TIsHUG NEWS DIGEST

TISHUG, PO Box 149, PENNANT HILLS, NSW,Aust.2120. Registered by Australia Post Publication #NBH5933

TISHUG NEWS DIGEST

TIsHUG NEWS DIGEST
Vol.5 No.5 June'86

The official Newsletter of the T.I.'s Homecomputer Users' Group (Australia).

Editor and Layout designer:
SHANE K. ANDERSEN
Assistant Editor & Reporter:
KEIR WELLS
Photographer:
MAURICE STEWARTSON

CONTRIBUTORS:

Revell Oatway
Dennis Hodgson
Terry Phillips
Jenny
Ross Mudie
Ben Takach
C M Ehninger
Geoff Trott
Shane Andersen
Fred Morris
Wade Bowmer
Rolf Schreiber
Steven Shraibman
Chris Buttner
Daniel Harris
Kevin Watts
Network

Published by:
TIsHUG - Australia

Printed by:
APPLE PRINT & COPY CENTRES
160 Castlereagh St,Sydney.2000
(02)264 8111

Club Address:
P.O.Box 149,
Pennant Hills,
N.S.W. 2120

Monthly Meeting Place:
WOODSTOCK COMMUNITY CENTRE
Church Street, BURWOOD.
First Saturday of each month
except January Public Holidays

Staring at 2pm except on FULL DAY TUTORIAL/WORKSHOPS.

Co-Ordinator:
FRED MORRIS . . . (02)871-3873
Secretary:
JOHN ROBINSON . . (02)848-0956

Up Front

WELCOME TO THE FIRST ISSUE OF THE TIsHUG NEWS DIGEST ! The Newsletter of TI's HOMECOMPUTER USERS GROUP.

This month there are two main events which will be worth attending. They are (1)The FULL DAY TUTORIAL WORKSHOP at our new meeting house..."WOODSTOCK COMMUNITY HALL". A grand old mansion which will be used to house our rather unique tutorial day for programmers and game players alike. See all the details about this on page 3.

(2)The Melbourne TI User group conducts its very first NATIONAL TI FAIR, and there will be many wonderful items on show, and introduced for the first time to this great country of ours. TIsHUG will have its own stand there, staffed be members of this group who plan to travel to Melbourne in a MINI BUS and/or fly down to share in this day...June 14th '86.

We will also be conducting our Regional Home Groups and Special Interest Groups and we hope that you will join in the fun of sharing, at these get-to-gethers.

I have prepared this magazine, in the hope that you will enjoy its contents, wit many varied topics,news & programs, with special thanks to all those this month who have taken the time to contribute. I have a stack of other files or contributions which have not been able to make this issue due to space...some of these articles that I plan to print in next months TND include the following...
(1)Review by Chris Buttner of the TULPI INTELLIGENT MODEM.
(2)Article by Terry Phillips on EIGHT DISK COPIERS FOR YOUR TI.
(3)Word Processing with Geoff Trott from our Illawarra Regional Group.
(4)and also by Geoff, PART#2 of a fantastic tutorial on Assembler.
(5)TIPS FROM TIGERCUB #23 by Jim Petersen.
(6)HOW TO USE MINIMEMORY .. from Sacremento Users Group.
Yes, these are just a few of those articles that have had to be held back to next months News Digest, but will be well worth waiting for. I am only sorry that we didn't have the room.

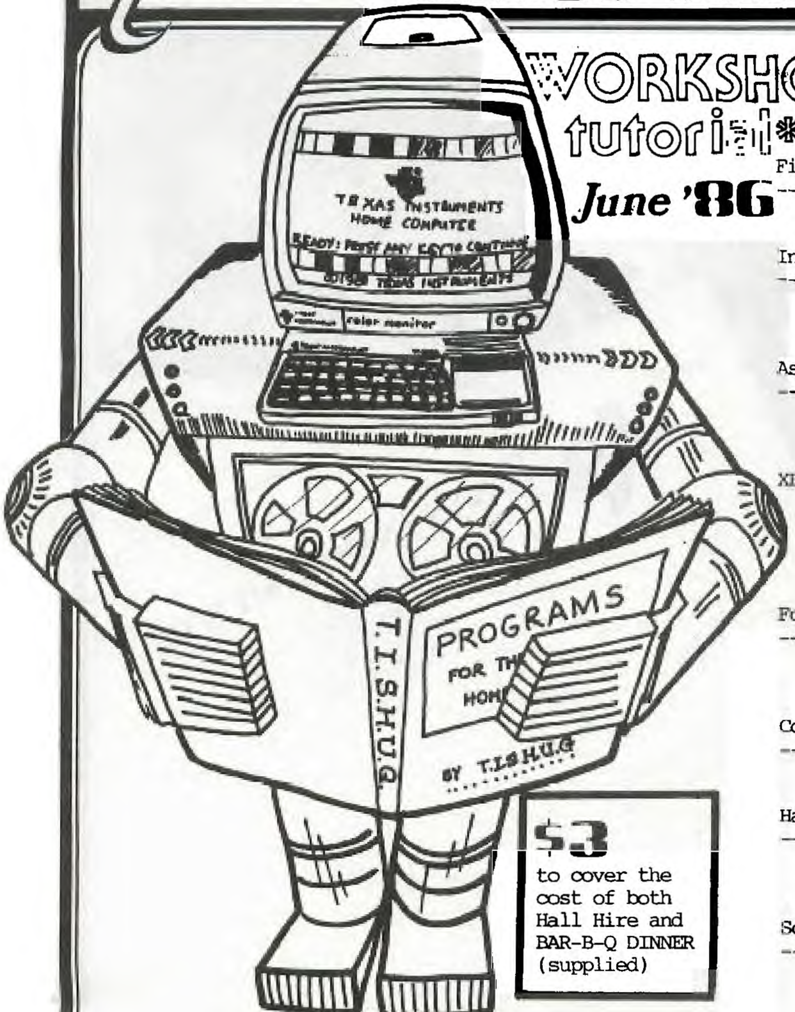
IN THIS ISSUE

I have also continued with the PULL-OUT CENTRE PROGRAM PAGES to place in that special binder of yours. It continues with such programs as..."CUBIC" Q*bert type program, "ON SAFARI", and "TI MARQUE".

Also, you will find a loose CUT-UP PAGE for you to use for ordering of your montly SHOP items and other areas of club activities.

Well, thats it from me. See you at the FULL DAY TUTORIAL WORKSHOP
Regards
SHANE ANDERSEN (EDITOR).
Shane Andersen

WORKSHOP tutorial* June '86



$3
to cover the cost of both Hall Hire and BAR-B-Q DINNER (supplied)

ALL DAY WORKSHOP JUNE 7

by Revell Oataway - EDUCATION CO-ORDINATOR

This year will be better than ever. A wide range of demonstrations, tutorials and workshops will cover most aspects of computing. We were fortunate to be able to obtain speakers and volunteers so that the full range of TIsHUG members, from beginner to expert; from developer to user, will benefit. At the time of writing, the following are definate. (More may be available on the day, as there is still time available)

Program Clinic

Graham Hollis, our "Crisis Line" operator wil be available to help with ailing B or XB programs. Graham has a tape based system, so bring a tape, or preferably a print-out of your program.

Music

Russel Welham will show how to make our computer sing.

Sprites

Chris Ryan will pass on the secrets of easy and effective programming of sprites.

Flow Charting

Ross Hardie will show how this fundemental part of program writing is both simple and essential.

XB Programming

Ben von Takach has written a data base in XB. He will demonstrate this program and describe the stages in building it as a guide to the construction of other complex program.

File Handling

Kevin Gardener will expand your knowledge of this most important area of large programs.

Information Processing

Fred Morris will discuss aspects of Navarone Data Base, Personal Record Keeper, TI Writer, Multiplan, and similar software.

Assembler For Rank Beginners

Mike Slattery will convince programmers that assembler is not difficult with his very successful introduction to this topic.

XB / Assembler Links

Ross Mudie will give a full tutorial on this most important area. A must for all advanced programmers that want to incorporate the speed of assembler into Basic programs.

Forth

Terry Johnson will introduce this powerful ability of our computer. Forth is nearly as fast as Assembler, but much easier to handle.

Communications

Shane Andersen will introduce and update everything you need to know about this.

Hardware Clinic

Robert Peverill and Peter Schubert will give advice and operate on hardware that does not need sophisticated testing equipment.

Software Demonstrations

Russel Welham and Keir Wells will demonstrate the best of the large variety of software available for our computer.

In addition, all the TIsHUG services will be available to members. Terry Phillips with the Software Library - BIG COPY SESSION planned, Chris Buttner wil have the SHOP going full tilt and Brian Graham will have the PUBLICATIONS LIBRARY for members use.

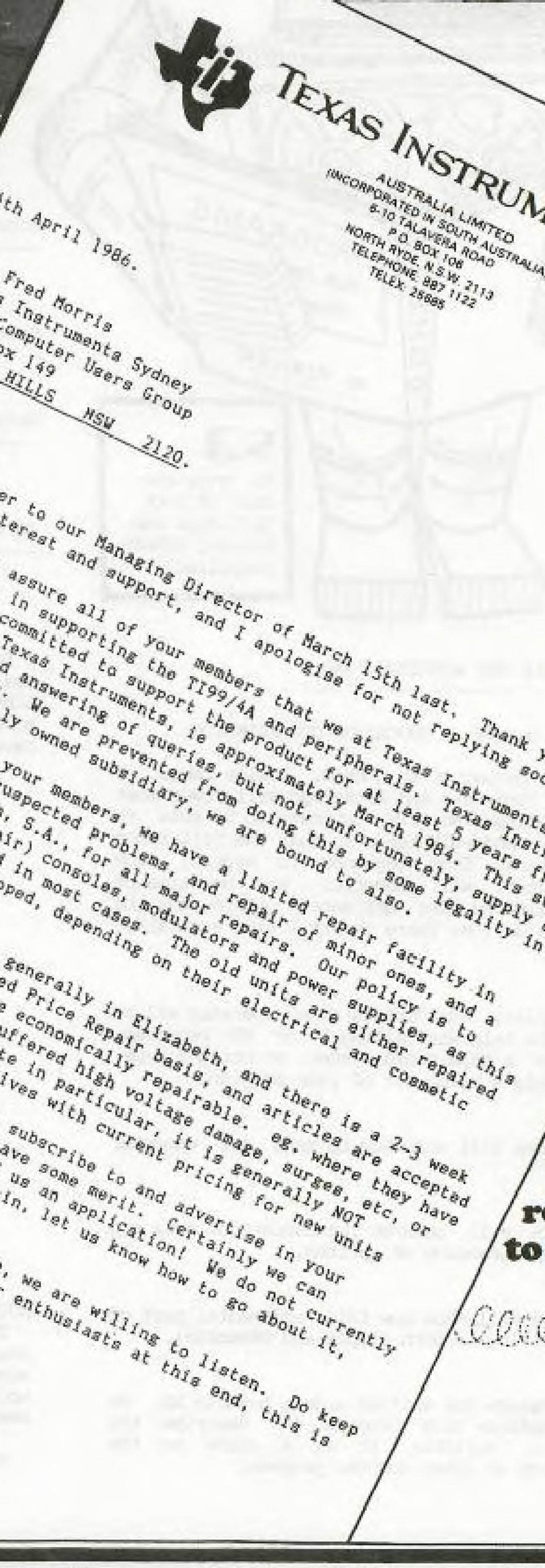
ACTIVITIES TIMETABLE

10.30 WORKSHOP START
11.00 Software Demonstrations.
Publications Library.
Software Library.
Software Copying.
Program Clinic.
Hardware Clinic.
Flow Charting.
Assembler-XB Links.
Assembler for Rank Beginners.
12.30 LUNCH - BARBEQUE FOR MEMBERS AND FAMILY
1.30 Program Clinic (cont)
Assembler-XB Links (cont)
Hardware Clinic (cont) Music; Sprites.
XB Programming.
Forth.
Information Processing.
3.30 AFTERNOON TEA/COFFEE
4.00 Assembler-XB Links (cont)
File Handling.
5.00 WORKSHOP CLOSE.

NOTES.

The schedule is arranged so that subjects in the same areas have been grouped together, and will be in the same room. Also, by the time the BIG DAY arrives I hope to have a few more activities scheduled for our members.

SEE YOU THERE!



TEXAS INSTRUMENTS

AUSTRALIA LIMITED
(INCORPORATED IN SOUTH AUSTRALIA)
6-10 TALAVERA ROAD
P.O. BOX 106
NORTH RYDE, N.S.W. 2113
TELEPHONE 887 1122
TELEX 25665

4th April 1986.

Mr. Fred Morris
Texas Instruments Sydney
Home Computer Users Group
P.O. Box 149
PENNANT HILLS NSW 2120.

Dear Fred,

Re your letter to our Managing Director of March 15th last. Thank you for your continuing interest and support, and I apologise for not replying sooner.

Firstly, let me assure all of your members that we at Texas Instruments have an ongoing interest in supporting the TI99/4A and peripherals. Texas Instruments itself is firmly committed to support the product for at least 5 years from last date of sale from Texas Instruments, ie approximately March 1984. This support includes service and answering of queries, but not, unfortunately, supply of hardware or software. We are prevented from doing this by some legality in the U.S., which as a wholly owned subsidiary, we are bound to also.

For the information of your members, we have a limited repair facility in Sydney for checking of suspected problems, and repair of minor ones, and a major set-up in Elizabeth, S.A., for all major repairs. Our policy is to exchange (rather than repair) consoles, modulators and power supplies, as this gives a same day turnaround in most cases. The old units are either repaired for later exchange, or scrapped, depending on their electrical and cosmetic condition.

Peripherals are repaired only, generally in Elizabeth, and there is a 2-3 week turnaround. Repair is on a Fixed Price Repair basis, and articles are accepted on the basis that they may not be economically repairable. eg. where they have been wrongly connected and have suffered high voltage damage, surges, etc, or are physically damaged. Please note in particular, it is generally NOT economical to repair floppy disk drives with current pricing for new units being what it is.

In regards to your suggestion that we subscribe to and advertise in your magazine I agree that this idea does have some merit. Certainly we can subscribe to your magazine, please send us an application! We do not currently have copy ready for advertising, but again, let us know how to go about it, and we will do what we can.

If you have any suggestions how we can help, we are willing to listen. Do keep in mind that we are not necessarily computer enthusiasts at this end, this is our livelyhood, not necessarily our hobby!

Your sincerely,
TEXAS INSTRUMENTS AUSTRALIA LIMITED.

Ron Brown
Consumer Marketing &
National Service Manager.

The smile you send out... returns to you!

SOFTWARE column by terry

Hi, and welcome to another software column. Here it is midway through the year again - how come they go so fast?

First up this month is news of another great Freeware release on disk and available at the club shop as from June. I guess all of you have at least two languages on your systems - Basic & Extended Basic - plus I suppose the majority have Assembly and quite a few have Forth. But who has "C"? Indeed you may ask what is "C". Let me tell you. It is another language (compiled) that is available with the help of this disk for you to play around with to see what you can do to enhance your programming skills. The disk is FULL containing 25 separate files including the full manual (DV80 format) which runs to 121 sectors on the disk. There are even some test programs on the disk which should be most enjoyable. One member who writes regular columns for this News Digest has had a copy of "C" for a few months and reckons it's the greatest. Don't be dissapointed - get a copy of "C" at the June meeting or order by mail from the Shop PO Box 149 Pennant Hills. Cost as usual will be $5.

My software mail bag hasn't exactly been bulging lately, and in fact I have only had one letter. Kevin Cox from Bayldon, up near Coffs Harbour, has written telling me of his problems loading Berlin off tape 1985/11. It seems Kevin can load Part 1 of the program but then has difficulties with Part 2. He has sent his tape back to me and it loads and runs OK off my system. Is anyone else having trouble with this program? Please write and let me know if you are. Kevin was also good enough to send me a corrected version of the 4 Stroke Engine demo and took the time to translate Palace of Ice from French to English. I also gather by his letter that he is interested in obtaining a Tunnels of Doom module. If you can help him out let me know please. Thanks Kevin, it's great to receive letters from members.

John McDonald from Ermington handed in a tape at the May meeting with some programs he has been writing. Unfortunately my recorder refuses to load John's tape but I will keep trying with other recorders. If no luck John I will need to get back to you to see what can be done. Speaking of the May meeting, the Copy-Thon went off well with a number of members taking the advantage of bolstering their software libraries. I hope you all got what you wanted and that we can run another Copy-Thon soon.

Other software to be released in June will be Tape 1986/6 containing the following programs:

ALFA - Extended Basic - this one is from Italy and incorporates good graphics and colors. The idea is to catch objects being thrown by some sort of monster. Instructions are included in the program.

ELIZA - Either Basic - discuss your problems and get advice from you TI.

IN A MAZE - Extended Basic - another one from Italy. The idea being to find your way through the maze. A very colorful program which I think you will like.

INDIAN DICE - Extended Basic - there are a few differing versions of this theme around but this is one of the better ones. The idea is to gain points by throwing dice but only certain combinations pay off. Up to 9 players can be accommodated with this version.

NEW YORK - NEW YORK - Etended Basic - control busy streets of traffic by pressing the appropriate keys to change the traffic lights from red to green and vice-versa. A bit slow but a reasonably entertaining game to fill in time on a cold night.

SQUARES - Extended Basic - written by Mike Slattery of Tishug. You have probably all played this game before. This one has number, letter and color code choices. Sure to keep you occupied.

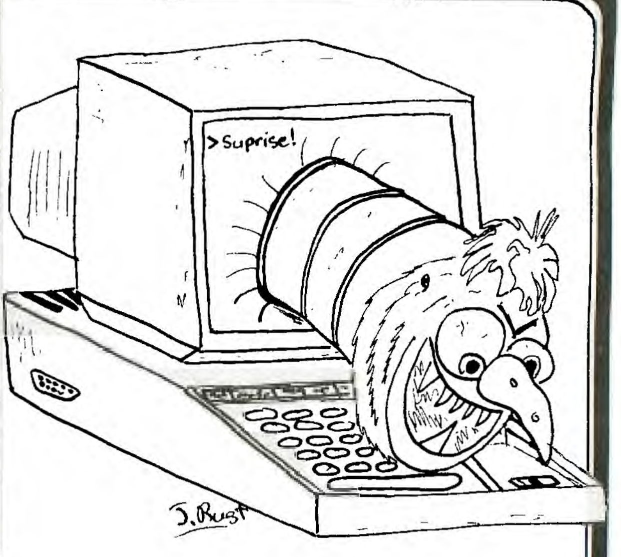
TI POKER - Extended Basic - traditional show poker game against the computer as your opponent. You dont see its cards so you won't know if your being cheated. Not the best card game I have seen but feel free to modify it to suit yourself.

WEBSTER - Extended Basic - Webster the spider drops on his prey, but watch out for scorpions.

The same programs will also be available on disk, but added to the disk are the two versions of Richard Stanford's DISKCAT program. The programs load and run through Option 3 of Editor Assembler.

Since I now have the 2 slimline drives installed, the external TI drive, purchased by the Group sometime ago is now for sale. The drive is still in good order and would be an inexpensive way for you to expand to two drives. The drive is a 110V version and would require a 240-110V step down unit to operate. Further particulars may be obtained by telephoning me on 02.7976313 or by writing to me at PO Box 149 Pennant Hills 2120.
At the last Committee meeting the idea was floated that tapes containing quality programs should be rented to members at meetings for a nominal sum ($2 or so) plus a deposit which would be refunded when the tape was returned at the next meeting. We have the equipment and necessary software to enable this to be done, but before doing so I would be interested in feedback as to whether it would be a going proposition. Again phone comments can be directed to the above telephone number or written comments would also be welcome. ●

If that ghost was Charlie Chaplin, he probably came for his royalties



Jenny's YOUNGER SET under 18's page

Hello to all my young friends. I hope you all enjoyed the recent school holidays and that many of you got away for a well earned break. During the past month I have received 3 letters and here they are.

From Peter Lees of Pennant Hills who wrote and sent me a tape with an entry for the Mouse competition plus a program called Young Ones which he says is a musical demonstration. Peter I am very sorry but try as hard as I could your tape would not load on my recorder. Never mind perhaps you could re-record your programs and send them to me again. Thanks anyway Peter, for the time you took to write.

Kresh Sprem from Penshurst also sent in a Mouse entry. Kresh, your poor mouse, it seems to want to go in several directions at the same time. On the same tape, Kresh sent in 2 good graphics demos, a plane and a racing car. Well done. You have certainly been busy over the holidays.

As I only got 2 entries, and one wouldn't load I guess I will have to call the competition off. I think at least 3 entries need be received before any prizes are awarded. Again thanks to those who did enter.

Also in my mail bag a long letter from Joshua Rust who tells me he has acquired a stand alone modem,TE2 module, Congo Bongo, Burgertime and Sneggit plus a host of other modules. Some of Joshua's high scores are Buck Rogers: 221991, Star Trek: 998300, Munchman: 187990, Blasto: 33720, Congo Bongo: 32820, Burgertime: 218997 and Car Wars: 21973. Joshua also sent me 2 cartons which I have passed onto Shane. Thanks Joshua. Can any of you Younger Set out there beat his high scores? Let me know if you do and you can have your name printed in this column.

Don't forget to come along to the big full day tutorial workshop to be held at Woodstock Community Centre on the Saturday of the forthcoming June long week-end. It will be a day full of fun and surprises and there will be a big bar-b-que at lunch time to keep the hunger pangs away. This is a day not to be missed.

That's all for this month. See you next time!

Jenny

```
10 !************************
12 !*    NUMBER GUESSER    *
14 !*by  DENNIS HODGSON    *
16 !*   TIsHUG JUNE 1986   *
18 !*      A ONE LINER     *
20 !*  EXTENDED BASIC RQD. *
22 !************************
24 !
26 !
28 !************************
   *Here is another 1 liner
   *combining learning with
   *a game.
30 !*The computer selects at
   *random a number between
   *1 and 99 and waits your
   *guess.
32 !*The computer shows if
   *your guess is higher or
   *lower and waits for the
   *next guess. The number
   *of tries is shown.
34 !*When you enter the mys-
   *tery number the TI99/4A
   *responds with "YES" and
   *starts a new game.
36 !*The objective is to
   *minimise "TRIES". Add
   *your own colours,sound
   *and voice if you like.
38 !
40 !****  INSTRUCTIONS  ****
42 !
44 !*Type the program with
   *out spaces except after
   *"FOR" ans before and
   *after "TO" down to and
   *including "::PRINT".
46 !*Press ENTER. Type 100
   *and press FCTN D until
   *cursor is after
   *"::PRINT" - continue to
   *type program until the
48 !*cursor stops (after 4
   *of the 5 commas). Press
   *ENTER. Proceed as above
   *until program entered.
   ************************
50 !
52 !**** NUMBER GUESSER ****
54 !
56 PRINT :: Q=1 :: J=99 :: R
ANDOMIZE :: A$(Z)="LOWER" ::
 A$(Q)="HIGHER" :: A$(2)="YE
S!" :: X=INT(RND*J)+Q :: FOR
 T=Q TO J :: INPUT "GUESS (1
-99) : ":G :: PRINT ,," TRIE
S=";T,A$(-(G<X)-2*(G=X)),,,,
,:: T=T-J*(G=X):: NEXT T ::
RUN
```

GETTING MORE OUT OF SPEECH - Some useful hints

Written by Software Co-ordinator - Terry Phillips

A lot of TI users haven't been associated with the group as long as me and probably have not had the opportunity to buy the module I will refer to later in these notes. The module was one of the first I bought back in 1981 long before I got an Extended Basic Module. The module is the SPEECH EDITOR a 1980 release number PHM3011. I have not seen it advertised in recent copies of Micropendium so I doubt if it is still available although you may be lucky and pick up one second hand. Mine, from memory, cost $50 when I bought it.

You are all aware that speech is available through Extended Basic provided of course you have a speech synthesizer attached. Pages 164 and 172 of the XB manual give a brief description of CALL SAY and CALL SPGET. Both pages direct you to the speech editor manual for a complete description of SAY and SPGET. Also on pages 203, 204 and 205 is the complete list of inbuilt speech synthesizer vocabulary.

To help you get more from your speech sythesizer here are some usefull tips from the speech editor manual.

The first thing that causes some problems is the use of phrases such as Texas Instruments, That is incorrect and so on. These must be programmed as follows :

```
CALL SAY("#TEXAS INSTRUMENTS#")
CALL SAY("#THAT IS INCORRECT#")
```

Another thing not mentioned in the XB manual is the use of Speech Separator Characters. You use the separator characters to give synthesized speech the cadence of natural conversation. Here is the complete list :

SPEECH SEPARATOR CHARACTERS

SYMBOL	PAUSE (Seconds)
+	0
space	.1
-	.2
,	.3
;	.5
:	.8
.	1.0

Try experimenting with these such as -

```
CALL SAY("HOW,ARE:YOU")
CALL SAY("ALL,,,TOGETHER--NOW+ONE;TWO;THREE")
```

What do speech code patterns look like? Type in and run the following short program to see.

```
10 CALL SPGET("HELLO",R$)
20 PRINT "START";R$;"STOP"
```

The words START and STOP are printed to show where the speech code begins and ends. While this short program is interesting it is not in itself very useful. Speech code patterns are made up of many characters that do not have a screen character representation. These data patterns can be translated into readable characters by using hexadecimal representation.

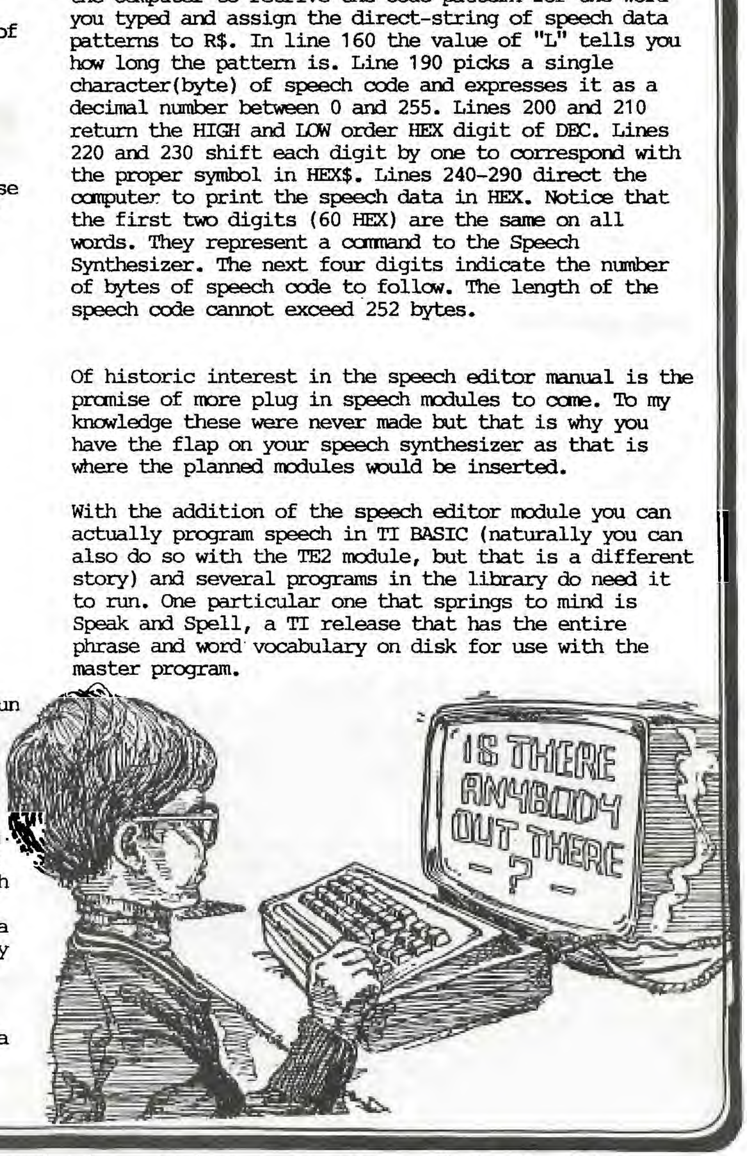
This next program converts a string of direct speech code into HEX. Enter the program and experiment with a variety of words from the resident vocabulary.

```
100 REM HEX DUMP OF SPEECH DATA
110 CALL CLEAR
120 INPUT "TYPE WORD: ":WORD$
130 CALL SAY(WORD$)
140 CALL SPGET(WORD$,R$)
150 HEX$="0123456789ABCDEF"
160 L=LEN(R$)
170 PRINT "LENGTH=";L;"BYTES"
180 FOR I=1 TO L
190 DEC=ASC(SEG$(R$,I,1))
200 HIGH=INT(DEC/16)
210 LOW=DEC-16*HIGH
220 HIGH=HIGH+1
230 LOW=LOW+1
240 PRINT SEG$(HEX$,HIGH,1);
250 PRINT SEG$(HEX$,LOW,1);
260 IF I/10<>INT(1/10)THEN 280
270 PRINT
280 NEXT I
290 PRINT
300 GOTO 120
```

Line 120 gives a prompt to type in a word selected from the resident vocabulary, then line 130 directs the computer to speak the word you typed. Line 140 tells the computer to retrive the code pattern for the word you typed and assign the direct-string of speech data patterns to R$. In line 160 the value of "L" tells you how long the pattern is. Line 190 picks a single character(byte) of speech code and expresses it as a decimal number between 0 and 255. Lines 200 and 210 return the HIGH and LOW order HEX digit of DEC. Lines 220 and 230 shift each digit by one to correspond with the proper symbol in HEX$. Lines 240-290 direct the computer to print the speech data in HEX. Notice that the first two digits (60 HEX) are the same on all words. They represent a command to the Speech Synthesizer. The next four digits indicate the number of bytes of speech code to follow. The length of the speech code cannot exceed 252 bytes.

Of historic interest in the speech editor manual is the promise of more plug in speech modules to come. To my knowledge these were never made but that is why you have the flap on your speech synthesizer as that is where the planned modules would be inserted.

With the addition of the speech editor module you can actually program speech in TI BASIC (naturally you can also do so with the TE2 module, but that is a different story) and several programs in the library do need it to run. One particular one that springs to mind is Speak and Spell, a TI release that has the entire phrase and word vocabulary on disk for use with the master program.



THE FOLLOWING IS A SPECIAL GUIDED TOUR OF THE LATEST HARDWARE AND SOFTWARE FOR THE TI-99/4(A) PREPARED BY OUR REPORTER - Steven Schraibman, WHO HAS JUST RETURNED FROM THE U.S.A AS OUR REPRESENTATIVE OF TIsHUG (AUST).

Following my article in the April TND comes this review of the Summer '86 catalogue from Tennex Computer Express and some other new releases from the states.

We start with Cor Comp peripherals. The only new release from Cor Comp is their new 256/512k memory expansion. This comes in two formats. One is a card for your PEB which replaces your 32k card. It comes in 256k ($US189) or 512k ($239) sizes. The second is a stand alone unit which plugs in between your PEB or the Cor Comp micro-expansion system. The advantage of the stand alone unit is that the 256k is recognized on top of your 32k card giving you 288k ($219) or a whopping 544k ($279) of memory. Some of you may say "So what? another 256k card, there's already several others on the market". Well not so!! This oneis different. Apart from the RAM disk feature there is that Cor Comp is developing word processor/spread sheet/data base software to recognize the 256k or 512k extra. That promises to be interesting.

Next we have Myarc. (By the way, I spoke to Jerry from Tex Comp and he told me that Myarc are next to bankrupt and that they probably won't last much longer. Who knows? Only time may tell.)

The latest news about the much heralded new computer comes from the New England 99 Fair in Boston. Myarc has decided to go two ways with the computer. One is an independent computer and the other is a "computer in a card" for the PEB. The card was displayed and operational at the faire however it short circuited after a while.

The card is configured like an Atari 520 or a full blown IBM PC with 128k VDP and 512k CPU RAM. It uses a sound chip similar to the 4A's with room to address other sound chips. It interfaces with other cards such as the RS232 and disk controller. The card features an MS mouse interface. The joysick interface is undecided, either the TI wiring or the standard Commodore/Atari one.

The computer features RGB and composite video output using the 99/38 chip. It has an 80 column text mode, 512x212 graphics mode and 512x424 interlaced mode. In the super high resolution mode each indivdual pixel can be one of 256 colours.

Under developement are BASIC III, an interface that allows you to use an IBM keyboard and software to save cartridges to disk as the computer doesn't recognize the cartridge and cassette ports.

Back to the catalogue, Tennex is already taking names for "reservations" on the Myarc Computer list, no obligations. They expect heavy demand. They project a $300 to $600 price range.

New on the scene is a German company, Mechatronic. They have Extended Basiic II+ ($79.95). This is TI XB plus new commands such as screen dump and disable quit key PLUS 40 hi-res graphics routines such as RECTANGLE, CIRCLE, ELLIPSE and TURN. Tennex offer a $20 rebate on your old XB module.

Also from Mechatronic comes a mouse interface ($119.95). The package includes the mouse, and a mouse based cataloging program, calculator and game. It is not compatible with present programs. They aslo have an 80 column card ($219.95) which is supposed to be compatible with most programs including TI Writer and Multiplan.

The next two pages are a bit of a let down as Tennex offer an Apple and an IBM compatible because "many of our customers have asked us to recommend other computer systems." There's something fishy going on!!

The rest of the magazine, some 40 pages do not offer anything new excepting a new old release from Britain. Sounds confusing? Well it is.

Long long ago Thorn EMI developed three games for the TI but when TI bailed out the games were never released. But recently a member of the UK users group got the rights to the games so now for $29.95 you may buy all three; Computer War is based on the movie War Games, Submarine Commander is a simulation of a submarine patrol mission and River Rescue where you save refugees from the river in your boat.

Tennex Computer Express, P.O. Box 6578, South Bend, IN 46660, USA. Phone: (219) 259-7051

Looking elsewhere we find Miller's Graphics. Miller's Graphics offer a variety of items. For $9.95 you may buy "The Orphan Chronicles", a book by Dr. Ron Albright that takes us from the inception of the 9900 chip right through to the 4A's demise and beyond.

The Gram Kracker allows you to save and edit modules and program in GPL. It also allows you to replace or change TI Basic and create your own "super modules" with several modules all in memory and on the master selection screen at the same time ($189.00 + $4 for s/h).

The DISkASSEMBLER ($19.95) allows you to disassemble right of the disk FIXED 80 or PROGRAM IMAGE files. It produces fully tagged object code with complete REFs, DEFs, EQUATEs and a Symbol Tables.

The EXPLORER ($24.95) emulates the 9900 processor allowing you to actually see step by step how a program, cartridge or even TI Basic work.

ADVANCED DIAGNOSTICS ($19.95) is the combination of a disk manger, disk fixer and disk tester all in one with added features such as being able to read and write tracks.

NIGHT MISSION ($19.95) is an extended basic game and tutorial in one. The five screened game won first place in a NY State competion against games for TI, Atari, Commodore and Apple. The manual takes you through the game line by line showing how it is done.

The latest release from Miller's Graphics is the IBM keyboard interface. It allows you to use an IBM keyboard as if it were the TI's. You can put the the TI aside but you still need its cartridge port. An IBM keyboard gives you a number pad, deicated cursor keys and programmable function keys.

PERFECT PARTNERS: ASSEMBLY AND EXTENDED BASIC

by ROLF SCHREIBER
TIsHUG/ILLAWARRA REGIONAL GROUP

A TI 99/4A with 32K memory expansion, disk drive and Extended Basic, can load Assembly Language programs into Low Memory (LO MEM) with the CALL LOAD("DSK1.file name") command. The XB assembly loader, which resides in LO MEM once CALL INIT has been executed, can only handle tagged object code. This means DIS/FIX 80 files which have been assembled without the 'C' option.

The area in memory where the loader places the files lies between >24F4 and >3FFF (9460 - 16383 in decimal). In the past, loading assembly files into High Memory (HI MEM) required special loaders or loading absolute origin (AORGed) code. THIS IS NOW NO LONGER NECESSARY!

What was never documented by TI in their XB manual, or anywhere else, was the fact that it is possible to use the resident XB loader to load relocatable tagged object code ANYWHERE in the 32K memory expansion (ie in both LO MEM and HI MEM). In order to load object code anywhere in HI MEM requires an adjustment to the pointers at >2002 and >2004 in LO MEM, which direct the XB loader where to start loading the files.

Linking to assembly routines loaded into HI MEM presents no problems, once a valid entry has been made in the DEF table and the loader pointers re-adjusted in LO MEM.

Enough of theory, here is a run-down of how to load a single assembly file into HI MEM and then to LINK to it from an XB program (also in HI MEM, but located at the "other end".

1. Initialize LO MEM with 'CALL INIT'. This will also allow you to execute the necessary CALL LOADs later on. Execute 'CALL PEEK(8194,G,H)'.
2. Load the XB program that you want to LINK with the assembly routine. RUN it, then BREAK the program with 'CLEAR' (FCTN 4). Enter 'CALL PEEK(-31866,C,D)'. This address points to the highest free address available to assembly language routines in HI MEM, once the XB program has reserved space for its numeric values.
3. Decide where in High Memory you want to start loading the assembly routine. Express the address in decimal as two consecutive bytes, A and B.
4. Adjust the pointers at >2002 and >2004 in LO MEM with CALL LOAD(8194,A,B,C,D), where A,B,C and D are the decimal values previously PEEKed in steps 2 and 3.
5. Load the Assembly Language file using CALL LOAD ("DSK1.XXXX"), where XXXX is the filename of the program to be loaded.
6. Execute a 'CALL PEEK(ADDR,E,F)' where ADDR is two bytes lower in memory than the highest free address determined in step 2, and 'E' and 'F' will contain the address of the routine's entry point. ADDR will be a negative number, viz ADDR=256*C+D-65538.
7. Poke the values for the program name and entry point into the DEF table. eg to enter the program name 'DEBUG', type 'CALL LOAD(16376,68,69,66,85,71,32,E,F)' <enter>, where the 68,69,66,85 and 71 represent the ASCII values (in decimal) for 'DEBUG', the '32' represents the space character to "pad" the name out to six characters and E and F, the address bytes, were PEEKed in step 6.
8. Update the pointers at >2002 and >2004 with 'CALL LOAD(8194,G,H,63,248)'. This restores the first free address in LO MEM and updates the last free address in LO MEM to >3FF8.
9. Link to your assembly language program from XB (immediate mode or from within a program) with a CALL LINK eg in our example 'CALL LINK("DEBUG").'

For people interested in finding out more, the Editor/Assembler manual pp262-5 is a good starting place. The procedure detailed above can be written as a program which in turn can easily be modified or expanded to load multiple files in both LO and HI MEM

Another fact not widely known is that the HI MEM portion of the 32K memory expansion can be made "invisible" to XB programs, so that they can be made to reside in, and RUN from, VDP RAM, even though the 32K is otherwise operational. This makes it possible to load up the full 32K with assembly language routines, make the HI MEM part "invisible" to XB, THEN load the XB program, so that it will RUN from VDP.

N.B. Since LO MEM is still recognized by the XB program, both the CALL LOAD and CALL LINK sub-programs are still operative.

If a CALL FILES(1) has been executed first, then 12869 bytes of memory would be available for the XB program in VDP, PLUS 6924 bytes in LO MEM and 24544 bytes in HI MEM for assembly language routines, ALL callable from the XB program while it is running. If you total all that up, you arive at 44,337 bytes or 43.3K of usable RAM, which is about 6K more than another well known (but vastly inferior) 64K home computer.

Perhaps the best compromise is for the XB program and the assembly routines (or at least that part of them which won't fit into LO MEM) to share HI MEM, the XB program occupying the > FFF end, and the assembly part occupying the >A000 end.

N.B. It would be a good idea to initially PRINT all PEEKed values and write them down on paper, at least until you are familiar with the method.

The card has a 32k buffer allowing for macros and batch files. (No price available yet but the keyboard will cost you $50-$100 on top of the card.) (Miller's Graphics, 1475 W. Cypress Ave., San Dimas, CA 9173, USA)

Ryte Data distribute Mechatronic in the US. The Mechatronic 80 column card (previously mentioned) has 256 colours and a resolution of 256x208. It alllows multi-coloured sprites in all 256 rows, with 192k of video RAM. It uses a version of the 99/38 without the MFX option. Also from Ryte Data besides Intern and the GPL Assembler (see elsewhere in the TND) is te GRAM-KARTE which combines th Gram Kracker and a 128k card. It is expandable to 512k and will soon have software to turn it into a RAM disk. (No prces available.) (Ryte Data, 210 Mountain St., Haliburton, Ontario, Canda KOM 1SO.)

From Texaments, TI ARTIST V2.0. This extremely friendly menu driven graphics package rivals GRAPHX. It has all the standard features plus a powerful fonts option and slides ($19.95). Also from Texaments Artist Companion #1 ($17.95), five disks of pictures, slides and fonts. Also Artist Companion #2, two disks at $9.95. Artist Extras has more pictures and fonts plus a DSR to use a Sketch Pad or a Mouse with TI Artist (2 disks, $6.95) and Display Master has several utilities to help use TI ARTIST pictures in presentaions ($14.95). (Texaments, 53 Center St.,Pachogue, New York 11772, USA.)

Not to be out done Asgard Software has its own Artist Companion, 2 disks for $7.00. Asgard also have a Graphx companion, 4 disks for $14.95. (Asgard Software, POB 10306, Rockville, MD 20850, USA.)

Well that about all for this month. I hope you have found someting of interest in this article. Remenber all prices are US $ and don't include postage.

Regards, STEVEN (TEXPAC BBS Username:SUS)

Welcome Back Steven!!!

Hello once again from the shop.

Efforts to source a 'cheap' disk have continued during the past month. We currently have a sample batch on trial. If these prove sucessful, they will be carried as a shop item. Price is expected to be around the $2 mark.

All copies of Navarone's Data Base Manager have now been sold. Sorry if you missed out. There are no immediate plans to import further packages of this item. We still have copies of Console Writer and Paint and Print. I you want one of these modules, order now to avoid disappointment.

If you are buying our premium quality disks, why not buy two boxes and take advantage of our promotional offer of a free digital watch (grey and attractively slim). This is a limited time offer.

Those of you who receive Micropendium will see in the April issue there is to be a price increase of US$2.00 on the annual subscription rate. Just how this will flow through to the club is still uncertain, but it will effect our selling price for copies from May onward. If you feel as I do that the magazine is good value, I recommend you take a subscription through the club. In this way we can be sure firstly that we order sufficient magazines and secondly we do this at the best price.

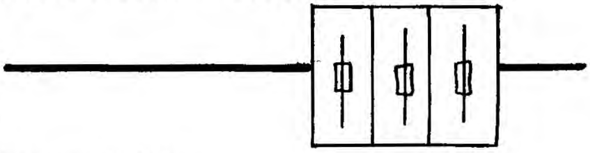
Over the past few months a number of you have requested Navarone Cartridge Expanders, and waited very patiently for them to arrive. They are currently in Sydney with the customs agent and by the time you read this they will be "on the shelf". If you still want the cartridge expander, please reconfirm your order now as we have only 12. Price including sales tax and customs charges should be less than $50.

An agent of the AMP Society contacted me about 6 weeks ago. The approach to the club initially was to offer a form of insurance cover for computers and peripherals. That initial approach has now developed into an offer to underwrite ANY form of insurance for members. Rates charged to members would be the standard company rate with a commission payable to the club. At the last committee meeting, this scheme was put into limbo pending our incorporation, however, if you are interested in taking up any of the policies offered by the AMP drop me a line. Incidentally, there is no obligation and the particular agent will not be approaching individual members until they request his services.

Elsewhere in this magazine (space permitting) you will see a review of a smart modem. This is one of two available through the club at special prices. Don't think you have to go outside the club to purchase supplies and peripherals just because they are not part of our regular stock. Check with us first. We may be able to come up with a super deal for you.

Regards, Chris Buttner

UPGRADING YOUR TI - 2 DRIVES IN THE PE BOX



By Terry Phillips

A couple of months ago I placed an ad in the SND seeking to buy 2 slimline drives for my PE Box. I had two main reasons for doing this. One being that a lot of others had done it and from what I had seen of their systems it looked very neat with 2 drives in the slot in the box. The second reason being that I had been running the 2 drives I had continuously for about 3 years and intermittent faults were developing mainly in the No. 1 drive in the box. When a lot os software is being copied for club members for a drive to suddenly pack it in would be disastrous.

Well my ad bore results. Fred Morris, our tireless Group Co-Ordinator, rang me to advise he had rounded up the 2 drives I needed and that they were mine. With a bit of sweet talking to the wife (it was around about my birthday) I became the owner of 2 slimline double-sided, doubled-density EPSON drives for the sum of $320. I should mention here that I had never heard of Epson drives. I knew they made good printers and had a vague recollection that they were marketing a PC compatable but that was about my sum knowledge of Epson.

I picked up the drives from Fred at the April meeting and took them home anxious to put them through their paces. I realised that I would need a special edge connector and power cable to instal both but at least I should be able to use one of the drives immediately. I hooked it up and tried it - it didn't work. I tried again but still no go. Very strange the light was coming on but it wouldn't read or write. Then it dawned on me. The drives have a push button mechanism which is akin to closing the door on the old drives. Put in the disk, push the button and presto! It worked. First job to initialise a disk - select double-sided, double-density and away it goes. After a few minutes the job is done and I have 1438 sectors available on my disk. What a saving this is going to be in disk storage space. Can't wait to get them both going.

Fred, as mentioned earlier, being a tireless worker, lets me know he will do the job for me, which he did on the Friday Anzac holiday. The job he did I don't recommend for those who would rather not have holes drilled in their PE Box, but this is necessary as the screw holes on the slimline drives do not align with the pre-existing holes where the old large drive fits. Anyway you can't see the holes once the top is back on the box. We only had one minor problem with instalation of both drives. When first tested they didn't work but with the addition of a 2 pin resistor pack in the first drive they both ran smoothly and very quiet. Those used to the clatter of the old drives switching on and off will really notice the difference.

Anyway, as mentioned both are now installed and operating very well. Disk storage space amounts to 720K and this is as much as the "big boys" are offering. So if you want to upgrade your TI get a couple of double sided slim line drives. Even if you don't have the Corcomp DD controller card your normal TI Card will double your disk capacity and in time save you a heap of $$ on disks. Fred, by the way, tells me that he can get more of these drives if you require them.

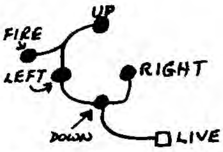
JOYSTICK CONVERTION
by Shane Andersen

Do you have an old pair of TI JOYSTICKS which no longer work, or don't work as good as you would like?

Now you can convert the 'Y' cable from those TI JOYSTS, and place Atari/Commodore style joysticks on them direct.

The ATARI P.C.Board...

FIRE	=	ORANGE
UP	=	WHITE
LEFT	=	GREEN
RIGHT	=	BROWN
DOWN	=	BLUE
LIVE	=	BLACK



TI JOYST WIRES...

FIRE	=	BLACK
UP	=	ORANGE
LEFT	=	BROWN
RIGHT	=	BLUE
DOWN	=	GREEN
LIVE	=	WHITE

You will notice that all motion on the screen will be smooth & the joysticks fun to use...

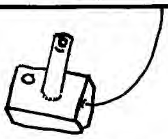
HOWEVER, when you try to move diaginal(both at the same time) all motion will slow to a stop. You then have to remove the diodes from the PCB tracks of the TI-JOYST, and place them on the same tracks of the Atary type joysicks. This can be done by drilling two small holes in each track, and cutting the track between the two holes before soldering the diodes in the right direction.

IF YOU DON'T HAVE TI JOYSTS for conversion, but want those other brand (Atari/Commadore style) to work on your TI, then follow these details...

All you need is (1)The Joysticks, (2) Ten 1n4001 diodes [available from any electronic component shops], (3) One 9 pin plug & case.

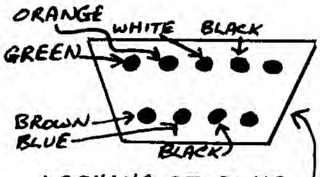
Your first step is to solder in the diodes, which can be done in one or two ways, either solder them in the p.c.board track of the Atari style joysticks. You need to drill 2 small holes in the track EXCEPT the track off the black lead where no diode is required here, cut the solder between the two holes in each track so as to make a bridge, then insert the diodes in each track, making sure to point the cathode toward the joystick, as this is most important or the joystick will not work. Then solder the diodes in, (you must do this to BOTH joysticks).

THE SECOND WAY: is to cut each wire off the joystick (except the black wire), cut the wire about 3/4" away from the

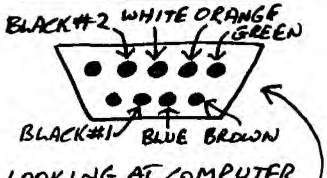


clip of the p.c.b then cut the wires of the diodes about 3/16" from each side off the body of the diodes, and solder the diodes in line, straight to the wires, remembering to point the cathode toward the clips, then put a strip of insulation tape around each diode, making sure to cover the naked wires. Screw back the joystick neatly.

Both methods have been tried and tested, successfully. The next step is to solder your plug together, firstly cut off the Atari plugs and disgard them. You must now solder the wires on both joysts together (i.e:white lead to white lead, orange lead to orange lead etc). Do this to all colours except BLACK, these must stay seperate. You can please yourself how you connect the plug together, but you must put the right colour wire to the right pinhole as shown on this diagram...



LOOKING AT PLUG



LOOKING AT COMPUTER OUTPUT

I suggest that before you solder your joined wires to the pins, that you solder one inch of wire to each join, making a 'Y' shaped union, this will make it easier for you to solder on the pins.

Before you put your plug together, remembr to put some insulation tape around all naked wires. Now put the plug together, and you're in business.

A LIGHT-PEN FOR YOUR TI-99/4A

In our club Software Library, we have a program entitled "LIGHTPEN TIC-TAC-TOE" or Naughts and Crosses with your Lightpen. It displays 9 black boxes in the middle of your screen with a white colour block zooming through them. You have to STOP that white block in its tracks, by placing the tip of your TI LIGHT PEN on a black square, where you want to place either a naught or a cross, and as soon as the white block passes in front of the pen, it stops and draws the desired symbol, ten the computer thinks and places the opposite symbol where it feels it should go.

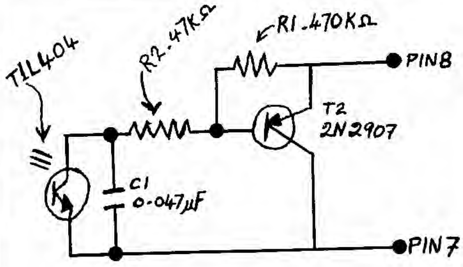
One other member of this club, has designed an assembly routine enabling you to draw on the screen with your light-pen, however, he kept it to himself, and we never saw it in action (HINT!!!!!)

The following LIGHT PEN diagram, comes direct from one of the biggest BBS'S in the U.S.A. (TEXNET-SOURCE)via Bernie Elsner and his modem in TI-UP...Please note that the *TIL404 transister has since been discontinued, however, there are alternatives which can be purchased from such places as Tandy's etc.

So, lets start with the LIGHT PEN ... first, you'll need the following parts:-

1> 9 pin "D" CONNECTOR to plug into your JOYSTICK port.
2> RG 174/U COAXIAL CABLE.
3> DA 20961 SHIELD.
4> FELT-TIP or BALL POINT PEN CASE.
5> 470 K-OHM RESISTOR.
6> 47 K-OHM RESISTOR.
7> .047uf CAPACITOR.
8> TIL404 TRANSISTOR.*
9> 2N2907 TRANSISTOR.

The photo-trasistor *(TIL404) is connected to one end of the wire and inserted into the pen case. It can be held in place by silicone glue about 5 to 10mm from the tip of the pen. The rest of the components can be connected together just before the joystick connector. It is added by using pins 8&7.



Terminal Emulator II W.P.

TE II - A WORDPROCESSOR TOO!

by Daniel Harris -TIsHUG

I am sharing my advice on HALF DUPLEX WORD PROCESSING which is available to all of you with a Terminal Emulatoe II and a Modem (The other requirement is ofcourse a RS232 interface). To get into HALF DUPLEX you select this option at the beginning of the job where it says on the Menu to choose Full or Half Duplex.

Once there, anything you type will appear on the screen. The good points are 40 characters visible and full screen editing! (Refer the TEII manual for cursor contols). eg. CTRL H enables your BACKSAPCE erase.

With the WRAP function ON you can type, almost endlessly and all the print will go into the buffer. You don't have to watch for the end of a line as you would when using a typewriter! However, you need to watch it when doing underlines - one character too many then the underline gets dragged down to the next line - complete with any nearby words!

If you want columns then CTRL J moves the cursor down to the next line. Press ENTER to start at the beginning of the new line or CTRL H to backspace to the starting position. The spacebar advances the cursor.

Begin every screen with a double row of asterisks or any other character you choose. The reason for this is because the top line is apt to disappear or get mangled somehow. This is a serious gremlin to get tricked by because there is no way to recover your work. You will have to start at the beginning - CTRL A does this as does WRAP ON/WRAP OFF. The entire buffer is cleared - ready for more mistakes?

If each screen is saved to cassette you can then edit it using a program knowing that DISPLAY 80 files are involved. A loop puts each string up on the screen - at this stage you have the chance to delete it, leave it ar change it. Whatever your choice you can then save it to CS2. You may the print the file using the "scrubbed" version from Cassette.

It is possible to use the screen to print a short pamphlet or if you go on typing until the buffer is full - checking for and correcting mistakes as you go, many copies may be printed without even saving it to cassette. Saving your work to cassette can be valuable - just in case! At least 200 lines can be stored in an array if you have 32K expansion. Using a good quality 45 minute tape (C90) you can store a lot of typing.

It has to be saved screen by screen, but once on tape and edited it will type the correct message through your printer. Remember, DISPLAY Variable 80 files! I have submitted a program that allows a screen to be edited before saving and/or printing. This program will enable any cassette saved material from the BBS to be retrieved and studied, screen by screen, with words broken up into the 28 character screen of TI BASIC. If required, also printed, spoken or copied onto CS2.

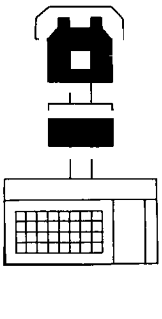
There are a lot of people out there who may not have a printer or simply do not want to waste paper. Thus by saving everything to cassettee first they can choose what to print. Moreover, they do not have to log on again to get a fresh copy. This applies to any data/text from any BBS as well as that which you write yourself using this HALF DUPLEX method. I am not saving this letter to casette bu WRAP ON/OFFing it to limbo!.

COMMENT.

DANIEL, I WISH YOU HADN'T "WRAP OFFED" IT BECAUSE ....SOMEONE ELSE..... HAD TO RETYPE IT FOR THE MAGAZINE. THUS YOU DEFEATED ONE OF THE MORE POSITIVE ASPECTS OF YOUR HINT! A GOOD IDEA THOUGH FOR MEMBERS WISHING TO SUBMIT ARTICLES TO TIsHUG NEWS DIGEST. ED.

Here is the program submitted with this article.

```
100 CALL CLEAR
110 DIM A$(26)
120 FOR N=1 TO 25
130 A$(N)=""
140 NEXT N
150 CALL SCREEN(14)
160 FOR SET=1 TO 13
170 CALL COLOR(SET,16,14)
180 NEXT SET
190 REM******************
200 REM*  D.N.HARRIS    *
210 REM* 20TH FEBRUARY  *
220 REM*     1986       *
230 REM*TI CONSOLE BASIC*
240 REM*YOU MUST ADJUST *
250 REM*THE CONTROL CODE*
260 REM*  IN THE PRINT  *
270 REM*CODE 10 IS LINE *
280 REM*FEED.           *
290 REM*CODE 29 IS CHANGE
300 REM*PEN COLOURS     *
310 REM******************
320 REM*This Programme  *
330 REM*enables Terminal*
340 REM*emulator Data   *
350 REM*to be stored on *
360 REM*Cassette and    *
370 REM*retrieved or    *
380 REM*copied in four  *
390 REM*ways.           *
400 REM******************
410 REM* CODE 11 BACKS  *
420 REM*SEE YOUR PRINT  *
430 REM*"MANUAL         *
440 REM*FOR THE CODES   *
450 REM*YOUR PRINTER    *
```



```
460 REM*OBEYS           *
470 REM******************
480 REM
490 CALL CLEAR
500 PRINT ,"PRESS 1 TO":"BAC
K PAPER_____________<_____]"
510 PRINT ,"PRESS 2 TO":"SCR
EEN PRINT___________<_____]"
520 PRINT ,"PRESS 3 TO":"HAR
DCOPY_______________<_____]"
530 PRINT ,"PRESS 4 TO":"CAS
SETTE COPY__________<_____]"
540 PRINT ,"PRESS 5 TO":"SPE
ECH OUTPUT__________<_____]"
550 PRINT ,"PRESS 6 TO":"END
 RUN________________<_____]"
560 PRINT ,"PRESS 7 TO":"REA
D TAPE OF SCREEN<_____]"
570 PRINT ,"PRESS 8 TO":"EDI
T___________________<_____]"
580 INPUT "NUMBER 1 TO 8>":R
590 IF R<1 THEN 580
600 IF R>8 THEN 580
610 ON R GOSUB 850,900,740,1
770,1840,2020,630,2040
620 GOTO 490
630 REM READ TAPE
640 OPEN #1:"CS1",INPUT ,FIX
ED 80,DISPLAY
650 R=0
660 FOR N=1 TO 24
670 N$=STR$(N)
680 Z=15-LEN(N$)
690 PRINT TAB(Z);N
700 INPUT #1:A$(N)
710 NEXT N
720 CLOSE #1
730 RETURN
740 REM   HARDCOPY**PRINTER
 CODES FOR LINEFEED CARRIAGE
 RETURN AND COLOUR  CHANGE**
****************************
750 FOR M=1 TO 24
760 OPEN #2:"RS232.LF"
770 PRINT #2:CHR$(11)
780 PRINT #2:A$(M)
790 PRINT #2:CHR$(29)
800 CLOSE #2
810 NEXT M
820 RETURN
830 INPUT R$
840 IF R$="Y" THEN 750
850 REM BACK PAPER
860 OPEN #1:"RS232.LF"
870 PRINT #1:CHR$(11)
880 CLOSE #1
890 RETURN
900 REM SCREEN PRINT
910 REM **COUNT LINES
920 FOR S=1 TO 24
930 REM INITIALIZE SUBSTRING
940 U$=""
950 V$=""
960 W$=""
970 X$=""
980 Y$=""
990 Z$=""
1000 REM HOW LONG A LINE?
1010 A=LEN(A$(S))
1020 IF A<28 THEN 1030 ELSE
1060
1030 L=28-A
1040 PRINT TAB(L);A$(S)
1050 GOTO 1730
1060 IF A<56 THEN 1070 ELSE
1290
1070 Q=INT(A/2)
1080 U=Q
1090 U=U-1
1100 IF U<1 THEN 1130
1110 TEST$=SEG$(A$(S),U,1)
1120 IF TEST$=" " THEN 1130
ELSE 1090
```

```
100 ! *******
110 ! *CUBIC*
120 ! *******
130 ! EXTENDED BASIC
140 ! AUTHOR UNKNOWN
150 ! TIsHUG JUNE 1986
160 CALL CLEAR
170 TURN=0 :: TU=28
180 BLF=5 :: BLF1=8 :: BLP=6
 :: BLP1=8 :: GR=2 :: GR1=6
:: BL=2 :: BL1=8 :: BZ=2 ::
BZ1=6
190 CALL CHAR(96,"0000000000
03070F1F0F070000000000000000
0000C0E0F0F8F0E00000000000")
200 CALL CHAR(112,"FFFFFFFFF
FFFFFFF"):: CALL COLOR(11,BL
F,BLF1):: CALL COLOR(10,BLP,
BLP1)
210 CALL CHAR(92,"01070E0701
3F2F232F2F0F1F1F080838C0F0B8
F0C0FEFAF2FAFAF8FC7C08080E")
220 CALL CHAR(108,"0103070F1
C3C3E7773793C1E07031E3CC0E0F
0F89C9EBEF7E7CF1E3CF0E03C1E"
)
230 CALL CHAR(140,"1F3066CE8
AC0E1F0FF7B3C1F1C187878F018C
CE6A2060E1EFEBC78F838181E1E"
)
240 CALL CHAR(104,"FFFFFFFFF
FFFFFFF")
250 CALL CHAR(105,"FF7F371F0
F070301")
260 CALL CHAR(120,"FF7F3F1F0
F070301")
270 CALL CHAR(106,"FFFEFCF8F
0E0C080")
280 CALL CHAR(121,"FFFEFCF8F
0E0C080")
290 CALL CHAR(122,"FFFFFFFFF
FFFFFFF")
300 CALL CHAR(123,"0103070F1
F3F7FFF")
310 CALL CHAR(124,"80C0E0F0F
8FCFEFF")
320 CALL COLOR(12,GR,GR1)
330 CALL CHAR(128,"0103070F1
F3F7FFF"):: CALL COLOR(13,BL
,BL1)
340 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
350 CALL CHAR(136,"0103070F1
F3F7FFF"):: CALL COLOR(14,BZ
,BZ1)
360 CALL CHAR(130,"80C0E0F0F
8FCFEFF")
370 CALL CHAR(113,"FF7F37170
F070301")
380 CALL CHAR(114,"FFFEFCF8F
0E0C080")
390 GOSUB 2230
400 A=18
410 B=-2
420 FOR I=5 TO 18 STEP 3
430 A=A-3
440 B=B+6
450 CALL HCHAR(I,A,104,B)
460 CALL HCHAR(I+1,A,104,B)
470 CALL HCHAR(I+2,A,104,B)
480 NEXT I
490 RESTORE 950
500 FOR I=1 TO 15
510 READ H,V,T
520 CALL HCHAR(H,V,112,T)
530 CALL HCHAR(H+1,V,112,T)
540 NEXT I
550 RESTORE 970
560 FOR I=1 TO 10
570 READ H,V
580 CALL VCHAR(H,V,112,3)::
CALL VCHAR(H,V+1,112,3)
590 NEXT I
600 FOR I=3 TO 27 STEP 6
610 CALL HCHAR(19,I,105):: C
ALL HCHAR(19,I+4,32,2):: CAL
L HCHAR(19,I+3,106)
620 NEXT I
630 CALL MAGNIFY(3)
640 RESTORE 980
650 FOR I=10 TO 23
660 READ HP,VP
670 CALL SPRITE(#I,96,7,HP*8
,VP*8+9)
680 NEXT I
690 RESTORE 990
700 FOR I=1 TO 15
710 READ HH,VV
720 CALL HCHAR(HH,VV,128)
730 CALL HCHAR(HH,VV+1,122,2
)
740 CALL HCHAR(HH,VV+3,124)
750 CALL HCHAR(HH+1,VV,120)
760 CALL HCHAR(HH+1,VV+1,122
,2)
770 CALL HCHAR(HH+1,VV+3,121
)
780 NEXT I
790 CALL HCHAR(7,18,136)
800 CALL HCHAR(10,15,136)
810 CALL HCHAR(13,12,136)
820 CALL HCHAR(16,9,136)
830 CALL HCHAR(16,15,136)
840 CALL HCHAR(13,18,136)
850 CALL HCHAR(10,21,136)
860 CALL HCHAR(16,21,136)
870 CALL HCHAR(13,24,136)
880 CALL HCHAR(16,27,136)
890 CALL HCHAR(4,18,130)
900 CALL HCHAR(7,21,130)
910 CALL HCHAR(10,24,130)
920 CALL HCHAR(13,27,130)
930 CALL HCHAR(16,30,130)
940 GOTO 1000
950 DATA 6,16,2,12,10,2,11,1
3,2,12,16,2,11,19,2,12,22,2
960 DATA 18,4,2,17,7,2,18,10
,2,17,13,2,18,16,2,17,19,2,1
8,22,2,17,25,2,18,28,2,0,0,0
970 DATA 9,13,8,16,9,19,14,1
0,14,16,14,22,15,7,15,13,15,
19,15,25,0,0,0
980 DATA 3,14,6,11,6,17,9,8,
9,14,9,20,12,5,12,11,12,17,1
2,23,15,2,15,8,15,20,15,26,0
,0,0
990 DATA 4,15,7,12,7,18,10,9
,10,15,10,21,13,6,13,12,13,1
8,13,24,16,3,16,9,16,15,16,2
1,16,27,0,0,0
1000 A1=3*8+1 :: B1=6*8+1 ::
 C1=9*8+1 :: D1=12*8+1 :: E1
=15*8+1 :: F1=18*8+1 :: G1=2
1*8+1 :: H1=24*8+1 :: I1=27*
8+1
1010 J1=14*8+1 :: K1=11*8+1
:: L1=8*8+1 :: M1=5*8+1 :: N
1=2*8+1
1020 BH=J1 :: BP=E1 :: PO=J1
 :: PV=A1 :: POL=J1 :: PVL=I
1
1030 DISPLAY AT(3,1)SIZE(11)
:"-----------"
1040 DISPLAY AT(4,2)SIZE(9):
"C U B I C"
1050 DISPLAY AT(5,1)SIZE(11)
:"-----------"
1060 CALL SPRITE(#24,140,16,
21*8+1,3*8+1)
1070 CALL SPRITE(#25,140,16,
21*8+1,9*8+1)
1080 CALL SPRITE(#26,140,16,
21*8+1,21*8+1)
1090 CALL SPRITE(#27,140,16,
21*8+1,27*8+1)
1100 CALL SPRITE(#1,140,16,B
H,BP)
1110 CALL SPRITE(#2,108,11,P
O,PV)
1120 CALL SPRITE(#3,92,10,PO
L,PVL)
1130 DISPLAY AT(3,18)SIZE(11
):"-----------"
1140 DISPLAY AT(5,18)SIZE(11
):"-----------"
1150 SC=0 :: DISPLAY AT(4,18
)SIZE(11)BEEP:"SCORE:";SC ::
 MOVE=1
1160 CALL KEY(0,TX,DF):: IF
DF=0 THEN 1160
1170 CALL KEY(0,K,S)
1180 IF K=ASC("W")THEN BH=BH
-24 :: BP=BP-24 :: GOTO 1660
1190 IF K=ASC("E")THEN BH=BH
-24 :: BP=BP+24 :: GOTO 1730
1200 IF K=ASC("S")THEN BH=BH
+24 :: BP=BP-24 :: GOTO 1800
1210 IF K=ASC("D")THEN BH=BH
+24 :: BP=BP+24 :: GOTO 1830
1220 CALL COINC(#1,#3,8,X7):
: IF X7=-1 THEN GOSUB 2090
1230 CALL LOCATE(#1,BH,BP)
1240 GOSUB 1860
1250 IF MOVE=1 THEN 1270
1260 CALL COLOR(#GOT,2):: IF
 GOT<>4 AND GOT>9 AND GOT<24
 THEN SC=SC+1 :: DISPLAY AT(
4,18)SIZE(11)BEEP:"SCORE: ";
SC :: MOVE=1
1270 CALL COINC(#1,#2,8,XXX)
:: IF XXX=-1 THEN GOSUB 2010
1280 GOSUB 1360
1290 CALL LOCATE(#2,PO,PV)
1300 CALL COLOR(#GET,7)
1310 GOSUB 1530
1320 CALL COINC(#1,#2,8,XXX)
:: IF XXX=-1 THEN GOSUB 2010
1330 CALL LOCATE(#3,POL,PVL)
1340 CALL COINC(#1,#3,8,X7):
: IF X7=-1 THEN GOSUB 2090
1350 GOTO 1170
1360 CALL CHAR(108,"0103070F
1C3C3E7773793C1E07031E3CC0E0
F0F89C9EBEF7E7CF1E3CF0E03C1E
"):: CALL PATTERN(#2,108)::
CALL COLOR(#2,4)
1370 IF PO=J1 AND PV=A1 THEN
 LET GET=21 :: PV=C1 :: RETU
RN
1380 IF PO=J1 AND PV=C1 THEN
 LET GET=28 :: PV=E1 :: RETU
RN
1390 IF PO=J1 AND PV=E1 THEN
 LET GET=22 :: PV=G1 :: RETU
RN
1400 IF PO=J1 AND PV=G1 THEN
 LET GET=23 :: PV=I1 :: RETU
RN
1410 IF PO=J1 AND PV=I1 THEN
 LET GET=19 :: PV=H1 :: PO=K
1 :: RETURN
1420 IF PO=K1 AND PV=H1 THEN
 LET GET=18 :: PV=F1 :: RETU
RN
1430 IF PO=K1 AND PV=F1 THEN
 LET GET=17 :: PV=D1 :: RETU
RN
1440 IF PO=K1 AND PV=D1 THEN
 LET GET=14 :: PV=E1 :: PO=L
1 :: RETURN
1450 IF PO=L1 AND PV=E1 THEN
 LET GET=15 :: PV=G1 :: RETU
RN
1460 IF PO=L1 AND PV=G1 THEN
 LET GET=12 :: PV=F1 :: PO=M
1 :: RETURN
1470 IF PO=M1 AND PV=F1 THEN
 LET GET=10 :: PV=E1 :: PO=N
1 :: RETURN
```

```
1480 IF PO=N1 AND PV=E1 THEN
 LET GET=11 :: PV=D1 :: PO=M
1 :: RETURN
1490 IF PO=M1 AND PV=D1 THEN
 LET GET=13 :: PV=C1 :: PO=L
1 :: RETURN
1500 IF PO=L1 AND PV=C1 THEN
 LET GET=16 :: PV=B1 :: PO=K
1 :: RETURN
1510 IF PO=K1 AND PV=B1 THEN
 LET GET=20 :: PV=A1 :: PO=J
1 :: RETURN
1520 RETURN
1530 IF POL=J1 AND PVL=I1 TH
EN POL=J1 :: PVL=G1 :: RETUR
N
1540 IF POL=J1 AND PVL=G1 TH
EN POL=J1 :: PVL=E1 :: RETUR
N
1550 IF POL=J1 AND PVL=E1 TH
EN POL=J1 :: PVL=C1 :: RETUR
N
1560 IF POL=J1 AND PVL=C1 TH
EN POL=J1 :: PVL=A1 :: RETUR
N
1570 IF POL=J1 AND PVL=A1 TH
EN POL=K1 :: PVL=B1 :: RETUR
N
1580 IF POL=K1 AND PVL=B1 TH
EN POL=L1 :: PVL=C1 :: RETUR
N
1590 IF POL=L1 AND PVL=C1 TH
EN POL=M1 :: PVL=D1 :: RETUR
N
1600 IF POL=M1 AND PVL=D1 TH
EN POL=N1 :: PVL=E1 :: RETUR
N
1610 IF POL=N1 AND PVL=E1 TH
EN POL=M1 :: PVL=F1 :: RETUR
N
1620 IF POL=M1 AND PVL=F1 TH
EN POL=L1 :: PVL=G1 :: RETUR
N
1630 IF POL=L1 AND PVL=G1 TH
EN POL=K1 :: PVL=H1 :: RETUR
N
1640 IF POL=K1 AND PVL=H1 TH
EN POL=J1 :: PVL=I1 :: RETUR
N
1650 RETURN
1660 IF BP<A1 AND BH<J1 THEN
 BH=BH+24 :: BP=BP+24 :: GOT
O 1220
1670 IF BP<B1 AND BH<K1 THEN
 BH=BH+24 :: BP=BP+24 :: GOT
O 1220
1680 IF BP<C1 AND BH<L1 THEN
 BH=BH+24 :: BP=BP+24 :: GOT
O 1220
1690 IF BP<D1 AND BH<M1 THEN
 BH=BH+24 :: BP=BP+24 :: GOT
O 1220
1700 IF BP<E1 AND BH<N1 THEN
 BH=BH+24 :: BP=BP+24 :: GOT
O 1220
1710 MOVE=0
1720 GOTO 1220
1730 IF BP>I1 AND BH<J1 THEN
 BH=BH+24 :: BP=BP-24 :: GOT
O 1220
1740 IF BP>H1 AND BH<K1 THEN
 BH=BH+24 :: BP=BP-24 :: GOT
O 1220
1750 IF BP>G1 AND BH<L1 THEN
 BH=BH+24 :: BP=BP-24 :: GOT
O 1220
1760 IF BP>E1 AND BH<N1 THEN
 BH=BH+24 :: BP=BP-24 :: GOT
O 1220
1770 IF BP>F1 AND BH<M1 THEN
 BH=BH+24 :: BP=BP-24 :: GOT
O 1220
1780 MOVE=0
1790 GOTO 1220
1800 IF BH>J1 THEN BH=BH-24
:: BP=BP+24 :: GOTO 1220
1810 MOVE=0
1820 GOTO 1220
1830 IF BH>J1 THEN BH=BH-24
:: BP=BP-24 :: GOTO 1220
1840 MOVE=0
1850 GOTO 1220
1860 IF BH=N1 AND BP=E1 THEN
 GOT=10 :: RETURN
1870 IF BH=M1 AND BP=D1 THEN
 GOT=11 :: RETURN
1880 IF BH=M1 AND BP=F1 THEN
 GOT=12 :: RETURN
1890 IF BH=L1 AND BP=C1 THEN
 GOT=13 :: RETURN
1900 IF BH=L1 AND BP=E1 THEN
 GOT=14 :: RETURN
1910 IF BH=L1 AND BP=G1 THEN
 GOT=15 :: RETURN
1920 IF BH=K1 AND BP=B1 THEN
 GOT=16 :: RETURN
1930 IF BH=K1 AND BP=D1 THEN
 GOT=17 :: RETURN
1940 IF BH=K1 AND BP=F1 THEN
 GOT=18 :: RETURN
1950 IF BH=K1 AND BP=H1 THEN
 GOT=19 :: RETURN
1960 IF BH=J1 AND BP=A1 THEN
 GOT=20 :: RETURN
1970 IF BH=J1 AND BP=C1 THEN
 GOT=21 :: RETURN
1980 IF BH=J1 AND BP=G1 THEN
 GOT=22 :: RETURN
1990 IF BH=J1 AND BP=I1 THEN
 GOT=23 :: RETURN
2000 GOT=4 :: RETURN
2010 CALL CHAR(108,"00000000
030301080C0603010000000000000
000606040081830E0C000000000
"):: CALL COLOR(#2,16):: CAL
L PATTERN(#2,108)
2020 CALL SOUND(220,200,0)
2030 CALL MOTION(#2,-20,0)
2040 FOR I=1 TO 250 :: NEXT
I
2050 CALL MOTION(#2,0,0)
2060 PO=K1 :: PV=B1
2070 SC=SC+10
2080 RETURN
2090 TURN=TURN+1
2100 IF TURN=2 THEN LET TURN
=0 :: TU=TU-1
2110 CALL DELSPRITE(#TU)
2120 CALL SOUND(-100,300,0):
: CALL SOUND(-200,400,0,800,
0,1000,0)
2130 IF TU=24 THEN 2160
2140 BH=14*8+1 :: BP=15*8+1
2150 RETURN
2160 FOR I=1 TO 200 :: NEXT
I
2170 CALL CLEAR
2180 FOR Z=10 TO 23
2190 CALL DELSPRITE(#Z):: NE
XT Z
2200 DISPLAY AT(12,1):"SCORE
:";SC
2210 FOR I=1 TO 400 :: NEXT
I
2220 END
2230 CALL CLEAR
2240 DISPLAY AT(10,3):" CUBI
C-ERASE RED DOTS (1 PT)EAT J
OKER (10 PTS) MOVE ONLYIN ST
EP...KEYS-W,E,S,D."
2250 DISPLAY AT(15,3):" JOKE
R-RESET RED DOTS..     HE IS
 WORTH(10 PTS) AND ONCEEATEN
 HE WILL START BACK     ALWAY
S AT HIS STARTING POINT"
2260 DISPLAY AT(20,3):" DEVI
L-WATCH FOR HIM,HE GOESROUND
 AND ROUND AND HE IS    OUT T
O GET YOU..GOOD SCORING"
2270 CALL HCHAR(10,2,140)::
CALL HCHAR(11,2,141):: CALL
HCHAR(10,3,142):: CALL HCHAR
(11,3,143):: CALL COLOR(14,1
6,8)
2280 CALL HCHAR(15,2,108)::
CALL HCHAR(16,2,109):: CALL
HCHAR(15,3,110):: CALL HCHAR
(16,3,111)
2290 CALL HCHAR(20,2,92):: C
ALL HCHAR(21,2,93):: CALL HC
HAR(20,3,94):: CALL HCHAR(21
,3,95)
2300 DISPLAY AT(24,1):"*****
******K E Y************"
2310 CALL KEY(0,F,M):: IF M=
0 THEN 2310
2320 CALL CLEAR :: CALL COLO
R(14,2,6)
2330 RETURN
```

●

```
100 REM ***********
110 REM *ON SAFARI*
120 REM ***********
130 REM TI BASIC
140 REM FOOLED ABOUT WITH
150 REM BY RUSSEL WELHAM
160 REM TIsHUG JUNE 1986
170 CALL CLEAR
180 CALL SCREEN(5)
190 FOR I=1 TO 14
200 CALL COLOR(I,16,1)
210 NEXT I
220 PRINT TAB(7);"**********
***":TAB(7);"* ON SAFARI *":
TAB(7);"*************": : :
: :
230 PRINT "YOU ARE AT YOUR B
ASE CAMP."
240 PRINT :"YOU ARE JUST LEA
VING TO FIND      THE TREASU
RE."
250 PRINT :"POSSIBLE EXITS:-
                SOUTH,EAST,WE
ST.": :
260 PRINT "WHAT SHALL I DO N
OW?"
270 INPUT "":WSN$
280 IF WSN$="GO SOUTH" THEN
420
290 IF WSN$="GO EAST" THEN 3
20
300 IF WSN$="GO WEST" THEN 5
20
310 GOTO 1420
320 PRINT :"    YOU ARE NOW
AT A              RIVER CROSS
ING."
330 PRINT :"ACROSS THE RIVER
 YOU SEE A         TREASURE C
HEST."
340 PRINT :"POSSIBLE EXITS:-
                CROSS RIVER,
 WEST."
350 PRINT :"WHAT SHALL I DO
NOW?"
360 INPUT "":WSN$
370 IF WSN$="GO WEST" THEN 2
30
380 IF WSN$="CROSS RIVER" TH
EN 400
390 GOTO 1440
400 PRINT :"AS YOU CROSS THE
 RIVER YOU  ARE ATTACKED BY
PIRANHAS    AND KILLED."
410 GOTO 910
```

⇨

```
420 PRINT :"YOU ARE NOW IN A
 BEAUTIFUL  CLEARING. YOU SE
E:-":"   A GUN,KEY,AMMO.": :
"POSSIBLE EXITS:-":"   EAST,
 NORTH, SOUTH."
430 PRINT :"WHAT SHALL I DO
NOW?"
440 INPUT "":WSN$
450 IF WSN$="TAKE GUN" THEN
930
460 IF WSN$="TAKE AMMO" THEN
 960
470 IF WSN$="TAKE KEY" THEN
990
480 IF WSN$="GO NORTH" THEN
230
490 IF WSN$="GO SOUTH" THEN
590
500 IF WSN$="GO EAST" THEN 5
70
510 GOTO 1460
520 PRINT :"YOU ARE NOW INSI
DE A":"    DAMP CAVE":"YOU S
EE:-A BEAR AND CUB": :"POSSI
BLE EXITS:-":"    SOUTH, EAS
T.": :"WHAT SHALL I DO NOW?"
530 INPUT "":WSN$
540 IF WSN$="GO EAST" THEN 2
30
550 IF WSN$="GO SOUTH" THEN
640
560 GOTO 1480
570 PRINT :"AS YOU WALK ALON
G YOU SINK  IN SOME QUICK SA
ND AND DIE."
580 GOTO 910
590 PRINT :"YOU ARE IN A EMP
TY CLEARING.YOU SEE:-A FEW R
OCKS.": :"POSSIBLE EXITS:-NO
RTH, WEST.": :"WHAT SHALL I
DO NOW?"
600 INPUT "":WSN$
610 IF WSN$="GO NORTH" THEN
420
620 IF WSN$="GO WEST" THEN 7
10
630 GOTO 1500
640 PRINT :"YOU ARE NOW AT Y
OUR           SUPPLY DEPOT.
": :"YOU SEE:-YOUR SUPPLIES.
": :"POSSIBLE EXITS:-NORTH,
SOUTH"
650 PRINT :"WHAT SHALL I DO
NOW?"
660 INPUT "":WSN$
670 IF WSN$="TAKE SUPPLIES"
THEN 1020
680 IF WSN$="GO NORTH" THEN
520
690 IF WSN$="GO SOUTH" THEN
710
700 GOTO 1520
710 PRINT :"YOU ARE NOW IN A
 SMALL      CLEARING. HERE Y
OU SET UP   CAMP TILL MORNIN
G.": :"POSSIBLE EXITS:-":"NO
RTH, SOUTH, EAST, WEST."
720 PRINT :"WHAT SHALL I DO
NOW?"
730 INPUT "":WSN$
740 IF WSN$="GO NORTH" THEN
640
750 IF WSN$="GO SOUTH" THEN
790
760 IF WSN$="GO EAST" THEN 5
90
770 IF WSN$="GO WEST" THEN 8
50
780 GOTO 1540
790 PRINT :"YOU ARE ON THE B
ANKS OF A   RIVER UP STREAM
YOU SEE A   WATERFALL.": :"P
OSSIBLE EXITS:-NORTH, EAST."
800 PRINT :"WHAT SHALL I DO
NOW?"
810 INPUT "":WSN$
820 IF WSN$="GO NORTH" THEN
710
830 IF WSN$="GO EAST" THEN 1
100
840 GOTO 1560
850 PRINT :"YOU ARE IN A LAR
GE CLEARING.": :"POSSIBLE EX
ITS:-EAST.": :"SUDDENLY YOU
ARE ATTACKED          BY LEOPA
RDS."
860 PRINT :"WHAT SHALL I DO
NOW?"
870 INPUT "":WSN$
880 IF WSN$="SHOOT LEOPARDS"
 THEN 1050
890 IF WSN$="GO EAST" THEN 7
10
900 GOTO 1580
910 PRINT : :"HA, HA, YOU LO
SE!               NOW TRY AGAI
N.": : :
920 GOTO 230
930 GU=1
940 PRINT :"YOU NOW HAVE THE
 GUN."
950 GOTO 440
960 AM=1
970 PRINT :"YOU NOW HAVE THE
 AMMO."
980 GOTO 440
990 KY=1
1000 PRINT :"YOU NOW HAVE TH
E KEY."
1010 GOTO 440
1020 SUP=1
1030 PRINT :"YOU NOW HAVE SU
PPLIES."
1040 GOTO 660
1050 IF GU=1 THEN 1080
1060 PRINT :"YOU HAVE NO GUN
 LEOPARDS            EAT YOU."
1070 GOTO 910
1080 PRINT :"LEOPARDS ARE DE
AD."
1090 GOTO 870
1100 PRINT :"YOU ARE AT THE
BOTTOM OF A  WATERFALL. BEHI
ND THE       WATERFALL IS A
CAVE.": :"POSSIBLE EXITS:-EA
ST, CAVE."
1110 PRINT :"WHAT SHALL I DO
 NOW?"
1120 INPUT "":WSN$
1130 IF WSN$="GO EAST" THEN
1160
1140 IF WSN$="ENTER CAVE" TH
EN 1270
1150 GOTO 1600
1160 PRINT :"YOU ARE ON THE
BANKS OF A          RIVER.":
 :"POSSIBLE EXITS:-NORTH, EA
ST.": :"SUDDENLY YOU ARE ATT
ACKED BY         PANTHERS."
1170 PRINT :"WHAT SHALL I DO
 NOW?"
1180 INPUT "":WSN$
1190 IF WSN$="GO NORTH" THEN
 570
1200 IF WSN$="GO EAST" THEN
570
1210 IF WSN$="SHOOT PANTHERS
" THEN 1230
1220 GOTO 1620
1230 IF AM=1 THEN 1260
1240 PRINT :"YOU HAVE NO AMM
O YOU ARE            EATEN."
1250 GOTO 910
1260 GOTO 1170
1270 PRINT :"YOU ARE IN A CA
VE YOU SEE:-       A LARGE C
HEST.": :"POSSIBLE EXITS:-WE
ST.": :"WHAT SHALL I DO NOW?
"
1280 INPUT "":WSN$
1290 IF WSN$="GO WEST" THEN
1330
1300 IF WSN$="OPEN CHEST" TH
EN 1350
1310 IF WSN$="TAKE CHEST" TH
EN 1390
1320 GOTO 1640
1330 PRINT :"AS YOU LEAVE TH
E CAVE YOU   SLIP AND BREAK
YOUR NECK."
1340 GOTO 910
1350 PRINT :"AS YOU OPEN THE
CHEST        EVERTHING SPINS
, AND YOU AREBACK AT YOUR CA
MP. WITH THE         TREASUR
E."
1360 PRINT :"YOU HAVE FINISH
ED, WELL DONE!!!!!"
1370 PRINT :"NOW TRY AGAIN"
1380 GOTO 230
1390 PRINT :"AS YOU TAKE THE
 CHEST A     GENIE APPEARS,
AND"
1400 PRINT "CAST A SPELL ON
YOU."
1410 GOTO 910
1420 PRINT :"CAN'T DO THAT"
1430 GOTO 260
1440 PRINT :"CAN'T DO THAT"
1450 GOTO 360
1460 PRINT :"CAN'T DO THAT"
1470 GOTO 440
1480 PRINT :"CAN'T DO THAT"
1490 GOTO 530
1500 PRINT :"CAN'T DO THAT"
1510 GOTO 600
1520 PRINT :"CAN'T DO THAT"
1530 GOTO 660
1540 PRINT :"CAN'T DO THAT"
1550 GOTO 730
1560 PRINT :"CAN'T DO THAT"
1570 GOTO 810
1580 PRINT :"CAN'T DO THAT"
1590 GOTO 870
1600 PRINT :"CAN'T DO THAT"
1610 GOTO 1120
1620 PRINT :"CAN'T DO THAT"
1630 GOTO 1180
1640 PRINT :"CAN'T DO THAT"
1650 GOTO 1280
1660 END
```

```
100 REM   **********
110 REM   *TIMARQUE*
120 REM   **********
130 REM   BY ANDREW ZAGNI
140 REM   TI BASIC
150 REM   TIsHUG JUNE 1986
160 GOTO 1470
170 P=96
180 Q=97
190 R=98
200 S=99
210 RETURN
220 P=96
230 Q=100
240 R=101
250 S=102
260 RETURN
270 P=103
280 Q=100
290 R=104
300 S=105
310 RETURN
320 P=103
330 Q=100
340 R=106
360 RETURN
370 P=96
380 Q=100
390 R=108
400 S=105
410 RETURN
420 P=96
430 Q=97
440 R=108
450 S=32
460 RETURN
470 P=103
480 Q=100
490 R=109
500 S=110
510 RETURN
520 P=111
530 Q=97
540 R=112
550 S=99
560 RETURN
570 P=113
580 Q=114
590 R=104
600 S=105
610 RETURN
620 P=32
630 Q=100
640 R=115
650 S=116
660 RETURN
670 P=111
680 Q=97
690 R=117
700 S=118
710 RETURN
720 P=119
730 Q=100
740 R=32
750 S=110
760 RETURN
770 P=120
780 Q=97
```

```
790 R=121
800 S=99
810 RETURN
820 P=120
830 Q=97
840 R=122
850 S=123
860 RETURN
870 P=103
880 Q=100
890 R=124
900 S=110
910 RETURN
920 P=96
930 Q=97
940 R=125
950 S=32
960 RETURN
970 P=103
980 Q=100
990 R=124
1000 S=126
1010 RETURN
1020 P=96
1030 Q=97
1040 R=125
1050 S=127
1060 RETURN
1070 P=128
1080 Q=129
1090 R=130
1100 S=110
1110 RETURN
1120 P=113
1130 Q=131
1140 R=104
1150 S=32
1160 RETURN
1170 P=119
1180 Q=100
1190 R=132
1200 S=110
1210 RETURN
1220 P=133
1230 Q=134
1240 R=135
1250 S=136
1260 RETURN
1270 P=137
1280 Q=138
1290 R=139
1300 S=140
1310 RETURN
1320 P=141
1330 Q=142
1340 R=117
1350 S=118
1360 RETURN
1370 P=143
1380 Q=131
1390 R=144
1400 S=32
1410 RETURN
1420 P=145
1430 Q=146
1440 R=147
1450 S=105
1460 RETURN
1470 DIM A$(53)
1480 DIM Z$(12)
1490 CALL CLEAR
1500 INPUT "SCREEN COLOUR (2
-16) :":SC
1510 PRINT ::
1520 INPUT "CHARACTER COLOUR
    :":CC
1530 IF CC=SC THEN 1490
1540 CALL SCREEN(SC)
1550 FOR AZ=1 TO 16
1560 CALL COLOR(AZ,CC,1)
1570 NEXT AZ
1580 CALL CLEAR
1590 PRINT "INITIALIZING PLE
ASE WAIT -AZ"
1600 PRINT ::::::::::::::
1610 A$(1)="FFFFC0C0C0FFFFC0
"
1620 A$(2)="C0C0C0C0C0C0"
1630 A$(3)="FCFC0C0C0CFCFC0C
"
1640 A$(4)="0C0C0C0C0C0C"
1650 A$(5)="C0C0C0C0FFFF"
1660 A$(6)="E0F0381838F8F0FC
"
1670 A$(7)="0C0C0C1CF8F0"
1680 A$(8)="FFFFC0C0C0C0C0C0
"
1690 A$(9)="FCFC"
1700 A$(10)="00000000FCFC"
1710 A$(11)="F0F81C0C0C0C0C0
C"
1720 A$(12)="0C0C0C1CF8F0"
1730 A$(13)="FCFC000000C0C0"
1740 A$(14)="FCFC000000FCFC0
C"
1750 A$(15)="0C0C0C0CFCFC"
1760 A$(16)="C0C0C0C0C0FFFFC
0"
1770 A$(17)="0C0C0C0C0CFCFC0
C"
1780 A$(18)="FFFF03030303030
3"
1790 A$(19)="03030303FFFF"
1800 A$(20)="C0C0C0C0C0C0C0C
0"
1810 A$(21)="C0C0C0C0C0C0"
1820 A$(22)="0C1C387060C0C06
0"
1830 A$(23)="603030181C0C"
1840 A$(24)="C0C0C0C0C0C0C0C
0"
1850 A$(25)="F0F0F8FCDECFC7C
3"
1860 A$(26)="3C3C7CFCECCC8C0
C"
1870 A$(27)="0C0C0C0C0C0C8CC
C"
1880 A$(28)="ECFC7C3C1C0C"
1890 A$(29)="FCFC0C0C0C0C0C0
C"
1900 A$(30)="F8FC0C0C0CFCF8"
1910 A$(31)="CCEC7C3CFCFC"
1920 A$(32)="C0E070381C0C"
1930 A$(33)="FFFFC0C0C0FFFF"
1940 A$(34)="00000000FFFF"
1950 A$(35)="FCFC000000FCFC"
1960 A$(36)="030303030303"
1970 A$(37)="0C0C0C0C0C0C0C0
C"
1980 A$(38)="C0C060603030181
8"
1990 A$(39)="0C0C07070303"
2000 A$(40)="0C0C18183030606
0"
2010 A$(41)="C0C08080"
2020 A$(42)="C0C0C0C0C0C0C3C
7"
2030 A$(43)="CFDFFDF8F0F0"
2040 A$(44)="0C0C0C0C0C0C0C8
C"
2050 A$(45)="CCECFC7C3C3C"
2060 A$(46)="C0C06030180F0F1
8"
2070 A$(47)="38306060C0C0"
2080 A$(48)="C0C06030180C070
3"
2090 A$(49)="0C0C183060C080"
2100 A$(50)="FFFF00000001070
F"
2110 A$(51)="3C78E0C0FFFF"
2120 A$(52)="FCFC0C3CF0E0800
0"
2130 FOR AZ=1 TO 52
2140 CALL CHAR(95+AZ,A$(AZ))
2150 NEXT AZ
2160 CALL HCHAR(8,1,32,32)
2170 Z$(1)=".     AZTEK"
2180 Z$(2)=".SIMULATIONS AND
"
2190 Z$(3)="  ALSO PROGRAMS"
2200 Z$(4)=""
2210 Z$(5)=" TI MICRO MARQUE
"
2220 Z$(6)=""
2230 Z$(7)=" BY ANDREW ZAGNI
"
2240 Z$(8)="     TASMANIA
"
2250 Z$(9)="    FOR TISHUG
"
2260 Z$(10)="    AUSTRALIA
 "
2270 Z$(11)=""
2280 Z$(12)=" CONTINUE Y OR
N"
2290 GOSUB 2820
2300 CALL KEY(0,K,S)
2310 IF S=0 THEN 2300
2320 IF K=ASC("N")THEN 1470
2330 Z$(1)=""
2340 Z$(2)=""
2350 Z$(3)=""
2360 Z$(4)=""
2370 Z$(5)=""
2380 Z$(6)=""
2390 Z$(7)=" INPUT ANY WORDS
"
2400 Z$(8)="  BUT MAKE SURE"
2410 Z$(9)="  THAT YOU ONLY"
2420 Z$(10)=" USE LETTERS AN
D"
2430 Z$(11)=" THE MESSAGE ET
C"
2440 Z$(12)=" WILL BE PRINTE
D"
2450 CALL HCHAR(13,1,32,384)
2460 GOSUB 2820
2470 CALL KEY(0,K,S)
2480 IF S=0 THEN 2470
2490 FOR V=1 TO 5
2500 Z$(V)=""
2510 NEXT V
2520 Z$(6)=" AT THE START OF
"
2530 Z$(7)=" EACH INPUT LINE
"
2540 Z$(8)=" PUT A FULL STOP
"
2550 Z$(9)=" IT ISNT PRINTED
"
2560 Z$(10)="  BUT IS NEEDED
"
2570 Z$(11)=" FOR PLACEMENTS
"
2580 Z$(12)=" OK BUD LETS GO
"
2590 CALL HCHAR(13,1,32,384)
2600 GOSUB 2820
2610 CALL KEY(0,K,S)
2620 IF S=0 THEN 2610
2630 CALL CLEAR
2640 INPUT "SCREEN COLOUR :"
:SC
2650 INPUT "CHARACTER COL :"
:CC
2660 IF SC=CC THEN 2640
2670 CALL CLEAR
2680 CALL SCREEN(SC)
2690 FOR W=1 TO 16
2700 CALL COLOR(W,CC,1)
2710 NEXT W
2720 PRINT "REMEMBER A DOT A
T THE START OF EACH LINE AND
 A MAXIMUM LENGTH OF 15 LETT
ERS A LINE"
2730 FOR V=1 TO 12
2740 INPUT STR$(V)&" :":Z$(V
)
2750 IF LEN(Z$(V))>16 THEN 2
740
2760 NEXT V
2770 CALL CLEAR
2780 GOSUB 2820
2790 CALL KEY(0,K,S)
2800 IF S=0 THEN 2790
2810 GOTO 2630
2820 C=2
2830 Y=1
2840 FOR AZ=1 TO 12
2850 CALL SOUND(100,550,5)
2860 FOR X=2 TO LEN(Z$(AZ))
2870 CHAR=ASC(SEG$(Z$(AZ),X,
1))
2880 IF CHAR=32 THEN 2950
2890 CHAR=CHAR-64
2900 ON CHAR GOSUB 170,220,2
70,320,370,420,470,520,570,6
20,670,720,770,820,870,920,9
70,1020,1070,1120,1170,1220,
1270,1320,1370,1420
2910 CALL HCHAR(Y,C,P)
2920 CALL HCHAR(Y+1,C,Q)
2930 CALL HCHAR(Y,C+1,R)
2940 CALL HCHAR(Y+1,C+1,S)
2950 C=C+2
2960 NEXT X
2970 Y=Y+2
2980 C=2
2990 NEXT AZ
3000 RETURN
```

```
1130 FOR P=1 TO U
1140 U$=U$&SEG$(A$(S),P,1)
1150 NEXT P
1160 V=U+28
1170 V=V-1
1180 IF V<1 THEN 1740
1190 TEST$=SEG$(A$(S),V,1)
1200 IF TEST$=" " THEN 1210
ELSE 1170
1210 FOR P=U TO V
1220 V$=V$&SEG$(A$(S),P,1)
1230 NEXT P
1240 FOR P=V TO A
1250 W$=W$&SEG$(A$(S),P,1)
1260 NEXT P
1270 PRINT U$:V$:W$
1280 GOTO 1730
1290 Q=INT(A/3)
1300 U=Q+28
1310 U=U-1
1320 IF U<1 THEN 1350
1330 TEST$=SEG$(A$(S),U,1)
1340 IF TEST$=" " THEN 1350
ELSE 1310
1350 FOR P=1 TO U
1360 U$=U$&SEG$(A$(S),U,1)
1370 NEXT P
1380 V=U+28
1390 V=V-1
1400 IF V<1 THEN 1430
1410 TEST$=SEG$(A$(S),V,1)
1420 IF TEST$=" " THEN 1430
ELSE 1390
1430 FOR P=U TO V
1440 U$=U$&SEG$(A$(S),P,1)
1450 NEXT P
1460 W=V+28
1470 W=W-1
1480 IF W>1 THEN 1510
1490 TEST$=SEG$(A$(S),W,1
1500 IF TEST$=" " THEN 1510
ELSE 1470
1510 FOR P=V TO W
1520 W$=W$&SEG$(A$(S),P,1)
1530 NEXT P
1540 IF A-W<28 THEN 1550 ELS
E 1600
1550 FOR P=W TO A
1560 X$=X$&SEG$(A$(S),P,1)
1570 NEXT P
1580 PRINT U$:V$:W$:X$
1590 GOTO 1730
1600 Q=INT((A-W)/2)
1610 X=W+Q
1620 X=X-1
1630 IF X<1 THEN 1650
1640 TEST$=SEG$(A$(S),X,1)
1650 IF TEST$=" " THEN 1660
ELSE 1620
1660 FOR P=W TO X
1670 X$=X$&SEG$(A$(S),P,1)
1680 NEXT P
1690 FOR P=X TO A
1700 Y$=Y$&SEG$(A$(S),P,1)
1710 NEXT P
1720 PRINT U$:V$:W$:X$:Y$
1730 NEXT S
1740 REM BEAUTY!
1750 R=0
1760 RETURN
1770 REM CASSETTE COPY
1780 OPEN #5:"CS2",OUTPUT,FI
XED 192,INTERNAL
1790 FOR C=1 TO 25
1800 PRINT #5:A$(C)
1810 NEXT C
1820 CLOSE #5
1830 RETURN
1840 REM SPEECH OUTPUT
1850 PRINT "IS TERMINAL EMUL
ATOR IN?"
1860 INPUT "Y/N":Q$
1870 OPEN #99:"SPEECH",OUTPU
T
1880 FOR T=1 TO 25
1890 A=LEN(A$(T))
1900 FOR B=1 TO A
1910 Z$=SEG$(A$(T),B,1)
1920 P=ASC(Z$)
1930 IF P<97 THEN 1960 ELSE
1940
1940 IF P>122 THEN 1960 ELSE
 1950
1950 P=P-32
1960 X$=X$&CHR$(P)
1970 NEXT B
1980 PRINT #99:X$
1990 X$=""
2000 NEXT T
2010 RETURN
2020 END
2030 REM M N S C T COUNTERS
2040 FOR ED=1 TO 26
2050 PRINT A$(ED)
2060 PRINT TAB(28);
2070 PRINT "PRESS ENTER IF O
.K."
2080 PRINT "ENTER REVISED VE
RSION IF    YOU HAVE ONE TO
ENTER, THEN PRESS ENTER."
2090 PRINT TAB(28);
2100 INPUT ED$
2110 IF ED$="" THEN 2140 ELS
E 2120
2120 A$(ED)=ED$
2130 REM EDITING LOOP*****
2140 NEXT ED
2150 RETURN
```

REGIONAL REPORT: Reports from our Regional Home-group leaders...

"See you there 'cause we care"

The TIsHUG Illawarra Regional Group

This group holds regular monthly meetings at Saint Matthews Church Hall, Philip Crescent, Mangerton, on the third Monday of each month (except January). We also hold occasional hardware and other special interest group meetings at irregular intervals. We are offering memory expansion and other simple hardware expansions upon request, and are working on software for systems without disks but with memory expansion.

The meetings normally start with a tutorial session on Extended BASIC, followed by a talk and demonstration of some other topic of interest. This leads to some refreshments while members meet each other and chat about problems. We maintain various libraries for the use of members.

The TIsHUG CARLINFORD Regional Group

The April meeting was held on 16 April. It was a free for all night with informal discussion and copying of programmes. Once again, shop items were available for purchase including the Micropendium magazine.

Ray Brown has now upgraded his system to two double-sided drives. This was very much a DIY effort. Any members in the area wishing to go down the same road can get the necessary information at Regional Group meetings.

The June meeting is to be held on:-

Wednesday 18 June : 141 Beecroft Road
Beecroft (Ray Brown).

If you plan to attend, please call Ray (86-4602) if possible. His home is on a battle-axe block on Beecroft Road (left hand side travelling to Epping just past the tennis courts and almost opposite the Nursing Home). Hope to see you there.

So you want to attend one of our TIsHUG Regional Home Groups and need more information...

GLEBE REGIONAL GROUP meeting with Mike Slattery (02)6920559 for more information.
PENRITH REGIONAL GROUP with Mel Copeland (047)351340
CENTRAL COAST REGIONAL GROUP with Russel Welham (043)924000
LIVERPOOL REGIONAL GROUP with Stan Puckle (046)256157
BANKSTOWN REGIONAL GROUP with David Mayo (02)7084293
ILLAWARRA REGIONAL GROUP with Rob Montgomery (042)286463
CARLINGFORD REGIONAL GROUP with Chris Buttner (02)8717753

MEMBERS LIVING IN SUTHERLAND SHIRE should contact PETER YOUNG on (02)5288775 with the view to forming a Regional Group in that area.
MEMBERS LIVING ON THE NORTH SHORE should contact KEITH DE HAAN on (02)4513032 with the view to forming a Regional Group in that area.

If you want to run a TIsHUG Regional Home Group and want to know what is expected of you...send in the special section on the latest version of the CUT-UP-PAGE and send it to P.O.Box 149, Pennant Hills, NSW 2120

SAVING PARTIAL PROGRAMMES

by Rolf Schreiber
Illawarra regional group - TISHUG

Sometimes a programme contains sections of code or useful subroutines that you might like to use in your own programmes. Unfortunately, saving only part of a programme is not as straight forward as entering the commands 'SAVE DSK1.filename:L1-L2'......or 'SAVE CS1:L1-L2' where L1 and L2 are first and last line numbers, respectively, of the section you want to SAVE. These commands are illegal in both BASICs, although a similar option is allowed with the 'LIST' command.

For the cassette-based programmer there is, unfortunately, no simple alternative to typing in that section of code, or else DELeteing all the unwanted lines. Either way is SLOW, unless many of the lines are very similar (eg CALL SOUND statements), in which case the REDO key (FCTN 8, XB only) will save a lot of typing in.

For disk-based users there are alternatives that are much less time consuming (on the part of the programmer), since the computer does most of the work. So, if you are disk- based, READ ON...

METHOD #1

This involves typing in the utility listing on p49 of MICROPENDIUM, DEC 1985, v2n11 and SAVing to disk in MERGE (DIS/VAR 163) format. Full instructions are included in the article. To use this program, do the following:

1) MERGE the utility with the main program already in memory.
2) RUN the resulting program and follow on-screen prompts.
3) When the cursor re-appears, SAVE the new file under a DIFFERENT filename using the MERGE option.
4) To check on the procedure, clear the memory with 'NEW', followed by 'MERGE DSK1.name', where 'name' is the filename under which the subroutine was saved.

It is useful to leave these subroutines (or partial programs) in MERGE format, since they can then be easily incorporated in your own programs.

METHOD #2

As luck would have it, the listing which needs to be typed in for this method is to be found in the same issue of MICROPENDIUM, on p51. Full instructions are shown on p50. Save the utility to disk using the filename 'XLATE', then use the following procedure:

1) With the main program in memory, determine the first and last line numbers of the subroutine to be saved.
2) In the immediate mode type in 'LIST "DSK1.name":L1-L2', where 'name' is a filename different from the original program and L1,L2 are the starting and ending line numbers of the subroutine in question. The file produced by this procedure is in DIS/VAR 80 format and could be READ with either TI WRITER or the E/A editor.
3) Clear the memory and load 'XLATE'. RUN the program and follow the on-screen prompts. Choose a filename for the output file which is different from the previous two. This latest file is in DIS/VAR 163 or MERGE format.
4) Clear the memory and MERGE in this latest file to check on the procedure.

The subroutine is best left in MERGE format, allowing easy incorporation in your own programs.

METHOD #3

This method involves using the 'EDITOR' program, which is on the Prograids3 disk (part of TI's PROGRAMMING AIDS III package). I won't detail the method since full documentation is supplied with the disk, but I will tell you that it is a more laborious procedure than either of the first two methods.

I have personally tried all the three ways and, in my opinion, Method #1 is the preferred one, in that it is the quickest way. The 'XLATE' utility involves an extra step; it is mainly used to convert BBS listings of BASIC programs back into program or 'memory image' format. The Prograids3 'EDITOR' utility is the most involved of the three methods; it is also capable of other functions, such as RESequencing line numbers in only a part of the program, and the ability of moving portions of code within a program.

'In my day, Mr Hobson, we used to throw chalk at them!'

★OOPS

OOPS - MINI MEMORY MODULE MANUAL.

By James Jackson

MMM Manual page 37 incorrectly lists the GPL status bits as:

H	GT	COND	CARRY	OVF	0	0	0
7	6	5	4	3	2	1	0

This is correctly listed in the EA Manual sec 16.2.1 page 250

H	GT	COND	CARRY	OVF	0	0	0
0	1	2	3	4	5	6	7

This inverted numbering of Hex>837C is repeated on page 38 under GPLINK Under the heading of linking to Grom-Resident Routines-XMLINK There is a routine noted as DATA >2300 Convert Integer to Floating Point(CIF) on page 50 of the MMM manual. This routine will only work with the Editor Assembler and Memory Expansion not on the MMM alone. Instead DATA >7200 should be used with the MMM, as this is the correct routine.

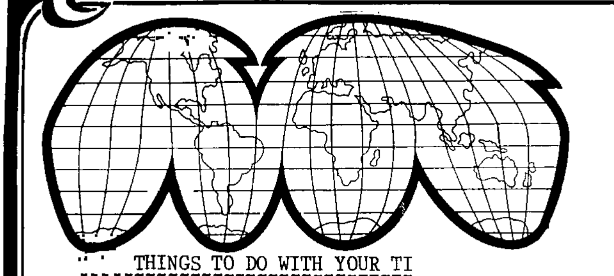
MONITORS vs TVs

I have noticed articles that mostly conclude that a TV "isn't up to video monitor quality." Well that "isn't necessarily so" it is more often that the TV is more complicated to adjust for monitor use.

You can set a TV for monitor use with the following XBasic commands: "FOR I=0 TO 14 :: CALL COLOR(I,16,5) :: NEXT I :: CALL SCREEN(6) :: ACCEPT AT(1,1):I" ; press <ENTER>. [if you're using C-Basic you will need to write a program, starting with:"100 FOR I=1 TO 16" ending with:"140 GOTO 140" instead of "ACCEPT AT(1,1):I", then type <RUN>, <ENTER>.

Now adjust the Fine-Tuning and the focus, the contrast and the brightness to get the sharpest view of the text on the screen. Next adjust the Colour and Tint to get the bluest colour for the lighter blue bands at the screen top and bottom. Finally press <CLEAR> and <QUIT> to get the main title screen, and give the colour a last adjustment if it is needed.

Retyped for TIsHUG by Kevin Watts from Sacremento Users Group Newsletter (NETWORK) via Aloha 99ers.



...THINGS TO DO WITH YOUR TI

Texpac BBS-May'86 & TND-JUNE'86
by Shane Andersen
##################################

On both this system last month , and in the current issue of the Sydney News Digest, I presented the first of my on-going articles on things to do with your computer. Two others...Laurie Marsh (VK2DWH) & Ross Mudie (VK2ZRQ - Username:MUDDY.

In this article I want to briefly look at other peripherals that many of you have never heard of before , which will connect to your TI-99/4A to assist you with International Information/Communication retrieval.

How about these for peripherals...

* PK-80 UNIVERSAL PACKET CONTROLLER.

* KANTRONICS CHALLENGER TERMINAL UNIT.

* THETA-777.

* UNIVERSAL TERMINAL UNIT.

* INTERFACE II

* Plus HAMSOFT for your TI-99/4A

I can just imagine you asking yourself ... What the heck is all this? I thought that the only peripheral for the TI were the 32k card,RS232,MODEM,DISK DRIVE & CONTROLLER,etc.

Ah! For those into Amateur Radio (HAMS) in this club , you will have some idea of what I am talking about.

Let's take a look at each one of the items listed above and see what they do, and how many of them are inter-related in one way or another with special software like HAMSOFT or TE#2...

The PK-80 is a glorified Modem (unlike the others I will mention) and priced at $472 aprox...its hardware features are:

* Z80A ONBOARD CPU
* 16K PROGRAM EPROM EXPANDABLE TO 32K
* 8K RAM
* HARDWARE HDLC FOR FULL DUPLEX (ESPECIALLY NICE FOR SATELLITE OPERATION).
* TRUE DATA CARRIER DETECT (DCD) DISTINGUISHES BETWEEN ACTUAL PACKET DATA AND INTERFERRING QRN.
* WORKS WITH 300 , 1200 , 2400 , 4800 , AND 9600 BAUD TERMINALS.

KANTRONICS CHALLENGER TERMINAL UNIT:
A quality product at a fair price is what Amateur operator wants. The Challanger includes many of the popular features you pay extra for on other units. Individual LEDS show both Mark and Space tones,with scope outputs available. Crystal controlled AFSK tones guarentee consistent RTTY (RADIO TELETYPE)/ASCII/AMTOR tones , and direst FSK keying is also available. Challenge transmits at the standard 170 Hz shift , and all shifts of RTTY can be received. Challenger can also be used to receive CW (MORSE CODE), but the unit does not have internal CW keying circuits for transmission. The Challenger is compatible with the TI-99/4A with the use of the #[illegible] cartridge.
The Challenger Terminal Unit is available for $245.00

THE THETA-777:
The revolutionary new Thete-777 allows AMTOR , RTTY and CW operation with a homecomputer. This unit has complete software communications which saves you (I believe) the trouble of having to load the software into your computer. Some of its features include...TWENTY-EIGHT LEDS with Bar Graph tuning , CW identification , Automatic Idle Signal Insertion , Echo function , Automatic CR/LF , CW RANDOM GENERATOR , VARIABLE CW WEIGHTS , TEST MESSAGE FUNCTION , CW PRACTICE FUNCTION , MARK-AND-BREAK SYSTEM , CRYSTAL CONTROLLED AFSK MODULATOR , etc. Confused? Well , read on and I will endevour to explain the uses for these peripherals with your TI. But before that...here is another unit called...

KANTRONICS "INTERFACE II":
Like others mentioned in this article , this unit sells for a mear $479.00 and features... DUAL TONE DETECTION , SELECTABLE SHIFT OPERATION for RTTY & CW , AM & FM OPERATION , QUARTZ SYNTHESIZED TONES , TWO CHANNEL OPERATION , RS232 Compatible. and last but not least...

THE UNIVERSAL TERMINAL UNIT:
A new concept in COMPUTER TO TRANSCEIVER(2 way radio)INTERFACING and priced at $399.00. With this unit , a standard TERMINAL EMULATOR program can be used, like our own TE#2 , FAST-TERM or 4A TALK... this is a real God-sent,as the #HAMSOFT TI CARTRIDGE sells for a very high $215.00 in the States.

Still confused? O.K. then let me remind you of that article I wrote last month on SHORT WAVE LISTENING and QSL'ing with overseas Radion Stations that you can pick up on a standerd Short Wave Radio. In between the transmissions of stations such as RADIO MOSCOW , RADIO AMERICA and all of the other countries , there are tones being transmitted the world over. And on those tones or carriers,are CW (MORSE CODE) , RTTY (RADIOTELETYPE or World News being telexed which you can convert to understandable and readable text on your computer screen and which can also be sent to your printer for later reading. Or...if you are an AMATEUR RADIO OPERATOR , you can not only RECEIVE these signals but also REPLY to them via your computer keyboard. One of our members in TISHUG has made his own mini peripheral and software to do just what I have spoken about, and it is hoped that we will hear more about his device either on TEXPAC BBS next month and/or printed in our TISHUG NEWS DIGEST (TND) this or next months issue. You have got to see it for yourself to really be thrilled with what can be done with your TI-99/4A and show you what you have been missing out on by not even knowing that these things even existed.

If you are interested in World events, and are sick of the either condenced or sensationalism of our Australian Media...then you will want to hear the truth and/or ALL THE FACTS as they are being transmitted from its source or received by our media. Items which our News media may think is too trivia to report , may be news which you need to hear and which you can receive on your TI-99/4A computer screen.

By the way,without sounding like an advert...each of the peripherals I have mentioned earlier in this article are available from EMTRONICS at 94 Wentworth Avenue Sydney 2000 Ph(02)2110988. You'll find JOHN GEORGE very helpful. And for the HAMS in this club , his CALL SIGN on air is VK2PCC. They also sell a huge rage of both SHORT WAVE RADIO's and 2 WAY RADIO gear.

Now , lets hear from you with what you are using your TI-99/4A for...drop me a line C/- the club address or on the TEXPAC BBS to Username:SHANE
Bye 4 now
Yours in Computing the TI WAY
SHANE ANDERSEN

```
100 REM ****************
110 REM *  FINANCIAL   *
120 REM *    MATH.     *
130 REM *C.M. EHNINGER*
140 REM ****************
150 OPTION BASE 1
160 DIM C$(8)
170 CALL CLEAR
180 CALL SCREEN(2)
190 DISPLAY TAB(10);"FINANCI
AL":TAB(12);"MATH."::::::::
::
200 FOR X=96 TO 152 STEP 8
210 Y=Y+1
220 C$(Y)=STR$(X)
230 NEXT X
240 FOR X=1 TO 8
250 C=VAL(C$(X))
260 CALL HCHAR(X+2,X+2,C,29-
2*X)
270 CALL VCHAR(X+2,X+2,C,22-
2*X)
280 CALL VCHAR(X+2,30-X,C,22
-2*X)
290 CALL HCHAR(23-X,X+2,C,29
-2*X)
300 NEXT X
310 FOR X=9 TO 16
320 CALL COLOR(X,1,1)
330 NEXT X
340 CALL SCREEN(8)
350 FOR X=1 TO 100
360 NEXT X
370 FOR Y=1 TO 2
380 FOR X=16 TO 9 STEP -1
390 CALL COLOR(X,X,X)
400 NEXT X
410 FOR X=9 TO 16
420 CALL COLOR(X,X-7,X-7)
430 NEXT X
440 NEXT Y
450 FOR X=16 TO 9 STEP -1
460 FOR Y=2 TO 9
470 CALL COLOR(X,Y,Y)
480 NEXT Y
490 NEXT X
500 FOR X=9 TO 16
510 FOR Y=2 TO 9
520 CALL COLOR(X,Y,Y)
530 NEXT Y
540 NEXT X
550 FOR X=1 TO 100
560 NEXT X
570 F$="00"
580 CALL CLEAR
590 CALL SCREEN(2)
600 DISPLAY TAB(6);"FINANCIA
L MATH":::TAB(9);"INTEREST":
:"SINGLE-VALUE PROBLEMS:"
610 DISPLAY "  1..COMPOUND V
ALUE":"  2..VALUE OF PRINCIP
AL":"  3..NUMBER OF PERIODS"
620 DISPLAY "  4..INTEREST R
ATE"::"ANNUITIES:":"  5..SUM
 OF ORDINARY ANNUITY"
630 DISPLAY "  6..VALUE OF A
NNUITY":"  7..VALUE OF PAYME
NT "
640 DISPLAY "  8..PRESENT VA
LUE OF SUM":"  9..NUMBER OF
PERIODS":" 10..INTEREST RATE
"::
650 DISPLAY " 99..TERMINATE"
:::"FUNCTION";
660 CALL SCREEN(8)
670 INPUT Q
680 IF Q=99 THEN 3770
690 IF (Q<1)+(Q>10)<>0 THEN
670
700 CALL CLEAR
710 ON Q GOTO 2000,2120,2240
,2370,2750,2920,3090,3200,25
80,3450
720 REM ************
730 REM *  SCREEN  *
740 REM ************
750 E=0
760 FOR X=YY TO YY+LEN(E$)-1
770 E=E+1
780 ED=ASC(SEG$(E$,E,1))
790 CALL HCHAR(XX,X,ED)
800 NEXT X
810 RETURN
820 REM  -----------
830 REM  SHOW VALUES
840 REM  -----------
850 YY=3
860 E$="PERIODIC AMOUNT:"
870 GOSUB 750
880 E$=STR$(R)
890 GOSUB 1390
900 GOTO 1340
910 YY=3
920 E$="FUTURE VALUE:"
930 GOSUB 750
940 E$=STR$(V)
950 GOSUB 1390
960 GOTO 1340
970 YY=3
980 E$="INTEREST RATE:"
990 GOSUB 750
1000 I=100*I
1010 E$=STR$(I)
1020 GOSUB 1390
1030 GOTO 1340
1040 YY=3
1050 E$="NO. OF PERIODS:"
1060 GOSUB 750
1070 E$=STR$(N)
1080 GOSUB 1390
1090 GOTO 1340
1100 YY=3
1110 E$="PRESENT VALUE:"
1120 GOSUB 750
1130 E$=STR$(V)
1140 GOSUB 1390
1150 GOTO 1340
1160 YY=3
1170 E$="SUM AMOUNT:"
1180 GOSUB 750
1190 E$=STR$(V)
1200 GOSUB 1390
1210 GOTO 1340
1220 YY=3
1230 E$="PRINCIPAL AMOUNT:"
1240 GOSUB 750
1250 E$=STR$(P)
1260 GOSUB 1390
1270 GOTO 1340
1280 XX=24
1290 YY=6
1300 E$="RETURN TO CONTINUE"
1310 GOSUB 750
1320 CALL KEY(0,B,C)
1330 IF C=0 THEN 1320
1340 XX=XX+2
1350 RETURN
1360 REM  ----------
1370 REM  NUM TO STR
1380 REM  ----------
1390 E$="0000000000"&E$
1400 F$=SEG$(E$,LEN(E$)-9,10
)
1410 E$=F$
1420 E=POS(E$,".",1)
1430 IF E>0 THEN 1460
1440 E$=E$&"."
1450 GOTO 1390
1460 G=ABS(8-E)
1470 IF G<3 THEN 1500
1480 E$=SEG$(E$,2,9)
1490 GOTO 1390
1500 E$=SEG$(E$,G+1,LEN(E$)-
G)&SEG$(F$,1,G)
1510 E$=SEG$(E$,1,1)&","&SEG
$(E$,2,3)&","&SEG$(E$,5,6)
1520 FOR X=1 TO 12
1530 IF SEG$(E$,1,1)="0" THE
N 1550
1540 IF SEG$(E$,1.1)<>"," TH
EN 1570
1550 E$=SEG$(E$,2,LEN(E$)-1)
1560 NEXT X
1570 YY=33-LEN(E$)
1580 GOSUB 750
1590 RETURN
1600 REM ***********
1610 REM *  INPUT  *
1620 REM ***********
1630 REM  ---
1640 REM  SUM
1650 REM  ---
1660 INPUT "ENTER SUM: ":V
1670 IF V<=0 THEN 1660
1680 RETURN
1690 REM  ---------
1700 REM  PRINCIPAL
1710 REM  ---------
1720 INPUT "ENTER PRINCIPAL:
 ":P
1730 IF P<=0 THEN 1720
1740 RETURN
1750 REM  --------
1760 REM  INTEREST
1770 REM  --------
1780 INPUT "ENTER INTEREST:
":I
1790 IF I<=0 THEN 1780
1800 I=I/100
1810 RETURN
1820 REM  -------
1830 REM  PERIODS
1840 REM  -------
1850 INPUT "ENTER NUMBER OF
PERIODS: ":N
1860 IF N<=0 THEN 1850
1870 RETURN
1880 REM  ----
1890 REM  RENT
1900 REM  ----
1910 INPUT "ENTER PERIODIC A
MOUNT: ":R
1920 IF R<=0 THEN 1910
1930 RETURN
1940 REM ****************
1950 REM *  PROCEDURES  *
1960 REM ****************
1970 REM  --------------
1980 REM  COMPOUND VALUE
1990 REM  --------------
2000 GOSUB 1720
2010 GOSUB 1780
2020 GOSUB 1850
2030 V=P*((1+I)^N)
2040 V1=INT(100*(V+.005))
2050 V=V1/100
2060 E$="COMPOUND VALUE"
2070 YY=10
2080 GOTO 2450
2090 REM  ---------
2100 REM  PRINCIPAL
2110 REM  ---------
2120 GOSUB 1660
2130 GOSUB 1780
2140 GOSUB 1850
2150 P=V/((1+I)^N)
2160 P1=INT(100*(P+.005))
2170 P=P1/100
2180 YY=8
2190 E$="PRINCIPAL AMOUNT"
2200 GOTO 2450
2210 REM  -------
2220 REM  PERIODS
2230 REM  -------
2240 CALL CLEAR
2250 GOSUB 1660
2260 GOSUB 1720
2270 GOSUB 1780
2280 N=LOG(V/P)/LOG(1+I)
2290 M=INT(100*(N+.005))
2300 N=M/100
2310 YY=8
```

```
2320 E$="NUMBER OF PERIODS"
2330 GOTO 2450
2340 REM --------
2350 REM  INTEREST
2360 REM --------
2370 GOSUB 1660
2380 GOSUB 1720
2390 GOSUB 1850
2400 I=((V/P)^(1/N))-1
2410 IN=INT(1E4*(I+5E-5))
2420 I=IN/1E4
2430 E$="INTEREST"
2440 YY=13
2450 XX=1
2460 CALL CLEAR
2470 GOSUB 750
2480 XX=5
2490 GOSUB 1160
2500 GOSUB 1220
2510 GOSUB 970
2520 GOSUB 1040
2530 GOSUB 1280
2540 GOTO 580
2550 REM -------
2560 REM  PERIODS
2570 REM -------
2580 GOSUB 1660
2590 GOSUB 1910
2600 GOSUB 1780
2610 N=LOG((I*V/R)+1)/LOG(1+
I)
2620 M=INT(100*(N+.005))
2630 N=M/100
2640 CALL CLEAR
2650 YY=8
2660 E$="NUMBER OF PERIODS"
2670 XX=1
2680 GOSUB 750
2690 XX=5
2700 GOSUB 1160
2710 GOTO 3370
2720 REM -------
2730 REM  ANNUITY
2740 REM -------
2750 GOSUB 1910
2760 GOSUB 1780
2770 GOSUB 1850
2780 V=R*(((1+I)^N)-1)/I
2790 V1=INT(100*(V+.005))
2800 V=V1/100
2810 CALL CLEAR
2820 XX=1
2830 YY=4
2840 E$="COMPOUND VALUE OF A
NNUITY"
2850 GOSUB 750
2860 XX=5
2870 GOSUB 910
2880 GOTO 3370
2890 REM --------
2900 REM  PAYMENTS
2910 REM --------
2920 GOSUB 1660
2930 GOSUB 1780
2940 GOSUB 1850
2950 R=V*I*((1+I)^N)/(((1+I)
^N)-1)
2960 R1=INT(100*(R+.005))
2970 R=R1/100
2980 E$="PAYMENTS TO RETIRE"
2990 CALL CLEAR
3000 XX=1
3010 YY=6
3020 GOSUB 750
3030 XX=5
3040 GOSUB 1160
3050 GOTO 3370
3060 REM ----------
3070 REM  RETIRE SUM
3080 REM ----------
3090 GOSUB 1660
3100 GOSUB 1780
3110 GOSUB 1850
3120 R=V*I/(((1+I)^N)-1)
3130 R1=INT(100*(R+.005))
3140 R=R1/100
3150 E$="PAYMENTS TO ACCUMUL
ATE"
3160 GOTO 2990
3170 REM -------
3180 REM  ANNUITY
3190 REM -------
3200 GOSUB 1910
3210 GOSUB 1780
3220 GOSUB 1850
3230 V=R*(1-(1+I)^(-1*N))/I
3240 V1=INT(100*(V+.005))
3250 V=V1/100
3260 E$="ANNUITY-PRESENT VAL
UE"
3270 CALL CLEAR
3280 XX=1
3290 YY=6
3300 GOSUB 750
3310 XX=5
3320 GOSUB 1100
3330 GOTO 3370
3340 REM ------
3350 REM  ANSWER
3360 REM ------
3370 GOSUB 850
3380 GOSUB 970
3390 GOSUB 1040
3400 GOSUB 1280
3410 GOTO 580
3420 REM --------
3430 REM  INTEREST
3440 REM --------
3450 GOSUB 1660
3460 GOSUB 1910
3470 GOSUB 1850
3480 T=4
3490 B=0
3500 TT=R*N
3510 IF TT>=V THEN 3710
3520 TST=ABS(T-B)
3530 IF TST<1E-5 THEN 3620
3540 IN=(T+B)/2
3550 IT=((1+IN)^N)-(V*IN/R)-
1
3560 IF IT=0 THEN 3620
3570 IF IT>=0 THEN 3600
3580 B=IN
3590 GOTO 3520
3600 T=IN
3610 GOTO 3520
3620 I=INT(1E4*(IN+5E-5))/1E
4
3630 CALL CLEAR
3640 XX=1
3650 E$="INTEREST RATE"
3660 YY=10
3670 GOSUB 750
3680 XX=5
3690 GOSUB 1160
3700 GOTO 3370
3710 CALL CLEAR
3720 XX=23
3730 YY=3
3740 E$="INTEREST RATE 0 OR
NEGATIVE"
3750 GOSUB 750
3760 GOTO 3400
3770 END
```

Diagnosing hardware malfunctions

by Geoff Trott
Illawarra Regional Group - TISHUG

It is becoming more apparent that when troubles strike our computer, it can be quite difficult to find out what the problem is so that a repair can be attempted. With TI not providing a very useful service in this regard, we are going to have to do something ourselves. If the computer is running, it is easy to trouble shoot a peripheral which is not working. It only requires a program which tests all its functions and reports any errors. Several of these are available for RS232, P-code card, Expansion memory, and so on. There is even a module which runs tests on the VDP memory, keyboard, sound and arithmetic functions. However none of these will help if the computer will not get past the title screen, or not even get to a title screen. For these cases all would appear to be lost, but is it?

The 9900 processor has two non-maskable interrupts. The first of these is the RESET, which is the one used to get into the system monitor, using two vectors at addresses 0 and 2. This is the one which happens at power up and when a cartridge is plugged in. If the console is not working then this is no use to us. The second one is the LOAD interrupt, which is not used in the system, but does come out the IO port at the side of the console. This uses two vectors at FFFC and FFFE which are at the top of memory expansion. This interrupt could be used to force a program stored at the top of memory to start up, and this program could be a diagnostic program for the console!

The hardware would be quite simple, consisting of an EPROM containing the program and the vectors and occupying the last 8K of the expansion memory address space, a RAM chip in the next to last 8K of memory (or less if available), a push button and circuitry for the LOAD signal, address decoding for the EPROM and RAM, and some simple way to enable some LEDs to be turned on. It could all fit into a small box which would plug into the IO port and use the 5 volt supply from the console. If a console was in trouble, any internal memory expansion would need to be removed before this box was attached to ensure no address conflicts.

The operation would be as follows. The console with the box attached would be powered up and the LOAD button pressed. This would start the diagnostic program, and if the processor was working one of the LEDs would turn on. Then the system RAM would be checked and if OK another LED would turn on. Then the VDP RAM would be checked to see if data could be stored and retrieved and a third LED would then turn on. If all is OK so far, the programme would set up the VDP with a screen for the diagnostic program, with perhaps a menu. It would need to check the system ROM against a check sum and then the GROMS to see if they have the correct data and are behaving properly. Once the screen is working the results of these tests can be displayed on the screen to give more information about the problems as found. The keyboard, sound, joystick and cassette functions could also be checked.

This would provide a simple, portable and relatively cheap way of checking out consoles, and providing information to determine exactly what is not working or program loops to enable troubleshooting with CROs etc. It could be easily upgraded as more experience is gained. I will be working on such a device in my limited free time, but the more who contribute their expertise to such a project, the quicker and better the result should be.

SPECIAL NOTE:
GEOFF, HAS LEFT YOU ALL WITH A THOUGHT STARTER FOR A PROJECT, NOW.... LET'S SEE IF OUR GROUP IS ABLE TO "WIN THROUGH". OVER TO YOU -- ED.

LINKING TO ASSEMBLY from EXTENDED BASIC.
by Ross Mudie of TISHUG.
Part 4 in a series of articles. June 1986 issue TND.

Unfortunately the program for ALPHA LOCK detection was omitted from last month's column so this month it is in the middle of the article.

One question which I received recently was "How can assembly be used with extended basic, since you can't plug in extended basic and the minimem at the same time?" .... The assembly programs which are to work with an extended basic program are prepared using the TI Editor Assembler (e/a) or BEAXS. The assembly source file is saved to disk and assembled in the normal manner for an assembly program, except that the compressed format cannot be used. The assembly program is loaded from disk into the extension memory after extended basic is loaded. It is also possible to include the assembly program information in the extended basic program as a series of CALL LOADs, programs have appeared in the pages of the SND from time to time in this format. The 32K extension memory is required to use assembly with extended basic.

I will be presenting tutorial material on linking to assembly from extended basic at the Full Day Tutorial Workshop on 7th June 1986 at Woodstock. I will provide printed tutorial information for people in my group. Don't forget to study all the references to linking in the e/a manual before the day, including the following pages in the e/a manual: 273 to 290 & 410 to 419. Full details of the Tutorial day appear elsewhere in this issue of the TND. Remember to bring a note pad, pencil, e/a manual and your lunch to the Tutorial Day.

If you have been trying to use XMLLNK with extended basic, the DATA equates are to be found on page 416 of the e/a manual. Disregard the note on the bottom of page 261 of the e/a manual, the XMLLNK CIF routine works fine with the data >20.

The GPLLNK utility presented in this month's article responds to the DATA values stated in pages 251 to 257 of the e/a manual, but don't forget if passing values from extended basic to use the equivalent decimal values from x/b.

Both these programs are stand alone routines. These programs and a number of others will be available on disk MUDIE_86/3 from the club shop in July. The cost of the disk will be the club's usual media fee of $5.

```
* ALPHA LOCK DETECTION for extended basic by
* Ross Mudie of TISHUG, 27th March 1986.

* Extended basic format:
* CALL LINK("ALPHA"[,Line #])
*    This module tests if the ALPHA LOCK key, (TI99/4A
* only), is operated. If the alpha lock is operated it
* provides a message:  "** RELEASE ALPHA LOCK KEY **"
* and waits until the alpha lock key is released.
*    If the optional line number is a number from 1 to
* 24 inclusive the message will be displayed on that
* line if necessary. If the line number is outside the
* valid range or is not specified the default value
* will be 24.
*    The module saves the screen information present
* where the message will be displayed and restores the
* information when the alpha lock key is released.
*    This module is designed as a stand alone program
* for the either the 32 or 40 column mode.  Change the
* LI value in the line with the lable MODE to 32 or 40
* as required for the environment of your x/b program.

*    The following x/b program will  allow  convenient
* testing of the assembly module in graphics mode.

* 100 ! SAVE DSK1.LOAD
* 110 CALL INIT
* 120 CALL HCHAR(1,1,46,768)
* 130 N=2
* 140 CALL LOAD("DSK1.AL")
* 150 CALL LINK("ALPHA",N)
* 160 DISPLAY AT(8,1):"PRESS ALPHA LOCK IN AGAIN"
* 170 DISPLAY AT(10,1):"ENTER NEXT LINE NUMBER  ";N
* 180 ACCEPT AT(10,26)VALIDATE(DIGIT)SIZE(-2):N
* 190 FOR L=8 TO 10 STEP 2
* 200 DISPLAY AT(L,1):"............................"
* 210 NEXT L
* 220 GOTO 150

        DEF  ALPHA    source ALOCK object AL

VMBW    EQU  >2024
NUMREF  EQU  >200C
XMLLNK  EQU  >2018
KSCAN   EQU  >201C
VMBR    EQU  >202C
FAC     EQU  >834A
KEYVAL  EQU  >8375
STATUS  EQU  >837C
GPLWS   EQU  >83E0

BUFFER  BSS  28
MYWS    BSS  32

D24     DATA 24
BH40    BYTE >40
FF      BYTE >FF
        EVEN

* Screen message MSG is in hex with hex 60 offset added
*            * *     R  E L   E A   S E    A  L P
MSG     DATA >8A8A,>80B2,>A5AC,>A5A1,>B3A5,>80A1,>ACB0
*            H A    L   O C   K    K E   Y     * *
        DATA >A8A1,>80AC,>AFA3,>AB80,>ABA5,>B980,>8A8A
*                                              28 bytes
ALPHA   LWPI MYWS

        CLR  R0        Use for CB then NUMREF element zero
        CB   @>8312,R0 Any arguments?
        JEQ  DEFALT      No
        LI   R1,1        First argument
        BLWP @NUMREF     Get numeric first argument
        CB   @FAC,@BH40 Is arg value in range 1 to 99?
        JNE  DEFALT      No
        BLWP @XMLLNK    {Convert floating point number in
        DATA >12B8      {FAC to integer in FAC
        C    @FAC,@D24   Is line number greater than 24?
        JGT  DEFALT       Yes
        MOV  @FAC,R1      Place line number in R1
        JMP  DECLNE      Avoid default value
DEFALT  MOV  @D24,R1     Places default line # in R1
DECLNE  DEC  R1          Dec R1 for line number in assy
MODE    LI   R0,32    32 for GRAPHICS, 40 for TEXT MODE
        MPY  R0,R1       Multiply line number by number of
*                        columns in line for MODE
        MOV  R2,R0      Multiplication result into R0
        INCT R0         Start at col 3 in designated line
*                  R0 contains where to read from for VMBR
        LI   R1,BUFFER Where to save line from the screen
        LI   R2,28       Number of bytes to be saved
        BLWP @VMBR       Save the line into the buffer

* The TEST routine responds to the release of the alpha
* lock & keys 1 to 5. The routines at NOTON & WAIT trap
* a number 1 to 5 being pressed, sending the program
* back to TEST when a number key is released.

TEST    CLR  R12         CRU Base address of zero
        SBZ  21          See editor assm manual page 409
        NOP
        TB   7           Test for alpha lock
        JEQ  NOTON       Jump if ALPHA LOCK is off
```

```
       LI   R1."         Address of message
       BLWP @VMBW        Write message
       JMP  TEST         Go and test for alpha lock again

NOTON  BLWP @KSCAN       Check no other keys operated
       CB   @KEYVAL,@FF  Any key operated?
       JEQ  SCREEN   If NO then go on to restore screen

WAIT   BLWP @KSCAN         Wait for operated key release
       CB   @KEYVAL,@FF
       JNE  WAIT
       JMP  TEST           Go to test alpha lock release
*            RO & R2 values are used unchanged by VMBW
SCREEN LI   R1,BUFFER To clear message & rewrite
       BLWP @VMBW        previous info on screen line

       LWPI GPLWS        Restore x/b work space
       CLR  RO             To prevent false indication of
       MOVB RO,@STATUS   errors on return to x/b
       RT                Return to x/b

       END
```

```
*          Source GPLLINK  object GPL
* GPLLNK routine for extended basic

UTLWS  EQU  >2038          Utility workspace
SUBST  EQU  >8373          SUBroutine Stack Pointer
GRMRA  EQU  >9802          GRoM Read Address
GPLWS  EQU  >83E0          GPL Work Space

GPLLNK DATA UTLWS       BLWP @ GPLLNK, vectors:Workspace
       DATA GPLLN1                           Entry Point

GPLLN1 MOVB @GRMRA,RO     {Save Grom read address in RO
       SWPB RO            {
       MOVB @GRMRA,RO     {
       SWPB RO            {
       AI   RO,-3          Back up to the XML instruction

       MOVB @SUBST,R1     {Get the stack pointer
       SRL  R1,8          {
       AI   R1,>8300      {
       INCT R1            {Push XML address for return
       MOV  RO,*R1        {
       SWPB R1            {
       MOVB R1,@SUBST     {

       LI   R3,>2000    Load R3 with address of XML link
       MOV  *R3,R2         Save current XML link address

       LI   RO,GPLLN2     {Load new XML address at >2000
       MOV  RO,*R3        {
       MOV  *R14+,@>83EC Place GPLLNK data val at >83CE
*                          & advance saved PC past DATA

       LWPI GPLWS          Load GPL/xb workspace
       B    @>0060         Go to routine in console GROM
*                          chip 0

GPLLN2 LWPI UTLWS         Should return here, reload
*                          utility workspace
       MOV  R2,*R3 Reload original XML address in >2000
       RTWP        Return to caller after GPLLNK & DATA

*Extended basic format:
* CALL LINK("EXBGPL",#) Accesses the GPLLNK utility
*from extended basic. Decimal values for # which do not
*require other values to be loaded or retrieved.
*   32  PWRUP  Branches to master screen in power up
*              routine
*   52  TON1   BEEP sound.
*   54  TON2   HONK sound.
* CALL LINK("BIGCHR") Loads the MASTER TITLE SCREEN
*characters in place of the normal upper case chars
* CALL LINK("UPPCHR") or CALL CHARSET to restore
*normal upper case characters. ("UPPCHR" does not
*restore default colours as CHARSET does).
* CALL LINK("LOWCHR") Loads the normal lower case
*(small capitals) characters.
```

```
       DEF  EXBGPL,BIGCHR,UPPCHR,LOWCHR
FAC    EQU  >834A
NUMREF EQU  >200C
BIGCHR LI   R1,1024   Start in VDP RAM at ascii char 32
       MOV  R1,@FAC      Place start location in FAC
       LI   R1,>16    Data required for GPLLNK, Hex 16
       MOV  R1,@DATA     Place data in word following
*                        BLWP @GPLLNK
       JMP  CHRENT    Jump to CHaRacter ENTry

UPPCHR LI   R1,1024   Start in VDP RAM at ascii char 32
       MOV  R1,@FAC      Place start location in FAC
       LI   R1,>18    Data required for GPLLNK, Hex 18
       MOV  R1,@DATA     Place data in word following
*                        BLWP @GPLLNK
       JMP  CHRENT    Jump to CHaRacter ENTry

LOWCHR LI   R1,1536  Start in VDP RAM at ascii char 96
       MOV  R1,@FAC      Place start location in FAC
       LI   R1,>4A    Data required for GPLLNK, Hex 4A
       MOV  R1,@DATA     Place data in word following
*                        BLWP @GPLLNK
       JMP  CHRENT    Jump to CHaRacter ENTry

EXBGPL CLR  RO           Element 0 in array from link
       LI   R1,1         First argument
       BLWP @NUMREF      Get value from LINK in x/b
       CLR  @DATA       To ensure left byte contains 00
       MOVB @FAC+1,@DATA+1 Gets the value from FAC &
*                          places in word after GPLLNK

CHRENT BLWP @GPLLNK        Executes GPLLNK
DATA   BSS  2            Data previously loaded here
       RT                Returns to calling program

       END
```

TI staff given a security warning

Texas Instruments has warned all its staff to be more conscious of personal security in the future.

The warning has been issued from the company's head office in Dallas, Texas, following the discovery of a missile bearing the company name in Libya after the recent US raids.

Texas Instruments employ around 1,000 people in Australia and according to John Mitchell, national accounts manager for TI, staff are a lot more conscious of security and safety following the memo but there is no paranoia in the company.

He said:"We have initiated some additional security precautions in the Melbourne office following the warning but I can't disclose what they are."

He said no additional security was being provided for senior executives of the company in Australia and he was unaware of any additional security for American executives.

The Texas Instruments defence systems electronics group is the largest section of the company and manufactures the guidance systems used in many missiles and modern day aircraft.

MicroPENDIUM - April 1986

What's in it?

Myarc's Card Computer debuts
Subprograms and routines
Australia's TI Fair
Techie Corner
Structured Basic Programming
Finding that User Note
Reviews - US Open Tennis & PRBASE (This latter one is held in the TIsHUG Library and will soon be released)
Newsbytes & User Notes

Again it's chock full of info, news and lots of ads for you to write away and get some great software. Make sure you get your copy at the club shop.

Those with a Corcomp Triple Tech Card will find a great listing for a directory program, which will give you day, date and time on your disk directories. Shades of Ross Mudie, there is also a listing which counts the number of words in a TI Writer file. An English ad tells of 3 new 100% machine code games available - Computer War, Submarine Commander and River Rescue. 35.00 is the suggested price for the 3 on one disk (English not US $). Corcomp have a new 512K expansion card available, price mis $229.95.

EXTENDED TUTORIAL

Tidbits II

A series of Extended Basic Tutoials for the beginners

Younger Set member helps out!

Hello again! I pity those who, on seeing this article, say to themselves: "Oh no! Not again!". (as if anyone would!) But I warmly welcome those who welcome my little (!?!) article!

If you wish to send me anything, address it to:

Wade Bowmer
Lot 11 Yanderra Ave.
Bangor
NSW 2234

This month I've got 4½ quirks. (4½? You'll see why later.)

Both BASICs allow you to add a remark after a line number.(Didn't know that, did you?) Here's an example:

```
GOTO 1000,SOMEWHERE
```

But there are always exceptions:

```
ON A GOSUB 1000,2000,STATEMENT
```

That is a no-no. But it works after nearly every line number, after GOTO, GOSUB, & IF..THEN..ELSE. But this remark is treated like a variable; so no spaces! There is, however, a bonus:

```
100 GOSUB 1000,TEST :: PRINT "YAY!!"
```

Yes, that's right, a remark in the middle of a line!

(This appeared in COMPUTE! January 1985, Pg 16)

Now for #2. Have you ever found DISPLAY AT useful? I have. (I prefer it for certain things sometimes, too, instead of CALL CLEAR and CALL HCHAR.) SIZE is an almost indispensable clause in DISPLAY, as without it, DISPLAY erases the rest of the line. So SIZE clears a window. (You should already know that...)

But, apart from separate DISPLAY ATs, what if the display list is complex, like A$;":";K ? Well, you could do

```
DISPLAY AT(1,1)SIZE(LEN(A$&":"&STR$(K)))
:A$;":";K
```

but this is cumbersome, using 37 bytes! Alternatively you can guess and do:

```
DISPLAY AT(1,1)SIZE(10):A$;":";K
```

or

```
DISPLAY AT(1,1)SIZE(10):USING"##### :
####":A$;":";K
```

which use 27 and 41 bytes, respectively. But the best way is:

```
DISPLAY AT(1,1):A$;":";K;
```

using just 21 bytes.

The secret is the final semi-colon. When a display-list ends in a semi-colon, the rest of the line is left untouched!! So the display-list doesn't take any more space than it needs.

My third quirk is also to do with DISPLAY AT: the colon. If you use

```
DISPLAY AT(22,1):"a string constant"
```

then use

```
: : : :
```

to scroll the screen up two lines and then "another string constant" my advice is to use

```
DISPLAY AT(22,1):"a string constant"::
PRINT : :"another string constant"
```

instead because otherwise "a string constant". will remain on line 23 and "another string constant" will end up on line 2!

The colon goes down a line, fair enough, but you'd expect the screen to scroll when it reaches the bottom. No, It doesn't! It comes back on the top of the screen (line 1) instead of scrolling!

(Incidently, DISPLAY AT with SIZE can be used to blank out part or all of the text row (columns 3 to 30) with the space character. If you wish to

COLOR EDITOR

The following program allows you to mix any two colors and displays a large block of the new color. After using it for a little while we made a couple of small modifications that allowed it to run a little faster. We left the original code in place with REMs in front of it so you could see the difference. There are many ways to get the computer to do something and its always fun to see how another way works.

This program places 15 double sized sprites down the right hand side of the screen to form the color selections. You are then allowed to move another sprite, the marker, up and down this column to make a selection. When you are on the color you want just press the fire button on the number one joystick or any key but the up and down arrow keys. The program then uses this color as the foreground color and the next color chosen will be the background color. It is fun to play with and it shows you which colors mix well to form a new shade for use in another program. We would like to thank Mr. Reitinger for sending us this program. We also hope he doesn't mind the minor modifications to his program.

```
1 !    COLOR EDITOR
   for mixing any desired
   2 colors with joyst or
       keyborad.

2 !   With Greetings

      E.H.REITINGER
     Vienna, Austria

3 ! TI99-Journal-Klub
      A-1150 Wein
    Felberstrabe 24/26

10 CALL SCREEN(16):: CALL CL
EAR

20 M$="55AA55AA55AA55AA" ::
A=122

30 CALL MAGNIFY(2):: CALL CH
AR(64,RPT$("F",16),34,"FF818
1FFFFFF",128,"FFFFFFFFFFFFFF
",73,M$):: CALL COLOR(3,16,2
,4,16,2,6,1,1,5,2,1)

40 CALL VCHAR(1,27,64,192)::
 CALL HCHAR(23,1,64,162):: H
=1

50 G=-2 :: FOR I=3 TO 16 ::
CALL SPRITE(#I,64,I,(G+I)*12
,230):: NEXT I :: CALL SPRIT
E(#2,34,16,5,230)!#16,128,16
,17,230)

60 CALL SPRITE(#1,42,2,A,231
)

70 FOR S=4 TO 22 :: CALL HCH
AR(S,3,73,24):: NEXT S

80 CALL JOYST(1,X,Y):: ON (S
GN(Y)+2)GOTO 90,130,110

90 A=A+12 :: IF A>170 THEN A
=2

100 CALL LOCATE(#1,A,231)::
GOTO 130
```

clear a whole text line, then just use
DISPLAY AT(row,1)
or if you wish to clear any column to 30 then use
DISPLAY AT(row,column)
It uses less memory than CALL HCHAR!)

And for #4: XBASIC's RND function is slow. I'm sorry, I don't know why. But I do know how to get RaNDom numbers faster!

First of all, there is CALL PEEK(-31880,variable) which returns a value in the range of 0 to 99, but you must execute a RANDOMIZE between each CALL. If you require a different range, then multiply by the (range required)/100, or work that out beforehand.

Secondly,CALL PEEK(-31808,varaible1,variable2) which returns two numbers in the range 0 to 255. Once again, RANDOMIZE is needed between each CALL. If the range you need is different, multiply by range/256, or work it out beforehand.

Incidently, it doesn't matter whether RANDOMIZE has a seed (numeric-expression after it) or not! Even if it's the same seed the numbers returned will be different every time!

The third one is CALL PEEK(-31879,variable) which also returns a value range 0 to 255. But this one doesn't need RANDOMIZE between each CALL. Why?

Location -31879 (>8379) is the VDP (Video Display Processor) interrupt counter. It counts 0 to 255, continuously. But...

Where I read about this location (SND Sept. 85 Pg 13) it said that it takes the computer 4¼ seconds to complete a cycle, but on my computer it takes just 5 seconds! Why?

Now for my ½ quirk: the sound processing routine uses the VDP interrupt to time the sound. That's why when I enter CALL SOUND(4250,440,2) it takes about 5 seconds to complete the note, instead of 4¼.

Finally, my end-of-article question: Why does it take 5 seconds instead of 4¼?

Incidently, last article's problem had nothing to do with the computer, rather, the TV! Sometimes, it loses some Red signal..

* SPECIAL NOTE *

These TIDBITS articles are aimed at the programmer in general. There are beginner programmers, advanced programmers, and many others in between: I cater for all (or at least I try to...)

If you read something you already know, remember that there will be other readers who hadn't know it before.

If you find something to difficult to understand, remember that someday you will understand it, and that there are other readers who already do.

PREVIEW:

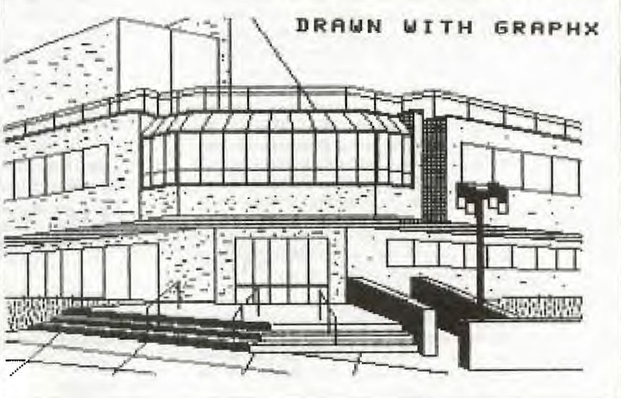
Starting next article, I will print "Beginners Tips", specially written for those just learning to program.

OFFER:

I wish to offer my services in the following way: If you have a meagre or mediocre program that you wish inproved (or "jazzed up"), you can send it to my address (already printed). Send it in the following way: SAVEd at least twice on a cassette, preferably with different head azimuth settings- but don't go to extra trouble! And please include as much documentation you think the program needs, it's possible that I will ask for more, but please don't overdo it!

At present, I can only do BASIC & XBASIC programs on a "bare-bones" console with joysticks, but I will eventually be able to do programs with the MiniMem. You may include a list of improvement suggestions, so don't hesitate to (I may not improve it to your liking, otherwise...). I would prefer the programs didn't use files, either (partly because I don't have a remote socket on my cassette recorder...). And include you address and telephone number please! I will contact you when I've finished to negotiate a fee. ■

```
110 A=A-12 :: IF A<0 THEN A=
170

120 CALL LOCATE(#1,A,231)

130 CALL KEY(1,K,S):: IF S=0
 THEN 80

140 IF K=5 THEN 110 :: IF K+
1=1 THEN 90

145 F=INT(A/12+2):: CALL SOU
ND(200,660,2):: GOTO 180 ! T
his was inserted to replace
line 150,160 & 170 Ed.

150 !FOR F=2 TO 16 :: CALL C
OINC(#1,#F,3,C):: IF C THEN
CALL SOUND(200,660,2):: GOTO
 180

160 !NEXT F

170 !GOTO 80

180 CALL COLOR(6,F,H):: DISP
LAY AT(24,9)SIZE(7):USING "
## ## ":F,H :: H=F :: GOTO 8
0
```



RS232 TUTOR

The method for accessing the RS232 interface in Assembly language is not very difficult once you have gleaned all of the pertinent information out of the EDITOR/ASSEMBLER manual. The information is in the manual but it is not all in one place. And unlike what we encountered in BASIC with various configurations and examples with explanation of each, in Assembly, we have no examples, just the information with nothing to really tie it all together for us.

Let's go through this information in the EDITOR/ASSEMBLER manual and begin building a general purpose routine that we can use for many applications.

First of all, if we try to look in the index of the manual for pages where the RS232 interface is referenced, we will come up empty-handed. Instead, we are going to have to use a little logic and intuition to locate the pertinent information.

First of all, if you will remember how operation of the RS232 interface is handled, this gives us a big key on where to look. That's right — the RS232 interface is handled just like a file, i.e., open, close, file #, etc. So, if we look under file in the index we find several places we can look To save time, and as you probably know, I have already looked at all of these references. So, let's look at the section on file management.

It is in this section we begin to see some things that really help us to better understand the operation of our computer and its peripherals. We learn that all devices, with the exception of the keyboard and its screen, are handled with DEVICE SERVICE that DSRs are accessed via a utility program called DSRLNK. By setting up a series of parameters in certain locations of memory, we can use this routine to gain control of the RS232 interface.

Then we are informed that we must have something called PERIPHERAL ACCESS BLOCK (PAB) set up VDP RAM before we call this routine. Well, this little bit of info sends us back to section 18 on file management.

In subsection 18.2, we find the definition of the PAB and how to set it up. We find the PAB is a series of bytes by which we can pass data to the device we are trying to control or access.

The first byte of the PAB must contain the op-code. These codes are explained in subsection 18.2.1. Here we see that the operation of the RS232 is going to be similar in function to the way it is operated in BASIC. That is, we are going to have to first perform an opening operation for the RS232 interface and then change the op-code so we can use the RS232 for output. And when we are finished we are going to have to close the RS232 operation.

The next byte of our RS232 PAB will contain the information on the file characteristics. Here we will use the default characteristics that are used for the RS232 operations in BASIC. So, we will set up our RS232 PAB for variable record length, display type data, update operation, and sequential file.

Next we need to decide where we are going to put the data we are going to send to the RS232 interface. This data must be in VDP RAM. So, not only do we need to decide where in VDP RAM we are going to put this data, we must now make a note that any data we will be sending to the RS232 interface will need to be placed in the VDP RAM before we can send it For the purpose of this routine we will use VDP RAM address >0002 for the location of our data. This is also the third location of the screen image table. So we will be able to see the data that we are sending to the RS232 interface.

The fourth byte will contain the value 80. This is the default value of the record length for the RS232 interface when we are using BASIC. The next byte contains the information on the number of bytes to be transferred Therefore, this byte will contain the length of our character string we will be sending to the RS232 interface card.

We will put zeros in the next three bytes, since we don't need to specify the record number when the file is sequential and we're not using the cassette interface.

The next byte contains the length of the file descriptor. The file descriptor itself is identical to the information we enter in BASIC to access our printer and RS232 interface. In our example, the file descriptor is RS232.BA =1200. So, the value in this byte will be 13. And since I have already given our file descriptor, it will start in the byte following the name length byte; in other words, byte number 10 of the PAB.

Now that we have built our PAB, let's go back to the section on the DSRLNK, ad continue developing our RS232 accessing routine.

The next piece of information we get from the section of the DSR link utility is the CPU address " >8356 and >8357 must contain" the VDP address of the name length byte in the PAB. Since we will be putting the PAB in VDP RAM at address >1000, the name length byte will reside at VDP address >1009. As a note of interest, I have found that you need to replace this pointer each time you call the DSRLNK because you can never be sure of its value after you have returned from DSRLNK.

Now, we see that the manual informs us of an area of memory called UTLTAB, but this information does not pertain to us in this application.

Lastly, we see that upon returning from the DSRLNK, if an error occurred the equal bit of the CPU STATUS REGISTER will be set. So, if we test the equal bit after executing DSRLNK, we can jump to an error routine which will display the error. If an error occurs, the error code is placed in workspace register 0 of our workspace registers.

We now have all the information we need to write our routine. I have written a sample routine for this article which is included for you. This routine was written for the EDITOR/ASSEMBLER system.

Those of you that own MINIMEMORY systems will need to make a few changes in order for this program to work with the LINE-BY-LINE Assembler. The three labels at lines 8, 9, 11 and 18 will need to be changed to two character symbols, such as M, M0, M1, etc. Also, the symbols VMBW, VSBW, and DSRLNK will need to be substituted with their equivalent addresses, >6028, >6024, and >6038, respectively And of course, you will not need the REF and DEF statements on lines 6 and 7.

Lines 8 through 18 form the calling part of the program. Once the program is executed, each time you press the ENTER key, the string will be sent to the RS232 interface. When you press FCTN, the program will terminate by returning to the calling program.

Lines 19 through 35 write the string to the VDP RAM, write the PAB to VDP RAM, open the RS232 interface as a file, write the string to the RS232 interface, close the RS232 file, and return to the calling program.

Lines 36 through 41 store the value of the VDP RAM address of the name length byte in the PAB to CPU address >8356, and then calls the DSRLNK routine after which the equal status bit is checked to see if an error occurred during the DSRLNK. If not, the routine returns to the calling program.

Lines 42 through 53 are called if an error is detected after returning from the DSRLNK. This routine displays the error on the video screen and waits for you to terminate the program by pressing "FCTN= ".

The remainder of the program contains the text for the error routine, the PAB data, and the string that is transmitted to the RS232 interface.

RS232 PROGRAM

```
ØØØ1   ****************************************
ØØØ2   * RS232 PRINTER SUBROUTINE             *
ØØØ3   * THIS ROUTINE WAS WRITTEN FOR THE     *
ØØØ4   * EDITOR/ASSEMBLER SYSTEM.             *
ØØØ5   ****************************************
ØØØ6          REF  VMBW,VSBW,DSRLNK
ØØØ7          DEF  RUN
ØØØ8   RUN    MOV  R11,R1Ø    ;SAVE RETURN POINTER
ØØØ9   RUNØ   CLR  R12
ØØ1Ø          LIMI 2
ØØ11   RUN1   TB   7          ;WHEN "FCTN" PRESSED
ØØ12          JNE  RUN2       ;EXIT ROUTINE.
ØØ13          TB   5          ;WHEN "ENTER" PRESSED
ØØ14          JEQ  RUN1       ;PRINT ANOTHER LINE.
ØØ15          LIMI Ø
ØØ16          BL   @PØ
ØØ17          JMP  RUNØ
ØØ18   RUN2   B    *1Ø
ØØ19   PØ     MOV  R11,R9     ;SAVE RETURN
ØØ2Ø          LI   RØ,>ØØØ2   ;WRITE TEXT TO VDP
ØØ21          LI   R1,T1
ØØ22          LI   R2,62
ØØ23          BLWP @VMBW
ØØ24          LI   RØ,>11ØØ   ;WRITE PAB TO VDP
ØØ25          LI   R1,PB
ØØ26          LI   R2,29
ØØ27          BLWP @VMBW
ØØ28          BL   @P3        ;OPEN FILE "RS232.BA=12ØØ"
ØØ29   P1     LI   R1,>Ø3ØØ   ;SET WRITE OP-CODE
ØØ3Ø          BLWP @VSBW      ;TO PAB.
ØØ31          BL   @P3
ØØ32          LI   R1,>Ø1ØØ
ØØ33          BLWP @VSBW
ØØ34          BL   @P3
ØØ35          B    *R9       ;RETURN
ØØ36   P3     LI   R3,>11Ø9 ;SET PAB POINTER
ØØ37          MOV  R3,@>8356
ØØ38          BLWP @DSRLNK
ØØ39          DATA >8
ØØ4Ø          JEQ  P4
ØØ41          RT
ØØ42   P4     CLR  R2        ;I/O ERROR ROUTINE
```

```
                MOVB R0,R2
                SWPB R2
                MOVB @NO(R2),R1
                LI   R0,49
                BLWP @VSBW
                LI   R0,42
                LI   R1,ER
                LI   R2,7
                BLWP @VMBW
                LIMI 2
                JMP  $
       ER       TEXT 'ERROR=  '
       NO       TEXT '0123456789ABCDEF'
       *        PERIPHERAL ACCESS BUFFER (PAB)
       PB       BYTE 0              ;OP-CODE
                BYTE >10            ;FLAG/STATUS
                DATA >0002          ;VDP BUFFER
                BYTE 80             ;RECORD LENGTH
                BYTE 62             ;# OF CHARACTERS
                DATA 0
                BYTE 0
                BYTE 13             ;NAME LENGTH
                TEXT 'RS232.BA=1200'
       T1       TEXT 'The quick brown fox jumps over the lazy dog''s'
                TEXT ' back. 1234567890'
                EVEN
                END
```

```
100 CALL CLEAR
110 INPUT "SPEED (100) FAST
(500) SLOW?":F
120 PRINT "2 OR 3 NOTE SEQUE
NCE":::::::::::::
130 CALL KEY(0,X,Y)
140 IF Y=0 THEN 130
150 IF X<50 THEN 130
160 IF X>51 THEN 130
170 ON X-49 GOSUB 440,460
180 CALL COLOR(2,5,5)
190 CALL COLOR(3,7,7)
200 CALL COLOR(4,11,11)
210 CALL CLEAR
220 CALL SCREEN(16)
230 FOR A=220 TO AA STEP 110
240 FOR B=220 TO AA STEP 110
250 FOR C=220 TO AA STEP 110
260 FOR D=220 TO AA STEP 110
270 FOR E=220 TO AA STEP 110
280 CALL HCHAR(4,1,A/11+20,6
4)
290 CALL HCHAR(7,1,B/11+20,6
4)
300 CALL HCHAR(10,1,C/11+20,
64)
310 CALL HCHAR(13,1,D/11+20,
64)
320 CALL HCHAR(16,1,E/11+20,
64)
330 CALL SOUND(F,A,2)
340 CALL SOUND(F,B,2)
350 CALL SOUND(F,C,2)
360 CALL SOUND(F,D,2)
370 CALL SOUND(F,E,2)
380 NEXT E
390 NEXT D
400 NEXT C
410 NEXT B
420 NEXT A
430 END
440 AA=330
450 RETURN
460 AA=440
470 RETURN
```

"If we want to get rid of this stuff, we'll have to get wise on the jargon."

CLASSIFIED ADVERT SECTION

FOR SALE:

Editor Assembler ..	$70
32K Chips	$25
Extended Basic . ..	$40
Speech Synthesizer	$65
5 Games Modules	$10 each
K/board + all cables	$110
Joysticks	$20

Or the lot for $320.00
Please phone J.Richards
on (02)863860 a/h.

EXTENDED BASIC, National
Cassettes + Books, Console,
Music Maker module all for
$120.
Please phone (02)6008932
after 6pm and ask for
Rolf Wissel.

WANT TO BUY:

I want to buy MICROSOFT MULTIPLAN
If you have one for sale, please
phone JIM on 305694 a/h

Second lease-of-life for Teledata e-mail

New money to the tune of over $1 million has been breathed into the Teledata electronic network service by three Melbourne businessmen and a MIC.

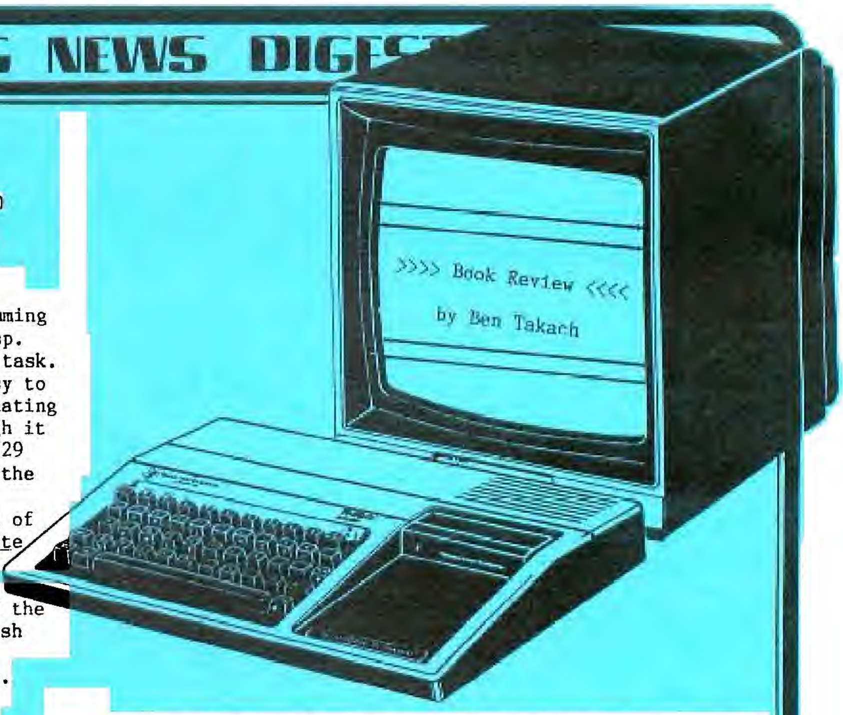
David Frenkel, formerly a general manager with Computervision, is the new president. The MIC is Western Pacific Investment Company, and the two other private investors are Peter Joss and Jim Marks, who established McIllwraith Davey Industries before selling it to Email in 1984.

Teledata's service has 2,500 business and home subscribers. It offers electronic mail, telex, Canberra news and electronic shopping services, plus gateways to other international networks.

Noel Fenton, vice-president, operations, said the investment "gives us the ability to fund overseas and Australian developments. Before we didn't have the money and we were struggling."

Frenkel said Teledata "intends to become the premier public dial-up access information and communications utility."

The new owners plan to develop and extend the company's services.



>>>> Book Review <<<<

by Ben Takach

VIDEO DISPLAY PROCESSORS PROGRAMMER'S GUIDE

Dated Aug.84. A quarto (11" X 8 1/2") format 110 pages TI publication.
The volume is kept in the TISHUG library and is available to registered library users.
It is an invaluable aid to graphic programming.
The rather cryptic references to graphic programming contained in the user's manuals of the TI-99 resp. the XBasic makes graphic programming a daunting task.
This very well prepared volume written in an easy to follow style takes the reader through the fascinating world of the TI Video Display Processors. Altough it concentrates on the newer generation TMS9118/28/29 VDPs, it also covers their predecessors used in the TI-99: the TMS9918A/28A/29A.
The reader is advised to consult the Data Manual of the TMS9918A/TMS9928A/TMS9929A also, to appreciate the differences of the two VDP series also available from the TISHUG library).
The book explains the interface communication of the CPU - VDP - EPROM and VRAM (dynamic screen refresh memory or V(ideo)RAM). Description of the VDP registers and their protocol is a useful chapter.
The publication deals with assembly language programming only.
Other chapters deal with CREATING PATTERNS, DISPLAY MODES (e.g. graphicsI, graphicsII, text and multi colour modes) and SPRITES.
Horizontal and vertical scrolling as well as animating sprites are discussed under the heading PROGRAMMING TIPS. This chapter also explains the function of the sprite coincidence flag. Appendices take up almost one half of the volume. A number of useful data tables and a complete graphics mode character set is included. Finally assembly language program examples are given in 6502, 8088, TMS7000, TMS9995 and TMS9900 assembly languages.
As mentioned earlier this book should be read in conjunction with the TMS9918A/28A/29A VDP Data manual. Altough the data contained in this edition is very similar to that of the Programmer's Guide, it is solely devoted to the VDPs used in the TI-99, thus there is no ambiguity. It is also a more technical publication primarily written for system designers. This volume is a UK edition dated 1982. I have found both books very informative, easy to digest and a useful aid to understand the TI-99 architecture.

TEXAS INSTRUMENTS TERMINAL EMULATOR PROTOCOL MANUAL.

Dated May 1981. A quarto format booklet of 53 pages produced by direct printing of a word processor prepared original.
The manual is a companion volume to the Terminal Emulator II owners manual. It provides an indepth explanation of the features of the TEII module. The book deals with the video control, the graphics and text mode supported by the module. The protocol to change from one mode to the other is explained.
Detailed description is given of the sound generators and the speech option.
A very clearly written chapter deals with the disk file organisation of the TI-99/4. This chapter is universally applicable to the TI-DOS and contains valuable information for the users of the TI-99 disk systems. Section 3 explains the data coding process performed by the terminal emulator. Further chapters detail the control characters and escape sequences one may receive with the terminal emulator. This chapter is very concise. It does not mention that in addition to the characters listed all other control codes in the range of ASCII 1-31 may be sent and received. Some not listed in the manual are definitely supported by the TE-II (e.g. CTRL S and CTRL Q, X-off resp. X-on).
Character definition, sound and speech command sequences are also detailed together with screen colour change commands. One could not hope to write programmes around the TE-II without these rather complex command sequences. These are contained under a chapter heading EXTENDED WRITES (I would really like to have an explanation of this expression). An indepth description of the file transmit / receive protocol concludes the main body of the book, followed by appendices. Two very useful appendices, App.B & C, deal with examples of host to remote resp. remote to host file transfers. Who should read it? Users of the TEII, programmers, and anyone who wants to gain a better understanding of the architecture of the TI-99/4.
The volume is kept in the TISHUG library and is available to registered library users.