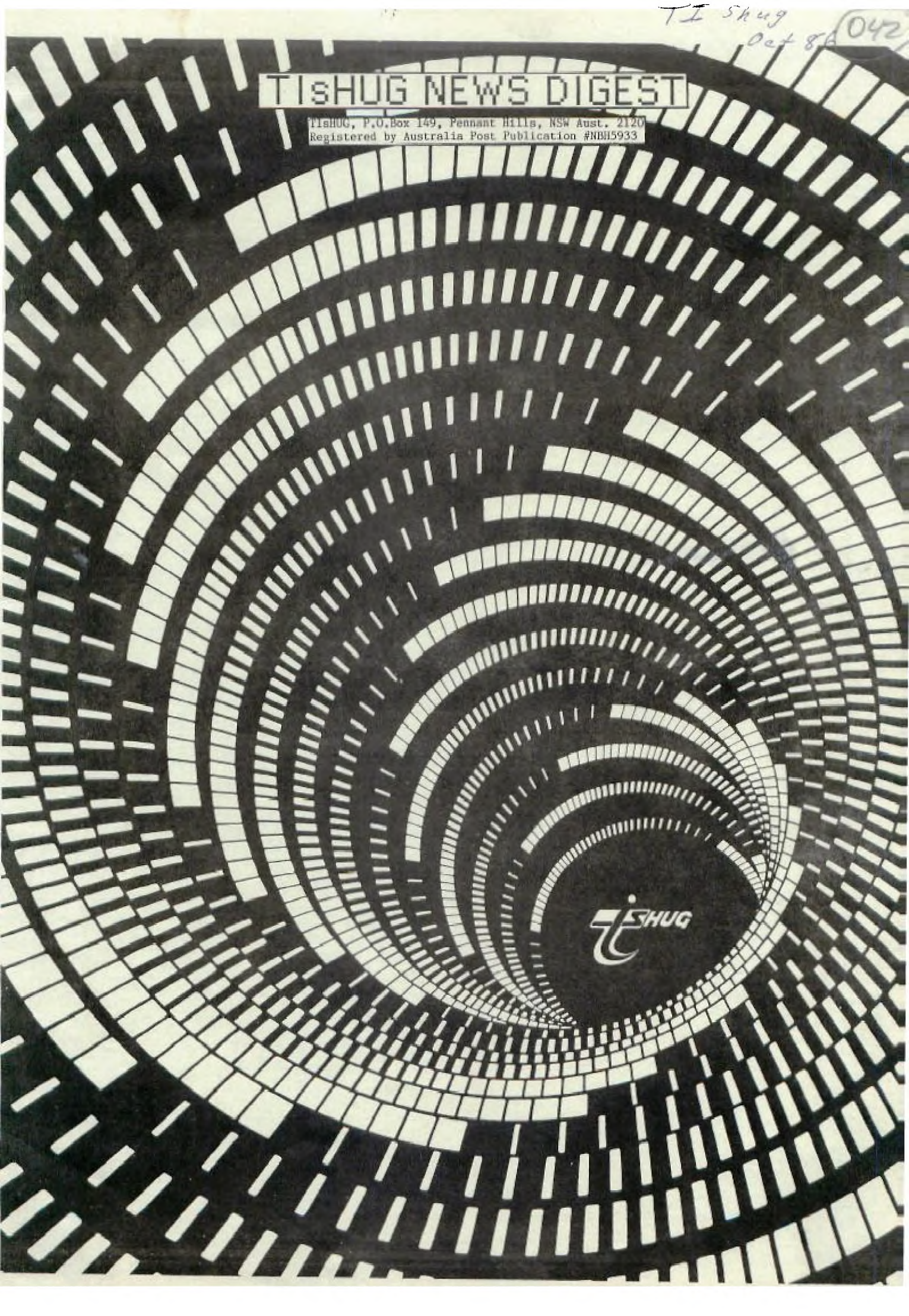


TJ Shug  
Oct 86 (042)

# TISHUG NEWS DIGEST

TISHUG, P.O.Box 149, Pennant Hills, NSW Aust. 2120  
Registered by Australia Post Publication #NBN5933



*TJ* SHUG



TISHUG NEWS DIGEST  
Vol.5 No.9 Oct '86

The official Newsletter of  
the TI's HOME COMPUTER  
USERS GROUP'S-Australia.

Editor & Layout designer:  
SHANE K. ANDERSEN

Editorial Staff:  
TERRY PHILIPS

Photographer:  
MAURICE STEWARTSON

Artwork in this issue:  
STEPHEN JUDD

Co-Ordinator:  
FRED MORRIS  
(02)871-3873

**Monthly meetings:**

Shirley House,  
Ethel Street, Burwood,  
(behind WOODSTOCK CENTRE)  
2pm - 4:30pm  
except for full day  
tutorial special events  
and January Public Hols.

**TISHUG REGIONAL HOME GROUPS**

**GLEBE:**  
Mike Slattery (02)6920559  
**PENRITH:**  
Mel Copeland (047)351340  
**CENTRAL COAST:**  
Russel Welham (043)924000  
**LIVERPOOL:**  
Stan Puckle (046)256157  
**BANKSTOWN:**  
David Mayo (02)7084293  
**ILLAWARRA:**  
Rob Montgomery (042)286463  
**CARLINGFORD:**  
Chris Buttner (02)8717753  
**SOUTHERLAND:**  
Peter Young (02)5288775  
**NORTH SHORE:**  
Keith De Haan (02)4513032

SHARE WITH US AT OUR  
REGIONAL HOME GROUPS!

TISHUG Aust. Club Address...  
P.O. Box 149 Pennant Hills  
N.S.W 2120



**IN THIS ISSUE**



From the Editor  
SHANE K. ANDERSEN

As you have read in the Face to Face column with our Co-Ordinator, the next club activity will be another big AUCTION AFTERNOON.

A great chance to get some real bargains in both hardware and software for your TI-99/4(a) computer.

With such an event, there has to be guidelines which we ask that you follow strictly...and they will be explained in this article.

Inserted in this issue, you will find a lift-out AUCTION LIST. You are asked to bring it, or photocopies of it, along to the next meeting.

- (1) Any person NOT supplying an Auction List and obtaining a LOT NUMBER from the Auctioneer will not have his/her goods Auctioned on this day.
- (2) Only MODULES & PERIPHERALS will be allowed for Auction. Disc/Cassette software will not be permitted for sale, swap or action.
- (3) Each and every item you bring into the hall MUST be tagged or labelled with your NAME on them. A record will be kept of all buying and selling.
- (4) ALL ITEMS SOLD: via CASH or Club BANKCARD/MASTERCARD ONLY. Regarding Club Bankcard/Mastercard: This service will be available for the convenience with a service charge of 10% to cover handling costs...and AS-IS basis. The Club takes no responsibility as to the condition of goods sold.
- (5) ALL SALES will be finalised at the fall of the hammer. No bid shall be retracted, and in the case of dispute, the Auctioneers decision shall be final.
- (6) Each person will be allowed to Auction five (5) items. If there is time at the end, then anything else you have will be put up also...and an Action List will then be provided.
- (7) The seller does not have to sell, unless he/she is happy with the price.
- (8) CHARGES: Those bringing MODULES - \$2.00 (any quantity) PERIPHERALS-\$2.00 EACH ITEM.

So there you have it. It should be a fun and productive afternoon for all, and a great time to thin-out or stock-up goodies for your TI-99/4(A) Computer. As they say on TV ... "BRING YOUR MONEY WITH YOU" ... BUT DON'T FORGET YOUR ACTION LIST.

This time, 99/4 (A) Consoles will be permitted for sale, for those who want a second (spare) unit, or for those who have a friend who also wants to share in the fun of TI-99/4(A) programming.  
Please ensure that you also bring your Membership Card/Badge to be worn on this day, as this activity is a special service to club members only

**IN THIS ISSUE:**

We have a lot of creative and talented members in TISHUG, and this issue of TND will show you some of these talents. We have the new MUSIC KEYBOARD INTERFACE & SOFTWARE for the TI by Aro Heino, news of our MINI PERIPHERAL BOX by Peter Schubert, a MORSE CODE INTERFACE & SOFTWARE by Ross Mudie, how to make your own LOAD INTERRUPT SWITCH with software with Paul Charlton ... plus, information about the new TEXPAC NUMBER & EXPLODING TRIPLE TECH CARDS, the commencement of the BIG CHRISTMAS COMPETITION (just two months away ya know!).

Check out the pictures created on our TI-99/4(A) of Digitized photographs which shows that the 99/4(A) is alive and well and very powerful...plus we have more great reading with our reviews, articles and programs to type in.

Once again, I think you will be very happy with what has been prepared for you this month.

★REMEMBER THAT THE NEXT MEETING WILL BE HELD ON THE SECOND SATURDAY OF OCTOBER, DUE TO THE LONG WEEK-END!★

Bye 4 now  
SHANE ANDERSEN (EDITOR)







## FACE TO FACE

A VIEW FROM THE TOP!  
Co-Ordinator's Report

WOODSTOCK REVISITED! (or, TISHUG goes where angels fear to tread - 24x32K expansions in one day and, they all work!)

On September 6th, reminiscent of the 1960s when 1000's converged on Woodstock to bear their souls and minds (bodies too, in some cases) to the beat of Rock 'n Roll, we saw, not the 1000's, but TISHUG members turning up to the TECHO WORKSHOP day for the purpose of breathing new life into their TI.99/4A computers. On the day 24 members installed a 32K memory expansion into their computers - what is more they all worked! I wonder - does this constitute a TI. User Group record?

Congratulations go to the TECHO team for their efforts and to the members who participated in what could be described as one of the more successful workshops held for some time! (The potential for disaster existed in proportions too horrendous to contemplate!) Starting at around 11.00am, until 7.00pm, it was heads down and bottoms up - the whole scene looked like the repair centre for that other machine - "Whateveritis 64" - the major difference being that the heart of the TI.99/4A now beats bigger and harder than ever before! Well done all of you!

I am certain that many of the members who expanded their machines enjoyed the experience. Possibly some would not like to go through it again - yet, are thankful for the opportunity to explore some of the, until now, hidden powers of the TI. Already Robert and his team are preparing to take on the next wave of members wanting to expand their machines - this time however, with possible modifications to both the 32K kit and timing for installation. Any members still interested in this expansion should forward a deposit of \$5.00 to the club shop so that we can gauge the overall club requirement for expansion kits.

While on the subject of expansion, Chris will be looking at a proposition that the club shop import expansion boxes from the U.S.A. We understand that these boxes are now available at around \$20 U.S. If so, then the landed cost of the upgrade will be cheap enough for many TISHUG members to afford to expand their computing horizons. More on the subject later. Other news is that Peter Schubert and team are designing an expansion box "motherboard" for those wishing to make their own expansion boxes. This will not have as many card slots as the box made 2 years ago by the "ABC" group but will nevertheless cater for most foreseen requirements. Also, the Hunter Valley group are ready to go into production of a similar expansion box motherboard - contact the club shop for details. Lastly, Ben von Takach will be returning from Germany, in time for the October meeting, bringing some German made TI.99/4A products. Who knows, we could be seeing quite a lot of this gear arriving in time for Xmas?

### BULLETIN BOARD

As mentioned last month, we have indeed run into a problem relocating the bulletin board. John Paine has advised that due to a change of work responsibilities - likely to require more time away from home - he is unable to take on the BBS. As a temporary measure, Ross Mudie has agreed to look after the BBS - it is up and running on telephone number (02)-3191009. Please note that it will not be attended after business hours so care must be taken not to "crash the system".

### SYSOP VACANCY

In view of this situation we now need a member to take on responsibility for the BBS - Yes, this means you! Should you like a tour of BBS duty please contact me, or any other member of the committee, for further details.

### OCTOBER MEETING - October 11th, 1986

Another AUCTION DAY planned. A good time for you to buy and/or sell items related to computing generally - so bring them along! The club, as mentioned previously, will be auctioning some items too - so come ready to make an offer! Items being sold by the club include:-

1 x TI. Impact printer	B/V \$345.26
1 x Brother HR25 printer	B/V \$424.80
2 x TI-PHP 1850 Disk Drives	B/V \$427.68
2 x TI-Internal drives (SS)	B/V \$244.80

B/V represents the current book value of club assets - offers will be considered. To view, prior to the auction, contact Terry re the TI. Printer on (02)-7976313 and contact John Robinson on (02)-8480956 regarding the Brother HR25. The disk drives will be available at the October meeting. We look forward to seeing you again.

### FORTH SPECIAL INTEREST GROUP

Terry Johnsen has agreed to co-ordinate a Forth special interest group and will make his home available to members wishing to "get into Forth". Contact Terry on (02)-802317 to get the ball rolling. Have you got your copy of Terry's Forth Spell Checker? And, have you made contact with the SYDNEY FORTH INTEREST GROUP which meets at Sydney University? Well Terry can help.

### ASSEMBLY SPECIAL INTEREST GROUP

I have decided to follow my own advice regarding Assembly - remember I said that the only way to learn Assembly programming was to have a task then tackle it? Any do? Well, to further my own interest in Assembly I will host an Assembly SIG. Interested then phone me on (02)-8713873. Let's get into it!

### REGIONAL GROUPS

Brian Graham tells me that Country members have not exactly rushed to become involved in the COUNTRY REGIONAL GROUP. Remember, it exists so use it - I am sure you will find Brian to be most helpful. Also, Russell Welham's Central Coast Regional Group will be holding a full day workshop tutorial on October 18th, 1986. For details contact Russell on (043)-924000. North Shore members have you contacted Keith de Haan yet? Don't forget to contact Peter Young if you live in the Sutherland Shire. Woollongong Regional Group will see us soon!

### CORCOMP - SERVICE

CorComp are still "on the ball"! Recently two Australian users had need to contact CorComp related to service advice and/or parts. They report that it took CorComp approximately 1 week to respond to their enquiries. This response time included sending parts to Australia which were necessary to fix the users difficulties. Congratulations CorComp!

### SPECIAL PROGRAMMING PROJECTS

Lastly, we need some programming projects. What would you like to do with your computer (other than that!) but cannot find the right software to do the job? Write to us explaining the problem maybe we can help. Your TI. is capable of a lot more than you think! So, drop us a line.





**BOOM!**

URGENT NEWS FOR THOSE WITH THE TRIPLE-TECH CARD!

Has your Triple Tech card had a blowout yet? No, I'm not referring to a FLAT tyre! Owners of the Cor-Comp card are starting to report that the lithium battery, used to maintain the time in the clock function, is EXPLODING on them. One Triple Tech owner said the noise was LOUD, like a gunshot. Some have stated that damage was done to the card, while others apparently suffered no fatal damage directly, but caused damage to the PEB box, or an adjacent card.

The card is not inside the familiar clam-shell case, as other cards for the expansion box, and is usually "sandwiched" between two other cards for stability. This places the battery in a position of being only mere milli-meters from sensitive chips in the adjacent cards. The force of the explosion is apparently great enough to cause various amounts of damage to that adjacent card.

It's hard for me to believe that Cor-Comp could have designed something with a major flaw, but the cause is apparently due to the fact that lithium cells are not rechargeable style batteries, and that's apparently what IS happening to it while expansion box is turned on. The battery is removable, and perhaps should be REMOVED in order to protect your investment, until a cure for this problem is announced.

If indeed this is a design flaw, then it would seem to dictate a product recall by Cor-Comp. All it would need is for a lithium battery to be replaced by a nickel-cadium battery. However, I have not heard any official comment from Cor-Comp regarding ANYTHING about the problem, much less that the problem even exists.

This article has been reprinted from the April '86 Dallas TI Home Computer Groups' newsletter, and then in the Sacramento 99'ers Users Group Newsletter of August '86. Haven't seen any further information in their newsletters or elsewhere.

MODIFICATION TO TRIPLE TECH CARD TO OVERCOME EXPLODING BATTERY PROBLEM FROM TISHUG(AUST)

by Ross Mudie...Oct'86 TISHUG NEWS DIGEST

When the battery was removed from the Triple Tech card and the expansion unit was turned on I read 5.1 volts across the battery holder terminals which could account for the reports of these non-rechargeable 3 volt lithium cells exploding.

To overcome this problem I connected a BAV45 (very low leakage) diode in series with the positive side of the battery. This prevents charging of the battery but still allows the battery to run the clock chip when the P.E. box power is turned off. A convenient mounting position for the diode is in series with R7 on the Triple Tech PCB. I unsoldered the end of R7 nearest to the battery and placed the anode end of the diode in the vacated hole in the PCB. The cathode end of the diode (the lead away from the pip on the can) was soldered in mid-air to the free lead of R7. The anode of the diode connects to the positive side of the battery. The leads should be cut suitably short.



[02]  
319 1009  
TEXPAC



Texpac BBS has been going through some changes over the past month, with the stepping-down of its Original System Operator, Shane Andersen. He said it could be done, and proved it, with the help of Robert Crago and helpers. Two years later, it's still one of the fastest methods to obtain News, Jokes & programs.

Over the next month or two, while we wait for Telecom to get their act together and give us a new line at the new permanent address, Ross Mudie has stepped in as acting Sysop to ensure that our modem Communicators don't loose interest while the relocation is in progress, has provided fellow modem users with an alternative phone number...(02) 319-1009.

While this stage was being prepared, all calls were being redirected to one of Australia's most powerful systems called THE BLACKBOARD, on (02)526-1343.

We wish to thank Will Black (a member of TISHUG) and sysop of The Blackboard BBS for his intervention in providing a special TI SECTION on his board. It is hoped that members will continue to use, and contribute to both systems.

By the way, you may be interested to know that Ross has decided to make TEXPAC BBS a 24 hour service, 7 days a week, and asks that all members come on line to 3191009 as often as possible, as you may miss out on some news. Instead of updating the system once a month, he'll be adding and deleting information on a regular basis throughout the month.



NOTE that a normal diode; e.g 1N914 is not suitable due to the reverse leakage being too high. This modification should only be done by a competent person under anti-static working conditions using a very fine tipped soldering iron. Always turn the PE box power OFF and wait 2 minutes before removing or inserting any card in the PE box.

The author and TISHUG will take no responsibility for any damage which may occur as a result of application or use of this information.

## ASSEMBLER.

### Interrupt switch

LOAD INTERRUPT SWITCH+SOME SOFTWARE TO USE IT WITH.



Our beloved 4A runs along doing what it was told to do unless something interrupts it. Pushing a module into the slot will cause a "reset" interrupt. On a reset the computer knows that it must look at a certain area of memory to get a new <W>orkspace <P>ointer and <P>rogram <C>ounter and proceeds to execute from there. Trouble is, these interrupt vectors are in ROM and can't be changed. However, there is an exception to this rule. It is called the "LOAD" interrupt. This causes the computer to look at >FFFC for a new WS and >FFFE for a new PC.

The interrupt is generated by momentarily shorting out ground and pin 13(LOAD) on the I/O port. The way that I did this is with a common doorbell button mounted on my speech synthesizer. Just drill a small hole for the switch and run 2 small wires from it to pin 13 and ground. Pins 21,23,25 and 27 are all ground. You can use any one of them. The odd numbered pins are on the bottom side of the synthesizer. Be careful-try it. If I can do it, anybody can.

The November 1983 Enthusiast 99 had some of the above and also the test program below.

```

REF  VMBW
ST   LI  0,302
     LI  1,1X
     LI  2,4
     BLWP @VMBW
     LIM 2
     JMP $
TX   TEXT 'TEST'
WS   BSS 32
     AORG >FFFC
     DATA WS,ST
     END
    
```

Assemble and load into memory then go to the title screen, push the button and test should appear on your screen. Hint:look close! Corcomp disk controller owners will have no trouble finding it.

The following is Compuserve message by Paul Charlton of Fast-Term fame.

```

#: 46939 S5/Tips & Tricks
20-Jul-85 18:08:33
Sb: #2 NEAT LOAD INTS!
Fm: Paul Charlton 75136,2256
    
```

HERE ARE TWO NEAT LOAD INTERRUPT ROUTINES. FIRST CHANGES TE2 CART TO 1200 BAUD (DON'T HOLD YOUR BREATH THOUGH) AND SECOND TO CHANGE NAME OF PRINTER FOR TAX INV. RECORD KEEPING

\* LOAD INTERRUPT ROUTINE TO PUT  
\* TE2 CART INTO 1200 BAUD  
\*

```

REGS  BSS 32
ENTER CLR @LOADWP
      LWPI REGS
      CLR RO
E1    DEC RO
      JNE E1
      LIM 0
      LI  R12,>1340
      SBO 31
      LDCR @CNTRL,8
      LDCR @INTVL,8
      LDCR @RDR,11
      LDCR @XDR,12
      SBO 18
E2    STWP RO
      MOV RO,@LOADWP
      RTWP
    
```

```

CNTRL BYTE >83
INTVL  BYTE 1600/64
RDR
XDR    DATA >1A1
*
      AORG >FFFC
LOADWP DATA REGS
      DATA ENTER
      END
AND THE NEXT ONE...
*
* LOAD INTERRUPT ROUTINE TO PUT
* A NEW PRINT FILE IN TAX INV. RECORD KEEPING
*
REGS  BSS 32
NAMLEN BYTE 3
NAME  TEXT 'PIO' | CHANGE FOR ANOTHER PRINT DEVICE
      EVEN
ENTER CLR @LOADWP
      LWPI REGS
      CLR RO
E1    DEC RO
      JNE E1
*
      LIM 0
      LI  RO,>0589
      ORI RO,>4000 'FOR WRITE TO VDP
      SWPB RO
      MOVB RO,@>8C02
      SWPB RO
      MOVB RO,@>8C02
      LI  R1,NAMLEN
      MOVB *R1,RO
      SRL RO,8
      LOOP1 MOVB *R1+,@>8C00
      DEC RO
      JOC LOOP1 'LEN + 1 # BYTES TO REPEAT
*
      STWP RO
      MOV RO,@LOADWP
      RTWP
*
      AORG >FFFC
LOADWP DATA REGS
      DATA ENTER
      END
    
```

OK??? THE 1ST YOU MUST ASSEMBLE THEN LOD FROM E/A 3. THEN REMOVE E/A CART (DONT TURN P-BOX OFF) AND INSERT TE2 CART GO INTO TE2 (ALL THE WAY TO THE BLANK SCREEN WITH CURSOR IN CORNER AND PRESS YOUR LOAD INTERRUPT BUTTON (EZ TO ADD IF YOU DONT HAVE ONE, ASK!)

TI/RK...SAME PROCEDURE AS ABOVE..BUT GO ALL THE WAY TO PRINTER (Y/N) ANSWER YES THEN TYPE RS232 (OR "TP") HIT ENTER THEN HIT LOAD INT BUTTON....VO LA! HELPFULLY, PAUL...

Lastly, we have my first Assembly program.

<-FOR THE MINI-MEMORY CARTRIDGE->

THIS PROGRAM WILL JUMP TO ITSELF WHEN A LOAD INTERRUPT IS GENERATED IT WILL PUT A MENU ON THE SCREEN ALLOWING YOU TO SELECT EDITOR OR FORMATTER IT DOES NOT RESET THE PROGRAM NAMES IN THE REF/DEF TABLE, BUT BRANCHES DIRECTLY TO THE PROGRAM'S START ADDRESSES. THE START ADDRESSES FOR THE EDITOR AND FORMATTER DEPEND ON THE VERSION OF TOM KNIGHT'S LOADER THAT YOU ARE USING. TO USE: USE A BASIC PROGRAM TO LOAD THIS AND TOM KNIGHT'S TI-WRITER LOADER INTO MEMORY, CALL LINK("START") AND AWAY YOU GO. AFTER USING EITHER THE EDITOR OR FORMATTER AND YOU WANT TO GO TO THE OTHER ONE, JUST HIT THE LOAD INTERRUPT BUTTON.

2





```

DEF START
REF VSBW,VMBW,KSCAN
AORG >FE06
MSG1 TEXT 'Editor or Formatter'
MSG2 TEXT 'Press E or F'
EDI B @>FB2C *START ADDRESS OF EDITOR
FOF B @>FB9A *START ADDRESS OF FORMATTER
START LWPI >2000
LI R0,>2000 *W'SPACE POINTER/THIS PROGRAM
MOV R0,>FFFC *W'SPACE POINTER WHEN LOAD
        INTERRUPT BUTTON IS PRESSED
LI R0,>FE2E *START ADDRESS OF THIS PROGRAM
MOV R0,>FFFE *PROGRAM COUNTER WHEN LOAD
        INTERRUPT BUTTON IS PRESSED

JMP CLEAR
DISP LI R0,265
      LI R1,MSG1
      LI R2,19
      BLWP @VMBW
      LI R0,324
      LI R1,MSG2
      LI R2,11
      BLWP @VMBW
CURSOR LI R0,337
        CLR R9
        BLWP @VSBW
        LI R1,>5F00
GETKEY SB @8374,@8374
        BLWP @KSCAN
        MOV B @>837C,R0
        COC @KEYMSK,R0
        JNE CURSOR
        MOV @>8374,R9
        CI R9,65 F
        JEQ EDI EDITOR
        CI R9,70 F
        JEQ FOF FORMATTER
CLEAR CLR R0
      CLR R1
LOOP BLWP @VSBW
      CI R0,767
      JEQ DISP
      INC R0
      JMP LOOP
KEYMSK DATA >2000
      END
    
```

The program to load the above:

```

100 CALL INIT
110 CALL CLEAR
120 CALL LOAD("DSK1.TKADD")
130 INPUT "HAS TKWRITER BEEN LOADED? ":A$
140 IF A$="Y" THEN 210
150 IF A$="N" THEN 180
160 CALL CLEAR
170 GOTO 130
180 CALL CLEAR
190 PRINT "LOADING TKWRITER":::::
200 CALL LOAD("DSK1.WRITER/EA")
210 CALL LINK("START")
220 END
    
```

A couple of notes on the int button. DO NOT USE IT WITH THE P-CODE CARD! Tends to wipe them out, I'm told. Also, the Debugger supplied with the E/A makes use of this button. Comes in handy when you are snooping around in the memory.



Your Committee is eager to give you a nice Christmas present, all you have to do is earn it. Here is the story, it is a true story and I didn't even bother to change the names to protect the innocents. Mrs. T. greeted me with the words one day: "I need the help of a genius." Such introduction always leaves me rather cold. It generally ends with yours truly dismantling the S-trap in the bathroom to retrieve Mrs. T.'s wedding ring or doing some other unpleasant tasks. Alas, this call for a genius was an exception. She handed me a fistfull of tickets and a list of numbers with the explanation: -If you can make up the exact amount of one of these prizes from these tickets then I can claim it at the shop.

There were 17 prizes on her list: \$1001, \$490, \$1387, \$558, \$86, \$410, \$1533, \$12, \$10, \$474, \$366, \$1387, \$246, \$726, \$678, \$1131 and \$1275.

The amounts shown on the tickets were: 140, 496, 68, 92, 8, 100, 224, 336, 906, 306, 496, 152 and 20.

I have crossed out a number of the prizes, commenting, you can not win these, all your tickets are even numbers; you can also cross out the prizes \$10, \$12 and \$68, these are too small, one can not make them up from the tickets you have. After that the elimination got more tedious.

Well the quickest way to solve this problem is by the computer said Mrs. T. This was a remarkable comment from someone, who so intensely mistrusts computers that she would not even go near one with the feather duster. It is not quite as simple, I do not have a program to solve such a problem, was my reply.

Neither have you out there! To make things more interesting, your Committee has set a prize for the best program.

The impartial judges will award points according to the scale below:

Program description	1-3 points
Logic flow diagram	3-6 points
Algorithm	7-10 points
Program	10-20 points

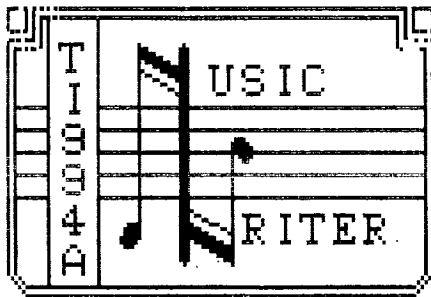
Thus a perfect score is 39 points and those not entering will get 0 points.

The prize and the closing date is announced elsewhere in this issue.

By the way, any entry, which states the correct answer without an attached program will also earn 0 points.

P.S. Since an entry by yours truly would be unethical, my well written program is available to the highest bidder. Send your offer to the Editor.





MUSIC WRITER

BY ARTO HEINO 1986

At the time of this article the program is still mostly in XBasic, but with the help of a basic compiler such as 9900BASIC, I will speed up all the graphic routines.

Requirements to run music writer are EXBASIC, 32K and DISK system.

Listed below is some of the programs capabilities:

- 1) MUSICAL NOTATION
- 2) TWO AND A HALF OCTAVES ON SCREEN
- 3) JOYSTICK OR KEYBOARD EDITING
- 4) HERES DUMP
- 5) CREATE BASIC MERGE FILE
- 6) ABLE TO USE EXTERNAL MUSIC KEYBOARD ATTACHED TO THE JOYSTICK PORT
- 7) SAVE AND EDIT THE MUSIC YOU PLAY ON EITHER KEYBOARD

Some of the commands available will be of great use to the musically minded, all these commands can be used on any portion of your score.

- 1) REPEAT
- 2) REVERSE
- 3) INVERT (melody)
- 4) CHANGE DURATION
- 5) CHANGE PITCH
- 6) INPUT TO CHANNEL 1,2 OR 3

The most interesting will be the joystick music keyboard. The types of uses it lends it self to are:

- 1) PLAY THE MELODY AND ADD HARMONY AFTER
- 2) PLAY ALONG WITH OTHER TUNES YOU WRITE
- 3) ROUGH OUT A TUNE AND CAREFULLY EDIT IT IN THE EDIT MODE
- 4) HAVE A HARD COPY OF YOUR LOOSE JAM

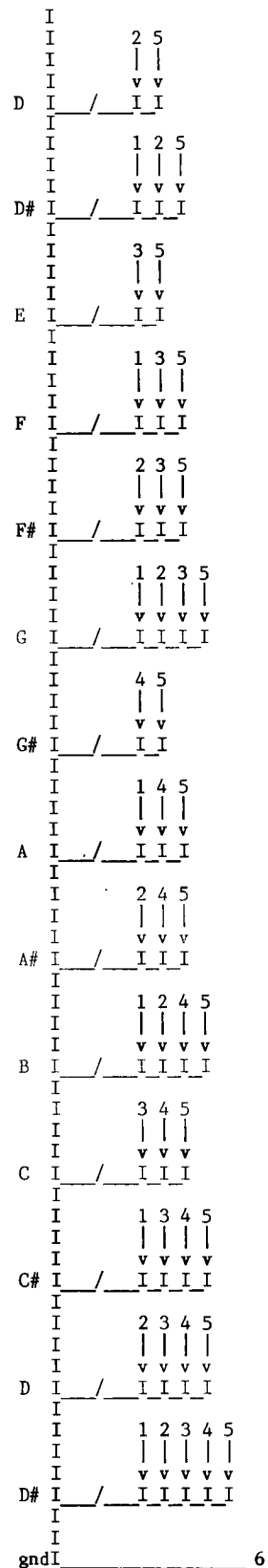
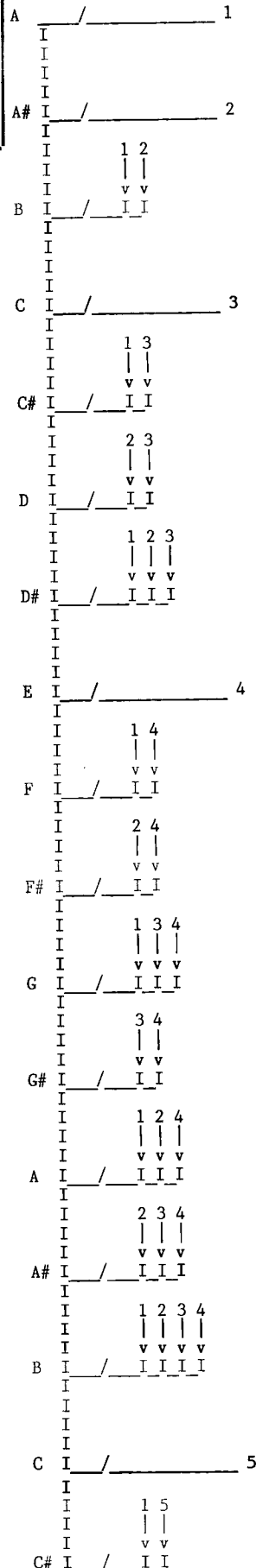
I built the musical keyboard using diodes and an old organ I had. I am working on using chips for the next one I build and a have been trying to track down a source of keyboards. If you are interested let me know.

If you want to build the device here is the parts list:

AMOUNT	PART
1)	75 1N914 DIODES
2)	1 9 PIN FEMALE PLUG
3)	1 SMALL VERO BOARD
4)	2ft LENGTH OF 6 WIRE RIBBON
5)	75 ASSORTED LENGTHS OF WIRE
6)	1 MUSICAL KEYBOARD 32 KEYS STARTING AT A natural

Here is a schematic of the circuit:

/\_/\_ = KEYBOARD SWITCH | = DIODE  
v



JOYSTICK PORT

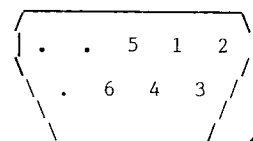






Figure 1 shows the dotmatrix screen dump of 4 screens. A total of 13 screens are available.

TITLE=DEMO1



Figure 2 is the same music but has been converted to a merge file. The option to turn your tunes to basic are on the main menu and also allow you to have a starting line number.

```

100 ! DEMO1BAS
110 ! BY ARTO HEIND 1984
120 CALL SOUND(240,391,4,261,8,195,8)
130 CALL SOUND(240,439,4,261,8,174,8)
140 CALL SOUND(480,523,4,261,8,195,8)
150 CALL SOUND(120,659,4,523,8,391,8)
160 CALL SOUND(180,439,4,349,8,219,8)
170 CALL SOUND(60,587,4,493,8,293,8)
180 CALL SOUND(120,391,4,329,8,195,8)
190 CALL SOUND(120,219,4,130,8,110,8)
200 CALL SOUND(480,523,4,261,8,195,8)
210 CALL SOUND(240,391,4,261,8,195,8)
220 CALL SOUND(240,439,4,261,8,174,8)
230 CALL SOUND(480,587,4,329,8,219,8)
240 CALL SOUND(120,659,4,493,8,349,8)
250 CALL SOUND(180,439,4,349,8,219,8)
260 CALL SOUND(60,587,4,493,8,293,8)
270 CALL SOUND(120,391,4,329,8,195,8)
280 CALL SOUND(60,261,4,195,8,164,8)
290 CALL SOUND(60,391,4,329,8,261,8)
300 CALL SOUND(480,523,4,329,8,261,8)
310 CALL SOUND(120,659,4,219,8,44000,30)
320 CALL SOUND(120,587,4,219,8,44000,30)
330 CALL SOUND(120,523,4,219,8,44000,30)
340 CALL SOUND(120,493,4,219,8,44000,30)
350 CALL SOUND(120,439,8,195,4,44000,30)
360 CALL SOUND(120,439,8,174,4,44000,30)
370 CALL SOUND(120,439,8,164,4,44000,30)
380 CALL SOUND(120,439,8,146,4,44000,30)
390 CALL SOUND(120,293,4,174,8,44000,30)
400 CALL SOUND(120,329,4,219,8,44000,30)
410 CALL SOUND(120,349,4,261,8,44000,30)
420 CALL SOUND(120,391,4,219,8,44000,30)
430 CALL SOUND(60,439,8,195,4,44000,30)
440 CALL SOUND(180,439,8,174,4,44000,30)
450 CALL SOUND(240,391,8,146,4,44000,30)
460 CALL SOUND(480,523,8,130,4,44000,30)
    
```

Here it is October already. The year sure has gone by quickly.

I hope you all enjoyed last months selection of software both on tape and disk. This month an equally impressive range will be available from the shop, including another tape of memory image assembly files especially for those whom have just had the expansion memory fitted to their computers.

Here's the titles -

On Disk:

1. PR Base - this is a personal record keeping utility and will come on a "flippy" disk with documentation in DV80 format on one side. The documentation is quite lengthy and occupies 275 disk sectors. Be warned this program is very complex and a thorough reading of the docs is recommended prior to trying to use it. Full system required with Editor Assembler or equivalent.

2. Utilities - I have just received this disk and it contains some very good software, which I think is exceptional value. Titles on the disk are:

O(Zero) Slash - a resident program to place a / through zeros while programming in XB

Clock - a resident digital clock that appears to keep fairly accurate time.

Disk Manager - another resident disk manager.

Disk Sector Copier - for backing up your disks.

Read DV80 - to read this type of file to screen or send to printer.

Screen Dump - a resident screen dump that works very efficiently.

A full system is needed for all software on this disk

3. Printer Pictures - again I have only just received this disk, containing a program called MAX/RLE,(and I have no idea what the RLE stands for) but this is what the program does. Load MAX using Option 3 of Editor Assembler - program name is START - then load any of the 9 picture files on the disk and in a few moments you will have it on the screen. Pressing "P" will send the picture to the printer, while pressing any of the numeral keys will change background screen colors. Similarly keys A, D, E, F and C also change colors and by pressing various combinations you can get some real wierd effects. Pressing key "S" will allow you to transfer the picture files - D/F 128 format - to GRAPHX format and they can then be accessed through that software package. I am unable to say whether the reverse can be done as all key combinations so far don't appear to allow it. Naturally, for this disk you will need Editor Assembler, or equivalent, and 32K memory expansion.

4. A Disk of Games/Music - all requiring 32K and Extended Basic, all, unfortunately, too big to put on tape in their present format.

Billy Ball at the Hatchery - another in this series. This is a challenging game but pretty easy to learn how to play.







Floor 2 - is the follow on from Flooraway which was released on tape 1986/9A last month. This is a very difficult and challenging game and you will need to be precise in planning your moves.

Greensleeves - a great musical version of this popular tune with some very nice graphics.

Marilyn Quiz - this one has some trivia questions and ends with a sort of nude picture. Not in the best of taste I suppose but if you don't like it then you can easily delete it from the disk.

Cannonball Chess - this is a fairly complex game with the idea being to capture your opponents cannons, gold and food stockpiles. There are some quite good graphics together with lengthy instructions. I would imagine that to complete a game would take the best part of an hour.

5. More of the Same - again disk based only.

Axel Foley - this is the music from Beverly Hills Cop and it is extremely well programmed. Good foot tapping stuff!

Roman War - another complex and challenging game. Read the instructions on the disk at least twice before even attempting to play the game.

Texas Rangers - this is a keyboard manouvering game that is reasonably difficult to play. Instructions are included in the game.

Rising Sun (House of the) - this is probably the best piece of music every written on the TL. This version has been enhanced with the addition of graphics.

Zodiac - an astronomy quiz and tutor.

## On Tape

1. Tape No. 1986/10 will contain the following programs. The programs will also be on disk for those preferring that format.

The Game of 1 or 2 - in Basic - is a game for 2 players, the idea being to be the one to pick up the last stick. Full instructions are included in the program.

Asteroid Mission - in Extended Basic - is a moonlander type of game but you have to avoid the asteroids while trying to land on the pad. There are 3 pads - the smaller one attracting the most points.

Death Mobile - in Basic - manouver the Death Mobile to hit the screen object. The game gets harder the more times you hit it. Instructions are in the program.

Grandfather Clock - in Extended Basic - not a game but a screen clock which keeps fairly close time. You can set alarms and have chimes on the hour.

Patscram Mission - in Extended Basic - a multi screen space game which should keep you occupied for a fair while. Instructions are included as a separate program.

Lets Play Trains - in Extended Basic - this is one for the "littlies". By pressing keys you can change shape of one of the wagons, add a watertower and train station and make the train whistle. The young kids should get a kick out of it.

Radio Ratings Game - in Extended Basic - the object is to get more listeners to tune into your station than your opponents by putting the radio into the ears. A 2 player game with joysticks.

School Daze - in Extended Basic - instructions included but basically you have forgotten to do your homework and are being pursued around the school by 6 of your teachers. Good fun for the kids as they can use their own teachers names!

2. Tape No. 1986/10A will contain the following assembly games and is only suitable for those with 32K memory expansion, either inside or on an expansion card.

Kluuto Empire - when this game opens (after you press ENTER when it has loaded) you control the RED ship and have to destroy all the buildings by hovering over them and pressing the FIRE button on your joystick. Sounds easy, but the WHITE ship follows your every move - avoid it at all costs!

Mastermind - an old favourite color code game. The on screen messages are in French but the game is easy to learn. First select how many colors you want in your code, 2 to 10. With 2 there are 32 possible combinations, while if you select 10, there are 100,000. Good luck if you select 10. On the main screen the number of colors you have selected are displayed with a number beside each. To enter your guess, press the number corresponding to the color. Press ENTER when all selections have been made. Errors in your logic will be "rewarded" with musical bells. Have fun and enjoy the challenge.

Pinball - when loaded, press "8" to reset and start a new game, "0" to fire the ball and "1" and "=" to wick the flippers. Unlike Mastermind, no thinking needed but still good fun.

Video Vegas - a very colorful "poker machine" which should keep you out of the club for a while. I actually got 3 bars when test playing. Now if only it was real money. You can play with \$1, \$2 or \$3 "bets".

3. Character Definition Utility by Craig Sheehan. An excellent utility for designing screen graphics with a whole host of interesting features, like inverse characters, mirror image, rotate left and right and the ability to save your creations to tape. The program comes with a 13 page instructional document, a quick reference card and 2 lots of 8 demo files on the tape. This is really good material from a TISHUG member and I thoroughly recommend you get a copy. It will probably cost you a little more than the normal tape due to having to re-produce the documentation, but at any price it is well worth purchasing. Extended Basic is required

## Software News

I have received 2 disks of assembly utilities, one from the Lehigh User Group (USA) and the other from TICHUG (Canberra). Thanks in particular to Peter Hancock of TICHUG for sending me his work. I have given both disks to our resident assembly expert, Ross Mudie to have a look at and evaluate for me. Some of the programs on the Lehigh disk appear to have duplicated some of Ross' earlier efforts with disks Mudie 1, 2 and 3. It is likely that both these disk will be released very soon.

I have received a copy of Funlwriter 3.3D which appears to have been slightly updated. If anyone would like a copy, 1 disk double sided or double density else 2 disks please see me to order at the October meeting. I will copy the disk(s) for you and have them available at the November meeting. If you are unable to attend the meeting you may order by mail.

From Barry Gibbins of Wyoming (Gosford) comes a great Extended Basic game called QBIT. Barry tells me he designed the game using a black & white monitor. It sure comes up good in color, Barry. Watch for this game on tape and disk 1986/11.

Scott Becker of Albuquerque, New Mexico, was kind enough to send me a copy of his great disk cataloguing utility. Watch for it also at the November meeting.



Terry Johnsen is releasing his FORTH spelling checker as public domain software. Terry's was the only entry in the recently concluded FORTH competition. As configured presently the program will only work with 2 double sided, double density drives (Corcomp only, not Myarc). If you have the necessary hardware see me for a copy.

Ross Mudie has finished his newest disk MUDIE 86/5 which deals extensively with morse and more linking to assembly from extended basic. He has also supplied a version of his Connect 4 game for use with modems. This interesting software will be available at the November meeting.

Arto Heino has given permission to release his excellent Extended Basic games, Across the Tasman, Draughts, Tarkon, Tiptoe Tom and Roulette. These will be released on tape 1986/11.

### Problem Department

I wonder if anyone can tell me what's going on here. This problem has cropped up a few times recently when I have been backing up disks using the File Copy option of DM1000 - the version on Funlwriter 3.3 to be precise. The problem happens when I am copying from a SSSD disk to a DSDD disk and occurs when I am copying about the 4th disk to the new one. DM1000 finishes copying all the files I ask it to and then starts again, and, according to the catalog of the disk the duplicate files are there TWICE, i.e. not overwritten! To steal a phrase, "Why is it so?" To the original question - can someone tell me what's going on?

That's it for this month. More news next time.

*Terry*



These pictures, and others, throughout this issue of TND done with MAX/RLE, which will be released at the Club Shop in October.



## TECHO TIME



### MINI P.E. BOARD

by Peter Schubert

I have just completed the first Mini Expansion Box based on the design of the Poormans Disc System in last months issue.

The problem I could see facing these constructors was the soldering of many wires onto the connectors, and trying to accomodate an extra slot for an RS232 card.

So what I did was place all the connectors AND the power supply onto one printed circuit board. It has 2 slots, one for the disc-controller, and one for any other TI card, or the RS232 card I am designing. The cards must be 'fully buffered' type as there is no buffering on the flex-cable to the console.

At present the ribbon cable plugs straight into the console, however a small PC board will be made for the console end to provide the thru-way connection for other accessories such as the Modem/RS232 unit I have made. The flex-cable itself is a 40-way ribbon with crimp-on connectors at both ends--\*\*No Soldering needed\*\*. An extra bonus on the thru-way connector is the inclusion of my Load-Interrupt Switch design (published in an earlier TND), and I am presently debating whether to add a Data Bus Buffer chip to allow longer cables (extra cost about \$5 and 30mA from console power supply). At present my cable is tested to 75 cm length.

The power supply is capable of supplying 2 drives as well as the Mini PE Box and the whole unit adds very little to the cost of the Poormans Disc System.

Additional parts needed are-

- P.C. Motherboard
- Extra 60-way connector
- 2 crimp-on 40-way ribbon connectors
- 40-way ribbon cable
- 2 diodes and caps for neg supply

Size of my unit mounted into a box is 180W x 120H x 240D. Further details and possibly a kit of parts in next months magazine. In the meantime if you wish further details you can contact me on (02)358 5602, or send a reply envelope to :

Peter Schubert  
P.O.Box 28  
Kings Cross 2011

Full constructional details can then be posted to you as soon as they are available. For those who do not feel up to the soldering work I may offer pre-assembled boards and cables ready to mount into your own box soon. If you are in the Sydney region then come along to the next meeting of TISHUG at Burwood and see it on DEMO.







## REVIEW

USING THE FUNLWRITER V3.3

by Ben Takach

This soft-ware development was mentioned more than once in this publication. I was not anxious to use it or even to try it in the past. TI-Writer did do all I wanted, it was no point to switch. Thus I was unaware of its potentials.

The chance to try Funlwriter coincided with the aquisition of the Myarc 512 kB ramdisk. These two make a mighty powerful combination! Funlwriter elevated the humble 99/4A into the professional class. One DSDD disk filled to its capacity of 1280 sectors was just a few sectors short to exploit all the potentials of Funlwriter. I managed to get the 43 files on 1 disk using the Myarc 18 sector/track format. Naturally all non essential files, like the extensive documentation files, have been left off the working disk. My fully blown Universal work-horse fills 1295 sectors.

Files APATCH and UPATCH are not needed on the working disk, thus if these are left off, then the remainder would just fit on a standard 16 sector/track DSDD disk.

What can it do? Absolutely everything one is likely to wish. Once it is loaded into the Ramdisk it takes 3-4 seconds to access TI-Writer, or about 3 seconds to load the Viatel program. Switching from a given user's program to one of the Disk Managers (either DM1000 or Myarc) takes less than 3 seconds. It would not be possible to describe all the available options within the constraint of a review. The depicted flow diagram is more appropriate.

The flow diagram is very much simplified and is aimed to highlight the possibilities -so to say an outside view-, rather than an attempt to portray the actual program structure. The later would be far to complex for the purpose of this review. Tony McGovern comments on the flow diagram indicated an over emphasis of the 2 paths to the Central Menu Screen. In fact option 1 and 2 are an extension of the switch cycle to provide a direct access to the Editor.

Naturally not all of these program files are supplied with Funlwriter, however the "hooks" were there to hang on any program one wishes to incorporate. The published file directory listing, complete with language and size indication as well as comments, gives an overview of the files which make up the package.

I can assure you, it works very well and verry fast!

### THE MASTER DISK DIRECTORY.

The MASTER DISK currently is loaded with a massive 320736 bytes in 43 files. It has space for a further 36608 bytes. The disk was formatted by the new MYARC Disk Manager Supreme Level III disk control card in a non standard DSDD 18 sectors per track mode. The disk capacity of this optional format is 360 kB, which is equal to the capacity of 4 SSSD disks.

DSK1-DISKNAME= FNLWR3/3-2 FORMAT 18 DSDD  
 AVAILABLE= 143 USED= 1295 43 FILES  
 FILENAME SIZE TYPE P LANG. COMMENTS LENGTH

FILENAME	SIZE	TYPE	P	LANG.	COMMENTS	LENGTH
APATCH	10	DIS/FIX 80	-	ASS.	FNLWRASS/2.P	
A .:	33	PROGRAM		ASS.	TI-E/A FILE	8192
A ':	22	PROGRAM		ASS.	TI-E/A F.(MOD.)	5230
C99B	2	PROGRAM		ASS.	C-COMPILER	154
C99C	33	PROGRAM		ASS.	C-COMPILER	8192
C99D	33	PROGRAM		ASS.	C-COMPILER	8192
C99E	29	PROGRAM		ASS.	C-COMPILER	6994
CAT	12	PROGRAM		XBAS	BETA-CATALOG	2630
CHARA1	5	PROGRAM		ASS.	TI-WRITER	1024
CHARA2	5	PROGRAM		ASS.	FNLWR.CHAR.SET	1024
DEFAULT	24	PROGRAM		XBAS	FASTERM	5716
DICT1	242	DIS/VAR 80		DATA	SPELL-CHECK	
DICT2	213	DIS/VAR 80		DATA	SPELL-CHECK	

DM	4	DIS/FIX 80		ASS.	MYARC-DSK.MAN.	
DM1	55	PROGRAM		ASS.	MYARC-DSK.MAN.	13592
DM2	21	PROGRAM		ASS.	MYARC-DSK.MAN.	5048
DP	16	PROGRAM		ASS.	TI-DISKPATCH	3778
EAU	8	PROGRAM		ASS.	FNLWRT.	1660
EDITB1	33	PROGRAM		ASS.	TI-WRT.F.(MOD.)	8190
EDITB2	6	PROGRAM		ASS.	TI-WRT.F.(MOD.)	1198
FORMB1	33	PROGRAM		ASS.	TI-WRT.F.(MOD.)	8192
FORMB2	15	PROGRAM		ASS.	TI-WRT.F.(MOD.)	3514
FORTH	6	DIS/FIX 80	Y	ASS.	MYARC	
FORTHSAVE	39	PROGRAM	Y	ASS.	MYARC	9552
rWRMM	5	DIS/FIX 80		ASS.	FNLWR.	
FWSAVE	6	DIS/FIX 80		ASS.	FNLWR.	
LOAD	39	PROGRAM		XB/M	PGM.LOADER(FNLWR)	9557
MGR3	33	PROGRAM		ASS.	DM-1000	8192
MGR4	27	PROGRAM		ASS.	DM-1000	6424
MYDICT1	26	DIS/VAR 80		DATA	SPELL-CHECK	
QD	9	PROGRAM		ASS.	QUICK-DIRECT.	2036
SEEDGEN	7	PROGRAM		ASS.	SPELL-CHECK	1519
TECHIE1	33	PROGRAM		ASS.	FASTERM (MOD.)	8192
TECHIE2	33	PROGRAM		ASS.	FASTERM (MOD.)	8186
TERM1	33	PROGRAM		ASS.	FASTERM (STD.)	8192
TERM2	32	PROGRAM		ASS.	FASTERM (STD.)	7930
UPATCH	5	DIS/FIX 80		ASS.	FNLWR.	
UTIL1	33	PROGRAM		ASS.	FNLWR.	8056
UTIL2	33	PROGRAM		ASS.	FNLWR.	8191
UTIL3	5	PROGRAM		ASS.	SPELL-CHECK	848
VTXBW	25	DIS/FIX 80		ASS.	VIATEL (MONO)	
XB	2	PROGRAM		XBAS	XB-EXIT	29
XBDM	10	DIS/FIX 80		ASS.	FNLWR.	

I was unable to find any obvious program-bugs during the days long thrashing associated with adding and integrating the additional program segments. However I did not run any TI-Forth programs, nor was any assembly programming tried (I am only a learner of the later). One could say the program is almost idiot proof. Extensive error trapping is employed, leaving only a very slim chance for any inadvertent user's-error. Some small changes could actually make it idiot proof.

If one attempts to run DSK1.UTIL2 of the UTILITY option's Text mode then the system will lock up. Off course there is no reason to do this, -nor any excuse for doing it- nevertheless it could be removed from the list of possible sources of error for the sake of perfection. The documentation files do explain the conditions under which the program will lock up. Some of the above stated as well as other points which may not be apparent at the first glance are worth commenting.

### COMMENTS

1. Illegal or incorrect file-load commands entered in response to program prompts may result in a system lock up. It is not possible to recover from a lock-up by key stroke sequence. The system has to be switched off and on again. Users of the Navarone cartridge expanders can return to the main TI-Title screen by pressing the reset button. Data in the memory will be lost.

2. THE INVISIBLE BUG. Funlwriters LOAD program may be edited to suit the needs of the individual user. User programs and/or files may be included, and the printer default can also be changed. Beware of inadvertent insertion of "screen transparent" control characters, these will corrupt the program, which result in an instant lock-up during the initial loading sequence. Make a back up copy of the disk before any changes are made to any of the original program files. The program LOAD is a composite BASIC and Machine language file and inexperienced users should seek help if not sure of the eventual result of a program modification.

3. Attempts to run DSK1.UTIL2 from the Utility option will result in an immediate system lock-up. Utility pgm. files 1 & 2 are strictly for internal use by the program.





4. Assembly type user programs are generally named UTIL1 and UTIL2 if the program is longer than 8kB. These have to be renamed before copied to the master disk. Spell-Checker has only 1 UTIL program file, thus this may be renamed UTIL3. Programs with two or more should be renamed to follow the original sequence. Fastern and the modified Tishug/Techie version may be loaded through the Assembly-Utility option of FUNLWRITER (option 3: Program E/A). The program files of both is named UTIL1 and UTIL2. These can conveniently be renamed TERM1 & TERM2, respectively TECHIE1 & TECHIE2.

5. The SPELL-CHECKER program changes the screen colour after it is loaded. The resulting colour combination is not very satisfactory, but it can not be changed after it is loaded. Change the screen colour combination before loading this program.

6. There are many ways to access XBasic mode without removing the FUNLWRITER disk from its drive. One may use one of the unused options of the user's list (1 to 5). One could simply be calling for a non existing program file. The subsequent break and error message places the system automatically in the XBasic mode. A more elegant solution is by a one liner:  
10 CALL CLEAR

This is filed by the name XB. The LOAD program will run this, and will branch into XBasic. It involves less key strokes and the master disk may be left in drive 1. Off course FUNLWRITER need to be rebooted again if it is so desired.

7. The Utility option 4 (Load/Run E/A) may be used to load the Black & White version of the E.Crago VIATEL program (VIXEW). The colour version (VIX) will not load. It will return a MEMORY OVERFLOW error, and the system has to be rebooted, because part of the memory will be overwritten. I was able to load and run VIX using the "invisible" UTILITY options (6 or 7 or 8), however the 43 program files of the FUNLWRITER package residing in Ram-Disk were hopelessly corrupted. Off course this is of no consequence if FUNLwriter is run from one of the conventional disk drives.

This is not a short-coming of the program, but rather highlights the limitation of the system's architecture. Lengthy assembly language programs will inevitably cause the corruption of any other program which is resident in RAM. In such cases the Ram-Disk has no advantage. It is worth noting that VIX is also too long for XB, although it loads and runs ok through BEAX-CC.

8. Tony McGovern mentioned that further improvements have been incorporated in the package since June 86 and apparently some of my comments are not applicable to the current issue. The package as is will also load from TI-WRITER, E/A and the Mini Mem modules. This list is augmented by the current issue to include E/A and Mini Mem Basics, E/A Load & Run, the Myarc disk controller card (CALL LR) as well as the Myarc and Corcomp disk managers with no module at all in the console.

9. The package embodies many improvements not contained in the original program segments, these are too numerous to list in a review which is subject to space limitation. Nevertheless I pick a few at random. FUNLWRITER remembers the drive it was loaded from, and will continue to operate from that drive. TI-WRITER's Show Directory function is much improved, albeit it still lacks the print-out option. A similar disk directory may be invoked by AID (FCIN 7). This feature is almost global.

### POSSIBLE IMPROVEMENTS.

There is not much more one can wish for. An almost instant access to 350kB of program plus the expanded RAM capacity and two additional disk drives is equal to the power of the most respectable big Blue selling for three times the price of this fully expanded 99/4A. Some cosmetic embellishing on the other hand would make this remarkable package even more desirable. I miss the option to cross over from the E/A-TI-Write- environment to the User's side without the need to QUIT and reboot. An additional switch option to step into User's list would be handy. This is not much problem if the program is in the Ramdisk, but quite a delay if the disk is loaded from a drive. The documentation provided is very comprehensive. These are on seven DIS/VAR-80 files, which may be printed out or viewed on the screen. The print-out fills 28 quarto pages.

Summing it up, the Funlwriter V3.3 is one of the most useful program development, which enables the user to run the 99/4A using the XB cartridge only, without the need to switch to E/A, Disk Man., TI-Writer or the TE-II module. Further the package offers improvements which are not incorporated in the original modules mentioned.

The package is distributed as a FAIRWARE. This concept is yet another attempt to overcome the problems of software piracy. Personally I do not think it will solve it, but this argument should be a subject of another chapter.

The authors did a splendid job, my congratulations to Tony and Will McGovern.

A simplified flowchart of the FUNLWRITER package is provided on the next page...



## REGIONAL REPORT.

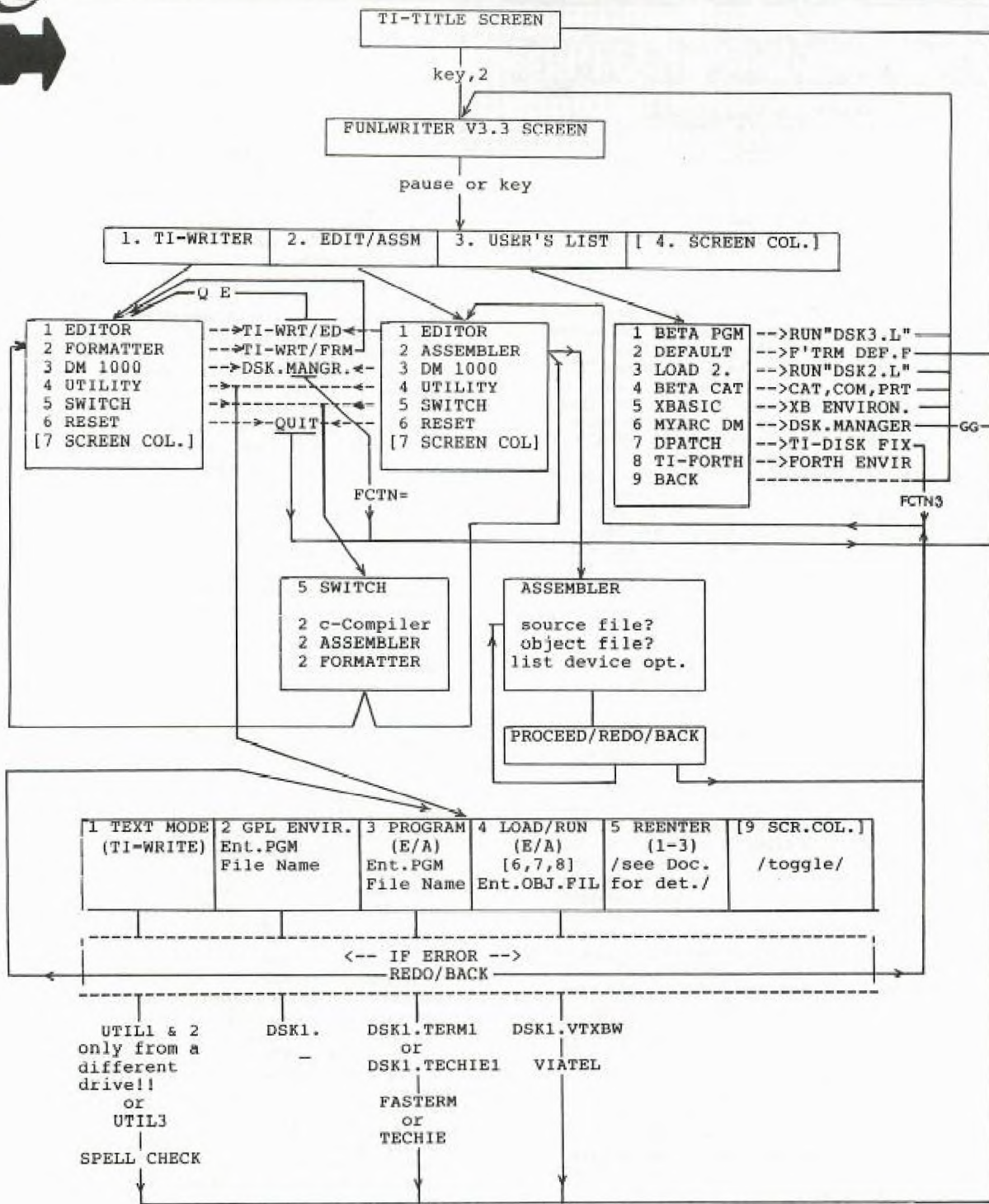
### CENTRAL COAST REGIONAL GROUP

This group will hold our annual full day tutorial workshop on Saturday, 18 October, 1986, commencing at 10.30am. The venue will once again be the Gorokan High School, Goobarabah Avenue, Gorokan. Bring your own computer and have a great fun day learning and chatting to fellow TI'ers. To cover costs of hall hiring there will be an entrance fee of \$2 per family with some refreshments being provided. For further details contact Central Coast Regional Group Co-ordinator, Russell Welham on (043) 92-4000.



Regional Report continued Page 17





The simplified Flowchart of the augmented program package. Items enclosed in [ ] are "invisible" options. These are available to the user but not shown on the screen.





# DID YOU EVER WONDER WHY?

Did you ever wonder why it was difficult to get a square or round shape on the screen, and if you managed to do that why it printed out rectangular or oval? This is particularly annoying in TI-LOGO and GRAPHX.

There is a logical reason for this if we look at the way the picture is put on the screen. The Video processor produces a picture using 256 dots across the screen and 192 dots down the screen. This corresponds to 32 columns and 24 rows of 8 dot by 8 dot cells. If each of these cells (or dots) was an exact square, then the area produced by the video processor would have an aspect ratio of 3 units to 4 units. This is the aspect ratio of television screens, so to get the cells to be an exact square, the picture would have to fill the whole screen. We all know that it does not, but in fact has large gaps at the top and the bottom. Why did TI do such a foolish thing? In fact TI did a reasonable job for the American television system, as can be seen if you examine closely a picture of a screen produced there. Their picture almost fills the television screen in the vertical as well as the horizontal direction. To understand why there is a difference, we must consider how the picture is produced on the screen.

A television picture is made up a number of horizontal lines, upon which the dots are produced as the electron beam moves across the screen. That is, there are a fixed number of lines and hence dots, in the vertical direction. In the horizontal direction the signal can change at any time so that the number of dots is dependant on other factors like bandwidth. Because of the fixed number of lines in the vertical direction, each dot must take one line and cannot use parts of a line. If we want 192 dots in the vertical direction, we must use 192 lines for them. The American NTSC system has 262 lines (525 with interlacing), and the 192 dots use 73% of the height of the picture. If the horizontal signal is arranged to also use 73% of the total width (it actually uses 256 out of 342 dots which is 75%), the picture would come out with square cells. Television sets are normally adjusted so that the edge of the picture does not appear on the screen so that this 73% would probably fill most of the screen.

In the PAL system there are 313 lines (625 with interlacing), and so the 192 dots use only 61.5% of the height, while the horizontal signal is unchanged. This gives the rather squashed effect we are so used to, and makes it difficult to get the same effect from a screen dump to a printer as is on the screen. The only solution is to reduce the width and increase the height of the television, which would limit its usefulness as a normal receiving set.

Did you ever wonder why clock programmes and timers using the CALL SOUND routine run slow? The reason is that the timing for the processor comes from the interrupts generated by the video processor. These are generated at the end of every vertical scan of the screen and

occur normally at approximately the frequency of the AC mains. In America and some European countries the frequency is 60Hz, while here it is 50Hz. This means that if the software counts the number of these interrupts to determine when one second has elapsed, we must count 50 interrupts while in America they must count 60. If the software counts 60 interrupts at a frequency of 50Hz, it will take 1.2 seconds and not 1.0 seconds. The CALL SOUND duration timing is set up to be correct for 60Hz and so will be 20% longer than expected here.

It is interesting that the RS232 requires a clock to derive the timing for the transfer of data in serial mode. The clock signal that is provided is that of the 9900 cpu clock, nominally at 3.0 MHz. This is divided down by the necessary number to provide the correct frequency for the RS232 operation. The software that does this has the choice of two numbers for each required frequency, and the ratio of these pairs of numbers is 5 to 6. This must be a relic of the 99/4 days, when the video processor was the same for our computer and that sold in America. To provide for both 50Hz and 60Hz picture scanning rate the processor must have been slowed down for us. The software switch for selecting which clock is in use is in the system ROM.

Did you ever wonder about the peculiarities of the character sets of BASIC? Why do the character codes for BASIC have 96 added to them and why only the characters up to 159 for BASIC and 143 for Extended BASIC can be used? The answer to these questions is contained in the Video processor chip and the way it uses its memory for the screen and character definitions. The VDP memory is used for all the screen functions and also for storage of programmes and for buffer areas for input and output (Disk, cassette, RS232). This makes it a very busy memory in a system without memory expansion, and it is important that no space is wasted. With this in mind let us look at the requirements of the video processor in regard to the screen functions. The normal mode that is used in BASIC is called graphics mode. This sets the screen up as 24 rows and 32 columns of cells containing 64 dot positions each (8\*8). Each of these cells can be made to show a shape by storing a code in an area of memory called a screen image table. The position of the screen image table is defined by a number stored in VDP register 2, and is given by that number times 1024. The screen image table takes up 768 bytes of memory. The code which defines the shape to be shown in a cell can be any number from 0 to 255. Each of these possible codes could be defined as a different shape by storing 8 bytes with a 1 for a dot and a 0 for no dot in the respective 64 bit positions. This means that to define a different shape for each of the 256 possible codes would require 2048 bytes of memory. The area of memory used for this pattern descriptor table is defined by a number stored in VDP register 4, and this number is multiplied by 2048 to give the first address of the pattern descriptor table. So far we would need 3072 bytes to give the full capabilities of the graphics mode. In this mode the colours of the dots and the background can also be specified, but only in groups of 8 pattern codes at a time.





There are 16 possible colours and these can be stored in 4 bits with the dot colour and background colour in one byte. This means that 32 bytes are needed to specify the colour combinations for the 256 possible patterns. The colour table memory area is defined as 64 times the value stored in VDP register 3. In graphics mode sprites can be used. It is possible to define 32 sprites at a time, and this is done by the sprite attribute table. Each sprite requires 4 bytes, 2 for the position of the sprite on the screen, 1 for the pattern code, and 1 for the colour and early clock attribute. This means an area of 128 bytes is needed for the sprite attribute table, and its location is defined as 128 times the number in VDP register 5. The sprites also need an area for their shapes to be stored. Although there are only 32 sprites possible at a time, the pattern code can be from 0 to 255, and each sprite needs 8 bytes for standard size and 32 bytes for double size. This means an area of 2048 bytes should be set aside for the sprites although only 1024 bytes could be used at any one time. The sprite pattern area is defined as 2048 times the number stored in VDP register 6.

If we add up the amount of memory required to allow the full capabilities of the VDP processor it comes to 4000 bytes or about 25% of the total memory. Rather than allow this, TI decided to minimise the use of memory for the screen functions and yet still provide most of the functions of the VDP processor with a few limitations. Let us examine what has happened in BASIC and Extended BASIC.

**BASIC:**

Screen image table .... : 0 to 767  
 Colour table ..... : 768 to 799  
 Pattern descriptor table: 0 to 2047

Sprites are not used in BASIC, but their attribute table and pattern table addresses are still set up.

Sprite attribute table : 768 to 895  
 Sprite descriptor table : 0 to 2047

As you can see some of the tables overlap and the maximum memory used is 2048 bytes. To solve the problem of the overlap, the patterns we normally want to use, starting at ASCII code 32, begin at address 1024, that is for code 128 or 96 added to the ASCII code of 32. This means that the ASCII code of the last pattern we can use is  $255 - 96 = 159$ , as it says in the BASIC manual. There are some patterns defined for codes 126 and 127 (ASCII codes 30 and 31) for the cursor and screen surround characters. The first half of the colour table is not used, while the second half determines the colour of our characters and any other patterns we define. There is a conflict between the colour table and the sprite attribute table which would make it difficult to use sprites.

**Extended BASIC:**

Screen image table .... : 0 to 767  
 Pattern descriptor table: 0 to 2047  
 Colour table ..... : 2048 to 2079  
 Sprite attribute table : 768 to 895  
 Sprite descriptor table : 0 to 2047


The sprite motion table uses addresses 1920 to 2047 to store the information needed by the system to calculate the position of the sprites from their velocities.

With the 96 added to the character code, the first pattern for the cursor starts at  $126 * 8 = 1008$  for both BASICs. Here is a table of the VDP memory which attempts to summarize the information above.

Address	BASIC use	Ex BASIC use
0 to 767	Image table	Image table
768 to 799	Colour table	Sprite Attribute table
800 to 879	Crunch buffer	
880 to 1007	System area	System area
1008 to 1919	character 30 to character 143	character 30 to character 143
1920 to 2047	character 144 to character 159	Sprite motion table
2048 to 2079		Colour table

By this means we are given the maximum memory for storing our programs and data in the VDP RAM.

## 'tip corner'



If you have some early TI or other Command Modules which only have the provision for RS232 (serial) menu choice for a printer and your printer happens to be a PIO (parallel) input only, there is a way you can generally get it to work:

- (1) When you set up your files, tell the computer that you are not using a printer, and create your data files as if you weren't using a printer.
- (2) During the review of the file after its creation, then put in a totally false printer call out (like RS-232/9) and the computer will then give a DEVICE NOT FOUND error message. At this point, enter PIO and generally the computer will access the parallel printer.

The procedure might vary some depending on the module, but the important thing is to avoid entering a printer identification when you are entering the data, putting it in later (falsely) during the recall and review.

Kent Maxwell





TIPS FROM THE TIGERCUB  
for TISHUG  
#25

Copyright 1985

TIGERCUB SOFTWARE  
156 Collingwood Ave.  
Columbus, OH 43213

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

It seems that I had better clear up a few misunderstandings. The "freeware" offers I have mentioned in past Tips are NOT available from me - send your disk and returnable mailer AND RETURN POSTAGE to the author of the program.

And, my copyrighted Tigercub Software programs are NOT freeware. They can only be legally obtained by mail order from me - if you copy them from anyone else, you are stealing!

As for the programs which I write and publish or distribute without copyright, they are also not Freeware, they are FREE. I don't want to be paid for them, and I don't think anyone else should be paid for them.

Some users groups are putting my copyrighted programs, and those of other programmers, in their software library, "for use but not copying" or "for review and evaluation only". Who do you think you're kidding? I know I won't sell any software to members of pirate clubs, so why should I support them?

If you didn't solve the Long Division Puzzle in Tips #24, try dividing 230709 by 835. As for the solution to the Tigercub Challenge, it was right on the same page! Try creating those DATA statements with the LINEWRITER routine. I don't know why it works, but it does.

I've been asked to print more information on the "program that writes a program". I don't have room for a detailed account, but here are the basics. If you tried my TOKENLIST routine

in Tips #23 you already have a list of the token codes you will need.

I won't go into the way that the computer squishes a program line number into only two characters, but you can accomplish it with `DEF L$=CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256))`, where LN has been predefined as the value of the line number.

If you need to refer to a program line in a statement, as in `GOTO 500`, use `DEF R$=CHR$(201)&CHR$(INT(RN/256))&CHR$(RN-256*INT(RN/256))`, RN being the line number.

To print a statement or command, simply print its token character. For instance, the token for `DATA` is 147, so you would print `CHR$(147)`. Note that all the punctuation marks used in programming, such as ( and +, are also represented by token codes which are NOT the same as their keyboard ASCII value.

To print a variable name, either numeric or string, just enclose it in quotes, "A" or "A\$".

To print a value, or an unquoted string (as in a `DATA` statement), or the word which follows a `CALL`, you must print `CHR$(200)` followed by a token giving the number of characters to follow, such as `CHR$(5)` for a 5-character word such as `CLEAR`, then the value in quotes. For instance, the token for `CALL` is 157, so `CALL CLEAR` is `CHR$(157)&CHR$(200)&CHR$(5)&"CLEAR"`.

You can simplify that by predefining `DEF U$(V$)=CHR$(200)&CHR$(LEN(V$))&V$`, and then simply print `CHR$(157)&U$("CLEAR")`.

A quoted string is handled in the same way except that it is preceded by token 199 instead of 200, so you can predefine it as `DEF Q$(V$)=CHR$(199)&CHR$(LEN(V$))&V$` - the computer will take care of the quote marks.

Each program line must end with `CHR$(0)`, and the last record you print must be `CHR$(255)&CHR$(255)`.

A `MERGE` format file is `D/V 163`, so open the file with `OPEN #1:"DSK1.MERGEFILE",VARIABLE 163`.

Don't print more than 163 characters in a record or the computer will blow its mind! You can print multiple-statement `XBasic` lines, but be sure to use the double-colon token `CHR$(130)` as the separator,

not two of the `CHR$(181)` colon tokens.

Any errors you make will usually not show up until you try to `MERGE` or use the program you have created. `I/O ERROR 25` means that you forgot the final `255 & 255`; `DATA ERROR` or `SYNTAX ERROR` probably means that you left off a `CHR$(0)` or gave the wrong count of characters after `CHR$(200)`.

Here's a bit of psychedelic blues - -

```
100 REM - FRANKIE & JOHNNIE
    by Jim Peterson
110 DIM S(12)
120 CALL SCREEN(2)
130 FOR R=1 TO 12
140 CALL COLOR(R+1,1,1)
150 FOR T=R TO 25-R
160 CALL HCHAR(T,R,32+R*8,34-2*R)
170 NEXT T
180 NEXT R
190 DATA 262,294,311,330,349,392,440,494,523,587,40000
200 FOR N=1 TO 11
210 READ S(N)
220 NEXT N
230 FOR J=1 TO 110 STEP 2
240 CALL COLOR(A+1,1,1)
250 READ T,A
260 CALL COLOR(A+1,A+2,A+2)
270 FOR TT=1 TO T
280 CALL SOUND(-999,S(A),0)
290 NEXT TT
300 NEXT J
310 RESTORE 330
320 GOTO 230
330 DATA 2,1,2,2,2,4,2,7,1,1,1,1,7,2,6,4,4,2,1,1,1,1,13,1
340 DATA 2,1,2,2,2,4,2,7,1,1,1,1,7,2,6,4,4,12,1
350 DATA 1,11,3,1,2,5,2,6,2,7,2,9,1,11,1,9,2,10,4,7,1,9,1,11,7,9
360 DATA 4,7,2,8,2,9,1,11,3,9,1,11,1,9,4,8,2,7,6,6
370 DATA 4,4,1,11,3,4,4,4,3,16,2,1,11,4,7,2,6,4,7,4,6,20,1,8,11
```

You can too have a blank space in your disk filenames! Just use `FCTN V` for the blank, instead of the space bar. You can even have a diskfull of 10 programs with invisible filenames consisting of 1 to 10 of those `FCTN V`'s.

However, those invisible characters can do strange things when you list your disk catalog to a printer.

If you want to `INPUT` a string with leading and/or trailing blanks, just enclose the whole works in quotation marks. Try this -

```
100 INPUT A$ !type TEST
110 PRINT A$;LEN(A$)
120 INPUT A$ !type " TEST "
130 PRINT A$;LEN(A$)
140 GOTO 100 !you can even
    input a blank string of 136
    characters
```



I really shouldn't tell you this, but if you want to make it difficult for someone to LIST your program, just insert a garbage line, every 5th line or so until you run out of memory, consisting of REM followed by 4 or 5 lines of random characters typed with the CTRL key held down.

Here's a program that can actually read your mind!

```

100 CALL CLEAR
110 PRINT "TIGERCUB MIND READER PROGRAM":
120 PRINT "I'll bet you a dollar I can guess what you are thinking.":
130 GOSUB 440
140 PRINT "And I'll bet another dollar I can tell if whether you are thinking is correct.":
150 GOSUB 440
160 PRINT "And I'll bet another dollar I'm right BOTH times.":
170 GOSUB 440
180 PRINT "And I'll bet one more dollar I can guess what you'll be thinking a minute from now.":
190 GOSUB 440
200 PRINT "OK....":
210 GOSUB 480
220 PRINT "You're thinking that a computer can't possibly know what you are thinking.....right?":
230 GOSUB 480
240 PRINT "So I told you what you were": "thinking.....right?":
250 GOSUB 480
260 PRINT "You owe me a buck.":
270 GOSUB 480
280 PRINT "And you're absolutely right..I can't read your mind.":
290 GOSUB 480
300 PRINT "So I told you correctly that": "what you were thinking was": "correct.....right?":
310 GOSUB 480
320 PRINT "You owe me another buck.":
330 GOSUB 480
340 PRINT "So I was right BOTH times...right?":
350 GOSUB 480
360 PRINT "That makes three bucks you owe me.":
370 GOSUB 480
380 PRINT "And now it's a minute later": "and you're thinking you've": "been played for a sucker....": "...right?":
390 GOSUB 480
400 PRINT "...so you owe me four bucks.":
410 GOSUB 480
420 PRINT "NEVER NEVER bet against a computer!!"
430 END
440 PRINT "Want to bet? Type Y(yes)":
450 CALL KEY(3,K,ST)

```

```

460 IF (ST=0)+(K<>89)THEN 450
470 RETURN
480 FOR D=1 TO 800
490 NEXT D
500 RETURN

```

Since the manual doesn't mention it, some folks don't know that you can use IMAGE and PRINT USING for output to the printer. Try this -

```

100 OPEN #1:"PIO"
110 INPUT "NAME? ":N$
120 INPUT "AMOUNT? ":A
130 PRINT #1,USING "#####.###":N$,A
GOTO 110

```

Of course, you could also add a line -

```

105 IMAGE "#####.###"

```

And change line 130 to

```

130 PRINT #1,USING 105:N$,A

```

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a returnable mailer WITH RETURN POSTAGE, at least - and I hope you'll also include something more!

Attention, assembly programmers! Fred Hawkins of the Lehigh UG is trying to coordinate a project of documenting the operating system by breaking the console ROM down to pages of 256 bytes so that each individual or group can work on just one page. Only those who participate will share in the results! All this is far beyond me, but if you want in, send an SASE and a SSSD disk with return postage and mailer to Fred Hawkins, 1020 N 6th St, Allentown PA 18102 - soon!

If you have a program on disk which is so long that you must type CALL FILES(1) before you can load it, add several program lines to it consisting of REM and any key you want to hold down for 5 lines. Then SAVE it back to the disk; it will now be in INT/VAR 254 format and will load without CALL FILES(1). If you then need sometime to make a cassette copy, just delete those lines and SAVE it back to disk again.

If a program loads, but gives you a MEMORY FULL IN LINE ... when you try to run it, it has used up all available memory while reading DATA into arrays or performing other internal calculations. If it runs for some time and then gives you the MEMORY FULL message, it is because you have repeatedly jumped out of a FOR...NEXT loop with an IF...THEN...GOTO before the loop is completed. This rarely happens but it can, especially when you repeatedly jump out of the innermost of several nested loops.

## MEMORY FULL

Jim Peterson

REGIONAL REPORT continued from page 12

## LIVERPOOL REGIONAL NEWS

The next meeting will be held on Friday, 10 October, 1986 at Cyril Bohlson's home, 4 Madeline Avenue, Northmead commencing at 7pm. Telephone Cyril on 639.5847 for further information if necessary.

At the last meeting, Arto Heino demonstrated his musical keyboard interface and software which is still under development at this time.

Richard Stanford of Camden has updated his disk cataloguing program, "DISKCAT" to now over-ride previous problems associated with program names on disk using control characters, e.g. the MDI disk. Copies of this updated version are available from the Liverpool Regional Group.

Robert Peverill demonstrated his latest cataloguing program. Robert's program will be available through the club shop shortly.







OLD MODULES - BUT STILL GOOD FUN.

By Terry Phillips - Software Co-Ordinator

Sydneysiders, at least, will remember the "big wet" early in August when it seemed like it was going to rain forever. Well on one of those wet nights I pulled out some of my old modules, games I hadn't played for some time, just to see if they were still as enjoyable as they were when I first got them. They were! So from a nostalgic viewpoint here's a short review of three that have brought me a great deal of pleasure. I doubt if you will now find them in any stores, but you may be lucky and pick them up, assuming you like the following descriptions, at the big auction day coming up in October.

First up .... Video Games 1. This was the first module I ever purchased, way back in May 1981. As I recollect the asking price was \$50 or thereabouts. At that time it was the only piece of software that I owned. Anything else I wanted to run was typed in out of the users manual. I didn't even own a tape recorder at this stage. How things have changed. Here I am sitting at a full system plus disk file boxes bulging with programs. Ah well! I guess you have to start somewhere. Anyway back to Video Games 1 - why the "1" I don't know. I have never seen a "2". The module contains 3 separate games - Pot Shot, Pinball and Doodle.

Pot Shot - as the name implies it is a shooting game. The idea being to place your sight on either white rabbits, flying ducks or deer and fire. Rabbits are easy to hit and consequently are worth only 1 point, deers are a little harder and are worth 2, while ducks are a bit of a challenge and earn 5 points per hit. Your "gun" is a six shooter but if you fire to fast it jams. Game length can be anything you like up to a maximum 120 seconds I suppose it's good harmless fun, and no worse than the myriad of other shooting games, but then little kids might find the idea of shooting defenceless animals a bit much. Graphics are a bit on the chunky side but very colorful.

Pinball - this to my mind is the best game in the package. An absorbing game to play and quite difficult to attain a big score. One reason for this is that occasionally it cheats and gobbles up one of your supply of 5 balls through an unseen hole at the top of the graphics screen. You only have 1 flipper, but this adds to the challenge. To score really big points you must light the bumpers - 3 sets - and then aim to bounce the ball around them for as long as possible. I have hear of some lucky players getting the ball stuck among the bumpers and getting up over 1 million points. Not me though!

Doodle - has many options, but basically there are two main scenes. 1 player versus the computer and a 2 player options enhance the games versatility. The idea is to outmanouvre you opponent and trap him so that he cannot move. As you move around you leave behind a trail upon which you cannot go back over. You can select obstacles or just a plain screen. Guaranteed you will get hours of fun out of this game.

Next ... A-MAZE-ING ... I can't remember how long I have had this module but it must be nearly as long as Video Games 1. The game has the honor of being the only one my wife will play on the computer. There are so many options and variations in this one that I doubt if anyone would every get tired of playing. According to the instructional booklet there are over 5000 possible mazes all with varying degrees of difficulty. You can have Escape Mazes, Cheese Hunt Mazes either competitive or cooperative, mazes with obstacles, invisible mazes, in fact you name it, the variation is almost endless. The basic task is to guide your mouse through the maze in the quickest time possible. Very mean cats are another hazard that the player can select. The cats can be programmed to pounce any where within the maze, adding further to the challenges of this great game. One option I have never had the opportunity to use on this module is to print the mazes to a printer. For this option you need the old TI Thermal Printer. I doubt if there are any of these around now.

Lastly ... Hustle ... an early Milton Bradley release for the TI. 3 options are available, Hustle, Snakefight and Blockade. You can either play against the computer or a friend. All options are basically similar but my preference is Snakefight where you guide your arrow snake around trying to avoid boxes that contain random numbers of points and making the other player crash into them. All the time you have to avoid the second players snake for if he hits you he scores all the points on display. Quick joystick action is essential if you want to get a big score.

Well there you have it. I certainly enjoyed rekindling my fondness for these games. If you have them and haven't played them for a while, why not do it now?

**OLD MODULES  
BUT STILL  
GOOD FUN!**

MUSIC MAKER.

TOUCH TYPING TUTOR

PARSEC

A-MAZE-ING

BLASTO

MIND CHALLENGES

HANGMAN

VIDEO CHESS



## LINKING EXT'D BASIC-ASSEMBLY

### #7 LINKING IT. #7

BY ROSS MUDIE.

--\*-- October 1986 --\*--

#### EXTENDED BASIC & ASSEMBLY MORSE PROGRAM.

This article introduces a morse code program and details a special KEY SCAN program which was developed to meet exacting requirements of a real time application. The Key Scan directly interfaces the keyboard using CRU bits. The discussion on CRU bits may provide readers with a better understanding of the operation in assembly of part of the CRU Input / Output.

I would like to hear from user group members if this type of article is fulfilling a need.

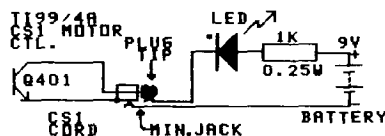
I have recently developed a keyboard to morse code program suitable for the console with a 32K memory expansion. This program should be popular with TI99/4A owners who hold novice or full amateur radio operator licences. The program allows the message to be typed on the keyboard and simultaneously be transmitted. The morse output can be heard in tone form from the computer. By connecting the CS1 motor control to the transmitter, the computer can directly key the transmitter, providing that the computer - transmitter interfacing is compatible.

The main program is written in assembly but I have used extended basic to make easy the task of allowing versatile options which can be edited and resaved as the user's requirements dictate. The cassette version of the program loads the assembly in CALL LOADS first and then loads the extended basic program, allowing loading from tape on a 32K expanded console. The disk version loads the extended basic first and then loads the assembly with CALL LOAD("DSK1.XBMO").

#### CRITICAL PROGRAM REQUIREMENTS.

In the assembly part there are multiple tasks which seem to be operating simultaneously. The KEY SCAN is a fast task which must be responsive to the keyboard. The sending of the morse code is a slow task which must maintain pulse duration accuracy for good morse to be produced. When a key is pressed it must be displayed on the screen and placed in a buffer which allows the operator to get ahead of the morse code output. The timing is critical in the program which maintains a speed accuracy of better than 1% at the speeds which divide evenly into the cycle time. These speeds are any integer number value between 5 and 40 which divide evenly into 840, (the magic number).

In development of the program I found that the inbuilt KSCAN utility gave a variable execution time, dependant on the key pressed and for how long it was pressed. To overcome this problem I wrote my own KEYSKAN program which had virtually no difference in its execution time



regardless of which key was pressed. This part of the program uses the CRU bits SBO, SBZ, and TB to set up & scan the keyboard. (I will discuss SBO, SBZ & TB later in this article).

The program also had to output a control line for the radio transmitter. For this purpose I chose to use the CS1 motor control. It would have been more convenient if I had used the CS2 motor control for this, however the beige coloured consoles, even if they have a double ended cassette control cord, are not fitted with the appropriate components in the console main PCB.

The MORSE program including the source file, extended basic program and this article will be available from the TISHUG shop on disk MUDIE 86/5 for the usual media fee. The cassette version will be a user program only, without any source file or article. To load CASSMORPT1 which is part 1 of the cassette based program on a disk system CALL FILES(1) must be invoked.

#### CASSETTE MOTOR CONTROL.

Controlling the cassette motor output is simple as the following example shows:

```
CLR R12
SBZ 22      Turns CS1 motor control on.
SBO 22      Turns CS1 motor control off.
```

If writing a program for a black TI99/4A console then SBZ 23 & SBO 23 will control CS2 motor off & on.

The following sample program turns the CS1 motor control ON & OFF at a slow rate to demonstrate the method.

```
DEF GO                      Program entry point
WS BSS 32                   Allocation of register work space
GO LWPI WS                   Use the allocated workspace.
CLR R12                      Establish correct base for CRU bit
LOOP0 SBZ 22                 Turn off CS1 motor control
LI R2,60000                  Value for delay loop timing
LOOP1 DEC R2                 Count down
JNE LOOP1                    Finished yet ?
SBO 22                       Turn on CS1 motor control
LI R2,60000                  Value for delay
LOOP2 DEC R2                 Count down
JNE LOOP2                    Finished yet?
JMP LOOP0                     Go and do it again.
END GO                       Auto start once the program is loaded
```

To monitor the operation of the switching at the CS1 motor control, I used a Light Emitting Diode, a 1K ohm resistor and a 9 volt battery. These components are connected as shown in figure 1 to a suitable miniature (2.5mm) jack socket which plugs onto the cassette cord.

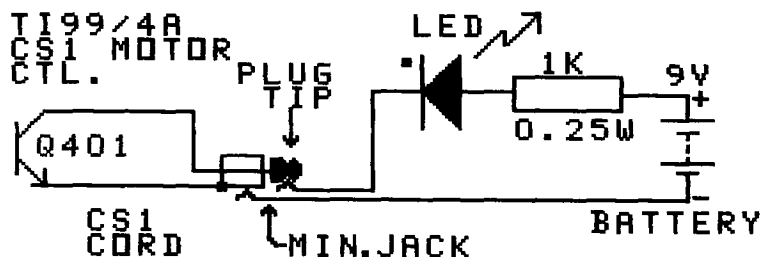


Figure 1.

Once a CRU bit has been turned ON or OFF it will stay that way until it is swapped to the other state.



## KEYBOARD SCAN.

To scan the keyboard there are three outputs from the TMS9901 communications controller labled P2, P3 & P4. These are CRU bits 18, 19 & 20 and their function is keyboard select bit 2, bit 1 & bit 0. These 3 outputs are connected to a binary to octal decoder. What this means is that by changing these 3 TMS9901 outputs in a binary number pattern there will be one of eight outputs ON at a time out of the octal decoder. If you have a technical manual for the computer console, look at the drawing which shows the outputs to the joystick and keyboard. Six of the outputs from the octal decoder drive the keyboard whilst the other two outputs drive transistors Q300 & Q301 to the Joystick Port.

There are 9 inputs from the keyboard which connect to the computer via the TMS9901 controls INT3 to INT10 & P21. In the assembly program these are tested by the TEST BIT instructions TB 3 to TB 10 & TB21.

Table 1 may help you to understand the relationships of the CRU bits set and scanned to test for operated keys on the keyboard.

CRU OUTPUTS	/-----\			TB	INPUTS			-----\		
20 19 18	7	8	9	3	4	5	6	10	21	
0 0 0	FCTN	SHFT	CTRL	=	SPACE	ENTER			A/L	
0 0 1	2	S	W	.	L	O	9	X		
0 1 0	3	D	E	,	K	I	8	C		
0 1 1	4	F	R	M	J	U	7	V		
1 0 0	5	G	T	N	H	Y	6	B		
1 0 1	1	A	Q	/	;	P	0	Z		
Joyst1										
1 1 0	UP				FIRE LEFT	RIGHT DOWN				
Joyst2										
1 1 1	UP				FIRE LEFT	RIGHT DOWN				

Table 1.

To scan the keys in the first row of table 1, (FCTN to A/L) first set register 12 to zero as follows:

```
CLR R12
```

To set the CRU outputs to Zero's, use Set Bit Zero as follows:

```
SBZ 20 )
SBZ 19 ) Refer to table 1
SBZ 18 )
```

The following examples assume these BYTES:

```
B13 BYTE 13 Ascii key values in decimal
B71 BYTE 71
```

To test if the ENTER key is operated:

```
TB 5 If the key is not operated then the
JEQ NOTOP Equal status bit is set
MOVB @B13,@>8375 Store contents of B13 in >8375
```

NOTOP

Another example, to test for the G key:

```
SBO 20 )
SBZ 19 ) Refer to table 1
SBZ 18 )
TB 8 )
JEQ NOTOP
MOVB @B71,@>8375 Store contents of B71 in >8375
```

NOTOP

If you have made it this far through the article you may be ready to understand a real program.

\* To allow this program to work under extended basic,  
 \* Remark out the REF VSBW,VMBW & enable the EQUates  
 \* for VSBW & VMBW. Change GO LI RO,>OC00 to >0700 .  
 \* Enable AI R1,>6000 in DISP after MOV B @KY,R1 .  
 \* Save as DSK1.KK . Assemble as DSK1.KO . Enter ext'd

\* Basic, type CALL INIT then CALL LOAD("DSK1,KO") ,  
 \* when loaded type CALL CLEAR then CALL LINK("GO") and  
 \* try with keyboard or joystick.

```
IDT 'KKmudie' Special Keyboard scan for
critical timing applications.
DEF GO Source KK Object KO
REF VSBW,VMBW Editor Assembler option 3
*VSBW EQU >2020 )For linked x/b, remark out previous
*VMBW EQU >2024 ) line and allow these two lines.
KY EQU >8375
```

```
JOYCHR DATA >183C,>7EFF,>1818,>1818 UP Joystick
DATA >1818,>1818,>FF7E,>3C18 DOWN direction
DATA >1030,>70FF,>FF70,>3010 LEFT characters
DATA >080C,>0EFF,>FF0E,>0C08 RIGHT
DATA >F080,>C09C,>921C,>1412 fr FIRE BUTTON
```

\* The following look up table determines the character value for each key press detected.

\* Control + anything & fctn X give the copyright symbol  
 \* (e/a option 3 only).  
 \* Lower case is not implemented.  
 \* The TB number remark shows which TB tests each group.

```
* TB7 fctn 2 3 4 5 1 up Functions
D2 BYTE >FD,50,51,52,53,49,128,128,>FF,4,7,2,14,3
* @ # $ % ! Ctrls
BYTE >FF,64,35,36,37,33,>0A,>0A,>0A,>0A,>0A,>0A
```

```
* TB8 shift S D F G A fctns { }
S BYTE >FC,83,68,70,71,65,>00,>00,>FF,8,9,123,125
* |
BYTE 124,>FF,83,68,70,71,65,>0A,>0A,>0A,>0A,>0A
BYTE >0A
```

```
* TB9 ctrl W E R T Q fctn ~ [ ]
W BYTE >FE,87,69,82,84,81,>00,>00,>FF,126,11,91,93
BYTE >FF,>FF,87,69,82,84,81,>0A,>0A,>0A,>0A,>0A
BYTE >0A
```

```
* TB3 = . , M N / fire Functions
M BYTE 61,46,44,77,78,47,132,132,>FF,>FF,>FF,>FF
* + > < - Ctrls
BYTE >FF,>FF,43,62,60,77,78,45,>0A,>0A,>0A,>0A
BYTE >0A,>0A
```

```
* TB4 space L K J H ; left Functions
L BYTE 32,76,75,74,72,59,130,130,32,>FF,>FF,>FF
* Shifts : Ctrls
BYTE >FF,>FF,32,76,75,74,72,58,32,>0A,>0A,>0A
BYTE >0A,>0A
```

```
* TB5 enter O I U Y P right ent ' ?
O BYTE 13,79,73,85,89,80,131,131,13,39,63,95,>FF
* "
BYTE 34,13,79,73,85,89,80,13,>0A,>0A,>0A,>0A
BYTE >0A
```

```
* TB6 9 8 7 6 0 down Functions
D9 BYTE >FF,57,56,55,54,48,129,129,>FF,15,6,1,12
* ( * & ^ ) Ctrls
BYTE >FF,0,40,42,38,94,41,>0A,>0A,>0A,>0A,>0A
BYTE >0A
```

```
* TB10 X C V B Z fctn `
X BYTE >FF,88,67,86,66,90,>00,>00,>FF,10,96,>FF
* \
BYTE >FF,92,>FF,88,67,86,66,90,>0A,>0A,>0A,>0A
BYTE >0A,>0A
```

EVEN

\* Subroutine for Test Bits

```
SUBTB
TB7 TB 7
JEQ TB8 If status register is not equal
then a key press has been detected by this
CRU bit. Further test & storage follows.
*
CB @D2(R2),@D2 Test for FCTN key on first pass
JEQ FCTN FUNCTION key detected get offset
MOVB @D2(R2),@KY Place byte for key in KY= >8375
```



```

TB8  TB  8
     JEQ TB9
     CB  @S(R2),@S    Test for SHIFT key on first pass
     JEQ SHIFT        SHIFT key detected get offset
     MOVB @S(R2),@KY

TB9  TB  9
     JEQ TB3
     CB  @W(R2),@W    Test for CTRL key on first pass
     JEQ CTRL        CONTROL key detected get offset
     MOVB @W(R2),@KY

TB3  TB  3           Remarks for all Test Bits are
     JEQ TB4         similar to TB7. If a key is
     MOVB @M(R2),@KY not detected then jump to the
                        next test or the end of sub.
*
TB4  TB  4
     JEQ TB5
     MOVB @L(R2),@KY

TB5  TB  5
     JEQ TB6
     MOVB @O(R2),@KY

TB6  TB  6
     JEQ TB10
     MOVB @D9(R2),@KY

TB10 TB  10
     JEQ ENDSUB
     MOVB @X(R2),@KY

ENDSUB INC R2       To point to next entry in table
      RT

FCTN  LI  R2,8       Table offset for FUNCTION
      JMP TB3

SHIFT LI  R2,14      Table offset for SHIFT
      JMP TB3

CTRL  LI  R2,20      Table offset for CONTROL
      JMP TB3

*For linked x/b change value for R0 from >0C00 to >0700
GO    LI  R0,>0C00    Load joystick special characters
      LI  R1,JOYCHR   from character 128 up
      LI  R2,40
      BLWP @VMBW

      CLR R12         Entry point
      CLR R2
      LOOP
*     MOVB @X,@KY    Clears KeY Buffer before each scan
*     Remove first * in previous line to
*     clear character when key is released

```

```

TEST0 SBZ 18
      SBZ 19         000=0
      SBZ 20
      BL  @SUBTB
*
TEST1 SBO 18
      SBZ 19         001=1
      SBZ 20
      BL  @SUBTB

TEST2 SBZ 18
      SBO 19         010=2
      SBZ 20
      BL  @SUBTB

TEST3 SBO 18
      SBO 19         011=3
      SBZ 20
      BL  @SUBTB

TEST4 SBZ 18
      SBZ 19         100=4
      SBO 20
      BL  @SUBTB

TEST5 SBO 18
      SBZ 19         101=5
      SBO 20
      BL  @SUBTB

TEST6 SBZ 18
      SBO 19         110=6
      SBO 20
      BL  @SUBTB           Joystick 1.

TEST7 SBO 18
      SBO 19         110=7
      SBO 20
      BL  @SUBTB           Joystick 2.

DISP  LI  R0,>0170    Screen location to display key
      MOVB @KY,R1     Byte to be displayed
*     AI  R1,>6000    Let this line in for linked x/b
      BLWP @VSBW     Show which key has been pressed
      JMP LOOP        Go and scan again

      END GO

```

\* The preceding program is intended to show you how to  
 \* use the CRU bits to scan the keyboard and joysticks.  
 \* It does not contain keyboard debounce since this type  
 \* of program is intended to work within another program  
 \* which provides protection against keybounce problems.  
 \* The STATUS register is not used in this program to  
 \* indicate the detection of a pressed key at the end of  
 \* a key scan.  
 \* The status register is NOT used to indicate that a  
 \* key is pressed. The key value is returned as a byte  
 \* in >8375; if no key is detected the value is >FF.  
 \* END \*

# HELP

*who ya gonna call...*



## PROGRAMMERS CRISIS

**LINE Ph : 992229**

Special TISHUG service for the beginner with Basic, Extended Basic or Minimemory.



## Type It

Programs To Try!!

For those of you who assembled your own 99/4(A) LIGHTPEN mentioned in the News Digest a few months ago, here is another program for you to type in and use with it.

Have fun! If you have written and other program that a light pen can be used with, please share it with us.

```

100 !-----!
110 !
120 ! 0000 000 00000 !
130 ! 0 0 0 0 0 !
140 ! 0 0 0 0 0 !
150 ! 0 0 0 0 0 !
160 ! 0 0 0 0 0 !
170 ! 0 0 0 0 0 !
180 ! 0000 000 0 !
190 !
200 !
210 REM by: Edwin McFall
220 REM 2005 W. 6th
230 REM ABERDEEN WA.
240 REM 98520
250 REM TI 99/4A VER. 1.1
260 REM REQUIRES LIGHTPEN
270 REM AND EXTENDED BASIC
280 REM
290 REM
300 CALL CLEAR :: CALL SCREE
N(2)
310 FOR X=1 TO 10 :: CALL CO
LOR(X,5-11*(X>8),1):: NEXT X
320 RANDOMIZE
330 FOR X=0 TO 2
340 CALL CHAR(96+8*X,"3C7EFF
FFFFFF7E3C")
350 CALL COLOR(9+X,1,1)
360 NEXT X
370 PRINT " `hp`hp `hp`
`hp`hp` hp` p`h`
p`h` hp` h hp`
h p`h hp` h hp`
h p`h"
380 PRINT " hp` h hp`
h p`h hp` h hp`
h p`h hp` p`h`
p`h"
390 PRINT " `hp`hp `hp`
p`h` : : :
400 PRINT : "`hp`hp`hp`hp`h
p`hp`hp`hp`hp`p BY: EDWI
N MCFALL hh TOUCH DOT T
O CONTINUE. p`ph`ph`ph`p
h`ph`ph`ph`ph`"
410 CALL JOYST(1,X,Y):: IF X
<>0 OR Y<>0 THEN 480
420 FOR C=1 TO 3
430 CALL COLOR(9,7-4*(C=1)-8
*(C=2),1)
440 CALL COLOR(10,7-4*(C=2)-
8*(C=3),1)
450 CALL COLOR(11,7-4*(C=3)-
8*(C=1),1)
460 NEXT C
470 GOTO 410
480 CALL SOUND(100,440,0)::
CALL CLEAR :: SC=0
490 DISPLAY AT(12,9):"EASY
HARD"
500 DISPLAY AT(16,7):"SELECT
DIFFICULTY" :: DISPLAY AT(18
,1):"EASY=LARGE DOTS, HARD=S
MALL."

```

```

510 CALL HCHAR(12,9,112):: C
ALL HCHAR(12,19,104)
520 CALL COLOR(11,16,16,10,2
,2)
530 FOR I=1 TO 10
540 CALL JOYST(1,X,Y):: IF X
=4 THEN CALL MAGNIFY(2):: GO
TO 610
550 NEXT I
560 CALL COLOR(11,2,2,10,16,
16)
570 FOR I=1 TO 10
580 CALL JOYST(1,X,Y):: IF X
=4 THEN CALL MAGNIFY(1):: GO
TO 610
590 NEXT I
600 GOTO 520
610 CALL CLEAR
620 CALL SPRITE(#1,96,16,92,
124)
630 DISPLAY AT(16,6)BEEP:"TO
UCH DOT TO START."
640 CALL JOYST(1,X,Y):: IF X
<4 THEN 640
650 CALL SOUND(-100,220,5)::
CALL SOUND(-100,880,0)
660 CALL CLEAR
670 FOR L=1 TO 20
680 CALL SPRITE(#1,96,16,INT
(RNDO)+1,INT(RNDO)+10)
690 N=0
700 CALL JOYST(1,X,Y)
710 IF X=4 THEN 730
720 N=N+1 :: GOTO 700
730 SC=SC+N :: CALL SOUND(-1
00,440,5)
740 DISPLAY AT(1,1):"SCORE "
;SC
750 FOR X=1 TO 100 :: NEXT X
760 NEXT L
770 FOR Z=1 TO SC STEP 10
780 CALL SOUND(-100,Z+110,0)
790 NEXT Z
800 CALL CLEAR :: CALL SPRIT
E(#1,96,16,150,123)
810 DISPLAY AT(10,6):"YOUR S
CORE IS ";SC :: DISPLAY AT(1
8,3):"TOUCH DOT TO PLAY
AGAIN."
820 IF SC>150 THEN 830 ELSE
DISPLAY AT(12,6):" YOU CA
N'T FOOL ME!
YOU CHEATED!" :: GOTO 900
830 IF SC>200 THEN 840 ELSE
DISPLAY AT(12,1):" VERY GOOD
! YOU'VE MASTERED
THE DOTS." :: GOTO 900
840 IF SC>225 THEN 850 ELSE
DISPLAY AT(12,1):"GEE WIZ! I
DIDN'T KNOW THAT HUMANS COU
LD MOVE THAT FAST!" :: GOTO
900
850 IF SC>250 THEN 860 ELSE
DISPLAY AT(12,1):"NOT BAD, B
UT YOU'LL NEVER SWAT A FLY
AT THAT SPEED." :: GOTO 900
860 IF SC>300 THEN 870 ELSE
DISPLAY AT(12,1):"YOU NEED A
LOT MORE PRACTICE" :: GOTO 9
00
870 IF SC>350 THEN 880 ELSE
DISPLAY AT(12,1):"I THINK YO
U NEED A SLOWER GAME.
MAYBE CHECKERS." :: GOTO 900
880 IF SC>400 THEN 890 ELSE
DISPLAY AT(12,1):"DIDN'T ANY
ONE TELL YOU THAT YOUR TRYIN
G FOR A LOW SCORE!" :: GOTO
900
890 DISPLAY AT(12,1):"LOOKS
LIKE YOU NEED HELP. TRY PO
INTING THE PEN AT THE DOT ST
UPID!"

```

```

900 FOR Z=1 TO 500 :: CALL J
OYST(1,X,Y):: IF X=4 THEN 93
0
910 NEXT Z
920 CALL CLEAR :: DISPLAY AT
(12,1):" DOTS ALL FOLKS..."
:: END
930 CALL DELSPRITE(ALL):: GO
TO 480

```



```

100 REM *****
110 REM *BE FRUITFUL*
120 REM *****
130 REM IN TI BASIC
140 REM EDUCATIONAL SOFTWARE
150 CALL CLEAR
160 CALL COLOR(12,9,11)
170 PRINT "xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxx";TAB(28);"x": "x
BE FRUITFUL AND MULTIPLY xx
xx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
: : : : : : : : : : :
180 CLI=4
190 RANDOMIZE
200 CALL CHAR(128,"000000000
001070F")
210 CALL CHAR(129,"00221C1C3
CF0FOFO")
220 CALL CHAR(130,"1F3F3F3F3
F3F203F")
230 CALL CHAR(131,"D09010101
0D800FC")
240 CALL CHAR(152,"0986463C3
C24")
250 CALL CHAR(132,"0020181E3
EFOFOFO")
260 CALL CHAR(146,"000000000
020303F")
270 CALL CHAR(147,"00000COE1
C38FOFO")
280 CALL CHAR(148,"3F1FOFO70
1010300")
290 CALL CHAR(149,"FOEOEOEO0
00080")
300 CALL CHAR(153,"000020601
C1C08")
310 CALL CHAR(136,"162B7D4FB
55D3B5D")
320 CALL CHAR(137,"98B46E7CE
D76FCD4")
330 CALL CHAR(91,"0101010101
0101")
340 CALL CHAR(93,"8080808080
8080")
350 CALL CHAR(154,"1818183C3
C3C18")
360 CALL CHAR(155,"00183C7E7
E3C18")
370 PRINT "ENTER A NUMBER BE
TWEEN 2 AND9,": "I WANT TO
PRACTICE MY TIMES TABLES UP
TO ";
380 INPUT "NUMBER _":A
390 IF (A<2)+(A>9)THEN 370
400 A5=9
410 IF A<6 THEN 430
420 A5=12
430 CALL SCREEN(14)
440 FOR GRETCHEN=1 TO 8
450 ATT=0
460 ON INT(RND*6+1)GOTO 920,
940,960,980,1000,1030
470 READ A$,B$,MOM1,MOM2,MOM
3,MOM4,CHILD,CL1,CL2,CL4
480 X=INT(RND*(A-1))+2
490 Y=INT(RND*A5+1)
500 ANSWER=X*Y
510 QUEST=QUEST+1

```



```

520 CALL CLEAR
530 CALL COLOR(CLI,2,1)
540 CALL COLOR(16,2,1)
550 PRINT "IF "&STR$(X)&A$&S
TR$(Y)
560 PRINT B$:"WOULD THERE BE
";
570 SPOT=11-X
580 FOR I=1 TO X
590 CALL HCHAR(SPOT+2*I,8,M0
M1)
600 CALL HCHAR(SPOT+I*2,9,M0
M2)
610 CALL HCHAR(SPOT+1+2*I,8,
MOM3)
620 CALL HCHAR(SPOT+1+2*I,9,
MOM4)
630 CALL HCHAR(SPOT+1+2*I,11
,CHILD,Y)
640 NEXT I
650 INPUT GUESS
660 IF GUESS=ANSWER THEN 800
670 ATT=ATT+1
680 ON ATT GOTO 690,720,750
690 PRINT "NOT RIGHT. TRY HA
RDER.";
700 WRONG=WRONG+.6
710 GOTO 650
720 PRINT "WRONG. COUNT THE
M ";
730 WRONG=WRONG+.3
740 GOTO 650
750 PRINT "THE RIGHT ANSWER
IS";ANSWER
760 WRONG=WRONG+.1
770 CALL SOUND(4000,200,30)
780 CALL SOUND(100,200,30)
790 GOTO 850
800 CALL COLOR(CLI,CL2,1)
810 CALL COLOR(16,CL4,1)
820 FOR I=1 TO 10
830 CALL SOUND(100,30*I+100,
4)
840 NEXT I
850 NEXT GRETCHEN
860 PRINT : "DO YOU WANT TO S
TOP?"
870 CALL KEY(3,SCH,ENK)
880 IF SCH=78 THEN 370
890 IF SCH<>89 THEN 870
900 PRINT :QUEST;"QUESTIONS"
: : "SCORE=";INT(((QUEST-WR
ONG)/QUEST)*100);"%
910 STOP
920 RESTORE 1020
930 GOTO 470
940 RESTORE 1050
950 GOTO 470
960 RESTORE 1060
970 GOTO 470
980 RESTORE 1070
990 GOTO 470
1000 RESTORE 1080
1010 GOTO 470
1020 DATA " CATS EACH HAD ",
"KITTENS, HOW MANY KITTENS",
128,129,130,131,152,13,15,16
1030 RESTORE 1090
1040 GOTO 470
1050 DATA " CHICKENS EACH HA
D ", "CHICKS, HOW MANY CHICKS
",146,147,148,149,153,15,16,
11
1060 DATA " DOGS EACH HAD ",
"PUPPIES, HOW MANY PUPPIES",
128,132,130,131,152,13,7,10
1070 DATA " TREES EACH HAD "
,"APPLES, HOW MANY APPLES",1
36,137,91,93,155,14,4,9
1080 DATA " TREES EACH HAD "
,"PEARS, HOW MANY PEARS",136
,137,91,93,154,14,4,11
1090 DATA " HENS EACH LAID "
,"EGGS, HOW MANY EGGS",146,1
47,148,149,155,15,10,16

```

```

100 REM *****
110 REM *DRACULA ADVENTURE*
120 REM *****
130 REM IN TI BASIC
140 CALL CLEAR
150 CALL SCREEN(13)
160 CALL COLOR(1,16,13)
170 FOR A=2 TO 8
180 CALL COLOR(A,16,7)
190 NEXT A
200 PRINT " # ADVENTURE WIT
H DRACULA #"
210 PRINT
220 PRINT
230 PRINT " REMEMBER : "
240 PRINT
250 PRINT " DRACULA IS ALWA
YS BEHIND"
260 PRINT " YYYOOOUUU.....
....."
270 PRINT
280 PRINT " GO AWAY FROM DR
ACULA AS"
290 PRINT " LONG AS YOU CAN
!"
300 PRINT
310 PRINT " IT WILL BE MORN
ING SOON."
320 FOR A=1 TO 5
330 PRINT
340 NEXT A
350 PRINT " (SOFTWARE_BY_BE
RND)"
360 PRINT " (ZEUTSCHEL,_WES
T-GERMANY)"
370 PRINT " (SOFTWARERIGHTS
:)"
380 PRINT " (BRIAN_O_BRIEN_
)"
390 FOR A=110 TO 220 STEP 10
400 CALL SOUND(500,A,0)
410 NEXT A
420 CALL KEY(0,K,S)
430 IF S<>1 THEN 420
440 FOR A=1 TO 5
450 PRINT
460 NEXT A
470 PRINT " USED KEYS FOR M
OVING:"
480 PRINT
490 PRINT " W E R"
500 PRINT " S D"
510 PRINT " Z X C"
520 PRINT
530 PRINT " WITH <SPACE> YO
U STOPS."
540 PRINT
550 PRINT
560 CALL KEY(0,K,S)
570 IF S<>1 THEN 560
580 RANDOMIZE
590 DATA 0,0,0,1,1,1,1,0
600 DATA 1,-1,0,-1,-1,-1
610 DATA -1,0,-1,1
620 X$=" XCDREWSZ"
630 CALL CHAR(96,"183C187E7E
FFDBD")
640 CALL CLEAR
650 CALL SCREEN(7)
660 CALL COLOR(6,16,4)
670 CALL COLOR(1,2,7)
680 CALL COLOR(2,16,5)

```

```

690 CALL COLOR(9,2,16)
700 X=INT(RND*30)+2
710 Y=INT(RND*22)+2
720 A=INT(RND*30)+2
730 B=INT(RND*22)+2
740 FOR LOOP=1 TO 120
750 X1=INT(RND*32)+1
760 Y1=INT(RND*24)+1
770 IF X1=X THEN 750
780 IF Y1=Y THEN 750
790 CALL HCHAR(Y1,X1,79,1)
800 REM CALL SOUND(1,1500,0)

810 NEXT LOOP
820 CALL HCHAR(1,1,43,32)
830 CALL HCHAR(24,1,43,32)
840 CALL VCHAR(2,1,43,22)
850 CALL VCHAR(2,32,43,32)
860 CALL HCHAR(B-GY,A-GX,32,
1)
870 CALL HCHAR(B,A,96,1)
880 IF A<>X THEN 910
890 IF B=Y THEN 1280
900 REM +++++
910 GY=SGN(Y-B)
920 GX=SGN(X-A)
930 T=INT(RND*8)+1
940 IF T=7 THEN 1010
950 A=A+GX
960 B=B+GY
970 CALL GCHAR(B,A,J)
980 IF J=32 THEN 1040
990 A=A-GX
1000 B=B-GY
1010 GX=INT(RND*3)-1
1020 GY=INT(RND*3)-1
1030 GOTO 950
1040 SCORE=SCORE+1
1050 IF SCORE=200 THEN 1580
1060 CALL HCHAR(Y,X,42,1)
1070 CALL KEY(0,KEY,ST)
1080 IF KEY<1 THEN 1200
1090 IF ST<>1 THEN 1200
1100 K$=CHR$(KEY)
1110 RESTORE
1120 FOR L=1 TO 9
1130 READ X1,Y1
1140 IF K$=SEG$(X$,L,1)THEN
1170
1150 NEXT L
1160 GOTO 1200
1170 REM
1180 DX=X1
1190 DY=Y1
1200 CALL HCHAR(Y,X,32,1)
1210 Y=Y+DY
1220 X=X+DX
1230 CALL GCHAR(Y,X,W)
1240 IF W=32 THEN 860
1250 Y=Y-DY
1260 X=X-DX
1270 GOTO 860
1280 FOR C=1 TO 30
1290 CALL SCREEN(INT(RND*16)
+1)
1300 CALL COLOR(1,INT(RND*16)
)+1,INT(RND*16)+1)
1310 CALL SOUND(10,INT(RND*4
000)+111,0)
1320 NEXT C
1330 CALL CLEAR
1340 CALL SCREEN(2)
1350 FOR W=5 TO 8
1360 CALL COLOR(W,16,7)
1370 NEXT W
1380 CALL COLOR(1,7,2)
1390 PRINT "NOW DRACULA HAS
NO"
1400 PRINT
1410 PRINT "HUUNNGGGEER AN
YMORE !!!"
1420 FOR W=1 TO 10
1430 PRINT
1440 NEXT W

```



```

1450 PRINT "YOUR SCORE IS ";
SCORE
1460 PRINT
1470 FOR A=1 TO 10
1480 O=O+110
1490 CALL SOUND(100,O,0)
1500 NEXT A
1510 INPUT "AGAIN (Y/N) ? ";
X$
1520 IF X$<>"Y" THEN 1550
1530 SCORE=0
1540 GOTO 580
1550 PRINT
1560 PRINT "GOOD BYE, OLD BO
Y."
1570 END
1580 CALL CLEAR
1590 CALL SCREEN(6)
1600 FOR W=5 TO 8
1610 CALL COLOR(W,16,7)
1620 NEXT W
1630 CALL COLOR(1,7,6)
1640 PRINT "CONGRATULATIONS
!!!"
1650 PRINT
1660 PRINT "IT IS MORNING AN
D SO "
1670 PRINT "DRACULA MUST GO
TO"
1680 PRINT "BED. YOU HAD LUC
K."
1690 FOR A=1 TO 7
1700 PRINT
1710 NEXT A
1720 GOTO 1510
    
```

## INDEX

### CONTENTS OF THIS MONTH'S TND

- #2 UFFONI (Auction Day) with Shane
- 3 FACE TO FACE (Co-Ord; Report)
- 4 EXPLODING TRIPLE TECH CARD & COMMUNICATORS PAGE.
- 5 ASSEMBLER- Interrupt Switch.
- 6 CHRISTMAS COMPETITION .
- 7 ARTO'S KEYBOARD MUSIC WRITER
- 8 & 9 SOFTWARE CLIMB with TERRY
- 10 TEND TIME & DIGITIZING THE TI.
- 11, 12, 13 FUNWRITER REVIEW by Ben
- 12 REGIONAL REPORT
- 14 DID YOU EVER WONDER WHY?
- 15 TIP CORNER : faking I/O PORT.
- 16 TIGERCLUB PROGRAM TIPS #25
- 18 OLDSIES BUT GOODIES (modules).
- 19 LINK-IT : Morse Code Interface.
- 22 TYPE IT : Dot Lightpen & By PHILIP (Educational game)
- 23 TYPE-IT : DRACULA ADVENTURE.
- 24 ARTO SCI-FI CARTOON the start.

Ato Heino is one of Australia's Quiet achievers, a very talented member of TISHUG. He has contributed designs for our covers of SND & TND, and in this issue has designed an interface for your TI which enables a music keyboard to be connected to the joystick port, with software (as explained elsewhere in this publication). He now commences his TISHUG NEWS DIGEST SCI-FI CARTOON.

This SCI-FI cartoon will be continued over the next few issues, and was created by him, using TI-ARTIST. We know that you will enjoy this thrilling story, as it unfolds.

