
ISHUG

NEWS DIGEST

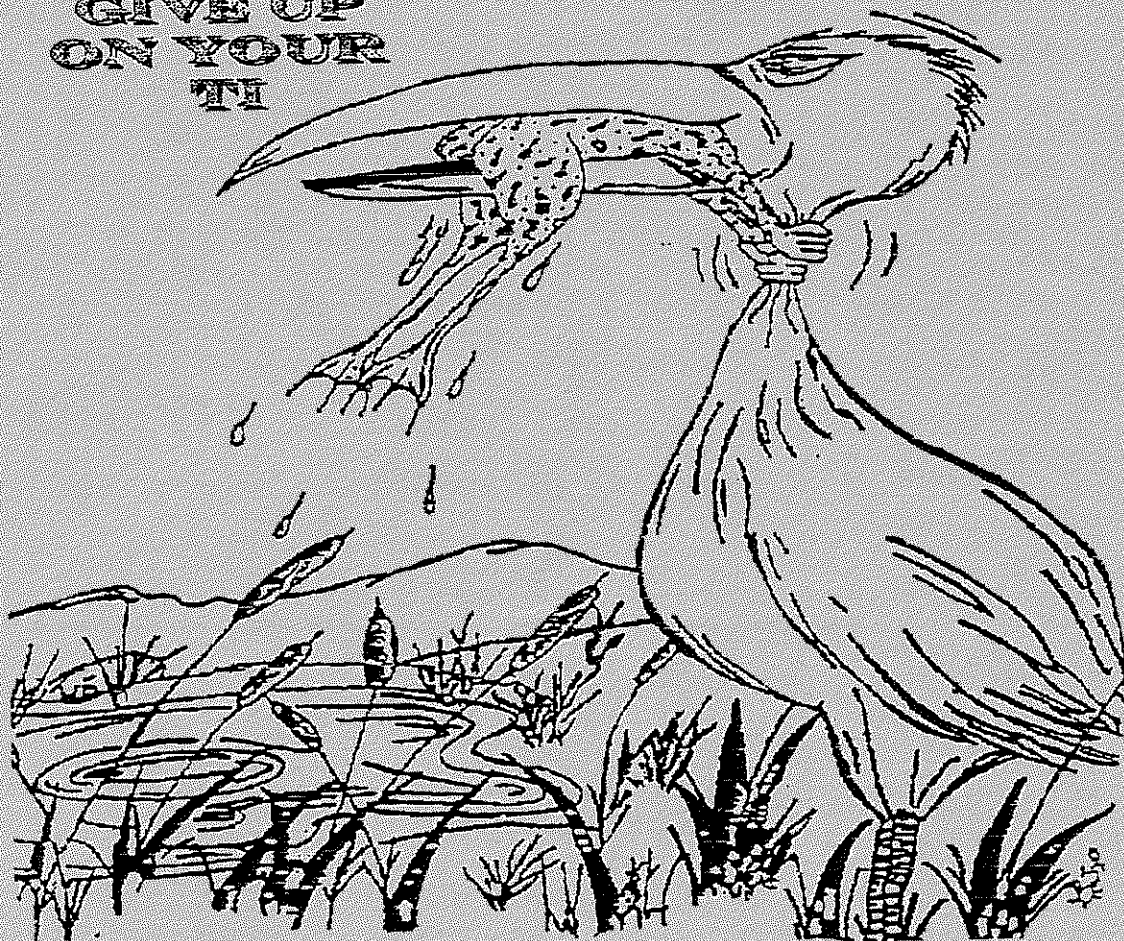
Focusing on the TI99/4A Home Computer

Volume 15, Number 5

June, 1996

PRINT POST Approved - PP244099/00016

**DON'T
"EVER"
GIVE UP
ON YOUR
TI**



Sydney, New South Wales, Australia

\$3

TiSHUG (Australia) Ltd.
A.C.N. 003 374 383

All correspondence to:
C/o 3 Storey St.
Ryde 2112 Australia

TiSHUG News Digest

ISSN 0819-1984

I N D E X

The Board

Co-ordinator

Dick Warburton (02) 9918 8132

Secretary

Percy Harrison (02) 808 3181

Treasurer

Cyril Bohlsen (02) 639 5847

Directors

Thomas Marshall (02) 871 7535

Loren West (047) 21 3739

Sub-committees

News Digest Editor

Loren West (047) 21 3739

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

TI Merchandising

Dick Warburton (02) 9918 8132

IBM Merchandising

Cyril Bohlsen (02) 639 5847

Software Library

Larry Saunders (02) 644 7377

Technical Co-ordinator

Geoff Trott (042) 29 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Glebe

Mike Slattery (02) 692 6162

Hunter Valley

Geoff Phillips (049) 42 8617

Illawarra

Geoff Trott (042) 29 6629

Liverpool

Larry Saunders (02) 644 7377

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00
Associate membership \$10.00
Overseas Airmail Dues A\$65.00
Overseas Surface Dues A\$50.00

TiSHUG Sydney Meeting

The June Meeting will start at
2.0 pm on the 1st June 1998
at Meadowbank Primary School,
Thistle Street, Meadowbank.

Printed by
Kwik Kopy West Ryde

Title	Description	Author	Page No
BBS DOWNLOAD	GENERAL INTEREST	LOREN WEST	10
CALENDAR PROGRAM	PROGRAM	ADRIAN ROBINSON	3
GPL REVIEW	REVIEW	MACK MCCORMICK	5
HELP LINE	USER SUPPORT	LOREN WEST	10
HOW TO FIT VERY LARGE FILES INTO TI WRITER	GENERAL INTEREST	CHARLES GOOD	15
IN RESPONSE	GENERAL INTEREST	RON WARFIELD	12
MONITORS COMPATIBLE WITH TI	GENERAL INTEREST	LOREN WEST	11
POWER YOUR ENTIRE TI SYSTEM	TECHNICAL	CHARLES GOOD	14
PUZZLE	PUZZLE	LOREN WEST	12
RARE OFFICIAL TI EDUCATIONAL CARTRIDGES	REVIEW	CHARLES GOOD	13
REGIONAL REPORTS	GENERAL INTEREST	VARIOUS	23
TISHUG SOFTWARE REVIEW	REVIEW	LARRY SAUNDERS	2
TRANSFERRING DATA BASE FILES	GENERAL INTEREST	LOREN WEST	4
TRIVIA QUESTIONS	GENERAL INTEREST	CHARLES GOOD	16

IBM INDEX

IBM SHOP	GENERAL INTEREST	CYRIL BOHLSSEN	18
TECHNICAL TIPS	TIPS	TECHNICAL TIPS DIGEST	19
USING A MICROPHONE WITH SOUND CARD	GENERAL INTEREST	LOREN WEST	18

IN MEMORIAM

It is with considerable sadness that we report the death of one of our TiSHUG members,
TREVOR WARHAM, who passed away 17th APRIL 1996.

Trevor was a member back in 1984, but let his membership lapse after a few years, he rejoined TiSHUG again in 1995 when he learned we now supported IBM computers.

Our sincere sympathy and condolences are extended to his Wife, Family and Friends, on this very sad occasion.

INSTRUCTIONS FOR MAX-RLE

Public Domain Program
by Travis Watford

RLE stands for Run Length Encoded. It is a program for preparing and viewing digitized pictures, both artwork and photographs, sent between computers over phone lines using a terminal emulator like Fast-Term. Many computers use this technique with the VIDTEX terminal emulator protocol, which permits viewing pictures on-line. For the TI-99/4A at present, pictures can be viewed off line only, but pictures can be exchanged with other brands of computers. The program supports four different formats - both TI-ARTIST and GRAPHX formats, as well as Display/Fixed 128, the usual format used in other computers, and Display/Variable 80 format.

LOADING MAX-RLE. The program is loaded using the Editor-Assembler module or equivalent, Option 3 - Load and Run. The Filename is MAX-RLE and the Program Name is START. The MAX-RLE title screen will then appear, asking for the name of the picture file you want to load.

RUNNING MAX-RLE. At the title screen, you have two options - you can load a picture or you can catalog a disk.

** To load a picture, just type the filename, for example, DSK1.PICTURE, and press ENTER. Whichever format the picture is in, the program will recognize it and load it. (NOTE: For TI-Artist files, omit the "P" and "C" at the end of the filename - the program will provide these automatically.) You will then see a grey screen for a short while as the picture loads. Give it a chance! It does not appear immediately, but there is nothing wrong with the program. The picture will appear all at once on the screen when it is ready. Note: The program supports the Horizon Ram Disk HD command.

** To catalog a disk, just type "DSKn.", where "n" is the drive number. Be sure you include the period. The catalog function does not work with the MYARC RAM Disk.

OPTIONS WITH PICTURE ON-SCREEN. There are three options when the picture appears - you can return to the title screen, print the screen on your printer, or save the picture in a different format to disk.

** To return to the MAX-RLE title screen, press ENTER. Be careful. An accidental pressing removes the picture.

** To print on your printer, press P. The default setting of PIO.CR will appear on screen. If you are using the parallel interface, use this. If you are using a serial interface (RS232), enter your printer's description just as you would, for example, using the TI-Writer Formatter. Your printer must be compatible with the GEMINI/EPSON family in its handling of dot-graphics.

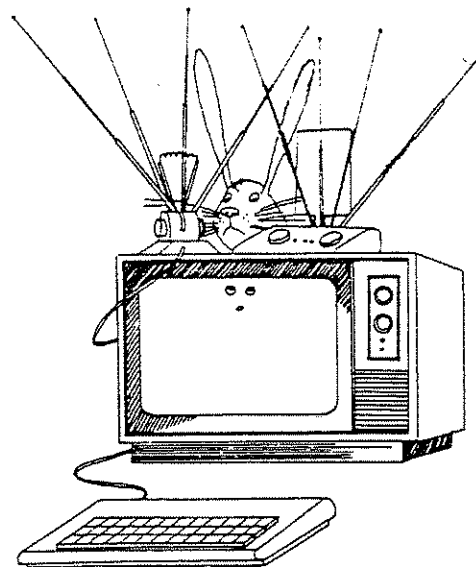
** To save to disk in a different Format, press S. The default setting of GRAPHX format will appear on screen. To save in a different format, press the space bar until the format you want to use appears. Then type the filename you wish to save to, for example, DSK1.MYPIC.

SENDING PICTURE FILES. Generally, pictures to be transmitted should be saved in the D/F128 format and uploaded with XMODEM transfers. This is the format used by other systems. Pictures can be sent in D/F80 format using ASCII (text) transfers, but they lack error checking in transmission and a noisy or weak connection can ruin the resulting picture.

PICTURES ON COMPUERVE. Pictures readable by MAX-RLE can be found in CompuServe in the TI Forum Data Libraries, the PICSIG, the ARTFORUM, and the CB simulator area. They are also starting to appear on many BBS.

-text adapted by Walt Howe from a version by Paul Grey, the author of the first TI-RLE program. This text is primarily for non-CIS subscribers.

END OF ARTICLE 



antenna

I hope you enjoy this disk collection. Since it was submitted to Micropendium for inclusion in their FAIRWARE section, there have been some additional programs added.

Reverserle- creates an inverse (negative image) of any Fixed 128 RLE picture.

IMPORTANT: The input file must be an RLE picture in df/128, no other type will be accepted by the program. If the picture is in another format such as Graphx or TI-Artist, first using the Max-rle program convert it to 128 format using the SAVE(S) command (paged with space bar). Read the doc's for Max-Rle for a complete description. The output filename of Reverserle may not be the same as the input filename, UNLESS it's sent to a different drive. (the program will issue an error if you attempt to call both input and output files the same name)

I have also included a copy of MAX-RLE, for those who don't have a copy.

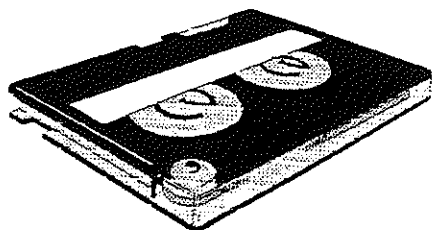
Besides the graphic screens, which are in RLE and GRAPHX, there is an additional program included called SHOWGX. SHOWGX loads the assembly program READG, this program unlike REVERSERLE is to run in BASIC with the E/A cartridge.

SHOWGX- Will scan any drive selected and look for GRAPHX pictures. When found it displays the name and loads it.

Press any key for it to go on to the next. When all pictures on that disk are loaded it asks for more?. If you respond yes, you can now load another disk from any drive.

DO NOT ATTEMPT to alter the basic loading program. If you have any questions please feel free to write me:

Stephen J. Tuorto
18-Chimney Lane
Bayshore, NY 11706



CALENDER PROGRAM

By Adrian Robinson

[In the May issue of our magazine we printed an article on "Calender programming", however the printing of this article wasn't the best, lets hope this copy comes out clearer for the members who would like to type the program in. We at the editorial apologise for any inconvenience caused.]

I would like to take this opportunity to thank the members of our club for their input and articles towards the production of this great magazine, month after month. (ED)]

```
10 ! SAVE DSK6.CALENDER
11 ! CALENDER PROGRAM
12 ! by Adrian Robinson
13 ! ROM Newsletter, August 1989
14 !
16 ON WARNING NEXT :: CALL CLEAR
18 DATA JANUARY,31,FEBRUARY,28,MARCH,
31,APRIL,30, MAY,31,JUNE,30
20 DATA JULY,31,AUGUST,31,SEPTEMBER,30,
OCTOBER,31,NOVEMBER,30,DECEMBER,31
22 W$=" SU ^MO ^TU ^WE ^TH ^FR ^SA
"&&RPTS("-",28) 24 M$="" :: FOR I=1 TO 31 ::
M$=M$&&RPTS("^",3-LEN(STR$(I)))&&STR$(I)&&" " :: NEXT I
26 INPUT " YEAR: " : Y :: IF Y<1583 THEN
PRINT"^YEAR^MUST^BE LATER THAN 1582": :
GOTO 26
28 OPEN #1:"PIO".DISPLAY .VARIABLE 28 :: PRINT
#1:Y: :
30 FOR M=1 TO 12 :: READ MO$,L :: CALL MONTH
((Y),M,B)
32 IF M=2 THEN IF (Y/4=INT(Y/4))-(Y/100=INT
(Y/100)) +(Y/400=INT(Y/400))THEN L=L+1
34 D$=RPTS(" ^^^".B)&&SEG$(M$,1,4*L)
36 PRINT " ";MO$:W$:D$: : : PRINT #1:"
":MO$:W$:D$: :
37 IF M=6 THEN PRINT #1:CHRS(12): : :
38 NEXT M :: CLOSE #1 :: END
40 SUB MONTH(Y,M,B)
42 F=365*Y+31*(M-1)+1
44 IF M>2 THEN F=F-INT(2.3+.4*M)ELSE Y=Y-1
46 F=F+INT(Y/4)-INT(Y/100)+INT(Y/400)
48 F=F-1 :: B=F-7*INT(F/7)
50 SUBEND
```

```
100 ! SAVE DSK6.CALENDER1
110 ! CALENDER PROGRAM
120 ! by Adrian Robinson
```



```

130 ! ROM Newsletter, August 1989
140 ! Printout mods by Ross Mudie
150 DIM P$(12,9)
160 ON WARNING NEXT :: CALL CLEAR
170 DATA JANUARY,31,FEBRUARY,28,MARCH,
31,APRIL,30, MAY,31,JUNE,30
180 DATA JULY,31,AUGUST,31,SEPTEMBER,30,
OCTOBER,31,NOVEMBER,30,DECEMBER,31
190 W$=" SU ^MO ^TU ^WE ^TH ^FR ^SA
"&&RPTS("-",28)
200 M$="" :: FOR I=1 TO 31 ::M$=M$&&RPTS("^",3-
LEN(STR$(I)))&&STR$(I)&&"^" :: NEXT I
210 INPUT " YEAR: ":Y :: IF Y<1583 THEN PRINT
"YEAR^MUST^BE LATER THAN 1582": : :: GOTO
210
220 OPEN #1:"PIO".DISPLAY ,VARIABLE ::
PRINT#1:TAB(28);CHR$(14);Y:
230 FOR M=1 TO 12 :: READ MO$,L :: CALL
MONTH(Y,M,L)
240 IF M=2 THEN IF^(Y/4=INT(Y/4))-
(Y/100=INT(Y/100))+ (Y/400=INT(Y/400)) THEN
L=L+1
250 D$=RPTS(" ^^^",B)&&SEG$(M$,1,4*L)
260 PRINT " ";MO$:W$:D$: : :
270 P$(M,1)=MO$ :: P$(M,2)=SEG$(W$.1,28)::
P$(M,3)=SEG$(W$.29,28):: P$(M,4)=SEG$(D$.1,28)::
P$(M,5)=SEG$(D$.29,28)
280 P$(M,6)=SEG$(D$.57,28):: P$(M,7)= SEG$(D$.85
,28) ::P$(M,8)=SEG$(D$.113,28)
290 IF LEN(D$)>141 THEN P$(M,9)=SEG$(D$, 141,28
) ELSE P$(M,9)=""
300 NEXT M
310 FOR M=1 TO 12 STEP 2
320 FOR L=1 TO 9
330 IF L=1 THEN PRINT #1:CHR$(14):TAB
(3):P$(M,L); TAB(22);P$(M+1,L)
340 IF L>1 THEN PRINT #1:TAB(2);P$(M,L);TAB(40);
P$(M+1,L)
350 NEXT L
360 PRINT #1: :
370 NEXT M :: PRINT #1:CHR$(12):: CLOSE #1 ::
END
380 SUB MONTH(Y,M,L)
390 F=365*Y+31*(M-1)+1
400 IF M>2 THEN F=F-INT(2.3+.4*M)ELSE Y=Y-1
410 F=F+INT(Y/4)-INT(Y/100)+INT(Y/400)
420 F=F-1 :: B=F-7*INT(F/7)
430 SUBEND

```

END OF ARTICLE 

Transferring Data Base Files From Our Trusty "TI" To the IBM

by Loren West

This task is not new, but it is to me, so I though I would share my problems and success with everybody.

The data base that I am transferring from is the popular PR BASE and to "ACCESS" data base program in Microsoft Office.

Access can receive data base files from various sources, also it can use text files somehow, this will be another story later.

Ok with text files in mind. How can I make text files from my PR Data base. Easy, well almost easy, To help me on the way, I spoke to Dick Warburton as he was doing the same. Dick was using a different method but the results turned out the same.

My method was similar to the method that I am using already, transferring text files from the TI to the IBM, (having a cable hooked up using the serial ports,) and that is to PRINT them to the RS232, then on the IBM side I use the Terminal program that comes with Windows, which receives the information from the serial port and then saves it as a text file.

On the PR base side of things, I had set up the program to print out all of the data that is in the data base as if you were expecting to see it on a printout from the printer. The only difference was in the direction of the printer, normally (PIO) But I changed it to (RS232.BA=9600.DA=7.LF).

Having received this information into the Terminal program, which appeared an the screen exactly as it would have if it were sent to the printer.

Now I have a text file containing all the data base information that I need, the next step is to move this text into the Access data base system, when this happens I will let you know.

See you all at the next meeting.

END OF ARTICLE 

GPL

A Graphics Programming Language (GPL)
Primer
Routine to Read/Edit Text from KSCAN
by Mack McCornick 74206,1522

I have always avoided delving into the study of GPL because I felt it was too difficult, cumbersome, executed too slowly, and had little to offer. Boy, was I wrong. It makes writing routines used by BASIC a snap in assembler. For example, I recently needed a routine to read text from the screen which would allow full editing including erase, insert, delete, quit, honk tone at right margin, and enter/up arrow/down arrow. I also wanted the neat auto-repeat feature used by TI where there is a slight pause before the key takes off repeating. I began to consider writing the routine but then remembered that an identical routine resided in GPL in GROM (Graphics Read Only Memory) in the console. I first considered using the routine from GROM but then remembered that it added the screen offset of >60 to each character and I didn't need that. I could have done some fancy trick to make it work but decided to convert GPL to 9900 assembler code.

You'll find two programs here. One to link you to the routine in console GROM from a CALL LOAD from E/A BASIC and the identical (almost) routine in 9900 assembler code ready to link into any program you may have that needs this utility. I've included in the 9900 routine the actual GPL code used by the Pre-Scan routine of the monitor so you may see what conversions were necessary. Try reading the GPL instructions (marked with three *) to get a flavor for GPL). If the GPL gets in the way if using the routine in your program you may delete the GPL statements though they will have no effect if they remain.

It has really become obvious to me why TI invented GPL though I used to condemn them for it. The major reason is that it saves about 41% more code than straight assembler. GROM as you may know is only used by TI and is a chip which supplies a byte at a time to a memory mapped address and auto-increments to the next byte (like VDP RAM) unless you change the address to be read from. It's a great way to save memory. TI calls it medium speed memory. It is 6K bytes big and resides on 8K boundaries. It is an ideal medium to hold the console BASIC routines because the TMS9900 CPU chip in the console can only directly address 64K. The GPL actually does not execute any code. GPL is interpreted in console ROM beginning at >0024 and extending to >D18. This interpreter is straight assembler code which acts as directed by the GPL bytes coming from the GROM.

Hence you see one reason TI BASIC is slow. It is interpreted by GPL and GPL is interpreted by assembler. Two interpretations! Instructions in GPL usually have two operands and most instructions can access RAM, GROM, or VDP RAM. Most instructions are single byte operands unless the operand is preceded by a D for double operand. GPL uses two stacks a data stack at >83A0 and a subroutine address stack at >8380 (this allows arbitrary nesting of subroutines). Here are a few types of instructions:

DATA TRANSFER -Single/Double Byte
-Block to Block
-Formatted Block Moves
ARITHMETIC -add, subtract, multiply, divide, negate,
absolute val.
LOGICAL -AND, XOR, Shifts.
CONDITION -Arithmetic and Logical
BRANCHING -Conditional/Uncond
BIT MANIPULATION -Set, reset, test.
SUBROUTINE -Call, Return
STACK OPNS -Push and Pop
MISC -Random Number, KSCAN, Coincidence Detection,
Sound, I/O

The closest language to GPL is assembler and any experienced assembler programmer should have little difficulty learning GPL. One major difference is the use of MACRO instructions by the assembler such as REPEAT...UNTIL and IF...THEN...ELSE. Very similar in this respect to 9900 assembly language.

A few words about how memory is addressed. Here are a few of the most common ways and their syntax. 5 represents the decimal byte 5. >33 represents hex 33. 110011 represents binary 110011. #5506 represents the decimal number 5506. :A: is the ASCII equivalent >41. Well this has been a *very* general overview of GPL. Let's look at some actual GPL source code and my interpretation of the 9900 assembler equivalent. This routine could have been shortened but I tried to keep it as close to GPL as possible. Hope you enjoy it. If you have questions just ask. My 6 GPL manuals cover thousands of pages and we have just skimmed the surface here. I plan to write a GPL disassembler and interpreter to convert GPL to 9900 object code within the next six months if my schedule permits. That should make the job easy!

```
DEF START  
REF KSCAN,VSBW,VSBR,GPLLNK
```

```
* JUST A LITTLE ROUTINE TO TEST SUBROUTINE *  
START LWPI WS  
MOVB @H00,@KEYVAL SCAN ENTIRE KEYBOARD  
LOOP BL @READLN  
DATA >002,>2FE START, END POSITONS  
JMP LOOP
```

```

*
* This is the console GPL READLN routine at
^^(>2A42 in GROM 1) converted to
9900
* assembler. Interprets BACKSPACE, INSERT, DELETE, and
      FORWARD. Uses SCRATCH
* PAD RAM. Total number of characters may be limited by
      changing the start
* value of ARG+2 (upper limit) and entering at READL1.
      VARW is the start
address
* of the field. VARA is the current highest write
      address.
* Entering at READL1 allows us to pre-specify the
      minimum number of characters
* to be read for default creation.
* Entering at READ00 allows specification of the
      initial cursor position. In
* this case ARG+6 has to be set to the cursor position
      and ARG+4<0.
* Programmer responsibility to insure that VARW <=
      ARG+6 <= VARA <= ARG+2
* ARG+4 indicates if the line has been changed. If so,
      ARG+4=0.
* This is a possible call:
* BL @READLN
* DATA >1DF,>35D LOWER,UPPER SCREEN LIMITS
*****
*
* EQUATES *
WS EQU >8300 MY WORKSPACE
ARG EQU >835C
VARW EQU >8320 ABS LOWER LIMIT
VARA EQU >832A CURRENT END OF LINE
TEMP EQU 0 RO USED FOR TEMP STORAGE
TEMP1 EQU 1 R1 USED FOR ADDL TEMP STORAGE
RILB EQU WS+3
TEMP2 EQU 2
TEMP3 EQU 3
TIMER EQU >8379 VDP TIMER INC EVERY 1/60 SEC.
KEYVAL EQU >8374 KEYBOARD TO SCAN
RKEY EQU >8375 KEY CODE
STATUS EQU >837C GPL STATUS BYTE

* CONSTANTS * (Should be EQU with byte values in code
^^^^^^^^^^^^^^^^to save memory.)
H00 BYTE 0
H01 BYTE 1
HFF BYTE >FF
H508 DATA 508
H60 DATA 60
H14 BYTE 14
H766 DATA 766

BREAK BYTE >02
DELETE BYTE >03
INSRT BYTE >04
CLRLN BYTE >07

```

```

BACK BYTE >08
FORW BYTE >09
DOWN BYTE >0A
MVUP BYTE >0B
CHRTN BYTE >0D
CURSOR BYTE >1E
SPACE BYTE >20
VARV BYTE 0 (This is at >8301 in GPL but I
              use >8300 for workspace)
VAR1 DATA 0 AUTO REPEAT COUNTER (This is 1
              byte at >830D in GPL)

EVEN

READLN
* The GPL code stores >35D at ARG+2 but to give more
  utility; replaced with the
* next two lines of code.
*** DST >35D,@ARG+2 GPL DOUBLE STORE
    MOV *R11+,@VARW START ADDRESS OF THE FIELD
    MOV *R11+,@ARG+2 UPPER LIMIT
*** DST @VARW,@VARA
    MOV @VARW,@VARA NOTHING ENTERED YET
* VARA SHOULD POINT TO A SPACE LOCATION OR END OF FIELD

READL1
*** ST 1,@ARG-4 STORE BYTE=1 TO ARG+4
    MOVB @H01,@ARG-4 MEANS NO CHANGE IN LINE

READL2
*** DST @VARW,@ARG+5 HAD TO USE ARG-6 BECAUSE OF
    WORD BOUNDARY PROBLEMS
    MOV @VARW,@ARG+6 POSITION CURSOR AT START OF
    FIELD

READ00
*** CLR @VAR1 CLEAR BYTE. I HAD TO USE WORD
    BECAUSE 9900 IS SO MUCH
    FASTER
    CLR @VAR1 COUNTER FOR AUTO REPEAT
* This is where we return to exit INSERT mode.

READ01
*** CLR @ARG-7 USED ARG+8 BECAUSE HAD TO USE
    ARG+6 & ARG+7 ALREADY
    MOVB @H00,@ARG+8 NORMAL OPERATION MODE
*** ST CURSOR,@VARV
    MOVB @CURSOR,@VARV VARV USED FOR CURSOR/CHARACTER

READS1
* Input 1 char and alternate cursor and character for
  blink
*** EX @VARV,RAM(@ARG+5) EXCHANGE @VARV WITH WHATS
    AT LOCATION ARG+5 IN

VDP
MOV @ARG+6,TEMP EXCHANGE VARV,ARG+6
BLWP @VSBR
SWPB TEMP1
MOVB @VARV,TEMP1
BLWP @VSBW
MOVB @RILB,@VARV

```

```

*** CLR @TIMER
MOV B @H00,@TIMER SET VDP TIMER TO ZERO

*** $REPEAT MACRO. REPEAT CODE UNTIL
$UNTIL IS TRUE
L00001 LIM 2 ENABLE INTERRUPTS SO THE VDP
TIMER (>8379) CAN INC

LIM 0 DISABLE INTERRUPTS SO THE VDP
WOH'T GET MESSED UP

*** SCAN SCAN THE KEYBOARD
BLWP @KSCAN SCAN FOR A CHARACTER
*** BS READS2 BRANCH ON COND BIT (EQ) SET
MOV B @STATUS,@STATUS EQUAL BIT SET?
JNE READS2 FOUND A NEW CHARACTER
*** INC @VAR1 INCREMENT THE BYTE @VAR1 BY ONE
INC @VAR1 INC AUTO-REPEAT COUNTER
*** SIF @RKEY .NE. >FF THEN MACRO. IF RKEY NOT EQ >
FF THEN EXECUTE THE
FOLLOWING CODE OTHERWISE SKIP
TO THE $END IF

TERMINATOR
CB @RKEY,@HFF OLD KEY?
JEQ L00002 YEP
*** SIF @VAR1 .HE. 254 THEN HIGHER OR EQUAL
C @VAR1,@H508 HOLD OLD KEY FOR A WHILE
* HAD TO DOUBLE 254 TO SLOW DOWN
ASSEMBLY CODE
JLT L00002 BEFORE STARTING REPEAT
*** SUB 30,@VAR1 SUBTRACT BYTE
S @H60,@VAR1 CONTROL REPEAT RATE
*** B READS3 UNCONDITIONAL BRANCH
JMP READS3
*** SEND IF
*** SEND IF
*** $UNTIL @TIMER .H. 14 TERMINATOR FOR REPEAT
UNTIL HIGHER THAN 14

L00002 CB @TIMER,@H14
JLE L00001 TIME NEXT CHARACTER SWITCH
*** BR READS1 BRANCH COND BIT RESET. USED TO
SAVE ONE BYTE OF

MEMORY
JMP READS1 RESTART CHAR BLINK CYCLE

READS2
*** CLR @VAR1
CLR @VAR1 CLEAR AUTO REPEAT COUNTER

READS3
*** SIF @VARV .NE. CURSOR THEN
CB @VARV,@CURSOR IF NE EXCHANGE AGAIN
JEQ L00003

*** EX @VARV,RAM(@ARG+5)
MOV @ARG+6,TEMP EXCHANGE VARV,ARG+6
BLWP @VSBW
SWPB TEMP1
MOV B @VARV,TEMP1
BLWP @VSBW
MOV B @R1LB,@VARV

*** $END IF
*** SIF @RKEY .L. : : THEN IF RKEY LESS THAN SPACE
THEN EXECUTE CODE
L00003 CB @RKEY,@SPACE IF .LT. SPACE THEN CONTROL CHAR
JLT L00004
B @L0000C

* THIS IS WHERE YOU WOULD TRAP ALL CONTROL CODES *
* HANDLE BREAK CHAR FIRST
* CB @RKEY,@BREAK
* JNE LABEL

* BACK ARROW - SPACE BACK ONE POSITION
*** SEND IF
*** SIF @RKEY .EQ. BACK GOTO RBACK GOTO's DO NOT
REQUIRE AN END IF TERM
L00004 CB @RKEY,@BACK BACK ARROW?
JNE B00002 TO FIX OUT OF RANGE ERROR
B @RBACK

* RIGHT ARROW - FORWARD SPACE
*** SIF @RKEY .EQ. FORW GOTO FORW
B00002 CB @RKEY,@FORW
JNE B00003 TO FIX OUT OF RANGE ERROR
B @RFORW

* INSERT *
*** SIF @RKEY .EQ. INSRT THEN
B00003 CB @RKEY,@INSRT
JNE L00005
*** ST 1,@ARG+8
MOV B @H01,@ARG+8 SET INSERT MODE FLAG
*** SEND IF

* DELETE - DELETE THE CURRENT CHAR
*** SIF @RKEY .EQ. DLETE THEN
L00005 CB @RKEY,@DLETE
JNE L00006
*** CLR @ARG+4
MOV B @H00,@ARG+4 INDICATE A CHANGE IN LINE
*** SIF @VARA .DNE. @ARG+6 THEN THE D MEANS DOUBLE #
OR WORD OF MEMORY
COMPARE
C @VARA,@ARG+6 EMPTY LINE?
JEQ L0001F YEP.
*** DST @VARA,@ARG
MOV @VARA,@ARG MOVE EVERYTHING FROM THE RIGHT
*** DSUB @ARG+5,@ARG DOUBLE BYTE (WORD) SUBTRACT
S @ARG+6,@ARG OF THE CURSOR TO THE LEFT
*** MOVE @ARG FROM RAM(1(ARG+6)) TO RAM(@ARG+5) THIS
IS A BLOCK MOVE OF
@ARG
*** BYTES OF VDP RAM FROM WHATS AT ADDR ARG+6 PLUS 1
TO WHATS AT ADDRESS

```



```

*** ARG+6. IN SHORT MOVE EVERYTHING ON SCREEN ONE
    BYTE LOWER.
MOV @ARG,TEMP2  COUNTER
MOV @ARG+6,TEMP

    INC TEMP      MOVE @ARG FROM RAM(1(ARG+6))
                  TO RAM(@ARG+6)

L00008 BLWP @VSBR
    DEC TEMP
    BLWP @VSBW
    INCT TEMP
    DEC TEMP2
    JNE L00008

*** DDEC @VARA    DECREMENT THE WORD (DOUBLE) AT
                  VARA
DEC @VARA        PRE-UPDATE END OF STRING

*** SIF RAM(@VARA) .EQ. : :-OFFSET GOTO READ01
                  OFFSET IS SCREEN OFFSET >60
MOV @VARA,TEMP
BLWP @VSBR
CB @TEMP1,@SPACE
JNE B00001      TO RESOLVE OUT OF RG ERROR
B @READ01

*** DINC @VARA    INCREMENT THE WORD OF MEMORY
                  AT VARA
B00001 INC @VARA

*** ST : :-OFFSET,RAM(@VARA)
L0001F MOV @VARA,TEMP
    LI TEMP1,>2000
    BLWP @VSBW
*** BR READ01
    B @READ01

* CLEAR - Clear the entire input line
*** SIF @RKEY .EQ. CLRLN THEN
L00006 CB @RKEY,@CLRLN
    JNE L00009
*** SREPEAT
*** ST : :-OFFSET,RAM(@VARA)
    MOV @SPACE,TEMP1
CLRLIN
    MOV @VARA,TEMP  SO WE CAN FIDDLE WITH VALUE
    BLWP @VSBW
    DEC @VARA      PRE-UPDATE END OF LINE
*** SUNTIL @VARA .DL. @VARW DOUBLE LESS THAN
    C @VARA,@VARW UP TO AND INCL FIRST POS
    JNE CLRLIN
*** DINC @VARA
    INC @VARA      UNDO LAST SUBTRACTION
    CLR @ARG+4
    MOV @H00,@ARG+4 INDICATE CHANGE
*** BR READL2
    B @READL2      RESTART EVERYTHING

```

```

*** SEND IF

* GENERAL EXIT POINT
*** SIF @RKEY .NE. CHRTH THEN
L00009 CB @RKEY,@CHRTH ONLY REACT ON CR/UP/DOWN
    JEQ L0000A
*** SIF @RKEY .NE. MVUP THEN
    CB @RKEY,@MVUP
    JEQ L0000A
*** SIF @RKEY .NE. DOWN GOTO READS1
    CB @RKEY,@DOWN
    JEQ L0000A
    B @READS1
*** $END IF
*** $END IF
*** SIF @VARA .DEQ. @ARG+2 THEN DOUBLE EQUAL
L0000A C @VARA,@ARG+2 CHECK FOR BLOCK ON LAST
                  POSITION
    JNE L0000B
*** SIF RAM(@VARA) .NE. : :-OFFSET THEN
    MOV @VARA,TEMP
    BLWP @VSBR
    CB TEMP1,@SPACE BLOCKED?
    JEQ L0000B
*** DINC @VARA
    INC @VARA      POINT BEYOND LAST CHAR IN LINE
*** SEND IF
*** SEND IF
L0000B RT          ENTER THE CURRENT LINE
*** SEND IF (THIS IS FROM THE SIF THAT CHECKED FOR
                  CTRL CODES)

* INSERT ROUTINE *
*** SIF @ARG-8 .NE. 0 THEN INSERT
L0000C CB @ARG-8,@H00 INSERT MODE
    JEQ L0000D
READS4
*** DST @VARA,@ARG
    MOV @VARA,@ARG USE ARG AS TEMP FOR INSERT
*** S WHILE @ARG .DH. @ARG-6
L0000F C @ARG,@ARG+6 MOVE EVERYTHING UP TO CURSOR
                  LOCATION
    JLE L0000E
*** DDEC @ARG
    DEC @ARG      COPY LOWER LOCATION TO HIGHER
                  ONE
*** ST RAM(@ARG),RAM(1:ARG)) GO FROM HIGH TO LOW
                  IN VDP RAM
    MOV @ARG,TEMP
    BLWP @VSBR
    INC TEMP
    BLWP @VSBW
    JMP L0000F
*** $SEND WHILE    TERMINATOR FOR WHILE
*** SIF @VARA .DL. @ARG+2 THEN
L0000E C @VARA,@ARG+2 ONLY UPDATE VARA AS UPPER
    JNE L0000D
*** DINC @VARA
    INC @VARA      HASN'T BEEN REACHED YET

```

```

*** SEND IF
*** SEND IF
*** ST @RKEY,RAM(@ARG+6)
LO000D MOVB @RKEY,TEMP1 DISPLAY THE CHARACTER
MOV @ARG+6,TEMP
BLWP @VSBW
*** CLR @ARG+4
MOV @H00,@ARG+4 INDICATE CHANGE IN LINE
READO5
*** SIF @ARG+5 .DEQ. @ARG+2 THEN
C @ARG+6,@ARG+2 HIT RIGHT MARGIN?
JNE L0002F
*** CALL TONE2 CALL ANOTHER GPL ROUTINE IN
THIS CASE BONK
MOV @H00,@STATUS CLEAR THE STATUS BYTE BEFORE
ACCESSING GPL
BLWP @GPLLNK GIVE A BAD RESPONSE TONE
DATA >0036
*** BR READS1
B @READS1 STAY IN CURRENT MODE
*** SEND IF
*** DINC @ARG+6
L0002F INC @ARG+6 UPDATE CURRENT ADDRESS
*** IF @ARG+6 .DH. @VARA THEN
C @ARG+6,@VARA CHECK FOR LAST NEW HIGH LIMIT
JLE L00010
*** DST @ARG+5,@VARA
MOV @ARG+6,@VARA UPDATE NEW HIGH LIMIT
*** SEND IF
*** SIF @VARA .DL. >2FE GOTO READS1
L00010 C @VARA,@H766
JHE L00011 TO FIX OUT OF RANGE PROBLEM
B @READS1 STILL SOME SPACE TO GO
L00011
* This is where we could scroll the screen if needed
* UPDATE POINTERS IF YOU SCROLL *
*** CALL SCROLL SCROLL THE SCREEN
*** DSUB 28,@VARA
* S @H28,@VARA BACK TO START OF LINE
*** DSUB 32,@VARW
* S @H32,@VARW BACKUP START LINE ADDRESS
*** DSUB 32,@ARG+2
* S @H32,@ARG+2 ABSOLUTE HIGH LIMIT BACKS UP
TOO
*** DSUB 32,@ARG+6
* S @H32,@ARG+6 CURRENT CURSOR POSITION ALSO
*** BR READS1
B @READS1 START WITH SOMETHING ELSE

* FORWARD CURSOR MOVE
RFORW
*** CLR @ARG+8
MOV @H00,@ARG+8 LEAVE INSERT MODE
*** BR READO5
B @READO5 USE REST OF LOGIC

```

```

* BACK CURSOR MOVE
RBACK
*** SIF @ARG+5 .DH. @VARW
C @ARG+6,@VARW CHECK BOTTOM RANGE
JLE L00012
*** DDEC @ARG+6
DEC @ARG+6
*** SEND IF
*** BR READO1
L00012 B @READO1

```

END

```

*****
* THIS IS A ROUTINE TO DIRECTLY ACCESS *
* THE GROM READLN ROUTINE. USE CALL *
* LOAD("DSKL.FILENAME") AND CALL LINK *
* ("DSKL.START") FROM E/A BASIC TO SEE *
* IT BECAUSE OF SCREEN OFFSET. *
*****

```

DEF START

```

GPLWS EQU >83E0 ADDRESS FOR GPL WORK SPACE
H00 BYTE 0
WS BSS >20 MY WORKSPACE
EVEN
START
LWPI WS POINT TO MY WORKSPACE
LI RO,>2
MOV RO,@>8320 START SCREEN ADDRESS FOR SCAN
MOVB @H00,@>837C CLEAR THE STATUS BYTE SO WE
DON'T GET AN ERROR
BLWP @GPLLNK LINK TO THE ROUTINE IN GROM
DATA >2A42
MOVB @H00,@>837C RETURN TOTHE CALLING PROGRAM
ON ENTER
LWPI GPLWS
B @>0070

```

```

* YOU COULD HAVE PLACED AN END STATEMENT HERE AND REF'd
GPLLNK INSTEAD
* OF USING THIS ROUTINE.
* GPLLNK ROUTINE *

```

GPLLNK DATA UTILWS,XGPL VECTOR FOR THE GPLLNK BLWP

```

UTILWS EQU >2094
SUBSTK EQU >8373
FLAG2 EQU >8349
SVGPRT EQU >2030

```

```

H20 BYTE >20
EVEN

```

```

XGPL
MOV @SUBSTK,R1
SRL R1,8
MOV *R14+,@>8304(R1)
SOCB @H20,@FLAG2
LWPI GPLWS
MOV @SVGPRT,R11
RT

```

END

END OF ARTICLE

HELP LINE

"USERS SUPPORT COLUMN"

We hope to be able to HELP anyone with problems. with programs, utilities, etc. In this column We are asking for those with answers to these problems to WRITE, RING or LEAVE a message on the BBS for the EDITOR, or SPEAK to me or any other Director of TisHUG.

The answers printed in this column are not the be all to end all, just answers merely to help each other with tried and used methods.

Question NO.1 *On an IBM computer what is the difference between (High memory, Upper memory) and how can they be used.*

Answer: This answer was taken from the book (memory management)

High memory

The first 64K of extended memory, which can be used almost As if it were extra conventional memory by utilising an advanced programming technique formally known as "smoke and mirrors." Smoke and mirrors works only on 386 or better computers. The 64K in question is often referred to as high memory area, or HMA.

Upper memory

A 384K area of memory that follows immediately after the 640K of conventional memory. On a 386 or better computer, upper memory can be used to hold programs that would otherwise consume conventional memory. This leaves more conventional memory available for DOS applications programs. (ED. Answer)

END OF ARTICLE

TEXPAC BBS PROGRAM DOWNLOAD AREA

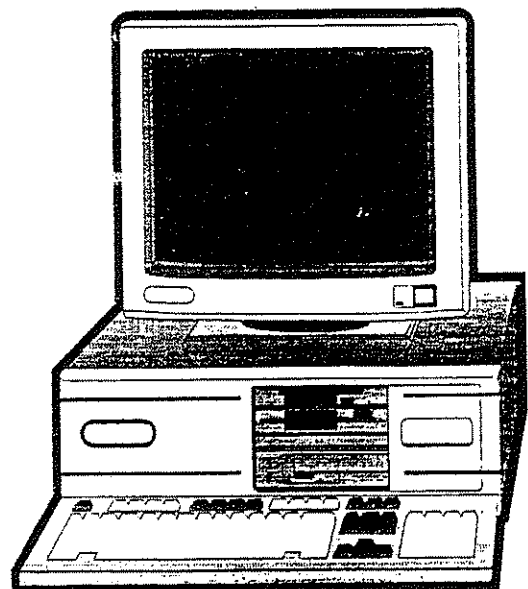
For info on programs read PROG_INFO.

No	NAME	SIZE	No	NAME	SIZE
1	16PUZZLE	(25)	2	AIDKEY	(35)
3	ALPHABET/I	(23)	4	BILLBOARD1	(3)
5	BLACKBOX	(37)	6	BOWLS	(38)
7	CARWASH	(26)	8	CATHAY	(46)
9	CRUTESTXB	(22)	10	CURSOR	(7)
11	DIABLO	(41)	12	FAM/TREE	(48)
13	FAMTREE/PG	(40)	14	GI/DWARF	(45)
15	GOthic	(48)	16	GRAPHPAPER	(4)
17	HELICOPTER	(43)	18	KJTESTXB	(14)
19	KRULL	(44)	20	MATCH-ANIM	(15)
21	MATCHMAKER	(35)	22	MERGE/DOC	(35)
23	MERGE/QL	(7)	24	MIDNITEXB	(35)
25	MIMIC	(12)	26	MOTHGOOSE	(37)
27	P/SCANLOAD	(5)	28	PROGALC	(35)
29	PSI	(45)	30	PSI/16	(32)
31	PSI/GEN	(43)	32	SING	(31)
33	SNAKE	(31)	34	STANLEY	(20)
35	SWORDSORCY	(46)	36	TIKEYS	(20)
37	TILO/V2	(43)	38	TRIANGLE	(43)
39	WAITABIT	(46)			

All of these programs are on the BBS just waiting for you to down load.

Who of you are having trouble with your IBM programs? well leave a message on **YOUR** BBS I' sure that there is somebody with some sort of answer.

Who would like a section on the BBS for IBM Text or Doc files ? Well lety ask at the next meeting.



> ! ! TEKNIKA IS A SUBSIDIARY OF FUJITSU. THIS MONITOR HAS NO !
 > NO ! RGBI/ANALOG ! AUDIO AND REQUIRES AN EXTERNAL SPEAKER-AMPLIFIER !
 > _____ !
 > ! ! THOMSON IS A FRENCH CONGLOMERATE. !
 > YES ! COMPOSITE. ! THIS IS A MARGINAL 80 COLUMN MONITOR. !
 > ! RGBI !
 > YES ! COMPOSITE. ! SAME MONITOR. HIGHER RESOLUTION PICTURE TUBE. !
 > ! RGBI !
 > _____ !

END OF ARTICLE 

PUZZLE

This months list of words is based around the subject
 of "GOLF"

C	P	A	C	I	D	N	A	H	P	Y	H	A	W	S
L	Y	A	W	R	I	A	F	U	E	I	D	R	I	B
U	W	H	D	I	V	O	T	G	N	R	G	D	B	U
B	A	L	L	S	U	T	O	E	A	O	O	N	A	L
H	R	G	S	R	E	B	H	Z	L	U	R	C	G	C
O	C	L	S	R	N	V	A	F	T	G	M	P	S	A
U	O	O	Y	G	F	H	I	G	Y	H	A	S	A	R
S	M	V	U	C	P	K	R	R	A	N	G	E	T	M
E	D	E	T	R	Q	A	O	E	D	L	J	U	C	N
L	X	O	I	Z	S	E	N	E	L	C	F	X	D	W
P	K	W	O	S	G	E	S	N	U	A	X	I	S	Y
J	J	X	H	W	J	S	L	J	V	L	X	J	R	E
B	H	Q	X	I	V	S	I	U	U	S	C	U	K	Q
R	D	U	C	N	F	D	C	A	R	T	X	Z	I	B
Y	E	H	Q	G	X	F	E	E	S	E	A	Y	D	M

Apron
Bags
Balls
Birdie
Bogey
Caddy
Cart
Club House
Clubs
Course
Divot
Drive

Eagle
Fairway
Fees
Flag
Foursome
Glove
Golf
Grass
Green
Handicap
Hazard

Irons
Penalty
Putter
Range
Relax
Rough
Rules
Score
Slice
Swing
Whiff
Woods

Find these hidden words

HOW TO PLAY

In this Puzzle there are (35) words somewhere
 horizontally, vertically, diagonally even backwards

GOOD LUCK !

IN RESPONSE TO YOUR PLEA

by Ron Warfield
 B.C. 99'ers Users Group

From the pages of the west penn 99ers NEWS.

Dear Mickey,

I have 2 HFDC cards and 1 FDC by Myarc and
 have also had the problems that you described.

The problem is that the Myarc controllers have
 difficulty reading the CorComp disk.

CorComp controllers put the disk information at
 the very top of the read sector and TI and Myarc leave
 space at the top.

TI had no problems with this because' they are
 slow, but Myarc controllers are so fast that they miss
 this info.

All I do is "REDISK" - copy the disk on a TI system
 and the problem goes away.

Good Luck.

RARE OFFICIAL TI EDUCATION CARTRIDGES :

THE SCOTT FORESMAN READING SERIES

reviewed by Charles Good

Lima Ohio User Group

These modules resemble PLATO software in that they present specific language arts concepts in a text format and then ask a series of questions to test the student's knowledge of the concept. Unlike PLATO software, the Scott Foresman modules make good use of music and colour graphics. The rare modules DO NOT make use of speech synthesis, unlike some of the more common cartridges in the Scott Foresman READING series. These cartridges seem to be designed for in classroom use, which may be why they were not made commonly available to the public. The "suitable age" designations below are taken from the Fall 1987 TRITON catalog which lists most of these modules.

READING TRAIL

Suitable for ages 8-12, this cartridge teaches about the characters, setting, and points of view in stories. Famous characters from the Wizard of Oz and a separate story about fishing are used to illustrate specific points.

READING POWER

Suitable for ages 8-12, this module teaches research skills involving the dictionary, encyclopedia, and library card catalog. Specifically the student learns how to find information that is organized alphabetically in these kinds of reference materials. A detective story called "The Lion's Charm" with colour animation and music is used in some of these activities.

READING RAINBOWS

This is one of the rarer of the "rare" READING modules. It has been listed in very few catalogs over the years. It teaches how things are alike, parts and wholes, and sizes. Speech synthesis is used effectively. My first grade daughter whipped through this in a very short time, so I assume it is designed for first grade (age 6).

READING WONDERS

Another of the more rare modules, READING WONDERS teaches the student to distinguish between various types of fiction and non fiction. Several colourful stories are used to illustrate what is and is not historical fiction, modern realistic fiction, science fiction, biography, autobiography, and information articles. I suspect that this is probably for ages 11-13

READING ADVENTURES

This uses a variety of stories to teach, within a paragraph, main and supporting details, drawing conclusions, and sequential relationships. I have seen this one mentioned, but not described in catalogs. It looks like about ages 8-10, but I am not sure.

READING CHEERS

I would have guessed this was for 2nd grade, but my 1987 TRITON catalog says ages 8-12 (2nd grade is age 7-8). The module teaches root words with endings (lazy and lazily), contractions, and compound words.

All of the above "rare" modules are c1983. To complete the record I will briefly describe below the more commonly available 1982 Scott Foresman titles in the READING series.

READING ON

Some nicely illustrated science fiction stories illustrate the use of maps, schedules, graphs, and why and how people use them. For ages 8-9

READING FUN

There is minimal use of speech synthesis in this 2nd grade level module. Four colourful stories illustrate problems and how people solve them, why things happen, and how characters feel.

READING ROUND UP

Four stories based on an "American Wild West" theme are used. Concepts taught are figures of speech, word meaning, and idioms. The module is designed for ages 9-10.

READING FLIGHT

For ages 11-12. A neat story about an archaeological dig on a south seas island called Bolo Island teaches classifying, summarizing, and outlining information.

END OF ARTICLE 

FAMOUS (?) QUOTES

An alcoholic is defined as someone you don't like who drinks as much as you do.

Dylan Thomas
(1914-1953)



POWER YOUR ENTIRE TI SYSTEM ALMOST ANYWHERE

WITH AN INVERTER

by Charles Good

Lima Ohio User Group

In the June, July, and August 1989 issues of *Micropendium* Jan Janowski had a series of articles describing how to run a small TI system in the field powered by a small 12 volt battery. (The articles originally appeared in the Chicago User Group newsletter.) Jan's "portable TI" included 32K, PIO port, and a ramdisk all in the console. There were no disk drives. All this required much complicated custom wiring and soldering. There is an easier way. With NO MODIFICATIONS WHATSOEVER you can run your entire TI system (console, PE box full of cards and two floppy drives, small monitor, and maybe also your printer) for between 30 minutes and several hours powered by a 12 volt automobile or (better) deep cycle battery. All you need is an appropriate inverter!

Inverters convert 12V DC current into 117V 60 cycle AC current. Any AC product can be run from a 12V DC power source such as a battery with a modern inverter. Inverter technology has improved considerably in the last 10 years. Old style inverters were very inefficient (50-60%) and produced "square wave" AC which is unacceptable to most electronic devices such as TVs, stereos, and computers. Modern inverters produce "modified sine wave" AC which is acceptable to almost everything electronic including TI99/4A equipment. One manufacturer now offers a true sine wave inverter said to be compatible with the most sensitive electronic equipment. These modern modified sine wave and full sine wave inverters are 80-98% efficient and draw only very small amounts of power when on standby.

The STATPOWER PROWATT 250

Is a 90% efficient modified sine wave inverter will power a complete TI99/4A system from a battery if you use a small monitor such as a 4" BW TV. The STATPOWER 250 will deliver the following AC current: 500 watt surge, 300 watts for 10 minutes, 250 watts for 30 minutes, and 200 watts continuously. Output voltage is very well regulated at 60 cycle and 115V +/- 5% until automatic shutdown if the battery voltage gets down to 10V. This is better voltage regulation than the power

company provides! Just plug in a multiple plug power strip into the single female 3 hole outlet at the AC end of the inverter, and plug the TI99/4A components into the power strip. You can run a console (40 watts), a full PE box (150 watts), and a 4 inch BW TV (16 watts) on the STATPOWER 250 powered by a car battery. I have done so. Run time is at least 30 minutes and may be greatly extended if you have a "switching power supply" in the console or if your small TV doesn't need the inverter because it can be run directly from a 12 volt source. (My small TV draws 16 watts AC from its inefficient (always warm) AC transformer, but draws only 9.5 watts when powered directly from a 12 volt battery). The main limit to run time is the power stored in the battery. If you use a 100 amp hour deep cycle battery (commonly available at department stores like Sears, Walmart, and Myers) where electric fishing "trolling" boat motors are sold, you can run all the above computer equipment for over 4 hours without discharging the battery more than 80%. Unlike automobile starting batteries, deep cycle batteries are designed to accept 80% discharges repeatedly (200+ times) with no significant loss of battery performance.

The STATPOWER 250 has a cigarette lighter plug and can be plugged into a car's cigarette lighter. However, most automobile lighter plugs are fused at 10 amps DC, which means a maximum of 120 watts of power is available from this source. To run a PE box equipped 99/4A you need to cable the STATPOWER 250 directly to a battery. The necessary 1 foot cable for car battery use with clamps for the - and + battery terminals on one end and a female cigarette lighter plug at the other end is available from Radio Shack, WalMart, and many auto supply stores.

As of early August 1993 catalog prices for the STATPOWER 250 vary from \$198-\$229 plus shipping. It is available from REAL GOODS (phone 800-762-7325) or SUNELCO (phone 800-338-6844). Phone these dealers and ask for their catalogs. They both sell inverters of larger and smaller capacity and both dealers can also set you up with a solar panel to charge the deep cycle battery you will use to power your TI system at remote locations. Smaller STATPOWER inverters not powerful enough to run a full TI system have recently appeared in the C.O.M.B. catalog. These smaller inverters can be used to run small AC electrical devices from a car's cigarette lighter. 140 watt AC continuous, item P3522-9506, for \$70. 125 watt AC continuous, item v3661-9505, for \$70. 50 watt AC continuous, item v3662-9505, for \$50. C.O.M.B.'s phone is 800-328-0609.

END OF ARTICLE 

HOW TO FIT VERY LARGE TEXT FILES INTO TI WRITER

ORIGINALLY PUBLISHED IN LIMA NEWSLETTER
MAY 1994

by Charles Good
Lima Ohio User Group

Text files ported over from other types of computers and converted into DV80 format for use with the TI are sometimes enormous, occupying several hundred sectors. Examples are Mike Wright's CYC and the Sherlock Holmes stories I have recently converted to DV80. Sometimes these large DV80 files are too big to fit into MYWORD on a Geneve, and they are certainly too big to completely fit into the memory available to 40 column 99/4A users. Users of 40 column systems can use DM1000 or DSKU to view these files on screen because only little pieces are loaded into memory at one time, but the view leaves something to be desired. Each 80 column line wraps around to a second line on the 40 column display and words are often split in odd places in the middle of a line.

If only 40 column users could load these very large DV80 files into the Funnelweb editor, or other version of Ti Writer. Then the text could be viewed in 80 columns by scrolling left/right. This can be done! I did it with the CYC. What you need to do is break the large DV80 files down into a series of smaller files, each small enough to fit into the TI Writer text buffer. Here is how to do this.

The key is to use the format "number space number space filename" when using L(oad)F(ile), S(ave)F(ile), and P(rint)F(ile) to split the large DV80 file into smaller pieces. This means, "The first line number of the text in the disk file or edit buffer, a space, the last line number of the text in the disk file or edit buffer, a space, the file name." Another thing to remember is that no version of TI Writer will handle line numbers greater than 9999. I am going to assume you are using any version of the Funnelweb text editor to split a file

- 1- Put a disk with the large TEXT file in DSK1. Put a blank initialized disk in DSK2. (You can use any other drive numbers you if you want).
- 2- Boot the Funnelweb text editor and from the command line type "LF" and <enter>. Then type "0001 9999 DSK1.TEXT" for a file name, complete with spaces, and press <enter>. TEXT

will load into memory until you get a BUFFER FULL message.

- 3- From the command line goto the end of the text in memory by typing "E" or "9999" followed by <enter>.
- 4- Probably the text in memory ends in the middle of a sentence. Move backwards through the text until you find a convenient place to end the small text file you are about to create. This can be at the end of a chapter, end of a paragraph, etc. Note the last line number of the part you want to save. Write it down. etc call it

LASTNUMBER.

- 5- Go to the command line and type PF and <enter>. (If you use PF rather than SF, then the original LF name stays in memory when using Funnelweb). Type "1 LASTNUMBER DSK2.TEXT-A" for the file name and press <enter>. Of course LASTNUMBER is a line number, not the letters L A S T etc. This will save the first chunk of your large DV80 file as a smaller DV80 file that will fit into TI-Writer's edit buffer.
- 6- Go to the command line and type "LF". The previous LF name "0001 9999 DSK1.TEXT" should appear if you are using Funnelweb. Change this to "LASTNUMBER 9999 DSK1.TEXT" and press <enter>. The computer count through the lines of the large file until it gets to line LASTNUMBER, and it will then load the next part of the large file into memory until you get a BUFFER FULL message.
- 7- Again go to the end of the text in memory by typing "E" or "9999" from the command line and pressing <enter>.
- 8- Move back through the text until you find a convenient end point (paragraph or chapter end) and write down this line number, calling it NEWLASTNUMBER.
- 9- Go back to the command line, type "PF" and <enter>, and type "1 NEWLASTNUMBER DSK2.TEXT-B" as the next file name before pressing enter.
- 10- This is important! You have to calculate the starting line number at which you will begin loading in the next segment of your large TEXT file. Use the formula $LASTNUMBER = LASTNUMBER + NEWLASTNUMBER$. Add these two numbers together to get the next LASTNUMBER.

- 11- Go to step 6 above and repeat steps 6-10 as many times as are needed to split TEXT up into TEXT-A, TEXT-B, TEXT-C, TEXT-D etc.

This is how I converted the BIBLE and the CYC into small files after using PC TRANSFER to convert large text files on a 360K IBM disk to a DSDD TI disk. You may want to do the same thing with some of the Sherlock Holmes stories and other book length manuscripts that are appearing in the Lima User Group software library.

END OF ARTICLE 

TRIVIA QUESTIONS

- 1. What was the 91st reason given in TI's November-December 1983 magazine advertisement entitled "91 REASONS TO BUY A TEXAS INSTRUMENTS HOME COMPUTER"?
- 2. What was the first software product released for the TI-99/4?
- 3. Name the first third-party software modules to be announced for the TI-99 Home Computer and who manufactured them?
- 4. What was the first software product released for the TI-99/4A that would not run on the 99/4?
- 5. What was the last software product scheduled to be released for the TI-99/4A (hint: it was never released)?
- 6. What was the best selling TI-99 entertainment software product?
- 7. What was the best selling TI-99 home use or personal productivity software product?
- 8. Which single TI produced / licensed software product carried the highest suggested retail price ever?
- 9. What is the single most expensive peripheral ever offered for the TI-99 by TI or any third-party manufacturer?
- 10. Name the company who in February, March and April 1983 claimed to be working on a version of Zaxxon for the 99/4A only to ultimately drop the project because they couldn't fit the program into an 8K module, and they didn't feel there were enough 99/4A disk systems sold to justify producing a disk version?
- 11. Name the author of Arcturus, the Zaxxon clone that was released as an I/O port module by Exceltec in 1985.
- 12. Name the person who "is" the company named Video Magic?
- 13. Who are Jim Chaffin and Bill Maxwell?
- 14. Name six Texas Instruments produced TI-99 products that were discontinued between 1979 and 1983?
- 15. Who is Bounty Bob?
- 16. Galen Read, the talented assembly language programmer who gave us WriterEase and Console Calc (aka TI-Planner), and who owned the now defunct Innovative Programming business, started out in the TI-99/4A business world as a partner in what firm?
- 17. What TI-99 program was Texas Instruments referring to in its February 1982 advertisement that read; "THE MONSTER IS FREE!"?
- 18. Who is Leslyn Tepper?
- 19. What is the name of the TI-99 product that carried product number PHA 2500?
- 20. Who is the author of TI Mini Writer PHT 6103, the Mini Memory based word processor that Texas Instruments announced but never released, that was picked up by TexComp and initially sold as part of an entry-level word processing system with the Mini Memory module and an Axiom GP-100TI printer making up the package?
- 21. On what piece of TI-99 hardware would one find the built-in EQUATION CALCULATOR?
- 22. What hardware peripheral did the TI-99/4 and the TI-99/4A support, that TI did not provide until January 1983?
- 23. What obvious visual feature separated TI's disk based software produced for use with the original stand alone Disk Memory Drive PHP 1850, from that which was produced after the Expansion Box Disk Drive was introduced?
- 24. What TI produced Solid State Software command module carries the earliest (oldest) copyright date?
- 25. Who was the 1st Place Winner in TI's Author Incentive Program that took place from August to November 15, 1980?

TRIVIA ANSWERS




- 1A: the \$50 rebate.
- 2A: According to the 1979 TI publication carrying Part # 221006, which appeared in August 1979, no one program can take credit for being first. The programs listed as being available at that time were; - Beginning Grammar, Early Learning Fun, Football, Home Financial Decisions, Household Budget Management, Number Magic, Physical Fitness, Video Chess and Video Graphs.
- 3A: Connect Four, Hangman, Yahtzee and Zero Zap by Milton Bradley. Announced in December 1979.
- 4A: TI-Writer, released in August 1982.
- 5A: Wingwar. Scheduled for released on December 8, 1983.
- 6A: Parsec
- 7A: Household Budget Management

- 8A: Speaking Scholastic Spelling Teacher Library PHL7008 at \$219.80
- 9A: The Myarc WDS/100 15mb Hard Disk System priced at \$3399.00
- 10A: DataEast of Chatsworth, CA.
- 11A: Bill Bies.
- 12A: John Phillips
- 13A: The founders of DLM Arcademics, manufacturers of Alien Addition, Dragon Mix and other educational software modules released in 1983.
- 14A: The TI-99/4 Home Computer, Disk Drive Controller PHP 1800, Disk Manager PHM 3019, Disk Memory Drive PHP 1850, Extended Basic v100 PHM 3026, 32K Memory Expansion Unit PHP 2200, MunchMan disk version PHD 5060, P-Code Peripheral PHP 2400, RS-232 Interface PHP 1700, Thermal Printer PHP 1900, Spell Writer Cassette version PHT6042, Telephone Coupler (Modem) PHP 1600, Thermal Paper PHA 1950, TI Invaders disk version PHD 5058, TI LOGO PHM 3040, Tombstone City: 21st Century disk version PHD 5057, Video Controller PHP 2300.
- 15A: The miner in Tigervision's MINER 2049er module.
- 16A: Silver Wolf Software.
- 17A: MunchMan.
- 18A: The editor of Mini-Mag 99.
- 19A: The Math Speech Dictionary or Math Speech module, which is listed in TI's 1980 Dealer Suggested Retail Price Lists, but no where else after that.
- 20A: Jerome Prinkl.
- 21A: On the original TI-99/4 Home Computer. It is listed as Option 2 from the Selection Menu under TI Basic.
- 22A: A Program Recorder (cassette tape player/recorder).
- 23A: The labels on the disks were upside down so they could be read correctly when being inserted into the horizontally placed disk drive.
- 24A: Home Financial Decisions-1978. Source is the 1980 TI Publication entitled APPLICATION PROGRAMS FOR THE TEXAS INSTRUMENTS HOME COMPUTER that carries Part # 221068.
- 25A: Charles Ehniger won the \$3000 first prize with his Household Inventory System entry.

END OF ARTICLE 



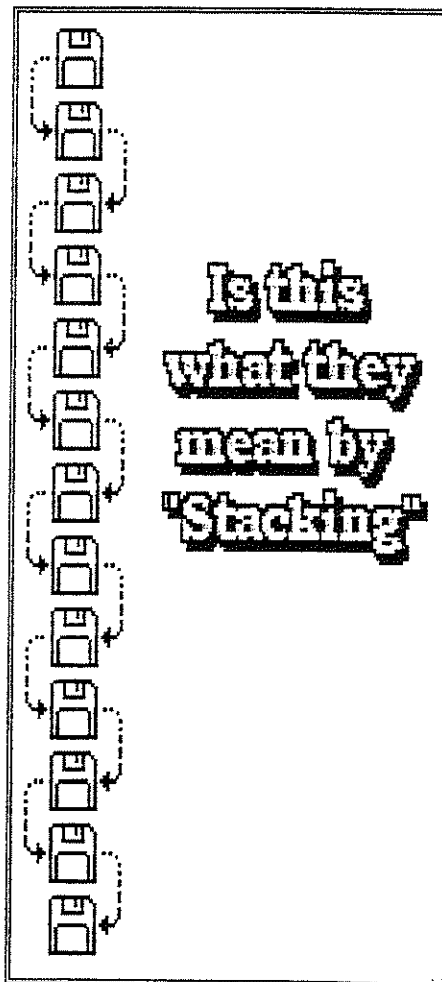
How to contribute to your Magazine

All  or  or  posted to C/O 3 Storey St Ryde 2112 Australia

We are able to publish articles forwarded to us in the following manner.

- Printed letters or articles
- Articles left on the TI, BBS, IBM or TI computers can leave TEXT files. (PH. NO.02 4564606) *Free to all TISHUG members*
- TI Computer floppy disks....5.25" DSDD or DSSD.....Text files, Funnelweb or TI Writer
- IBM compatible Floppy Disks....5.25" or 3.5", we can process - text files, Word for Windows ver: 1.0 - 6.0, WordPerfect, and Word for Macintosh ver: 5x. (on a IBM formatted disk)

These items can be posted to the above address or could be handed to the Editor or one of the Club Directors. Please put your name on the disk so it can be returned



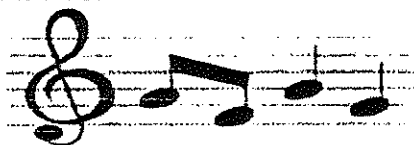
USING A MICROPHONE WITH YOUR SOUND CARD

By Loren West

What fun you can have with a microphone... This story is to point out that your sound card might have a "MIC" connection. There's nothing special about this but how many people have put this simple feature to use. I've used the Sound Blaster 16 sound card, which has some interesting program's included on the installation disk's.

Click on the "sound blaster 16 icon when you're at the program managers screen display and you should then be looking at a range of icon's with titles like "Soundo'LE", "Creative Mixer", "Creative Mosaic" "Wave Studio" plus more. Creative Mosaic is a tile type game with sound. Creative Mixer. I use to adjust the levels of the CD output MIC input. Volume etc.

First thing to do is connect a microphone to the MIC socket (3.5mm plug, mono) I used an old microphone that I pulled out of a radio cassette recorder, soldered a length of shielded microphone wire (purchased from Dick Smith's) and then the plug was connected. Plugged it in, adjusted the MIC level in the Creative Mixer program and presto I had sound coming through the speakers also hooked up to the sound card.



That was the first part of the fun.

Now to record your voice onto the computer. I used the Wave Studio, click the mouse cursor on the red LED to record, which brings up some other selections to fill out then you're on the air, when you are no longer talking or singing or what ever, you press stop.

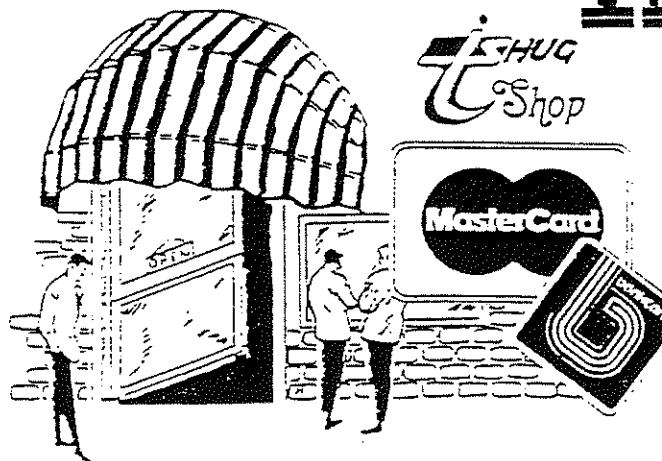
To play back what you have just recorded you press the play button, just as if you had a cassette recorder in front of you, while in the Wave Studio you can edit the sound that's been recorded, add echo, reverse the sound, increase the volume, etc. Then save the sound if you want or make other sounds.

I moved the sounds after they had been recorded, to the Windows directory so I can use them when Windows starts up or error messages, this amuses family and friends when they use the computer.

Of course this is only the start you can also record songs, sounds from a CD if you happen to have a CD ROM in your system, there is also a inlet to plug other sound devices into the computer via the LINE INPUT plug

Hope you have fun trying these simple idea's out, so let us know via the magazine how you went. See you at the meeting.

END OF ARTICLE



The IBM SHOP

with Cyril Bohlsen

42mB Seagate ST351A/X HDD IDE	\$ 45.00
5.25" Disk storage box (100 cap)	\$ 10.00
Parallel printer cable 2M	\$ 5.00
3 Button mouse	\$ 14.00
Mouse pad	\$ 1.50
3.5" FDD mounting kit	\$ 6.00
3.5" power adaptor cable	\$ 6.00
Mouse adaptors 25M to 9F/M	\$ 5.00
Mouse adaptor 25F/M to 9M	\$ 5.00
15-9 pin "D" adaptor for Monitor	\$ 6.00
Joystick "Blastick" (IBM)	\$ 20.00
CPU fan & heat sink	\$ 8.00
486 SX-25 CPU 5V (Intel)	\$ 20.00
486 DX-2/66 CPU 3.45V Texas Inst.	\$ 60.00
486 DX-2/66 CPU (IBM)	\$ 60.00
4mb Simm 72 pin 70ns with Parity	\$ 200.00
1mb Simm 30 pin 70ns with Parity	\$ 55.00
256k Simm 30 pin with Parity	\$ 20.00
30-72 pin Simm adaptor	\$ 25.00

KTX Work & Play Mega Multimedia Pack

Quad speed CD-ROM 16 Bit Sound Card	
Stereo Speakers, M/S Titles	\$ 300.00
Hex speed CD-ROM 16 Bit Sound Card	
Stereo Speakers, M/S Titles	\$ 360.00

For current pricing of items not listed please contact Cyril Bohlsen at the general meetings or Phone (02) 639 5847

NOTE : All prices listed are at time of printing, and may change at any time. Prices do not cover posting and packaging.

TECH TIPS



Tech Tips originally published in Your Computer January 1994:

- Quick text files
- Slow Memmaker
- DOS upgrade
- Saving CD games
- A bucket of 486 chips
- Compressed Pocket
- Hints for DOS utilities
- Switching off hard disk?

Quick text files

A few years ago I discovered a little program that has become very handy. This program is called Txt2com. This is a free program created by Keith P. Graham except that the resulting com files may not be included in a commercial package (not freeware, shareware or user supported).

Txt2com is a simple program which makes an ASCII text file into a full screen display. It is useful for documentation and for 'readme' files which can be shown on the screen and then printed out. The com file created by Txt2com is easy to use and intuitive. There is a help line on line 25 and an optional title on line 1. The cursor controls move around the file and the Esc key exits.

This is a great way to send your text files to another person so that they can read your

file presented in an attractive way, it beats using the DOS Type and More commands.

Creating a text com file is easy. Enter at the dos prompt:

Txt2com textfile comfile heading
where 'textfile' is the name of the ASCII textfile to be converted, and 'comfile' is the name of the com file to be created. If you leave off the *.com extension, the program will add one. Filenames can include disk and paths. As a practical limit, text files should be no larger than 50KB.

'Heading' is anything you want to appear at the top line of the display and can be any character (except DOS' pipe and redirect characters) and any number of words. Headings are automatically centred.

You can send the text file to a file or to the printer by choosing 'T' option while the com file is displaying the file. The program will ask for a file name. Simply enter the name of the file at the prompt. If you want it to go to a printer, enter LPT1 or LPT2.

You can control screen attributes through the use of the tilde character (~). The default colour is 07 which is white on black. The attribute digits will change this. In the readme file that accompanies Txt2com there are examples of screen attributes showing the sequence and a colour example of the result of that sequence. For example '~34' displays red text on a light blue background, and '~DB' displays flashing light blue text on a purple background!

I find it is so easy to change my ASCII text files to com files, and then for myself or a recipient to view the file they simply type the file's name (for example 'readme').

When I boot my system up I found I did not always want to work with a specific Windows application, DOS application or load a memory hungry menu system, I would rather work directly from the DOS prompt. I load my own menu, then I am taken to the DOS prompt where I can select my task from the command line.

The menu I created by writing instructions into a text file and then converting it into a com file using Txt2com. It's called WELCOME.COM, so I put the file name Welcome on the last line of my autoexec.bat file so after boot-up the customised menu is displayed on the screen.

(I have created a macro.bat file that loads all my saved Doskey macros and the instructions and descriptions on how to use them are contained within my simple menu. I find I work most efficiently this way.)

Robyn Dodson

I've put Txt2com on our January Utility Disk, available from the Sydney PC User Group. See 'What have you missed?' on page 35 for details on getting it.

Slow Memmaker

I have been reading and enjoying your magazine for about two years now. Of all the segments, I find 'Tech Tips' the most interesting and the Tech Tips Digest is one of my most valuable references. I have a question, though, that I have not seen dealt with.

I recently installed MS-DOS 6, and proceeded to run Memmaker, to optimise my memory. I chose express setup, which aborted the first time and suggested I try again with conservative settings. I did so, and when finished, the program informed me that it had changed my free base memory from 573,344 bytes to 623,344. It also claimed that it had saved my old autoexec.bat and config.sys files as autoexec.umb and config.umb which, true to form it would seem with this kind of thing, it failed to do.

Upon checking the speed with Landmark version 0.99, after the Memmaker changes I found that the processing speed had dropped from its usual 31.7MHz to 21.0MHz. Why is this so, and if this is the usual case, why on earth would I want to inflict this program on my system again?

Tim Browne

The only thing that Memmaker has done to your system is to load emm386.exe to allow drivers to be loaded into upper memory blocks (UMBs), and to load as many drivers and resident programs as it can find into these blocks.

The version of the Landmark test that you're using is also pretty old, and to be honest I've never had much faith in the figures that the Landmark test presents, especially for 386 and 486 machines.

However, I did try to reproduce your results with Landmark version 2.0, as well as Norton's Sysinfo 7, without success. In all cases the presence of himem.sys and emm386.exe made no perceptible difference to the benchmark results. All I can suggest is that one of these drivers is causing the timing loop in the old version of the test to take longer to execute, and presumably the new version uses a test that isn't affected by the presence of those drivers.

Something you always have to bear in

mind with benchmarks is that they're not real applications. They're written to model real applications to a certain extent, but as with all modelling, you have to stand back occasionally and ask whether the model is an accurate representation of reality. In this case, I suggest that Landmark 0.99 is probably not.

DOS upgrade

The May issue, in advocating shareware through the Sydney PC Users Group, improved your already excellent magazine. I was quite excited when I read the report on Dcopy 1.3 — here at last there seemed to be an answer to my problem.

My wife and I each have a 486SX/25 running DOS — the only difference being that I have both size floppy drives, and she only has a 3.5-inch drive. The DOS 6.0 upgrade originally came only on 5.25-inch disks, and I upgraded with no problems.

I tried to make a 3.5-inch copy to install it on my wife's machine, using every version of Diskcopy or Copy that I knew. I even tried, unsuccessfully, to format a 1.44MB disk to 1.2MB. In all cases, the new 3.5-inch disk would abort half way through the loading of the upgrade, asking that I install the very disk that was in the drive.

After getting a copy of Dcopy13, it ran through without a hitch onto a formatted 1.44MB disk. The result was identical to the previous six failures. My computer gets a lot of use with no other problems, and I have had a lot of successful use of Diskcopy and Copy. What am I doing wrong?

Bruce Norman

The reason you can't create a Diskcopy of a 1.2MB disk on a 1.44MB blank, is that the two have different numbers of sectors — a 1.2MB floppy has 15 sectors per track, a 1.44MB has 18. Even though you could conceivably copy a 15 sector image onto a 1.44MB disk, DOS, or rather the BIOS, won't allow it.

But aside from the hidden system files (io.sys and msdos.sys), there's nothing magical about the arrangement of the files on the DOS installation disk, and any old copy program will do for copying them from one floppy to another.

I suggest taking a blank formatted 1.44MB floppy, and first transferring the system files onto it. Since you already have DOS installed on one hard disk, place the blank floppy in the 3.5-inch

drive on that machine, and from the hard drive (C: prompt) type 'sys a:' (or sys b:, if that's the letter of your 3.5-inch drive).

Then use Xcopy to copy the remaining files on the first floppy onto the new system disk you've created. Xcopy is faster than Copy when you are copying from one floppy to another, since it will read as many short files into memory as it can before writing them to the other disk, rather than doing it one file at a time.

Repeat this Xcopy process for each successive disk in the set, except that you don't need to put the system files on them first. You'll waste a bit of space on each disk, but you shouldn't have any trouble installing from the copies.

By the way, I am assuming that you have two copies of DOS for the two machines — installing only one copy of DOS on two different machines would be a breach of the software licence.

Saving CD games

We have a number of CD-ROM disks, with programs which can be run directly from the CD. This caused problems if we wanted to save configurations or other data, or save games.

There is an answer, using DOS' Append. It enables programs to open data files in specified directories. I made a directory called 'cdgames' on the hard drive, and each game we want to play from the CD has its own sub-directory under cdgames.

Each game also has a batch file. For example, Monster Bash, which is in a directory called 3607 — on the World of Games CD — in the F: drive, has this batch file on the hard drive:

```
append /x:on
append f:\3607
bash1
```

Some of our games take a lot of disk space — Monster Bash takes over 1MB, and Cosmo takes nearly twice as much. Using Append we can easily play lots of games and still leave room on the hard drive for mum.

Richard Vagg

Append behaves in a similar way to the DOS Path statement, except that the latter only works for program files (those ending in *.com, *.bat, and *.exe) while Append works for all other files. By using the /x:on switch, Append can also search for program files, so you don't have to add the directory to the path as well. By

using Append in this way, any failed attempt at opening a file in the current directory will then try to open that file in the Appended directory or directories.

Be very careful when using the Append command, as it can sometimes behave in unexpected ways. For example, you can't create a new file in the current directory if a file of the same name exists in an Appended directory. But if you open a file in the Appended directory with a word processor, and then save it, the way that most word processors work means that the file will probably be saved to the current directory.

For this reason, it's a good idea to cancel any Append command as soon as you have no further use for it, using 'Append ;'.

A bucket of 486 chips

I've been reading the promotions for the different 486 chips and systems. But what is the bottom of the line performance difference between identical 486DX 33 and 50 systems, compared with 486DX2 50 and 66 systems?

Derek Barker

Intel has caused a fair bit of confusion with the introduction of its clock-doubled processors. Are they really twice as fast as a non-doubled chip or not? Excuse me while I make a brief historical diversion... The first 486 was a 25MHz chip, which built the functionality of a 386 (with considerable enhancements), a 387 maths co-processor, and an 8KB cache into a single chip for the first time. This was soon followed by a 33MHz version, as manufacturing bugs were ironed out and yields of the faster chip rose.

This isn't the place to go into the operation of caches. Suffice it to say, a cache is a relatively small chunk of fast memory that keeps track of recent accesses to a much larger block of slow memory, so that if a memory location is accessed, and the contents of that location are already in the cache memory, it can be read by the processor much faster than if it had to be read from main memory. When the cache is able to supply information in this way, it's called a cache 'hit', and if it's not in the cache, it's a cache 'miss'.

Just to really confuse everybody, Intel also started to sell chips which didn't quite run at the specified speed, and/or

in which the maths co-processor was faulty. This was the 486SX, which was originally launched as a 20MHz part, later supplemented by 25MHz and 33MHz versions.

The confusion arose because the SX suffix was previously used in conjunction with the 386, to signify a 16-bit external bus instead of a 32-bit one. Some people extended this interpretation to the 486SX, which has a full 32-bit bus, rather than the more general intention of a 'lower spec' chip. That's all by way of an aside, but it covers the other third of the 486 line-up.

Next kid on the block was the 50MHz 486DX (the DX suffix suddenly appeared in reference to full-spec 486s as soon as the SX was launched), which Intel decided was the next logical step after the 33MHz version. Trouble was, motherboard manufacturers had difficulty in producing boards that could operate at this speed, so not a lot of 'true' 486DX/50 machines were built, and those that were attracted a significant price premium.

So Intel launched its first clock doubled chip — the 486DX2/50. This chip was essentially a 50MHz chip in a 25MHz chip's clothing. It ran from a 25MHz clock, but internally the clock was doubled to run the thing at 50MHz. This meant that any processor operation which was totally internal to the chip proceeded at twice the speed of a 25MHz part, but external operations were handled at 25MHz speeds.

With any processor before the 486, the speed improvement would be marginal, if any, because the chip has to continually refer to external memory in order to retrieve the next instruction in the program, not to mention any data that also has to be transferred to or from main memory. But the 486 has an internal cache, so any instruction or data which is present in the cache when the processor calls for it, can be accessed at 50MHz.

It's only when external memory or I/O operations (which can't be cached) are called for, that the chip is limited by the 25MHz external speed. How often this happens depends on the software being run, and how frequently cache hits occur. Short repetitive routines like loops can often fit entirely in the cache, but 8KB is not very big, so a certain percentage of reads have to come from main memory.

So a DX2/50 is twice as fast as a DX/25 for instructions that are read out of the cache, but this is degraded somewhat by

the need to occasionally access main memory, so I use a ball-park figure of about 80% as the level of improvement of the clock-doubled chip over the non-doubled one. Now of course, we have the DX2/66 as well, which operates externally at 33MHz.

I'd rate the four chips you mention in the order DX/33, DX2/50, DX/50, and DX2/66. Using that 80% ball-park figure (some say it's closer to 90% for DOS applications), these chips have equivalent clock speeds of 33MHz, 45MHz, 50MHz, and 59MHz. So there's not a lot between them, but whatever figure you use for the performance improvement from clock doubling, the DX2/66 is somewhat slower than twice the speed of a DX/33, and the other two are in between.

Compressed Pocket

This is for those people who, like me, use a Phd (Pocket Hard Disk) and have upgraded to DOS 6.0. The Phd is a handy little device which serves both as a file backup unit, and a means of taking my computer environment with me to a foreign PC — in my case, to TAFE classes.

In an emergency (as in failure of my main hard drive) it becomes a reserve unit, holding all my main applications and data, requiring only a floppy boot disk to be up and running again. The boot floppy is needed because DOS does not allow a drive connected to the parallel port to be a bootable drive. Obviously, DOS 6's data compression is very useful on such devices, so I was looking forward to installing it. On my C: drive this proved straightforward, but not so on the Phd.

As you know, Doublespace creates a special file to hold the compressed data, and assigns a drive letter to it. Compressed data on drive C: is assigned the letter C:, and the real drive C: becomes drive H: (the host drive). So far, so good. Add an additional drive holding your compressed data, and two more drive letters are needed — this is automatically done for you, and DOS keeps a record that they have been assigned.

Now for some reason, DOS 6.0 treats the Phd as a 'removable media' drive, so you must add the line 'doublespace /mount x:' (where x: is the drive letter) to the autoexec.bat file. That's where the problem appears.

Each time the computer is switched on, the drive letter to be mounted changes; first it was I:, then J:, then K:, and so on. I was rapidly running out of alphabet. It appears

that the Phd device driver software assigns itself the next available drive letter; DOS 6 uses that and one more for the compressed drive, and stores it. Next time the Phd assigns itself the one after that.

With no successful suggestions coming from the suppliers, my eventual solution was a batch file, to be run as the last thing before switching off. There is no point including it in the autoexec.bat file, as by the time this starts both the Doublespace and device driver have run and the problem already exists.

The batch file (bye.bat) simply resets the dblspace.ini file to a default, so that the leapfrogging no longer takes place. It does this by keeping a standard copy of dblspace.ini (which I called dblspace.phd), which is copied back over dblspace.ini prior to switching off. This is made more convenient by linking the batch file to a function key (using ansi.sys); I also added a reboot option, when that is all that is required.

There may be a more elegant solution in the pipeline, but meantime this works, and my data is readily accessible again.

Bye.bat:

```
dblspace /unmount i:
attrib -r -s -h h:\dblspace.ini
copy h:\dblspace.phd h:\dblspace.ini
echo ... switch off, or press any key to
```

```
reboot
pause>nul
reboot
```

dblspace.phd:

```
MaxRemovableDrives=2
FirstDrive=D
LastDrive=H
MaxFileFragments=116
ActivateDrive=H,C0
```

Jim Kenworthy

I don't have a Phd here to test it with, but it might work in the autoexec.bat file also (though not with the 'unmount' command). Since Doublespace has already loaded by the time it is executed, changing the dblspace.ini file at that point shouldn't affect the current drive assignments, but the file will be there when you next boot up. That way you don't have to remember to type in 'bye' before turning the machine off — it also means that the drive letters will be correct after a crash.

The 'reboot' command Mr Kenworthy used is copyright, so we are unable to reproduce it here for copyright reasons. However utilities which perform this function are readily available from user groups and bulletin boards, and in any case, it's not central to the operation of the batch file.

Hints for DOS utilities

One problem I've found with a lot of farm programs, is that they ask you if you want to backup your work on exit. This is a good idea, but most look for the DOS Backup command, which in DOS 6 is Mbackup. My solution to this problem was to expand the DOS 5 backup.exe file to my DOS directory.

Another small problem I've run into is if Windows is installed after DOS 6, the path needs to be changed so that c:\dos is before c:\windows, otherwise commands like smartdrv or MSD call the (older) Windows version.

Allen Hill

Switching off hard disk?

I have a fax/voice/data modem and some fax software on my PC. To receive all incoming faxes and voice messages, I must leave my PC and modem turned on all the time.

A friend says that if I leave my PC powered on all the time, then the hard disk drive would only last 2 or 3 years, due to heat and wear caused by continuous rotation. On the other hand, I have been told that it's better to leave the PC powered up all the time, to prolong the life of the PC motherboard and cards, by not doing cold boots too often.

If both statements are true, I would like to leave my PC with the 40MB hard drive

(drive C:) switched on, with my 245MB hard drive (drive D:) disconnected. After booting up my PC with both hard drives connected, drive D: could then be unplugged whenever I am not around, and plugged back in on my return.

Extra cables would be added so that the disk controller and power connectors to the D: drive are outside the PC chassis. Indirectly it would also mean that my 245MB drive is inaccessible to PC hackers from the outside. Will there be any possible problem due to the BIOS being told that there are two hard drives connected?

Andrew Knox

Unfortunately neither the hard drives nor the controllers are designed to be connected and disconnected on the fly, as you want to do. There are slide-in assemblies which can be used to remove a hard disk from a computer for security or portability, but these are designed to be used only while power is turned off. So with one of these you'd be turning the power on and off twice as often as you would if you just turned the machine on at the beginning of the day, and off again at the end.

There are drives designed to be 'hot-swapped', and they are commonly used in redundant drive arrays, but they are not cheap, and are not a practical solution to your problem.

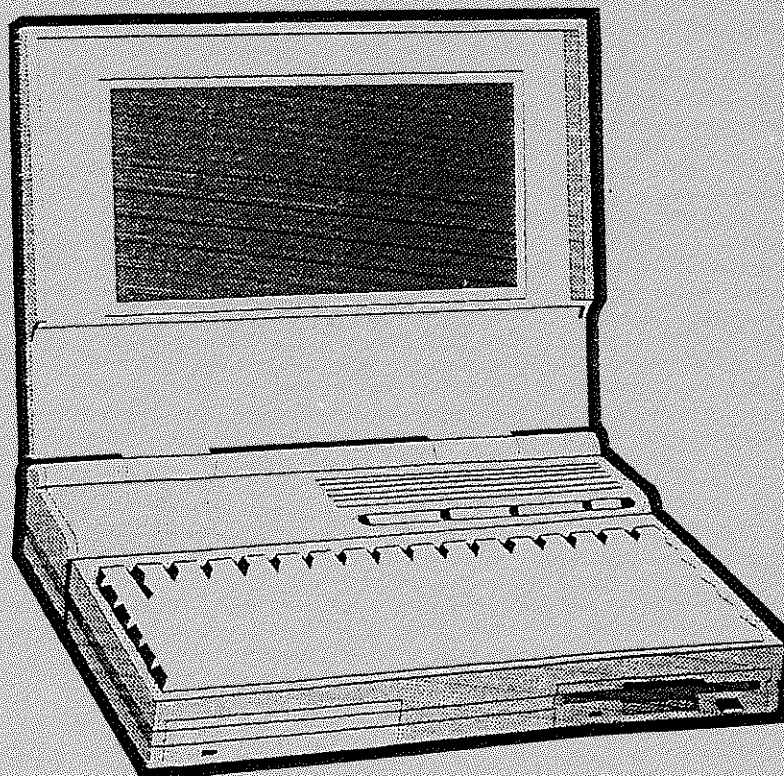
However, I don't see a lot of harm in

just leaving your PC and both drives running, except for the power consumption. Modern hard drives don't consume as much power as their older cousins, and it would undoubtedly be a small fraction of the total consumption of the machine. The ideal machine for your application would be one of the new 'Energy Star' models that are starting to appear, which power down to a low power state (including shutting off the hard drive) when not in use, and wake up almost instantly when they need to do something.

Hard drives do have a finite life, but I suspect any additional life gained by turning the drive off would be lost again due to the additional stresses imposed by starting it up next time, not to mention the wear on the heads and disk surface when the heads land as the drive slows down to a stop. Hard drive life is impossible to predict on a case by case basis, and failures do happen, but manufacturers typically quote MTBF (mean time between failure) figures in the order of tens of years of continuous operation.

Aside from the hardware problems your method involved, as you suggest, the BIOS would also get upset if one of the drives disappeared mid-stream, not to mention DOS and your application software.

Note: There is no Tech Tip of the Month this month.



REGIONAL GROUP REPORTS

Meeting Summary For JUNE

Central Coast	08/06/96	Saratoga
Glebe	08/06/96	Glebe
Hunter Valley	09/06	16/06/96
Illawarra	04/06/96	Keiraville
Liverpool	07/06/96	Yagoona West
Sutherland	21/06/96	Jannali

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 135B Arundel Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The Meetings are usually held on the second or third Sunday of each month at members homes starting at 3pm. Check the location with Geoff Phillips by leaving a message on (049) 428 617. Please note that the previous phone number (049)428176 is now used exclusively by the ZZAP BBS which also has TI support. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the first Tuesday of each month after the TISHUG Sydney meeting at 7.30pm, at the home of Geoff Trott, 20 Robsons Road, Keiraville. A variety of investigations take place at our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Geoff Trott on (042)296629 for more information.

LIVERPOOL Regional Group*

Regular meeting date is the Friday following the Tishug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home). After 9.30 PM or at work (02)708 1987

Liquorland YAGOONA for more information.

*** ALL WELCOME ***

7 th JUNE 1996
12th JULY 1996
9th AUGUST 1996
13th SEPTEMBER 1996
11th OCTOBER 1996
8th NOVEMBER 1996
13th DECEMBER 1996

Bye for now Larry
Liverpool Regional Co-Ordinator

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young.

TISHUG in Sydney

Monthly meetings start promptly at 2pm on the first Saturday of the month. They are held at the MEADOWBANK PRIMARY SCHOOL, on the corner of Thistle Street and Beimore Street, Meadowbank. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

JUNE MEETING - 1st JUNE

JULY MEETING - 6st JULY

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

15th JUNE FOR THE JULY MAGAZINE

These dates are all Saturdays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00 pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.