they were entered any time while editing a document, These keystrokes are used to combine several commands into one more sophisticated one. For instance the manual contains a macro for duplicating lines within a document and consists of 8 separate keystrokes. Most modern word processors support macros to some degree because they make editing text, and creating custom commands, much simpler.

Finally, Press has a well developed mail merge function which is completely compatible with the TI-Writer mail merge format. So, if you use this function in TI-Writer for form letters and business correspondence, you can use the same files in Press as is.

One of the most exciting capabilities of Press is not a function most users will see or use very often. Press can be configured to take advantage of virtually any system. Press will use to its advantage the full range of RAMdisks, superCarts, GRAM devices, all 80-column cards, keyboard enhancements, any number of drives and hard-disks, and (of course) the Myarc Geneve. It is axiomatic in Press that virtually any hardware you purchase for your system will result in an increase in performance, and in some cases, functionality. For instance, while the program will readily function with a single disk drive, it is very difficult to use the spelling checker with less then two. An extremely efficient TI99/4A system would be one with at least one disk drive and a 512K RAMdisk. You could put the entire program, dictionary and user dictionary on the RAMdisk for fast access, and use a floppy to store your document (where it would be safest in case of power problems anyway). Press could fully utilize a 9640 system with 720K floppies and a hard-drive or two. The more hardware you have, the more you can make Press do and the faster it will do it. Adaptable programs are also relatively rare in the TI99/4A world. Most programs are designed with a minimum system in mind and will rarely take advantage of more then perhaps a few disk drives.

Press is a fully featured word processor for the TI99/4A and 9640 that is on par with popular PC-based word processors. It was designed to do things that other word processors do, but does not owe much to any one specific program. It was designed to take best advantage of the capabilities of the TI99/4A and the Geneve, and was written within that context. Like TI-Writer, it is a word processor for a TI99/4A computer, not something ported to a TI99/4A computer.

While we believe (and you may agree) that it is the most capable word processor for the TI99/4A and 9640, Asgard do not pretend that it is the end in word processors.

There are many things on their list of features that did not make the initial release due to time considerations. We may periodically add these functions. It is here where the Asgard commitment to user service comes in. Asgard will mail announcements of upgrades as they occur to all registered users, and offer then at a price commensurate with the scale of the changes. In order to reach unregistered users (if any), Asgard will also announce upgrades in TI99/4A magazines.

Finally, Asgard will be running a regular column in the Asgard News Publication called Press Report (this is the first installment), oriented to the Press user. It will feature notices of updates, useful macros, tips and advice to Press users. o

continued from page 9

If you have a RAMdisk, save it on your boot drive (that is the one with the MENU program), as "XBASIC", then make it one of your menu selections. You will find that it will get you into Extended BASIC quickly without looking to drive 1 first. The program has had some delays added (FOR NEXT loops) to enable it to work with the fast 32K memory on the 16 bit bus with no wait states and the start address has been adjusted to make the second byte of the address zero to avoid a potential problem pointed out by George Meldrum.

```
100 CALL CLEAR
110 B=5 :: F=16
120 C=16*(F-1)+(B-1)
130 CALL INIT :: CALL LOAD(9976,C,C,C,C,C,C,C,C,2,0,
    7,15+B,4,32,32)
140 FOR I=1 TO 2 :: NEXT I ::
    CALL LOAD(9991,48,2,0,8,0,2,1,38,248,2, 2,0,8,4,32,
    32,36,2,0,8,8,4)
150 FOR I=1 TO 2 :: NEXT I ::
    CALL LOAD(10013,32,32,36,2,0,8,16,4,32, 32,36,2,0,
    8,24,4,32,32,36,4,91)
160 CALL LOAD(-31804,39)
170 CALL LOAD(-31952,255,231,255,231)
```

o

continued from page 1

I would like to thank a few other people besides Rolf who have made our job easier by their efforts. Terry Phillips has always provided his column in plenty of time and has also sent the mailing labels, normally without prompting. In fact he has always responded to requests of mine promptly and efficiently. Percy Harrison has kept me supplied with money to pay for postage and incidentals most efficiently. Vincent Maker has provided Jenny with most of the information for her Younger Set column and has done his best to try and get some communication with younger members going. Then there are the Sub-editors of the BBS who provide so much material for the BBS which then appears in the TND. These include Robert Brown and Stephen Judd for Games, John Ryan (Coffs Harbour) for retyping so many articles from other newsletters, Warren Welham for his efforts with the publications and getting them to us as soon as he could and of course Ross Mudie who manages to tie us all together with the BBS and still produce interesting and quality articles on his own wide range of interests. I think that the group would have a great amount of trouble keeping everything going without Ross doing his 200% all the time. And he is such a nice bloke too!

****

As I sit here in front of my screen (or is it in front of me?) my thoughts turn to the future. We are supposed to do that at this time of year with all the fortune tellers asked to predict the events of the coming year. I reckon I can do as good a job as they can so here goes with some thoughts about TIsHUG in 1989. There will be changes in TIsHUG this year (that is a good start!) with either the directors changing their approach to the operation of the group or the directors deciding to allow owners of other types of computer to become members of the group and to share in the assets of the group. After all, there is so much money in the bank and there are fewer TI99/4A products available to spend the money on and the number of members is reducing to the point where it is not possible to produce such a good magazine at the current subscription rate. Introduce a new source of members and off we go again. It has been too difficult to attract people to the group from those who still have a TI99/4A in the closet. It is more important to have a large membership from anywhere than to ensure that the members with interests in the TI99/4A have an excellent and viable group to turn to for information and products. It is an interesting phenomenon around the world that TI99/4A user groups are very friendly and cohesive. Very strange! If you are worried about my future for TIsHUG, perhaps you should ask the prospective directors to declare their interests in other computers before you vote them into office for a year. Remember, it only takes 3 out of the 5 to make a decision.

****

Talking to members recently, which happens more these days as editor of TND, there seems to be some discontent about the direction that TIsHUG has been going, or perhaps the lack of direction. I find myself not willing to criticise people who are giving their time freely to look after the interests of the group, but my worries and those of our regional group have been brought to two successive AGMs with no sign that the directors have even bothered to discuss these issues. I do not want to lose any members from the group so I do not want to upset anyone, but it is now clear that others are starting to see some problems. Let me ask you all to imagine that you were unable to communicate with the group except by mail. What would you want the group to provide for your membership? A quality, informative, mistake free newsletter? Cheap Public Domain and Fairware software? Ready access to Information from overseas? I believe that TIsHUG only properly provides the first of these and even Shane would agree that since the Illawarra Regional Group has been responsible, the quality, accuracy and amount of information has increased enormously. So the TND is then, at the moment, about the only thing that TIsHUG offers to its members who cannot get to Sydney and who do not want to pay STD rates to the BBS. (We have even attracted people from overseas to join TIsHUG, like Stephen Shaw and Jane Laflamme.) What does TIsHUG do to help those preparing the TND? It provides a printer and printer buffer and since I begged on bended knees, a guillotine. The software library has a system provided I believe, the BBS has a system provided I believe, but the TND editor is not even entitled to a miniPE system, which I also asked for. I believe that the Editor should be provided with a complete system (or a Geneve) with console, miniPE system with 2 DSDD disk drives, an 80 Column card and monitor and the program Press as well as the printer and printer buffer. From experience, if anyone else in the family is a user, then you need a second system for the editor. If I was to take on the Editor's job again, these would be my minimum requirements.

****

I start the look at the newsletters with May and June issues of the LeHigh group. For those whose geography is not good this is based in a steel town called Allentown, just north of Philadelphia. There is an article by Jack Zawediuk on connecting up a 3 1/2 drive to a disk controller. Jerry Boyer talks on the Geneve and on all the familiar software which runs on it. Jack Sughrue talks some more about the good old days, Chick De Marti explains about bits and bytes with BASIC and Tony Falco presents some Math one liners.In June there is a review of Spad XIII by Brad Snyder, more from Jack Sughrue on the good old days (should be read by all aspiring directors), a humorous approach to repairing, more on bits and nibbles by Jim Swedlow and some more one liners for graphing by Tony Falco.

October from Northern New Jersey contained the first of a series on disk drives by John Willforth, a program from Ed Machonis to print a tag, various snippets from Chick De Marti including the news of a Forth disk which loads into the 8K SuperCart space from Paul Newmeyer, 290 S. Ridge E, Geneva, Ohio 44041. In November there is a review of two BBS programs, MXT and Paradigm, by Art Byers.

The PUG Peripheral from Pittsburgh for September has TI-Writer tips from Stan Katzman, the second Forth tutorial by Lutz Winkler, more on cassette files by Mickey Schmitt, Jack Sughrue has more views of interest and a hint on naming letters so the cataloguing sorts them, Tips for beginners (15) by Frank Zic, Multiplan (10) by Audrey Bucher and Kiddie Corner by Sue Harper.

ROM for October from Orange County, California has TI Bits from Jim Swedlow And So Forth (35) by Earl Raguse. November's issue has TI Bits again And So Forth (36). They are all interesting.

Spirit of 99 of October from Central Ohio (CONNI) is based in Jim Peterson's home town has as article on Funnelweb by Charles Good, Jack Sughrue's Impact-99 column, more on cassettes by Mickey Schmitt and Stan Katzman on TI-Writer. November contains photographs of a number of well known TI99/4A people with a review of TI-Base by Jack Sughrue, a music program which won a contest called Sonata by Kevin Noesner, more from Mickey Schmitt on cassettes and from Stan Katzman on TI-Writer. In December Mickey Schmitt's series on cassettes continues as does those of Stan Katzman on TI-Writer and Jack Sughrue's review of TI-Base, Jim Peterson explains a bug in TI-Writer if you are trying to remove control characters from a file, Jean Hall reviews 1000 Words for converting picture files from TI-Artist to D/V80 for inclusion in a TI-Writer document, Dr Roy Tamashiro gives a program to help with retirement income, Mark Schafer introduces a rather neat way to save space on a disk if you want several versions of the same program and Phil Townsend gives some hints on fixing printer problems.

The November issue of the Tacoma Informer has a review of the Myarc HFDCC by Steve Mickelson and a review of TI-Base by Gene Kelly.

TI-SIG of the San Diego Computer Society for September has two of Tony McGovern's Assembly Squeezing articles and W. Hamilton talks about changing the underlining and emphasize characters in the formatter. In October, Waldo Hamilton discusses the construction of a ribbon cable adapter for the GROM port and also gives an interesting way to have BASIC issue multiple beeps.

TopIcs from Los Angeles for September mentions the visit by Peter Gleed (Melbourne) to LA, Beginning Forth (4) by Earl Raguse, Chick De Marti notes, EZ-BASIC and a review of Super Extended BASIC, BBSing with Danny and relative File Sorts in Extended BASIC. October has a

continued on page 43