

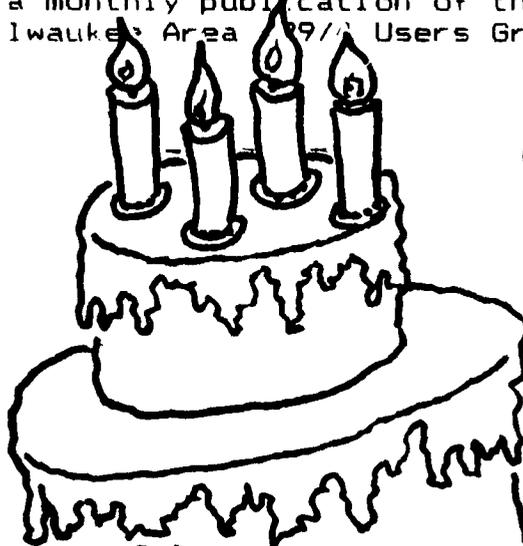
99

# HOCUS

Home Computer  
Users Spotlight

a monthly publication of the  
Milwaukee Area '89 Users Group

MAY 1985



## HAPPY BIRTHDAY MIL AREA UG

### NOISELESS PERIPHERAL EXPANSION BOX

By Bob Hubel MSP UG

Are you distracted & disturbed by the tornadic roar of the Peripheral Expansion Box? I engineered this box to provide sufficient cooling capacity for the most strenuous of circumstances -- all 8 card slots occupied and under heavy, continuous usage. Since my use didn't approach the design limits, I experimented with ways to slow the fan down, and I was successful in reducing that noise level to a barely perceptible purr! I have now been testing this change for a sufficient period of time, and I feel comfortable in recommending the modification to all. In fact I have even run under light loads for moderate periods of time without any fan at all, but I don't advise going to that extreme.

I had considered 2 options, buy a quieter fan or slow down the fan I have. A quieter fan costs about \$20 or more, so I quickly discarded that option. I could reduce the fan speed with either a special solid-state voltage regulation device or

merely install a power resistor in series with the fan. Since I have no experience with such solid-state devices and the components would as likely cost around \$10, I elected for the latter method at a cost of \$0, using components from my box of electronic junk. If you had to buy the components at retail, the cost would amount to no more than \$2-\$3.

The PEB fan is rated at 14 Watts. I have found that inserting a 500 - 700 Ohm power resistor, 10 Watts power dissipation reduces the speed to a very acceptable level. 700 Ohms is my own preference. Series of 2 or more resistors may be combined to add up to 700 Ohms. The resistor "steals" the energy that originally was intended for the fan, thus the fan doesn't work as hard. However in doing so, the resistor must shed the extra energy itself and does so by producing heat. Therefore you should mount the resistor outside of the PEB, immediately behind the fan thus allowing the circulating air to cool it. Mounting it inside the cabinet, although aesthetically more pleasing will add unwanted heat inside the cabinet. Procedure to disassemble the PEB to access the fan lead wires

- 1) Unplug power cord and remove lid
- 2) Disconnect and remove disk drive
- 3) Remove all slide-in cards
- 4) Remove cabinet screws (?!?) on bottom, back and sides
- 5) Slide cabinet base out
- 6) Disconnect 1 of the fan leads
- 7) Extend the 2 wires thru holes in the cabinet back
- 8) Splice resistor(s) in series and mount on box near exhaust
- 9) Reassemble box

Since the fan air intake is thru the card cage, I cut foam to fit in the empty slots at the far forward side of the card cage, forcing the reduced airflow thru the existing cards, increasing their cooling.

The resistors will run warm and possibly hot, this is normal. Don't cheat and use less than a 10 Watt resistor which may get too hot and cause problems.

You'll be amazed at the reduction in the noise pollution. If anyone knows of an inexpensive solid-state equivalent to perform the identical function, please notify me.

Good Luck !!

OFFICIAL GROUP ADDRESS  
4122 N. GLENWAY  
WAUWATOSA WI 53222

#### GROUP OFFICERS

PRESIDENT.....Jerry Trinkl 327-0170  
VICE-PRESIDENT...James Schroeder 264-4735  
TREASURER.....Phil Norton 462-8954  
COORDINATOR.....Gene Hitz 453-0499  
SECRETARY.....Jim Steinhardt 475-9028  
LIBRARIANS.....E.J. VonDerEhe 549-0593  
Fred Fabian 327-3618  
NEWSLETTER EDITOR.....\*\* STILL VACANT \*\*

#### MEMBERSHIP INFORMATION

Membership to the Milwaukee Area 99/4A Users Group is open to anyone who is interested in using and/or programming the Texas Instruments 99/4A Home Computer, and is willing to share his/her fellowship with other members. Annual dues for Individuals - \$10.00, Family membership - \$15.00. This fee helps to defray the expense of the publication of this newsletter and provide a library to members for their enjoyment.

#### MEETING INFORMATION

Meetings of the Milwaukee Area 99/4A Users Group are held on the SECOND SATURDAY each month in the lower level of Wauwatosa Savings and Loan, located at 7500 W. State Street in Wauwatosa, Wisconsin. Meeting times are from about 1:00 PM to 4:00 PM, depending on the content of the meeting for each particular month. Users are encouraged to bring their computers and/or related hardware and software to any of the meetings to assist members in utilizing their own equipment.

#### Book Review

Book: Compute!'s Beginner's Guide to Assembly Language  
on the TI-99/4A

Author: Peter M. L. Lottrup

Reviewer: Chris Maag

First, the facts on this book. The price is a reasonable \$12.95. There are 262 pages, and this book was written for the user who has the Mini-Memory/Line-by-Line Assembler. Still, the book will be valuable even to people who may have the Editor-Assembler package. It is basically a book for the beginning assembly language student.

The topics covered in this book include many of the things that will completely mystify the beginner. Some of these topics include sound and sound lists, sprites and how to use and control them, and the various graphics modes. It also covers the built-in utilities, linking with Basic, and some excellent programming techniques for assembly language.

I would say that this book is an excellent book for all beginning students. It covers all of the important topics very well. The book explains in great detail all of the necessary op-codes, and uses over 25 programs and routines to show you how to use them. These programs are in large print and are easy to follow. The op-codes that the book focuses on are very thoroughly explained by showing what they do and how they work. There is also a very nice index, as well as four handy appendices. The index is about four pages long, and cross-references all of the op-codes and the programs in the book. The appendices have a character chart with the ASCII codes in both hex and base-ten. The next appendix has a list of op-codes, their format number, and what they do. The third appendix is another list of the op-codes with a chart to tell you what operands are needed and where the result of the operation is placed.

One of the other topics that the book covers is how to mimic some of the X-Basic operations that are so handy. Examples of these are how to do Accept At, sprite commands, Call Key, and Call Joyst. The book also shows you how to do many of the things that are impossible to do in Basic or X-Basic, such as bit-map mode drawing, scrolling the text screens in four directions, and screen switching.

The only thing that I felt was not perfect in this book was that the book never explains hexadecimal arithmetic. I feel that this topic would be very confusing to the beginner and should be included in any assembly language book. This book is available at Walden's, B. Dalton's, and at many area computer stores. All in all, it is a 'must have' book for assembly language programmers.

#### Star Trek Strategic Operations Simulator Game Review

By Jeff Maag

Star Trek was authored by Sega and marketed by Texas Instruments. This game is found in the arcades and appears almost exactly the same on the TI-99/4A. First, the screen is divided into half vertically. The bottom half is the close up 3D view of your battle. The top half is divided into half horizontally, with the left side being your gauges, and the right side showing the whole sector of your battle. In my opinion, the 3D closeup viewer is useless since the sector wide viewer is so accurate and shows all the essential details. As a matter of fact, I don't even use the closeup viewer.

The object of the game is to clear out sectors, one at a time, of Klingons, Nomads, and antimatter saucers. You can destroy these by firing phasers or photon torpedoes at them, or you can run from them using impulse power, or warp drive. You start out with five photon torpedoes, five units of warp drive, and five shields for defense. You have unlimited phasers and impulse power. To replenish the photon torpedoes, warp drive, and shields you must dock with the square starbases where you get one more of each of these. If you clear out a sector without using a star base, you are awarded with extra points, so its not always advisable to use them if they're not needed. The Klingons become more numerous and grow more aggressive as you clear each sector.

The graphics and sound of this game are fair. It uses sprites for the sector wide viewer. The rest uses bit-map graphics. The 3D closeup viewer shows the ships getting larger when they get closer. This effect is pretty rough looking. The graphs showing the amount of photon torpedoes, warp drive units, and shields at first weren't legible at a quick glance. The sound was also fair with nothing out of the ordinary. This game has the option of speech, but since I don't own a speech sythesizer, I couldn't test this.

Overall the game is pretty good and is a tremendous improvement over the slow basic version, if you like arcade games. One problem with this game, along with a lot of arcade type games, is that it gets repetitive because each sector battle is basically the same except for increasing difficulty. Another problem, also usual for these types of games, is that after two weeks of playing the game (and getting a highscore of 650,000), I got sick of it.

The solution to this problem is that I checked Star Trek out of our new cartridge library. I gave the librarian five dollars and my membership card and next month I will get four dollars back. This way I will be able to buy something more worth-while, like the latest issue of "HOME COMPUTERS" magazine!?!

## TI FORTH DeBugs found

```
.... Jeff Stanford
Scr # 22 Line 5  BASE->R HEX ( 3800 ' SATR ! )
Scr # 23 Line 2  : CINIT 3800 DUP ' SPDTAB ! 800 / 6 VNTR 3800 ' SATR !
Scr # 28 Line 1  : EDT VDPNDE @ 5 = 0= IF SPLIT ENDIF CINI? 'CUR R/C CGOTOXY
Line 11 OF DF 5 0 SPRPAT CLS SCRNO DROP 300 ' SATR ! QUIT ENDDF
```

```
.... Tom Freeman
Scr # 53 , # 54 , # 55 Line 1  VDPSET2 --> SETVDP2
Scr # 58 Switch Lines 9 & 10
Change Line 9  VDPNDE 4 < IF SMTN 80 0 VFILL 300 ' SATR ! ENDIF
Scr # 59 Line 9  Between >R and SP put  B SLA SWAP OOFB AND DR
```

```
.... Everybody & his brother
Scr # 72 Line 5  PAB_3DGR --> PAB-ADDR
```

```
.... Jim Vincent
Manual Chapter 6 Page 10 Line 1  HEX 3800 ' SATR !
Chapter 10 Page 3 Line 20 : DOWN -100 ALLOT DROP ;
```

---

## FORCING PRINTER PAUSES

by Abdallah Clark

If you want to change, in the middle of your text, to another printwheel or differently colored ribbon, use the ALTERNATE INPUT Command at the point in your text where you want to make the switch. When used in text without a separate file being specified for a "Mailing List" option in the Text Formatter, the ALTERNATE INPUT Command stops the printer and then leaves TI-WRITER waiting for your input from the keyboard. Change your daisywheel or dot-matrix printer as needed, then press ENTER, and the printing continues to completion from that spot.

Since no harm is done by just pushing ENTER, you have an easy way of pausing, not aborting, the printout process! However, only one such change may be made per line of text, unless you do some really fancy work with the Transliterate Command.

Also, it seems the results will be more dependable if you put the DEFINE PROMPT Command (and its carriage return) on a separate line. If your ALTERNATE INPUT is on a line to itself, though, you will have a linefeed there by pressing ENTER. However, place it where needed or substitute them for a carriage return symbol or reduce the ".SP n" format command by one to retain your text's form.

It is also important to note that the ALTERNATE INPUT Command is the only Command that does not use a leading period as a signal to the Text Formatter that it is a special function symbol. This has two effects. One, don't let habit make you use a period, or you will have an unwanted period in your text. Two, you cannot use that combination of characters in your own document, unless indirectly by way of transliteration.

Another note to be mentioned about the ALTERNATE INPUT Command is that the digit used between the asterisks may only be used once, whether this pertains to a single document, or a series of documents "linked" by the INCLUDE FILE command. Be careful not to exceed the maximum of 99 for that digit in the ALTERNATE INPUT Command when you have a series of documents, too.

You may also use the DEFINE PROMPT Command in conjunction with the ALTERNATE INPUT Command, to compose a memory jogger message for the task needed. (Remember: always give the ".DP xxxxx" first, then the particular ALTERNATE INPUT Command.) This way, when the Text Formatter prints the document, the printer will stop where the ALTERNATE INPUT group of characters are located and your prompt appears on the monitor/TV display. It's even easier than you would think if you list all the prompts at the beginning of your document, because then you may move/rearrange text to your heart's content without having to worry about whether you're keeping the prompt ahead of the input. A handy convenience if you make more than one of these printer changes in the course of one document!

----- Abdallah Clark

RABID by Cullhane Gibbs

Here is a good game to play if you're feeling bloodthirsty. It was written by one of our members, Cullhane Gibbs, who is only 13 years old. What you have to do is defend yourself from the rbsid knife-wielding maniacs. You are equipped with a flamethrower. Good luck!

```

100 REM *****
110 REM *   RABID   *
120 REM *BY CULLHANE GIBBS *
130 REM *IN EXTENDED BASIC *
140 REM *JOYSTICKS REQUIRED*
150 REM *****
160 CALL MAGNIFY(2)
170 RANDOMIZE
180 CALL CHAR(47,"002277FFFF
2A0000")
190 CALL CHAR(94,"123422256F
800451")
200 CALL CHAR(64,"3838107C8B
292A28")
210 CALL CHAR(124,"20502070A
8B0AB44")
220 CALL CLEAR :: CALL SCREE
N(2)
230 FOR COLOUR=2 TO 12 :: CA
LL COLOR(COLOUR,COLOUR+1,2):
: NEXT COLOUR
240 PRINT "   RABID":"BY C
ULLHANE GIBBS":"NEEDS EXTEND
ED BASIC":"AND JOYSTICKS" ::
PRINT :: PRINT "INSTRUCTION
S:"
250 PRINT "AVOID RABID,":
"KNIFE WEILDING MANIACS.":"T
O DEFEND YOURSELF"
260 PRINT "PRESS YOUR FIREBU
TTON":"TO RELEASE A SHORT FL
AME":"WHICH WILL BURN YOUR":
"ATTACKERS.":"NEW SCREEN STA
RTS WHEN ALL"
270 PRINT "ATTACKERS ARE KIL
LED.":"ONCE YOU ARE KILLED T
HE":"GAME IS OVER.":"YOU CAN
FIRE ONLY":"IN ONE DIRECTIO
N-":"TO THE LEFT."
280 PRINT "PRESS ANY KEY"
290 FOR D=10 TO 50 :: CALL S
OUND(D,701,0):: CALL SOUND(-
100,-8,0):: CALL SOUND(-50,-
4,10):: CALL SOUND(-100,-2,0
):: NEXT D
300 CALL KEY(0,KEP,SEP):: IF
SEP=0 THEN 300 ELSE 310
310 CALL CLEAR
320 PRINT "SCORECHART:" :: P
RINT

```

```

330 PRINT "@ MANIAC=100" ::
PRINT :: PRINT "* YOU" :: PR
INT
340 PRINT "PRESS ANY KEY TO
BEGIN"
350 CALL SOUND(-1000,-8,0)
360 PRINT :: PRINT :: PRINT
:: PRINT :: PRINT :: PRINT :
: PRINT :: PRINT :: PRINT ::
PRINT
370 CALL KEY(0,K,S):: IF S=0
THEN 370 ELSE 380
380 CALL CLEAR :: PRINT "
GET READY,PLAYER!" :: PRINT
:: PRINT :: PRINT :: PRINT
:: PRINT :: PRINT :: PRINT ::
PRINT :: PRINT :: PRINT
390 FOR TIME=1 TO 500 :: NEX
T TIME
400 SCORE=0
410 CALL CLEAR
420 CALL SPRITE(#11,124,4,70
,200)
430 FOR SPRIT=1 TO 4 :: CALL
SPRITE(#SPRIT,64,5,121,89):
: CALL MOTION(#SPRIT,INT(RND
*10)+1,-INT(RND*10)+1):: NEX
T SPRIT :: MANIC=4
440 DISPLAY AT(1,3):"SCORE:"
;SCORE
450 CALL JOYST(1,X,Y):: CALL
MOTION(#11,-Y*2,X*2):: CALL
SOUND(-3,-3,0)
460 CALL POSITION(#11,YPOS1,
XPOS1)
470 CALL KEY(1,KE,ST)
480 IF ST=-1 AND XPOS1]24 TH
EN 490 ELSE 690
490 CALL SPRITE(#12,47,7,YPO
S1,XPOS1-16)
500 CALL COINC(#12,#1,20,A):
: IF A=-1 THEN 510 ELSE 540
510 CALL SOUND(1000,340,0)::
CALL PATTERN(#1,94):: SCRE
=SCORE+100 :: MANIC=MANIC-1
:: CALL DELSPRITE(#1)
520 CALL DELSPRITE(#12)
530 IF MANIC=0 THEN 430 ELSE
540

```

```

540 CALL COINC(#12,#2,20,A):
: IF A=-1 THEN 550 ELSE 590
550 CALL DELSPRITE(#12)
560 CALL SOUND(1000,340,0)::
CALL PATTERN(#2,94):: SCORE
=SCORE+100 :: MANIC=MANIC-1
:: CALL DELSPRITE(#2)
570 CALL DELSPRITE(#12)
580 IF MANIC=0 THEN 430 ELSE
590
590 CALL COINC(#12,#3,20,B):
: IF B=-1 THEN 600 ELSE 640
600 CALL DELSPRITE(#12)
610 CALL SOUND(1000,340,0)::
CALL PATTERN(#3,94):: SCORE
=SCORE+100 :: MANIC=MANIC-1
:: CALL DELSPRITE(#3)
620 CALL DELSPRITE(#12)
630 IF MANIC=0 THEN 430 ELSE
640
640 CALL COINC(#12,#4,20,C):
: IF C=-1 THEN 650 ELSE 700
650 CALL DELSPRITE(#12)
660 CALL SOUND(1000,340,0)::
CALL PATTERN(#4,94):: SCORE
=SCORE+100 :: MANIC=MANIC-1
:: CALL DELSPRITE(#4)
670 CALL DELSPRITE(#12)
680 IF MANIC=0 THEN 430 ELSE
700
690 CALL DELSPRITE(#12)
700 CALL COINC(#1,#11,16,T):
: IF T=-1 THEN 740 ELSE 710
710 CALL COINC(#2,#11,16,U):
: IF U=-1 THEN 740 ELSE 720
720 CALL COINC(#3,#11,16,V):
: IF V=-1 THEN 740 ELSE 730
730 CALL COINC(#4,#11,18,W):
: IF W=-1 THEN 740 ELSE 780
740 FOR DIP=1 TO 28 :: CALL
MOTION(#DIP,0,0):: NEXT DIP
:: CALL SOUND(-1000,-8,0)::
CALL PATTERN(#11,94):: FOR D
EL=1 TO 50 :: NEXT DEL
750 CALL DELSPRITE(#11):: DI
SPLAY AT(23,3):"GAME OVER-ST
ART AGAIN Y OR N" :: CALL KE
Y(0,P,S):: IF S=0 THEN 750 :
: IF P=ASC("n")THEN 770
760 IF P=ASC("y")THEN 400 EL
SE 750
770 END
780 GOTO 450

```

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.

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each! (plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Come on now, folks, don't you support your local schools? And don't you support those who support

you? There are thousands of schools which have TI-99/4A computers in the classroom, usually without disk drive and without Extended Basic. They could use some educational programs in Basic on cassette. They could probably use some of the public domain software in your library. Maybe they could use some of the educational programs I sell for just \$3 (and I authorize schools to copy them for use within the school). There is probably such a school in your area - is your group supporting it? In the last Tips, I asked the members of 101 users groups to give me the addresses of schools that had TIs, so I could send them a free catalog. How many addresses did I get? Zero to the power of zero times zero!

More on the pestiferous asterisk bug in TI-Writer. Dr. Guy-Stefan Romano has confirmed and explained it. If you are printing out of the Foreatter mode and your text contains an asterisk followed by two or more numeric digits - the asterisk and two digits will disappear! For instance, A#256 becomes A6, and I've noticed that A6 in programs published in several newsletters recently.

The TI-Writer program misinterprets the asterisk and two digits as an instruction to input data from a "value file" (see Alternate Input on p. 111 of the annual).

The solution to this bug is to type two asterisks followed by two dummy digits, then the actual digits. For instance, instead of A#256 type A##25256. Trouble is, the bug usually shows up in a program which has been LISled to disk and then

MERGED into TI-Writer, and is usually not noticed. The solution? Run the program through my 28-Column Converter (see Tips #18!).

Dr. Romano informs me that there is an even worse bug in the Transliterate command coding, erratic and sometimes destructive. It is triggered by certain sequences of characters, but these have not been documented.

Dr. Romano says that he does not use transliteration.

I would suggest that you also avoid the use of the & and @. The & will only underline a single word, unless you tie words together with the ^ sign. If you tie words together, the Fill and Adjust will leave gaping blanks in your lines and if you tie too many together the line will extend beyond the right margin! Also, the underlining is a broken line. It is better to use the escape codes CTRL U, FCTN R, CTRL U, SHIFT -, CTRL U, SHIFT A, CTRL U, which will give a solid underline until you turn it off with CTRL U, FCTN R, CTRL U, SHIFT -, CTRL U, SHIFT @, CTRL U.

The @ is handy to emphasize a single word, but if you want to double-strike a whole sentence or paragraph it is better to use the escape code CTRL U, FCTN R, CTRL U, SHIFT G, and turn it off again with CTRL U, FCTN R, CTRL U, SHIFT H.

The period bug is another killer - the Foreatter thinks that any line which begins with a period is a foreatter command, and deletes the whole line! If your text contains a decimal value such as .11 and the wraparound puts it at the beginning of a line, the

line disappears! There are two ways around this - put a 0 in front of all your decimals, as 0.11, or transliterate all your periods.

In all, the TI-Writer foreatter is a temperamental and unpredictable piece of software, prone to unwanted line feeds and unexpected paper-wasting form feeds. I like to use it to right-justify text back to the disk, but from then on I prefer to print it out of the editor mode, or out of my own program.

Designing downloadable characters for the Gemini printer (see page 115 of the manual) is a bit tricky because it is hard to visualize how the expanded pattern will appear in print. The following program will enable you to experiment with designs, dump them directly to the printer for viewing, then save them as a file. When you later dump this file into printer RAM for use, you must activate the download characters with the escape code - CHR\$(27);CHR\$(36);CHR\$(1).

```
100 CALL CLEAR :: CALL SCREE
N(4):: CALL CHAR(128,"FFB1B1
B1B1B1B1FF",129,RPT$( "F",16)
):: CALL COLOR(13,2,16)
110 FOR R=9 TO 15 :: CALL MC
HAR(R,11,128,9):: NEXT R
120 X=1 :: FOR R=9 TO 15 ::
DISPLAY AT(R,7)SIZE(2):STR$(
X):: X=X*2 :: NEXT R :: FOR
C=9 TO 17 :: DISPLAY AT(8,C)
SIZE(1):STR$(C-8):: NEXT C
130 DISPLAY AT(2,9):"TIGERCUB'S" :: DISPLAY AT(4,1):"GEMINI
CHARACTER DOWNLOADER" 'p
rogrammed by Jim Peterson fo
r the Public Domain
140 DISPLAY AT(17,1):" Move
cursor with W,L,R,S,D." : "2..
and C keys. Toggle on": "and
off with Q key. Press": "Ent
er when finished." :: "Pres
```

```

any key
150 CALL KEY(O,K,ST):: IF ST
=0 THEN 150 :: CALL MCHAR(17
,1,32,224)
160 R=9 :: C=11 :: CH=128
170 CALL MCHAR(R,C,32):: CAL
L MCHAR(R,C,CH):: FOR D=1 TO
10 :: NEXT D :: CALL KEY(3,
K,ST):: IF ST=0 THEN 170
180 ON PUS("MWERDCXZS"&CHR$(
13),CHR$(K),1)+1 GOTO 170,31
0,230,220,210,200,190,260,25
0,240,330
190 K=R+1
200 C=C+1 :: GOTO 270
210 C=C+1
220 R=R-1 :: GOTO 270
230 K=R-1
240 C=C-1 :: GOTO 270
250 C=C-1
260 R=R+1
270 R=R-(R/9)+(R/15):: C=C-(
C/11)+(C/19):: IF CH=128 THE
M 300 :: CALL BCHAR(R,C-1,6X
):: CALL BCHAR(R,C+1,6Z):: I
F (6X<>129)&(6Z<>129)THEN 30
0
280 DISPLAY AT(22,1):"You ca
n't have two in a row":horiz
ontally!" :: FOR D=1 TO 50
:: NEXT D :: DISPLAY AT(22,1
):: " "
290 CH=CH-1
300 CALL MCHAR(R,C,CH):: GUT
D 170
310 CH=CH+1+(CH=129)*2 :: IF
CH=128 THEN 320 :: CALL BCH
AR(K,C-1,6X):: CALL BCHAR(K,
C+1,6Z):: IF (6X<>129)&(6Z<>
129)THEN 320 ELSE 280
320 CALL MCHAR(K,C,LM):: GOTO
D 170
330 FOR C=1 TO 19 :: X=1 ::
FOR K=9 TO 15 :: CALL BCHAR
(R,L,b)
340 IF b=129 THEN A=A+1
350 X=X+2 :: NEXT K
360 FOR J=1 TO LEN(STR$(A))
:: CALL VCHAR(15+J,C,ASC(SEEK
(STR$(A),J,1))):NEXT J ::
M=M&CHR$(A):: A=0 :: NEXT
C :: A=0
370 DISPLAY AT(20,1):"Print?
Y/N Y" :: ACCEPT AT(20,12)V
ALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="N" THEN 470
380 IF F=1 THEN 390 :: F=1
:: DISPLAY AT(20,1):"Printer
name?" :: ACCEPT AT(20,15):P
$ :: OPEN #1:PS

```

```

390 DISPLAY AT(20,1):"ASCII
to redefine?" :: ACCEPT AT(2
0,20)VALIDATE(DIGIT)SIZE(3):
CH
400 DISPLAY AT(20,1):"Descen
der (0 or 1)? 0" :: ACCEPT A
T(20,21)VALIDATE("01")SIZE(-
1):D$ :: D=VAL(D$)
410 M$=CHR$(27)&CHR$(42)&CHR
$(1)&CHR$(CH)&CHR$(D)&M$
420 PRINT #1:M$ :: PRINT #1:
CHR$(27);CHR$(36);CHR$(1):
430 PRINT #1:RPT$(CHR$(CH),7
2):: PRINT #1:CHR$(14);RPT$(
CHR$(CH),36)
440 DISPLAY AT(20,1):"Save (
Y/N)? Y" :: ACCEPT AT(20,13)
VALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="N" THEN 470
450 IF F3=1 THEN 460 :: F3=1
:: DISPLAY AT(20,1):"File na
me? DSK" :: ACCEPT AT(20,14)
:F$ :: OPEN #2:"DSK"&F$
460 PRINT #2:M$
470 M$="" :: DISPLAY AT(20,1
):"Another (Y/N)? Y" :: ACCE
PT AT(20,16)VALIDATE("YN")SI
ZE(-1):Q$ :: IF Q$="Y" THEN
100
480 CLOSE #1 :: CLOSE #2 ::
END

```

Micropendium ran a contest to improve on a brief ingenious organ program. The winner was Michael Christianson, who wrote a superb program. You'll have to buy the January issue of the magazine to get it (you should be subscribing, anyhow!). I didn't enter the contest, of course, and my version is not nearly as good, but have fun -

```

90 CALL CLEAR
95 PRINT TAB(5):"MICROPENDIU
M ORGAN" :: : : : : : "Pl
ay bass with left hand": "o
n left side of keyboard." :
"melody on the right": :
100 REM - MICROPENDIUM ORGAN
modified by Jim Peterson
110 OPTION BASE 0
120 DIM NUT(20)
130 FOR A=0 TO 20
140 READ NUT(A)
150 NEXT A

```

```

160 DATA 40000,220,247,262,2
94,330,349,392,440,494,523,5
87,659,698,784,880,988,1047,
1175,1319,1397
170 CALL KEY(1,K1,S)
180 CALL KEY(2,K2,S)
190 CALL SOUND(-1000,NOTE(K2
+1),0,NOTE(K2+1)*1.01,5,NOTE
(K1+1)*3.75-ABS(K1+1=0)*1100
00,30,-4,0+ABS(K1+1=0)*30)
200 GOTO 170

```

A sprite routine that doesn't do anything but look pretty. I call it Patches.

```

50 CALL CLEAR :: CALL SCREEN
(5)
100 A$=RPT$("AA55",16):: B$=
RPT$("F",64):: CALL MAGNIFY(
4):: RANDOMIZE
110 FOR CH=40 TO 136 STEP 8
:: CALL CHAR(CH,A$,CH+4,B$)
:: NEXT CH
120 C=2 :: S=40 :: R=1 :: FO
R T=1 TO 24 STEP 2 :: COL=15
ORND+50 :: CALL SPRITE(OT,S
,C,R,COL,OT+1,S+4,C+1,R,COL)
:: S=S+8 :: C=C+1 :: R=R+15
:: NEXT T
140 FOR T=1 TO 50 :: CALL CO
LOR(#INT(24*RN)+1),INT(16*RN
D+1)):: NEXT T :: GOTO 120

```

This is one that I fancied up, based on a sprite routine written by a youngster named Andrew Sorenson, published in the Sydney Newsdigest from Australia.

```

100 ! WILL O' WISP
by Jim Peterson
based on
Andrew Sorensen's
sprite routine
110 CALL CLEAR :: CALL SCREE
N(2):: CR=48
120 FOR CH=48 TO 63 :: FOR L
=1 TO 4 :: RANDOMIZE :: X=IN
T(16*RN)+1)*2-1 :: X$=SEEK("
0018243C425A667E8199A5BDC3DB
E7FF",X,2):: B$=B$&X$ :: C$=
X$&C$ :: NEXT L :: CALL CHAR
(CH,B$&C$):: B$,C$="" :: NEX
T CH
130 FOR N=1 TO 28 :: CALL SP
RITE(ON,CR,INT(14*RN)+3),B$N
+20,120,5,0):: NEXT N :: IF

```

```

CR=64 THEN CR=48 :: T=T+1+I
2)*2 :: CALL MAGNIFY(T)
140 X=(INT(3*RN)-1)*4 :: Y=
(INT(3*RN)-1)*4
150 IF INT(10*RN+10)<>10 TH
EN 170
160 CR=CR+1 :: GOTO 130
170 FOR N=1 TO 28 :: CALL MO
TION(ON,-Y*20,X*20):: NEXT N
:: GOTO 140

```

Here are a few more enhancements to my Menu Loader, published in Tips #15. Delete line 150 and add

```

101 OPTION BASE 1 :: DIM P$(
127):: ON WARNING NEXT :: 6
DIO 110
105 @,A,A$,B,C,D$,FLAG,1,J,K
,KD,KK,M$,MN,P$,P$(1),Q$,S,
T,T$(1),TT,VT,X
CALL INIT :: CALL LOAD :: CA
LL LINK :: CALL PEEK :: CALL
KEY :: CALL SCREEN :: CALL
COLOR :: CALL CLEAR :: CALL
VCHAR :: CALL SOUND :: !@P-

```

The pre-scan will speed up run time by a worthwhile amount. The warning default will prevent a screen scroll on an erroneous Enter.

When you're finished printing strip labels, cut off the strip BEHIND the platen and roll it FORWARD! You'll waste a few labels that way, but if you try to roll backwards and get a quamy label stuck in the works, you've got trouble!

MEMORY FULL

Jim Peterson

the Tigercub