Chicago faire reports

NEW PRODUCTS DEBUT:

TI BINGO

MIDI PLAY IN

DISK MANAGER 2000

Bud Mills receives Birdwell award

Reviews

TI SPLASH SCENES

TI THEME FOR PCs

```
MENU:
A PROGRAM BY JPH AND BAT
FROM THE TRAVELER DISKAZINE
COPYRIGHT (C) 1987, 1988

COLORS:
 2 BLACK           10 LT. RED
 3 MED. GREEN      11 DARK YELLOW
 4 LT. GREEN       12 LT. YELLOW
 5 DARK BLUE       13 DARK GREEN
 6 LT. BLUE        14 MAGENTA
 7 DARK RED        15 GRAY
 8 CYAN            16 WHITE
 9 MED. RED

SUGGESTED: 16/5, 2/8, 2/12,
OR 16/2.

FOREGROUND (2-16)?:   16

BACKGROUND (2-16)?:    5

WANT LARGE LETTERS AND TRUE
LOWER CASE? Y
```

How about playing Ratinfest?

Extended BASIC

• USING IF-POS

• FILTERING CAPTURED FILE CONTENT INTO SEPARATE FILES

• MENU

CONTENTS

COMMENTS

Where do you go for repairs?

Where do you go when your TI or Geneve or other hardware goes south? The answer used to be simple. Send it to TI. Or send it to Myarc. Send it to the manufacturer. Those days are long gone. Today the choices are slim and none. As recently as last year you could rely on Cecure Electronics to repair Myarc and TI products. But that's no longer the case. Cecure is apparently no longer repairing TI or Myarc equipment. Readers are cautioned not to send money or hardware to Cecure without confirming the company is still servicing the TI community.

Fortunately, Tim Tesch continues to support Myarc hardware, including the Geneve and Hard and Floppy Disk Controller. He prefers that you query him before sending hardware for repair. He expects payment after the repair is made. For more information, contact him at 1856 Dixie Road; Port Washington, WI 53074; or e-mail him at ttesch@juno.com

I know of no one who is actively repairing CorComp products. The TI list-server has queried its subscribers without positive result. If your CorComp disk controller goes out you may have no choice but to find a replacement, such as the Western Horizon SCSI card or the German SCSI card developed by Michael Becker. Both of the SCSI cards continue to be supported by their developers. Of course, you can always try to find a used card.

Bud Mills says he's continuing to support Horizon RAMdisks (see the Chicago Faire article for more details).

That's about all that the support I'm aware of. If you know of others, by all means let us know so that we can pass it on.

NEW PRODUCTS DEBUT IN CHICAGO, BUD MILLS IS BIRDWELL WINNER

Maybe two dozen TI users attended the 1998 Chicago TI Faire in Evanston, Nov. 14. Which was fine. The faire was held in a church basement meeting room divided into a display area for vendors and a demonstration area for those who wanted to give presentations about their products. Several programs were introduced, including TI Bingo and MIDI Play In by Bruce Harrison, Disk Manager 2000 by George Kaal of the Dutch TI user group (Berry Harmsen represented the group), and a 32-bit version of PC99 by Mike Wright of CaDD Electronics. You'll find more information in the article that starts on page 5.

The biggest announcement to come from the fair wasn't about products. It was the naming of Bud Mills as the recipient of the 1998 John Birdwell Memorial Award for Excellence in Computing.

My thanks to Hal Shanafield for putting together another successful, albeit low-keyed, faire.

—JK

Clarification

First, a late update. After I'd sent off the MIDI-Master article, and too late for publication, I heard from Richard Bell that he's changed his e-mail address. Now it's wim4home@aol.com.

Second, a little clarification about the issue of program file headers, as mentioned in Jeff White's article about file compression. The correct value for the second word in the header of a Program file is the file's TOTAL length, including the six bytes of the header. I've never used PFC, but have heard of this particular problem before, and believe Jeff is right about there being a bug in PFC, which causes it sometimes to put the wrong value (six bytes less) in the file header. Barry Traver mentioned this problem to me years ago, but it's never been solved except by sector editing as Jeff did.

Finally, a minor error on page 22 at the bottom. In that table, 256K should equate to 10 groups of 24K. That should be pretty obvious, but just a slight misprint.

Bruce Harrison
Hyattsville, Maryland

Where can users get repairs?

Our TI, Myarc and CorComp systems are aging. Therefore, many of us either need or eventually will be needing to have repairs done on our hardware.

Please publish a list of all known current repair facilities for the three equipment manufacturers. You may have this information on hand, but may not. Perhaps you could first poll your readers for their input on this matter, and then publish a current repair facility listing. As the information changes, updates will be needed.

Oliver D. Hébert
Brewton, Alabama
See the Comments column.— Ed.

The only one?

There is very little interest in the TI99/4A around here at the present. Our group still refers to itself as a "TI99 User Group," because we all still own, and some actually use the TI for business around the home (besides games). I like to read about what others are doing with it, though, and it is encouraging to see that programs are still being developed, and improvements made for it.

Keep up the good work. When the Y2K bug strikes, the TI may be the only computer still in operation! My clone checked out as "compatible," but then who *knows*.

Gordon H. McCaa
Lugoff, South Carolina

Hanging in

You guys amaze me, but as long as you can hang in there so will I just for old times sake if nothing more. Where are your advertisers?

Sending my best to you and may your future be bright.

Gene Barrett
Redding, California

New software debuts
BUD MILLS NAMED BIRDWELL RECIPIENT
32-BIT VERSION OF PC99 DEMO'D

Story and Photos
BY CHARLES GOOD
The following was posted on the TI listserver.—Ed.

I have been attending the annual Chicago Faire for many years. It is always nice to actually talk to TI personalities that you normally only read about, and it is nice to see the new stuff that continues to appear for our ancient but favorite computer. This year was no exception. There was interesting new hardware and software.

The Faire is "International" because European user groups are usually represented. This year Berry Harmsen was there representing the TI Club Errorfree (Germany) and the TI Gebruikersgroep (Netherlands). He had a free disk of mostly new software from these groups that he gave to those attending the Faire. Contrary to pre-Faire publicity, this disk is copyable. Berry also had a European-made SCSI card which he ran on a 99/4A system. Attached to the SCSI card was a zip drive set as SCS7. He used this system to demonstrate the new European software. Unlike my early production Western Horizon SCSI card, the European card had connectors for both an internal SCSI cable and an external SCSI cable (some sort of D connec-

Berry Harmsen demonstrates Disk Manager 2000.

tor). The external zip drive was attached to the external connector of the SCSI card.

The most interesting piece of software Berry demonstrated was a 40-column disk manager that is fully compatible with floppy disks, with MFM hard drives controlled by Myarc's HFDC, and with SCSI hard drives. When you catalog the root directory of a hard drive the listing shows all the subdirectories at the end of the list. If you click on a subdirectory you get a listing of that subdirectory and at the top of the list you are given the opportunity to go back to the previous (next highest) directory. You can move up and down the entire directory structure. This disk manager lets you execute copy move rename and protect files, and

Berry Harmsen demonstrates
Disk Manager 2000.

The Faire ended with dinner and the announcement of Bud Mills as the 1998 Birdwell award winner.

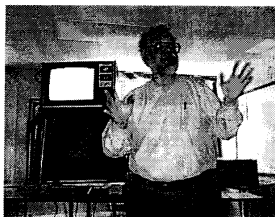Lew King talks about using the TI on the Internet.

Bruce Harrison introduced TI
Bingo and MIDI Play In.

CHICAGO TI FAIRE

Continued from page 5
you can view files in ASCII and hex. I have reviewed a number of 40-column disk managers in my MICROpendium column, and this seems to be the best such product to date. All 99/4A users with hard drives should check out this disk manager.

TI BINGO CALLS OUT THE NUMBERS

Bruce Harrison demonstrated his new TI BINGO software. It prints out bingo cards, up to four on a single 8.5 x 11-inch sheet of paper. It then speaks the bingo letter number combinations and puts the called combinations on screen for reference. You can run an entire large-scale bingo operation with this software. It is fun. Bruce also demonstrated the non-AMS version of his MIDI Play In software. He convinced a member of the public to play a short piece on a keyboard and then the 99/4A played it back. The non-AMS version stores up to 2,400 notes. This is actually quite a bit of music. The AMS version stores 2,400 notes in each block of six AMS memory pages.
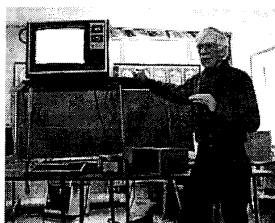
Lew King talks about using the TI on the Internet.

TERM-80 AND THE INTERNET

Lew King demonstrated how a 99/4A using Term-80 software can access the Internet, read the text of web pages, and read e-mail. The trick is to get an Internet access provider to provide a Unix shell account. The available Chicago Internet access provider could only provide a PPP connection, which we found won't work with Term-80. Lew ended up having to telephone long distance to a known Unix shell provider.

32-BIT VERSION OF PC99

Mike Wright of CaDD Electronics showed the latest development trend for PC99, a prototype Windows 95/98 version of PC99. This is a native windows 32-bit program that can make use of the various fancy features that windows provides. At the top of the PC99 window one of the options is "cartridge." Click on this, and a window opens up that allows you to select the cartridge you want to run with PC99. Another option is "size." Click on this and you

Bruce Harrison introduced TI Bingo and MIDI Play In.

CHICAGO TI FAIRE

The Faire ended with dinner and the announcement of Bud Mills as the 1998 Birdwell award winner.

get your choice of three different PC99 window sizes. Another option is "disk." When this is made operational you can click on it and from a window load any PC99 "disk" into your various PC99 disk drives. Mike demonstrated multiple PC99 programs running on screen simultaneously, each in its own window and each running a different cartridge. This prototype is not yet ready for prime time. It is version 0.1 and was not released. Mike makes no promises when, if ever, the windows version of PC99 will be available to the public. Now for the bad news. Mike said that development of 80-column 99/4A emulation has been suspended because the programmers can't figure out how to fully emulate the actions of the 9939/9958 video chip.

BUD MILLS NAMED BIRDWELL WINNER

Bud Mills was the best known TI vendor present. In response to a direct question from me, Bud said that he still supports his products and still repairs them. People are welcome to send him their Horizon RAMdisks, PGRAM cards, MEMEX cards for repair or upgrade. He said that his fees, payable after the repair is completed, would average $35 plus return shipping. He claims that, except for a couple of cards found in a corner of his home shortly before the show, all cards sent to him for repair have been returned to their owners. If you have questions about this, Bud's e-mail address is budmills@stax.net and his phone number is 419-385-5946. Bud had some PGRAMs for sale at his table.

At an informal dinner after the faire Hal Shanafield, faire organizer, presented the 1998 John Birdwell Memorial Award for Excellence in Computing to Bud Mills. This award is financed by shareware fees sent in by users of the late John Birdwell's DSKU (Disk Utilities) disk manager.

A new disk manager for TI/Geneve users

BY BERRY HARMSEN

One of the bonuses for those attending this year's Chicago TI Faire was a free copy of a new disk manager called Disk Manager 2000. The program was distributed by Berry Harmsen of the Dutch TI users group. — Ed.

DM2000

DM2000 was written by Fred Kaal for the TI, with support for Western Horizon SCSI devices as well as floppies and MDM hard drives.

As many readers know, the WHT SCSI controller card was first available to Geneve users. It took several years before the SCSI card could be used by TI99/4A users. This delay, of course, was due to development of an EPROM with DSR support for the TI.

With the SCSI-card came the sector-editor SCUZZY, written by Brad Snyder. A wonderful program! It has a routine to format SCSI devices and one to view the headers of the cards in use. And there were several existing programs that worked fine with SCSI. Very useful was the last version of BOOT, written by John Johnson of the Miami TI User Group. It could show and print SCSI directories and it could load program files right from the SCSI disk.

With the first SCSI EPROM came the first disk manager, written by David Nieters. This disk manager is quite useful, but it has its limits — you have to work your way file by file. How-ever, it remains the only program for initializing a SCSI device

Fred Kaal, of the Dutch TI-group (TI-Gebruikersgroep), is a Geneve user who needed a disk manager to control his Myarc Hard and Floppy Disk Controller. So he wrote one. At the point that the program was ready, he took home the two SCSI cards (the American card by WHT and the German version by Michael Becker) and an Iomega SCSI drive. After an extra month of pro-gramming he brought the result to a Dutch TIUG meeting. He called it Disk Manager 2K (two kilo; that's 2,000 grams). So a new disk manager was born.

Disk Manager 2000 is a file manager for use with floppy and hard disk drives controlled by HDFC and SCSI cards. It is possible to access files in a directory structure by entering a full pathname instead by drive number as used by programs written in the pre-directory age. There was no intention to replace the fabulous Disk Manager 1000 program. How-

The DM2000 help screen is available at the touch of a key.

The DM2000 execution screen lets users select tasks and then execute them.

ever, the need for a pro-gram like DM2000 for the TI, for SCSI-devices and more or less for the Myarc 9640 is high.

The help screen of the program gives you an idea of the possibilities of DM2K (see Fig. 1).

With Disk Manager 2000 it is possible to protect and unprotect files, rename files, copy files, move files, delete files, view a file in ASCII and hex (program files only in hex), create a subdirectory, remove a subdirectory, rename a subdirectory and print the contents of a subdirectory. It is not possible to copy subdirectories, delete subdirectories which are not empty, protect subdirectories, or create subdirectories on floppy disks.

The execution screen of Disk Man-ager 2000 gives you the device information, information on 12 files

at a time and the status of your computer. (See Fig. 2 for a screen-shot.)

At this moment the TI-99 user needs all four above mentioned pro-grams for controlling his SCSI device. To show the main functions of these programs this table makes this clear.

Disk Manager 2000 was programmed on the Geneve in GPL mode and tested on the TI99/4A with the WHT SCSI card and the SNUG ASCSI card. You need an E/A module to run it (option 5). TI users can receive a copy of Disk Manager 2000 by sending a self-addressed return label and $5 in U.S. currency to the Dutch TI Users Group in Amsterdam:

TI-Gebruikersgroep
Attn: Berry Harmsen
1e Oosterparkstraat 141E
NL 1091 GZ Amsterdam
The Netherlands
E-mail: berry.harmsen@wxs.nl

Frederik George Kaal
E-mail: fgkaal@gironet.nl

POS is one of most useful programming functions

BY W. LEONARD TAFFS

This article first appeared in the newsletter of the SouthWest Ninety-Niners user group.—Ed.

At our last general user's meeting, discussion of the use of "IF POS" in programming came up in the question and answer session.

In the course of this discussion someone was overheard to say "I don't know what (they're) talking about!"

> The function of an IF POS statement is to locate the POSITION of something, the "something" being a numeric or text character or group of characters.

In the Extended BASIC manual, POS is explained on page 145. At first I found this hard to understand but by practicing trial-and-error experiments, using this function, I began to make some sense of it. As the POS function is one of the most important and useful functions the TI-99/4A can execute, it is worth becoming familiar with if you have not used it before in your own programming.

In the first place, what use is IF POS? Well, the letters "POS" are the first three letters of the word POSITION, so the function of an IF POS statement is to locate the POSITION of something, the "something" being a numeric or text character or group of characters. This makes it an indispensable function for use in data search programs,

The following is developed from the XB manual, p.145.

```
1 REM [IFPOSXB145] 5-20-98
100 CALL CLEAR :: X$="1201,0
104,ABCO,67.34,036" :: Z$=X$
110 PRINT "X$=": : :X$:
120 S=POS(X$,CHR$(44),1)
130 IF S=O THEN CT=CT+1 :: P
RINT CT;X$ GOTO 170
140 CT=CT+1Y :: Y$=SEG$(X$,1,
S-1):: PRINT CT;Y$
150 X$=SEG$(X$,S+1,LEN(X$))
160 GOTO 120
170 PRINT : :
180 LINPUT "ITEM: ":IT$
190 IF POS(Z$,IT$,1)THEN PRI
NT :IT$;" is PRESENT in X$":
:ELSE PRINT :IT$;" is NOT f
ound in X$!": :
200 DISPLAY AT(24,1):"Press
<ANY KEY> to try again" :: C
ALL KEY(O,K,S):: IF S<1 THEN
```

```
200 :: CT,S=O :: GOTO 100
```

Up to line 160 is modeled after the page 145 example, and then I have added lines 170-200 to allow one to experiment entering any items to see if the IF..POS in line 190 will find what you enter or not.

For another example, say you want to locate the word "machine" in the phrase:

"The ultimate purpose of the machine was clear."

```
.......0....1....1....2....2...2.................
123456789012345678901234567893123456789412345678
```

The numbers below the phrase above show the first character of the word "machine" is located at the 29th column.

The following short program assigns the variable A$ to this phrase string, and the variable "P" is assigned to indicate what the POS status is. If it finds "machine," P will equal 1, if not, P will equal zero.

```
100 A$=(the sentence above)
110 P=POS(A$,"machine",1)
120 IF P THEN PRINT A$,P
```

Run this program and it will print your sentence, and print "29." The 29th character is where it found the beginning of the word. Now change the spelling of machine in line 110 to "Machine." When you run the program you will get nothing (P will equal zero), as POS(A$,"machine",1) is case-sensitive.

Using an IF-THEN statement, the above lines 110 and 120 could have been written:

```
110 IF POS(A$,"machine",1)
THEN PRINT A$
```

Note, a single POS statement only finds the first occurrence of the word which you will see if you try the following:

```
100 A$="The machine failed
the machine test."
        12345
```

The POS function finds the word at 5 but does not show you the second occurrence.

To locate the second occurrence, another IF POS is necessary (unless you develop the version shown above from the XB manual).

```
100 A$="The machine failed
the machine test."
110 A=LEN(A$) :: P1=POS(A$,"
machine",1)
120 A1$=SEG$(A$,P1+8,A)
130 P2=POS(A1$,"machine",1)
140 PRINT A$,P1 :
150 PRINT A1$,P2
```

In order to find the second occurrence, the separate string A1$ had to be set up to begin at the next word after the first occurrence of "machine" (or one can use the variation of it found in tbp XB manual version above). This illustration was what was being discussed at the last general user's meeting.

EXTENDED BASIC

Continued from page 13

```
3 ! Enter FREQUENCY
          DURATION
          DECREMENT
4 !
5 ! Suggested Start with
    FREQ 1000 Duration 10
    Decrement by 25
6 !
40 CALL CLEAR :: INPUT "ENTE
R FREQ: ":FR :: PRINT :: INP
UT "DURATION: ":X :: PRINT :
: INPUT "DECREMENT BY: ":DC
:: CALL CLEAR
50 C=FR
90 DISPLAY AT(24,1):"FREQ.:"
;STR$(FR);" DUR:";STR$(X);"
DECR:";STR$(DC):: CT=1
100 IF C<110 THEN STOP ELSE
CALL SOUND(X,C,30,C,30,C,30,
-1,1)
110 CALL SOUND(X,C,30,C,30,C
,30,-2,1)
120 CALL SaM(X,C,30,C,30,C
,30,-3,1)
130 CALL SOUND(X,C,30,C,30,C
,30,-4,1)
140 CALL SOUND(X,C,30,C,30,C
,30,-1.5,1)
150 DISPLAY AT(CT+1,1):RPT$(
" ",56):: DISPLAY AT(CT,1):C
:: IF CT=21 THEN CT=1
160 C=C-DC :: CT=CT+1 :: IF
C<100O THEN X=X+10
170 GOTO 100
```

As suggested in the REMarked lines at the beginning, you have three inputs to enter for Frequency,

Duration, and Decremental value. The program plays two reference tones and then the Frequency that you enter. The program continues sounding the decremented frequencies until it can go no longer. Chances are that as it reaches its lowest tones you will find it hard to tell the difference between the lowest notes. If you can hook up your computer to a good sound system you way get more surprising results. That's just my guess, as I have not tried hooking mine up to a sound system. This is just a beginning experiment of mine — you way be able to improve on it.

And on the subject of sound, I wanted a motorcycle sound. Don't remember seeing one though I'm sure they could be found in many game cartridges, etc. Anyway, not knowing where I could find one I tried this experiment. Again, maybe you can improve on it. I wanted to flash a couple of messages while the program was running, but when I tried putting then in they slowed up or messed up the sound too much.

```
1 REM (MOTORCYCLE) 5-4-98
    SOUND EXPERIMENT BY
    W.LEONARD TAFFS, SW99ERS
2 !
100 CALL CLEAR :: DISPLAY AT
(12,8):"MY NEW BIKE.."
110 X=110 :: DISPLAY AT(24,1
):"STARTING UP AND TAKING OF
F.. "
120 CALL SOLND(-500,440,30,4
40,30,X,30,-4,0)
130 ! test spare line
```

EXTENDED BASIC

```
140 CT=CT+1 :: X=X+15
150 IF Y THEN 180 ELSE IF X>
=1300 THEN X=X+25
160 IF CT=200 THEN Y=1
170 GOTO 120
180 CT2=CT2+1 :: IF CT2<35 T
HEN 120 ELSE 190
190 DISPLAY AT(24,1):"OOPS,
I'M RUNNING OUT OF GAS" :: X
=4000 :: GOTO 220
200 DISPLAY AT(23,1):" THAT'
S ALL THERE IS, FOLKS!":" HO
PE YOU ENJOYED THE RIDE! "
210 FOR DLY=1 TO 500 :: NEXT
DLY :: STOP
220 CALL SOUND(-500,440,30,4
40,30,X,30,44,0)
230 X=X-40 :: IF X<=110 THEN
    200 ELSE 220
```

The LOW NOTES program I demonstrated at the last general user's meeting was a beginning attempt to identify low Real Bass Notes approximately. For those who are familiar with musical notes, my use of the note C is a transposition from A. In other words, the descending scale approximations are those approximating descending scales from TI's lowest listed frequency of 110.

LOWOCTAVES

```
1 REM [LOWOCTAVES] 5-3-98 !2
04
2 !!131
3 ! Inspired by Chick De Mar
   ti's article in LA99er
   TopIcs 8810 issue p.6 !
```

```
047
4 !!131
5 ! by W.Leonard Taffs
    SW99ers, Tucson, AZ. !12
7
6 !!131
10 GOTO 100 !179
20 A,AB,B,BB,C,C1,D,DB,DLY,E
,EB,F,G,GB,K,S,X !252
30 CALL CLEAR :: CALL KEY ::
 CALL SOUND !025
40 @ !P- !064
100 CALL CLEAR ! A=110 is LO
WEST CALL SOUND Frequency pe
rmitted by TI !219
110 DISPLAY AT(3,2):"NOTES L
OWER THAN TI'S 110": :"CALL
SOUND"" LIMIT"" !" !210
120 DISPLAY AT(7,1):"By W. L
eonard Taffs, SW99ers": : :"
 Inspired by old article by
":" Chick De Marti, LA99ers
":" from TopIcs Oct. 1988 P.
9" !123
130 DISPLAY AT(24,1):"Press
<ANY KEY> to Continue" :: CA
LL KEY(0,K,S):: IF S<1 THEN
130 :: CALL CLEAR !016
140 A=110 :: DISPLAY AT(24,1
):"Lowest CALL SOUND: (A=110
)" :: CALL SOUND(1500,A,0)!1
30
150 FOR DLY=1 TO 400 :: NEXT
DLY :: CALL CLEAR !145
160 REM ** start down-step *
* !145
170 ! (A=DO) !005
```

EXTENDED BASIC

```
180 A=440 :: DISPLAY AT(24,5
)"Standard: A=440" :: CALL
SOUND(2000,A,0)!000
190 FOR DLY=1 TO 300 :: NEXT
 DLY :: CALL CLEAR !144
200 A=220 :: DISPLAY AT(24,1
1):"A=220" :: CALL SOUND(200
0,A,0)!145
210 FOR DLY=1 TO 300 :: NEXT
 DLY :: CALL CLEAR !144
220 A=110 :: CALL SOUND(2000
,A,0):: DISPLAY AT(24,11):"A
=110" !141
230 FOR DLY=1 TO 300 :: NEXT
 DLY :: CALL CLEAR :: DISPLA
Y AT(20,1):"START DESCENDING
 from 110": :" All BELOW 110
!":"frequencies are guesst
imates" !233
240 C1=1675 :: CALL SOUND(80
0,C1,30,C1,30,C1,30,-4,1)::
GOSUB 560 ! C1=A !227
250 B=1550 :: CALL SOUND(800
,B,30,B,30,B,30,-4,1):: GOSU
B 560 !126
260 BB=1500 :: CALL SOUND(80
0,BB,30,BB,30,BB,30,-4,1)::
GOSUB 560 !129
270 A=1400 :: CALL SOUND(800
,A,30,A,30,A,30,-4,1):: GOSU
B 560 !116
280 AB=1335 :: CALL SOUND(80
0,AB,30,AB,30,AB,30,-4,1)::
GOSUB 560 !131
290 G=1260 :: CALL SOUND(800
,G,30,G,30,G,30,-4,1):: GOSU
```

```
B 560 !144
300 GB=1180 :: CALL SOUND(80
0,GB,30,GB,30,GB,30,-4,1)::
GOSUB 560 !153
310 F=1100 :: CALL SOUND(800
,F,30,F,30,F,30,-4,1):: GOSU
B 560 !133
320 E=1040 :: CALL SOUND(800
,E,30,E,30,E,30,-4,1):: GOSU
B 560 !132
330 EB=980 :: CALL SOUND(800
,EB,30,EB,30,EB,30,-4,1):: G
OSUB 560 !103
340 D=925 :: CALL SOUND(800,
D,30,D,30,D,30,-4,1):: GOSUB
 560 !090
350 DB=870 :: CALL SOUND(800
,DB,30,DB,30,DB,30,-4,1):: G
OSUB 560 !097
360 C=840 :: CALL SOUND(800,
C,30,C,30,C,30,-4,1):: GOSUB
 560 ! C=A !182
370 DISPLAY AT(15,1):"COMPAR
ISON A'S Sounding...." !104
380 A=440 :: CALL SOUND(2000
,A,0):: A=220 :: CALL SOUND(
2000,A,0):: A=110 :: CALL SO
UND(2000,A,0)!080
390 FOR DLY=1 TO 300 :: NEXT
 DLY :: CALL CLEAR :: DISPLA
Y AT(24,1):"Resuming Descend
ing Notes..." !151
400 C=840 :: CALL SOUND(800,
C,30,C,30,C,30,-4,1):: GOSUB
 560 ! C=A !182
410 B=800 :: CALL SOUND(1000
,B,30,B,30,B,30,-4,1):: GOSU
```

EXTENDED BASIC

```
B 560 !116
420 BB=765 :: CALL SOUND(100
0,BB,30,BB,30,BB,30,-4,1)::
GOSUB 560 !134
430 A=725 :: CALL SOUND(1000
,A,30,A,30,A,30,-4,1):: GOSU
B 560 !118
440 AB=690 :: CALL SOUND(100
0,AB,30,AB,30,AB,30,-4,1)::
GOSUB 560 !127
450 G=650 :: CALL SOUND(1000
,G,30,G,30,G,30,-4,1):: GOSU
B 560 !139
460 GB=615 :: CALL SOUND(100
0,GB,30,GB,30,GB,30,-4,1)::
GOSUB 560 !148
470 F=580 :: CALL SOUND(1000
,F,30,F,30,F,30,-4,1):: GOSU
B 560 !137
480 E=548 :: CALL SOUND(1200
,E,30,E,30,E,30,-4,1):: GOSU
B 560 !139
490 EB=520 :: CALL SOUND(150
0,EB,30,EB,30,EB,30,-4,1)::
GOSUB 560 !140
500 D=490 :: CALL SOUND(1500
,D,30,D,30,D,30,-4,1):: GOSU
B 560 !134
510 DB=467 :: CALL SOUND(150
0,DB,30,DB,30,DB,30,-4,1)::
GOSUB 560 !146
520 C=448 :: CALL SOUND(1500
,C,30,C,30,C,30,-4,1):: DISP
LAY AT(24,1):" Lowest Audibl
e A" !065
525 FOR DLY=1 TO 400 :: NEXT
 DLY :: GOSUB 560 :: DISPLAY
```

```
AT(24,1):" Even Lower..." !
139
530 B=432 :: CALL SOUND(1500
,B,30,B,30,B,30,-4,1):: GOSU
B 560 !122
540 BB=420 :: CALL SOUND(150
0,BB,30,BB,30,BB,30,-4,1)::
GOSUB 560 !127
550 X=1 :: A=405 :: CALL SOU
ND(1500,A,30,A,30,A,30,-4,1)
:: GOSUB 560 !008
560 REM ** Comparison Tone *
* !115
570 IF X<>1 THEN 600 ELSE DI
SPLAY AT(22,1):"Nearing Ear'
s Low Threshold":"Use ""Q"
" to Leave Program." !049
580 CALL KEY(0,K,S):: IF S<1
 THEN 580 !140
590 IF (K=81)+(K=113)THEN ST
OP ELSE A,X=0 :: GOTO 140 !2
37
600 RETURN !136
```

Takach, TIsHUG member, dies

Benedek von Takach Dukai (Ben von Takach) died Oct. 16 in Australia. He was a member of TIsHUG and a programmer who had written articles for MICROpendium.

He was born Oct. 27, 1926. He was multilingual and a leader in the field of diecasting technology, according to the TIsHUG News Digest. His wife, five children and a number of grandchildren survive him.

Where is Bryn Mawr?

BY BRUCE HARRISON

Depending on which side of the Atlantic you're on, it's either in South Wales or on Philadelphia's Main Line. Of course in our case it's a trick question, because today's column deals with the subject of a word search program, not directly with geography. As we'll explain, however, this program could be used to help in teaching that subject.

This project was the direct result of a call for help from Ed Morse, of Phoenix, Arizona. Ed had been trying to modify an old TI BASIC program which generated word search puzzles to suit his own needs. For those unfamiliar with the concept, a word search puzzle (or word find puzzle) is a rectangular matrix of letters in which words are hidden. The words can run horizontally, vertically, or diagonally, in a total of eight directions through the matrix. There are also letters in the matrix which don't belong to any of the words.

Ed was trying to use this old BASIC program to create puzzles containing the names of members of his family. He had 108 names in all, and the matrix of only 25 by 25 was nowhere near large enough for that purpose. Further, the BASIC program required him to type in each name on each run. That's one whale of a lot of typing! Ed had expanded the original program's matrix, but still found that not all the names would fit.

Our first problem was to deal with all that typing on each run. Why would anyone design such a program to require typing in each name each time? Perhaps this was done for a console BASIC case in which no disk drive was available.

Our first order of business, then, was to fix the input section so that one could have the names in a D/V80 file, which only needed to be typed once, then have the program read the names from the file. We made that and a couple other small changes, making the matrix 32 by 32 letters, and passed a copy back to Ed. Still, this BASIC program would not always fit all of his 108 names into the puzzle, and it was painfully slow in trying.

ASSEMBLY TO THE RESCUE

Having failed to completely understand why the BASIC program couldn't do the job, we set out to make an assembly version that would. To do this, we borrowed source code from some of our other programs, then devised our own algorithm for placing the words. Like the BASIC version, ours used random numbers to decide in which direction each word would be placed. Since there are eight such directions, the random number had to range only from zero through seven. That much was easy.

We devised our scheme so that 0 meant straight across left to right, one meant downward and right, two meant straight downward, and so on through

seven, which meant upward and to the right. For each of those directions, we would establish a last row and last column for placement, based on the length of the word being placed. In other words the program would not try to place the word where there was not enough room. But we're already getting ahead of the story.

THE MATRIX STRUCTURE

In the special case for Ed's family, the matrix has to have 32 rows of 32 columns. In our assembly version, this is simply a block of 1024 (32 times 32) bytes which are all set to zeros before we start placing words. To place a word, we start with its first letter and a starting point in the matrix. If that element of the matrix is a zero or if it's got a matching letter, then we proceed to examine the next appropriate spot in the matrix. If that element of the matrix is a zero or if it's got a matching letter, then we proceed to examine the next appropriate spot in the matrix. This continues until we've found the end of the word's length or we've found a letter that doesn't match the word. If we got through the whole word, then the letters of that word are placed into the matrix, and we can move on to the next word. If that word wouldn't fit in that place, we try another starting point and repeat the process. If we find that this word can't fit in that direction, then we go back and try the next direction.

We make eight direction trials, so the word has every possible chance to be placed. Each time a word is fit into the matrix, it gets added to a list of used words in memory. This list gets used later by the program, to print out the list of words to be found. The used list is kept separate from the list as read in from the disk. That's important in case there are one or more words that won't fit into the matrix. In this "special for Ed Morse" version, that wasn't a problem, as all 108 names would fit nicely into the 32 x 32 matrix.

THE PUBLIC DOMAIN VERSION

For the more general-purpose release version, we used the original 25 x 25 matrix size as in the old BASIC program. The program starts by showing a title screen, which stays on screen until a key is pressed. The delay between startup and the pressing of a key is used to seed the random number process. Once a key has been pressed, the user is asked to supply a file name.

There are a number of ready made word files supplied on the disk, so the user can try out the program immediately. Each is a short D/V80 file (three or four sectors) with words related to a single topic. For example, the file ANIMALS contains the names of animals, COUNTRIES contains the names of countries from around the world, FOODS contains names of things we eat, and so on. Each file contains between 50 and 70 words, depending partly on the length of words involved. Files with many long words may have as few as 50 entries, while those with mostly shorter words may have 70 or more entries.

Given a complete file name (e.g. DSK1.FOODS), the program proceeds to open that file and read each word. As they're read, the words appear on the

screen and scroll up quickly. When the end of the file is found, the program reports on the screen how many words were found, then puts the legend "PLACING THE WORDS" on the screen. In the next couple of seconds, all of the words get placed into the matrix. Compared to the BASIC version, which takes a very long time to place words, this is practically instantaneous.

THE FIRST 11

Just to make the puzzles a bit tougher to solve, we take additional random numbers for the first 11 words in the list. Each of them gets a randomly chosen row and column for its starting point. Thus these 11 words are, in general, more hidden than the others. After the first 11 are placed, the others are fitted in according to an algorithm that places their initial starting points based solely on the direction the word is to go.

> Just to make the puzzles a bit tougher to solve, we take additional random numbers for the first 11 words in the list. Thus these 11 words are, in general, more hidden than the others.

For example, if a word is being placed down and right, its first trial placement will be at the upper left corner. If up and left, its first trial position will be at the lower right corner, and so on. The program then cycles through all possible positions where this particular word can fit, until either the word is placed or it's found that there's not room for this word in this direction. The program will then try placing this word in seven other directions, and will usually find a place for it.

Letters placed in the matrix can of course belong to more than one word where they match up. In many cases, for example, you may find a "C" at the upper left corner that's the first letter of three words, one going across left to right, one going straight down, and one on the diagonal down and right. The same sharing may happen anywhere, as words going in different directions may share a letter where they cross over each other. In most cases, all of this gets done in about two seconds, then the puzzle is almost ready for printing.

The last step in forming the matrix is the "random fill" process, in which the program examines each of the 625 bytes in the matrix. For each one that still contains zero, the program inserts a randomly chosen uppercase letter. This takes essentially no time because of the rapid and efficient way the program generates random numbers.

THE EXCEPTIONAL CASE

Sometimes a word list will contain long words toward the end, and it's possible that one of these won't be able to be placed because earlier placements have not left a suitable spot for this word. When that happens, the program stops and offers the user three options on which to proceed.

- Option 1 is to re-start placement.
- Option 2 is to skip this particular word and continue with the next one.
- Option 3 is to just stop and print the matrix with those words already placed.

If the user chooses Option 1, he actually has two possible ways to go after making that choice. The program asks whether the user wants to make a new try with the same data. If yes, the program re-clears the matrix to all zeros, then starts the placement process again, using the word list that's already in memory from the file. This is why the list of used words is kept separate from the original list read from the file, so that the placement process may start again without re-reading the file. If the user answers this "same data" prompt with any keypress other than "Y" or "y," the program goes back to the file name input prompt so a different file may be tried.

Usually even a difficult word list can all be placed in one or two more tries, simply because a different set of random numbers creates a new situation for each placement attempt. On rare occasions a word list may be found that may take five or six tries to get placed. When that happens to us, we go back and re-edit that word file, leaving fewer or shorter words in the list.

THE PRINTING PROCESS

Once the program has filled the matrix in memory, we're ready to print. The user gets a prompt for the printer file name, with the "PIO" option already in place as a default entry. If that's the user's desire, just pressing Enter will start the printing process.

When we first distributed a few test copies of this program, we had good reaction from almost every tester, but got one surprise answer. That was from John H. Bull, of Knoxville, Tennessee. Seems what he wanted was to make many identical copies of just one puzzle. To do that, he saved the puzzle to a disk file instead of sending it to the printer. In that early version, this idea worked, but the resulting disk file contained unwanted carriage returns and line feeds. In the latest update, that condition has been corrected, so that if the puzzle is sent to disk, no carriage returns or line feeds will be included.

The matrix itself gets printed centered in the 80 columns on the paper, with a space between each letter going across the paper. An escape sequence gets sent out before the matrix, to set the line feed so that the spacing between letters will be the same in both horizontal and vertical directions. That makes it easier to

THE ART OF ASSEMBLY   PART 73

find words along the diagonals.

As we all know by now, setting a different line feed amount for 9-pin printers and for 24-pin printers is a tricky process. Thus on the public domain release disk there are two versions of the main program, so that the disk can be used with either type of printer and gives satisfactory results in both cases. Another escape sequence is sent before the word list gets printed, to set the vertical spacing back to the normal one-sixth inch. These escape sequences are also included if the file goes to disk, so that a puzzle saved to and then printed from disk will look no different from one printed directly from the program.

The list of words gets printed in columns, after the program determines how many columns are needed and how many words are to be included in each column. Between the matrix and the word list, the program prints a line which says "FIND THESE XX WORDS IN THE PUZZLE ABOVE." The "XX" is of course the number of words actually placed in the matrix, which may not be the same as the number of words in the original word file. (Remember that the user can skip over words that didn'[t fit.) This way, the user is assured that all words printed in the list are actually somewhere in that puzzle above, hard though it may be to find them.

On rare occasions, included words may get placed over a longer "including" word. Suppose that a list included both the words "JONES" and "JONESBORO." It's possible, though unlikely, that the word "JONES" will get placed, then JONESBORO placed right on top of that, so that nowhere in the puzzle will JONES appear by itself. In hundreds of test runs with various word files, this has only happened to us once, but be warned that it can happen, and can drive the solver crazy.

THE FINISHING TOUCH

After the matrix and word list are printed, the program places the name of the original word file at the bottom of the sheet. This allows the user to quickly determine what kind of words he's looking for, as well as just knowing where the list came from. For example, one wouldn't be looking for ROBIN or MEADOWLARK if the file name was CARS. Of course there are some cases where a word may appear in more than one list. COUGAR, for example, is in the list file called CARS, and also in the one called ANIMALS. There are 21 word files supplied on the disk, including many topics. They range from ANIMALS through TREES, and include such things as FOODS, SPORTS, etc.

THE GEOGRAPHY LESSONS

Among the files supplied are ones that could be used as part of Geography lessons. For example, there's a file called CITIES, which includes names of cities large and small in the United States. For our friends overseas, there are ones called CANCITY (for Canada), BRITCITY (for the U.K.), and IRELAND and

THE ART OF ASSEMBLY   PART 73

GERMANY for those countries. There are also other geography oriented files such as LAKES (of North America), RIVERS (of the World), and MOUNTAINS (of the world). Using these, a student could first find the words in the puzzle, then go to the atlas and find the locations. Alex Trebek would be proud of you for doing this!

Just for the fun of it, there are files such as FOODS, which may have the side effect of making you hungry, and SPORTS, which includes esoteric ones like curling, biathalon, and so on. Of course this is all intended just to spark your own imagination. John Bull, for example, made a file of words for his wife's hobby of making miniatures. Over in Dorset, England, John Murphy made files of towns in England, Scotland, and Wales.

You'll notice that there's no sidebar this month. Mainly that's because the whole program is much too large to publish all its source code, but also because nothing exotic was done that hasn't been covered before in this column. The complete source code is included on the public domain disk for those who care, although it's not completely annotated. As always, the disk containing this program has been made available through the Lima user group, so it's accessible to the whole community. Contact Dr. Charles Good at P.O. Box 647, Venedocia, OH 45894.

EXTRA ADDED ATTRACTION

When we are making up word files for use with this program, we don't necessarily think up words in alphabetical order, but that's how we want them in the file. Thus we went back a long way in this series (to Part 24) and found the source for our File Sorter. We modified that slightly, and saved it as an Option-5 program file called simply "SORT." SORT is supplied on this public domain disk, so that once you've made a file of words, you can sort that file in a couple of seconds using this program.

The file gets sorted while it's being read from disk, so that as soon as the file closes on input, it's ready to be saved back to disk in sorted form, and its existing file name is already there in the input field for OUTPUT NAME. This sort program can actually be used for other purposes, as it will correctly sort any D/V80 file that's 100 sectors or less in length.

The limits on word files are fairly simple. The program will accept words of any length from two through 20 letters. If there are blank lines in the file, the program will ignore them, as it will words longer than 20 or shorter than two letters. All words should be in uppercase only, but the program doesn't check that. This could be a way to cheat, by using word files all in lowercase, as the random fill letters are all uppercase. (We don't recommend cheating, but it is possible.)

Next month we'll open a whole new world, concerning use of color printers to print TI-Artist picture files. See you then.

Adding 32K of SRAM to Geneve worth the effort

BY TIM TESCH

The following was posted on the TI list server.—Ed.

A few people have asked how to modify their Geneves with the extra 32K SRAM, to give them extra memory and let them use MDOS 2.50 and above. So. here we go.

Requirements:
• Soldering iron, 15-20watt max.
• Solder, use rosin core NOT organic core
• 8-inch piece of wire-wrap wire
• (1) 32Kx8 SRAM (Hitachi 62256 compatible)

DISCLAIMER: If you are not comfortable modifying your equipment, do not perform this upgrade. Neither the author nor MICROpendium will take responsibility for the outcome.

STEP 1

Carefully remove the 32K chip from the Geneve, noting the orientation. The square/half-circle should be facing the back of the card (joystick/mouse/video connectors.) If you are not sure, use a Sharpie or piece of tape to keep track of the position.

Use a small, wide-blade, flathead screwdriver. Apply gentle pressure to the top of the memory chip with your thumb, while gently prying between the memory and its socket. **Do not pry one side completely.** Doing so will bend the pins on the opposite

side. Pry the chip a bit at a time, switching from one side to the other until the chip is free.

This chip is located to the left of the battery, directly below the 12-volt regulator. If you do not recognize this chip — STOP! Send in your card

STEP 2

Stack the new 32K chip on top of the existing memory. Place the new chip on top of the old chip. The square/half- circles MUST lay on top of one another.

Bend pin No. 20 enough so that it does not contact the pin underneath. You may do so using a snall-blade screwdriver. Here is a visual representation:

```
 1 1 1 1 1
 4 3 2 1 0 9 8 7 6 5 4 3 2 1
+---------------------------+
:                           :
:    HM62256-10             (
:    (upside down)          :
+---------------------------+
 1 1 1 1 1 2 2 2 2 2 2 2 2
 5 6 7 8 9 0 1 2 3 4 5 6 7 8
              :
```
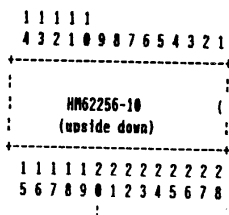
Do not solder tis pin to its neighbor below.

STEP 3

Insert stacked chips into socket. The chip should be inserted as shown above, half-moon to the right. Make sure pin No. 20 clears the EPROM so

that you can solder the wire to it as shown in step No. 4.

STEP 4

Solder wire to memory.

Strip and tin one end of the wire. Solder to pin No. 20, taking care not to short it with any adjoining pins or the EPROM directly below.

Take the wire and bring It to the underside of the board. You can go around the top or bottom. Do not go through the holes near the edge connector. While tempting, you run the risk of stripping wire and shorting something. Worst-case, you short the wire with +16 volts. This would not be a good thing.

STEP 5

Solder wire to underside of Geneve gate array.

This step MUST be done carefully and exactly. If you make a mistake, you risk blowing out your gate array. This chip will cost you about $45 to replace.

Strip the wire, exposing no more than one-eighth inch. We will solder the wire to the gate array as shown in the following graphic:

```
<<<- mouse/joystick/video<<<

 o o      Gate Array      o o
 o o      (underside of   o o
 o o           board)     o o
 o o                      o o
 o o                      o o
 o o o o X o o o o o o o o
 o o o o o o o o o o o o
```
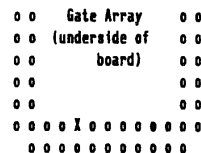
"X" marks the spot. Carefully solder the wire to this pin. Make sure that you do not short any of the three surrounding pins. In case you wish to count, it is the fifth pin from the left, top row.

STEP 6

Test the Geneve.

Replace clamshell and put into your PEB. Turn on the power and see what happens. If all went well, you should boot without any problem. Once booted, type CHKDSK at any valid disk. You should see:

589824 bytes of total memory.

STEP 7

Pat yourself on the back — you've successfully modified your Geneve! Go try MDOS 6.0!

Common Questions About PC99

Do you have questions about PC99, the TI emulator that runs on PCs? Here are some answers from the CaDD Electronics web site. The cost of the software is $47 for the light version and $94 for the full version.

DOES PC99 RUN UNDER WINDOWS NT?

PC99 runs under Window NT 4.0. However, sound using a sound card does not work. The reason is that PC99 directly accesses the AdLib I/0 addresses

PC99

0x388 and 0x389, which Creative Labs and most other sound card manufacturers incorporate on their sound cards (to make them AdLib compatible). NT prevents direct access of hardware ports by an application, and thus sound does not work.

There is reference on the web to a sound card emulator that runs under NT which allows direct access. We have not investigated this at this time, but if anyone has any feedback on it we would appreciate hearing about it.

WHY DOESN'T SPEECH WORK?

The TI Speech Synthesizer is partially emulated. The emulation allows you to run programs that require the Speech Synthesizer, e.g. Lasso; XB routines such as CALL SPGET work; and you can access the TI speech ROM directly with assembly language. The PC99 package also includes two utilities that allow you to examine the speech ROM. However, no audible speech is produced.

There are two requirements for audible speech to work.

1. The TMS5200 and Linear Predictive Coding (LPC) must be understood. We think we can do most of this and have a fair amount of documentation on it. However, there are some internal interpolations using matrices that may be difficult to figure out

2. The Digital Signal Processor (DSP) chip in the Sound Blaster must be programmed. We have had no success in generating speech in any form with the DSP using Creative Labs documentation and other information found on the web. It also may not be possible to program speech concurrently, i.e. the

PC99

execution of PC99 TMS9900 instructions may stop while speech is played.

Until we can solve these problems there can be no audible speech. We are investigating this when time permits.

CaDD has an open offer to any programmer: Write us a callable routine that will allow speech generation and we will attempt to include it in PC99.

WHY CAN'T TI DISKS BE READ DIRECTLY ON THE PC?

Our understanding is that we will not be able to do this reliably, if at all. The best we would be able to offer is a utility that will read a TI disk in a PC 5.25-inch drive and convert it to PC99 format.

The following points are offered to support this argument:

1. If the operation is unreliable, it becomes extremely difficult for us to support. We simply do not have the resources to deal with every PC and every PC 5.25-inch drive combination that might occur.

2. TI disks are 5.25-inch. Most of today's PCS no longer have a 5.25-inch drive.

3. We understand that "recent" PCs with 5.25-inch drives have had a signal line removed which makes it impossible for them to read TI disks. Vintage PCs may or may not work, since the PC controller has difficulty reading sectors numbered 0-8 (TI method) rather than 1-9 (IBM method).

4. We found a company in California that was able to read samples of TI disks we sent them. They were willing to sell their library for $2,500 with the caveat that it was not guaranteed to work with all 5.25-inch drives. The cost was prohibitive.

For these reasons no development is being done in this area. The PC99 product includes a way of moving your disk library to the PC. However, you must have a TI and PC connected together to do this.

For people who no longer have access to a TI system, CaDD provides a disk conversion service.

CaDD has an open offer to any programmer: Write us a utility that will read a TI 5.25-inch disk in a PC 5.25-inch drive and we will include it with PC99. You can do this as shareware and we will include the shareware notice with PC99 if you desire.

Note: There is at least one published routine written in C that we have tried that does not work on our test systems. We have also tested a package developed by Gary Bowser that yields poor results.

WHAT IS PC99 WRITTEN IN?

There are three code versions of PC99: PC99.EXE, PC99A.EXE, and PC99L.EXE. PC99.EXE is written 100 percent in C. PC99A.EXE and PC99L.EXE contain some assembly code to improve performance. All of the PC utilities supplied with PC99 are written in C. The Read Sector and Write Sector utilities

Continued from page 27

are TI programs and are written in TMS9900 Assembly Language.

WHAT DEVELOPMENT TOOLS ARE USED?

Compiler	Watcom 11.0a, using DOS/4GW extender
Assembler	WASM 11.0 (Watcom assembler)
Graphics Library	FastGraph 5.0 for DOS, from Ted Gruber Software
Comm Library	Comm Lib 5.20B, from Greenleaf Sound Creative Labs Software Development Kit. (The cfg program uses some of these routines to test for a Sound Blaster.)
Editor	Brief, from Underware, or **vi**

WHO ARE THE PC99 DEVELOPERS?

There are three developers: Greg Hill, Mark van Coppenolle, and Mike Wright.

Greg developed the PC99 architecture, the original 9900 processor and 9918A VDP emulation, and is responsible for the Mini-Screen code. Mark founded CaDD Electronics, designed The Gramulator, and is responsible for the low-level hardware emulation code and PC99A.EXE (accelerated version), Mike conceived the project, writes the utilities, writes the documentation, and is responsible for order handling.

CaDD Electronics can be reached 45 Centerville Dr., Salem, NH 03079-2674; phone — (603) 895-0119 or (603) 893-1540; web site — http:// pw2.netcom.com/~mjmw/index.html; e-mail — mjmw@ix.netcom.com

Using TI-BASE to create a public records database

BY DICK BEERY

This article has appeared in several user group newsletters.—Ed.

This month I would like to share with you one use I have found for TI-Base that I feel may be of interest to anyone who does research, although the reference is specifically to genealogy-based material.

I have written several earlier columns regarding TI programs that handle genealogical data. Among these are: Gene-III by Walter Davies of Salinas, CA, and Genealogical Record Keeping by Joe Wright of Australia. These two are the best of which I currently have knowledge. I still strongly support the use a program specifically designed for use with genealogy, although some people disagree, and prefer to use a database, word processor, spreadsheet, etc.

The use of TI-Base, or in reality any database, in genealogy that I as speaking

of here, is not for the organization of family data onto Family Group sheets nor four- (nor five-) generation charts. Rather, it is a way of organizing research data, culled from courthouse records, published town/county records, etc.

The data I have was first copied at the resource location by hand into notebooks or by using a copier. My problem was that I have in several instances many ancestors or people with similar surnames who resided in the same area, I found that I had births, marriages, deaths, and naturalization records scattered among many pages of information. Wouldn't it be nice, I said, if I could enter all this information, sort it, and thus have all four types of information together for, for example, the Binder family?

When I was unsure as to whether a particular person was a family member, comparing birth dates and other information might give me a clue for further research. If two ancestors with different surnames have the same European origin and were naturalized on the same date, might this not suggest checking the passenger lists to see whether they came over the ocean together? And I am sure that there are many more uses for this type of data organization that have not even occurred to me yet.

I set up the database as follows:

Type	1 (Enter N, B, M, or D)
Surname	15
Firstne	24
Father	24
Mother	24
Spouse	30
Eventdate	10 (e.g. 1989MAR22)
Agent	10 (e.g. name of sinister, JP, etc.)
Age	9 (e.g. 91-06-23 =91 years, six months, 23 days.)
Cause	15 (e.g. old age, dropsy, etc.)
Origin	10 (e.g. Germany)
Spparents	30 (i.e. parents of the spouse)
Residence	12 (at the date of death)
Occupation	12
Other	30
Reference	24

The "Other" field is used to hold such information as: b. Hocking Twp d. Lancaster OH for a death record; INT/LIC (intention given or license obtained) 1897JUN01. Likewise, anything else not covered in the specific categories listed. Reference would include notations such as FF CTY BIRTHS, followed by the volume, page and entry numbers. The numbers following the field names above represent the number of characters I have allotted to each field.

TI-BASE

Anyone familiar with the use of a database will immediately observe that there is no way all this information may be printed on the same line, using a standard-carriage printer. Because printing may be done selectively, one can choose to include or omit whatever is deemed necessary for the intended recipient or use of the printout.

One format that works for me is to use condensed print and to select only the following columns, in order: SURNAME, FIRSTNM, EVENTDATE, TYPE, SPOUSE, ORIGIN, AGE, CAUSE. This permits me to access all four types of record: birth, death, etc. and to present limited information about each. If I want to include nearly all of the categories in the database, I choose subscript printing and use one-eighth inch line spacing. This is not recommended for those with poor vision, although it prints bright and clear and well-spaced. I find that condensed print is unsuited for printing nearly all the information, unless space between lines is increased. Obviously, any time that most of the information is printed, data for any given individual will appear on two successive lines of print.

Some options to consider:

• Select all records where Type = B (or N, etc.), then print only the information pertinent to that type of record

• Make two or more printouts of the whole record, including the surname and firstnm with each, plus some of the other fields, until the whole set of information has been printed out

• Whatever your imagination or needs dictate.

Some cautions:

• Eventdate should be entered as e.g. 1875DEC16 if you wish to sort by date

• Age may be entered in the standard format: 27-03-16, that is, 27 years, three months and 16 days

• Enter spouse and parent names with the surname first, in case you should want to sort on that field. I first printed the "Type" field in the first column, but now feel it works better following the Eventdate and just preceding the Spouse category; your preference might be different.

I conclude with four or five sample lines bearing the limited information discussed several paragraphs above, as a way for helping you to decide whether this approach to keeping and combining records might or might not work for you.

SURNAME	FIRSTNM	EVENTDATE	TYPE	SPOUSE	ORIGIN	AGE	CAUSE
BINDER	FRANCIS	1884DEC12	D		06-04-06		CONSUMPTION
BINDER	HELEN RHEA	1887DEC06	B				
BINDER	MARGARET E.	1900MAR29	D		45-01-ILI		CONSUMPTION

TI-BASE

BINDER	MARY	1881APR16	D		60-00-03	CROUP
BININGER	CHARLES P.	1871JUL08	B			
BININGER	EDWARD LEO	1882OCT06	B			
BININGER	JOHN F.	1913JUN25	M	THIMMES,RACHEL FRANCES		
BINDER	GEO	1844OCT30	N	BADEN		

HARDWARE PROJECT

Delay the power and sync it right

BY WAYNE GARRISON

How many times have you powered up your TI99/4A and expansion box only to find that the system comes up in a locked condition? That's when everything appears to be running but is soon as you try to load something from disk, everything acts as if it were asleep. You may be answering "never," but for those of us who have all of our computer hardware (console, expansion box, printer, etc.) plugged into one of those powerstrips with one convenient switch to turn it all off or an it one time, we may have a different answer.

If one were to dig out that dusty old book which came with your expansion box and page through it you would come to the instructions on how to get started. It recommends that the expansion box be turned an first, then the console. If powerup isn't done in this manner, a lockup condition may occur. You have a 50/50 chance that something will go wrong if it is not done in that order.

This is due to the timing pulses which must synchronize between the console and the interface card in the expansion box before these two units can talk to each other.

This may sound like Greek to you, but imagine that you and a friend are talking to each other but instead of taking turns talking and listening, you are both talking at the same time and stopping to listen at the same time. What kind of conversation could you have? Certainly not a very good one. Well, this is basically the same with computers, except they talk an wires much like phone lines.

I get aggravated when my computer locks up for this reason but I like the convenience of my powerstrip. I thought, "What I need is a way to delay the A/C power to the computer console for just a few seconds until the expansion box his had enough time to power-up and prepare for synchronization."

It sounds easy enough but what's the best way to do it? Well, to answer

Continued from page 31
myself, there are a couple of ways of
doing it. I could design a circuit with
an R/C network to gate an SCR that
would hold the power off momen-
tarily. This would do it but I would
prefer something a little easier.

What I decided to do was to use a
relay. There is a device called a "time
delay relay." It does just as it's name
implies; it has a delayed response
when it is turned on. It is used to
switch electrical devices on or off.

I drew up a wiring diagram to
determine exactly what I would need
in the way of parts. After looking in
an electronics catalog, I realized there
was one slight obstacle; the cost of
such a relay. The prices ranged from
about $35 to about $100, depending
on the amount of delay and the
method of determining the delay
time. It wasn't long before I began to
wonder if it would be worth it.

Before giving up on my idea I paid
a visit to my favorite surplus elec-
tronics store to see if I could find a
bargain on a used time relay. I
rummaged through same old junk
and found just what I was looking
for. I also found a regular relay, which
I would also need for the project. The
guy at the sales counter let me have
these items for a mere fraction of
what they would have cost me new.

Now that I had the main ingredi-
ents for my new project, it was time
to start putting this thing together. I
went to the basement in search of
some sort of box or case to mount

this contraption in or on. I found a
couple of electrical junction boxes in
a bunch of junk I had left over from a
small electrical job I did.

Both relays had eight mounting
pins like the old vacuum tubes that
were once used to make our radios
and televisions work. I had some old
replacement tube sockets in my
electronics parts. I couldn't believe I
was having such good luck at finding
everything I needed.

Next I fastened the two "J" boxes
together, knocking out a hole
between them for wiring. My local
hardware store had a few other things
I needed, such as a "J" box cover with
the holes for mounting a wall outlet,
or as some refer to them, "wall plug."
I also had to get a cover which didn't
have any holes in it. I needed it to
mount the relay sockets. Using a drill
and a hole saw, I cut holes in the
blank cover to accommodate the relay
sockets.

If you are wondering what the
other relay is for, it is used to turn on
the time relay and the A/C power to
the expansion box. Since this relay
unit would probably be somewhat
unsightly and would be an eyesore on
my desk, it was necessary to have a
small plastic case with a switch to
operate my relay unit sitting on the
desk with a cable connecting it to the
relay unit sitting an the floor behind
my desk. I decided to use a couple of
push-button switches and some sort
of light or light emitting diode to
indicate when the power is on.

I separated the two outlets on the
receptacle so that one can be used to
plug in my powerstrip and the other
to provide the delayed A/C power for
the computer console. On most
receptacles, there is a small jumper
strip of metal that connects the wire
clamping screws together for both
outlets. These jumpers can be
removed with a small screwdriver or
wire cutters so that they will be
separate from each other. This is a
must to ensure power-up at different
times. The expansion box will plug
into the outlet
controlled by Rl.
The console will
plug into the
outlet controlled
by relay TDR.
(See the wiring
diagram.)

Wiring schematic for power delay unit

This little
device can be
used for other
things, too. I have
friends who are
high speed
computers which
power up and run
their diagnostics
and configuration
screens before the
monitor has had
enough time to
light up. This
would be perfect
for those purpos-
es. By the way, I
decided to
include a fuse
holder and fuse as
is standard
practice when
dealing with 120
volts A/C.
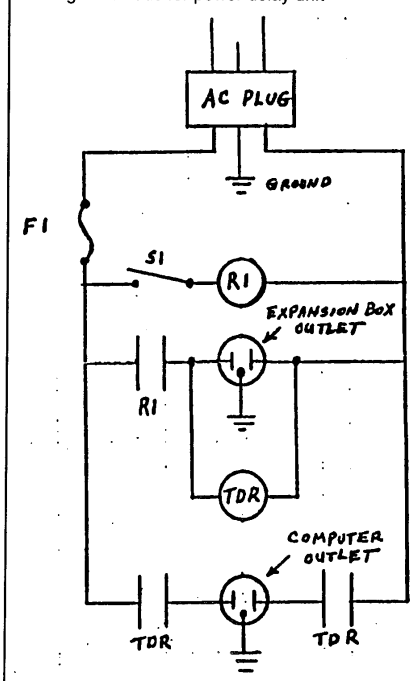
HARDWARE PROJECT

Continued from page 33

The way it works is simple. Plug this device in, then plug your expansion box into the appropriate outlet and do the same for your computer console. You may choose to plug a powerstrip into the outlet labeled computer outlet and have not only your computer, but also your printer and other equipment plugged into the strip. Therefore, all of these things would come an at the same time.

Make sure the switches on the expansion box and the computer console are in the "on" position. When you flip the switch (S1) on the delay unit, the relay R1 will energize which will close the set of contacts labeled R1. This will provide power to the expansion box outlet along with the TDR relay.

Now the time delay relay will start counting. When it times out to what you have it set for it will close the two sets of contacts marked TDR and will provide power to the computer outlet. As stated before, S1 should be mounted in a plastic box or case with a two-conductor cable connecting it to the relay unit which will sit on the floor behind your desk. The only difference between this proposed wiring and mine is that I chose to use two push-button switches wired to a set of holding contacts, and I included a small indicator light to mount on the switch box.

There are a number of different things you can do with this device.

For those who may have read one of my earlier articles addressing spike protectors and EMI filters, the components mentioned in that article can be integrated into this project.

As I stated before, this is not the only way to handle this small but sometimes annoying problem. The same results can be achieved with electronics possibly with a timing circuit and SCR (silicon controlled rectifier) or maybe a solid state relay. These relays are expensive and if I hadn't had such good luck finding one, then I would hive been forced to do it electronically.

If you were to open a time delay relay you would find a small electronic timing circuit. The way I did it was the easiest, just wire everything up. Once I had all the materials gathered up, it took me one evening to put it together.

This is a simple project if you like tinkering with electrical stuff and see a true need for this contraption. I did some checking around and was unable to find anything like it being sold commercially. It works great and I have found by using it that three to five seconds is sufficient time delay. There are a number of timing variances that one can come up with. Add a touch of automation to your system and save yourself a little aggravation and a couple of steps in powering up your TI. Delay the power and sync it right. I think you'll like it.

EXTENDED BASIC

Filtering captured file content into separate files

The programs included with this article were written by Ken Finto. They are similar and are used to convert telecommunications capture files into several files based on user-specified criteria, in this case category and topic number. The programs can be modified to accept other criteria. Examples used here come from the old GEnie BBS service but may have application to Usenet and other Internet information sources. The shaded portions are the text that actually are written to the separate text files. You'll notice that some of the extraneous matter is filtered out.

The programs are called GEXB-CVRT and GEABCVRT.

GEXBCVRT is written in Extended BASIC. This program extracts messages from a captured D/V80 file created with Fast Term, Telco, or other program. When this program runs, it creates new D/V80 files based on their respective Topic and Category (see samples). You can run file after file through the program and it will append rather than overwrite the data to each file it creates.

GEABCVRT is written to run in Advanced BASIC or Extended Basic.

Original capture file

Topic 6   Mon Jun 22, 1987
S.LISON   (Forwarded)
Sub: Mike Ballman 32K project
 TIMING PROBLEMS WITH THE 32K
MEMORY ON THE 16 BIT BUS.

1 new messages.
****

Category 4, Topic 6
Message 10   Sun Feb 26, 1989
STEVE.BL   at 09:24 MST

Concerning John Guion's article in the January MICROpendium about bypassing the 16 bit bus wait state .... <snip>

15 new messages.
****

Category 4, Topic 8
Message 24   Sun Feb 26, 1989
E.HALLETT [EDWARD]   at 19:23 MST

JJ, Thanks for the info on the Tandom TM 252.... <snip>

Category 4, Topic 8
Message 25   Sun Feb 26, 1989
JPLESLIE [Jonathan]   at 23:55 CST

Okay, so what settings should I use for my Seagate ST225 .... <snip>

Topic 6 capture file

Message 10   Sun Feb 26, 1989
STEVE.BL   at 09:24 MST

Concerning John Guion's article in the January MICROpendium about bypassing the 16 bit bus wait state ....

Topic 8 capture file

Message 24   Sun Feb 26, 1989

EXTENDED BASIC

Continued from page 35
E.HALLETT [EDWARD]   at 19:23 MST

JJ, Thanks for the info on the Tandom TM 252.... <snip>

———

Message 25   Sun Feb 26, 1989
JPLESLIE [Jonathan]   at 23:55 CST

Okay, so what settings should I use for my Seagate ST225.... <snip>

GEXBCVRT

```
100 OPEN #1:"DSK1.CAT4" !104
101 T1=1 :: T2=99 !063
102 !LINPUT #1:A$ :: IF A$="
Category 4,  Topic 13" THEN
120 ELSE 102 !058
110 LINPUT #1:A$ !187
120 IF SEG$(A$,1,8)="Categor
y" THEN GOSUB 1000 !069
130 IF EOF(1)THEN CLOSE #1 :
: END ELSE 110 !193
1000 GOSUB 3000 !019
1002 IF VAL(G$)<T1 THEN RETU
RN !178
1003 IF VAL(G$)>T2 THEN CLOS
E #1 :: END !208
1004 OPEN #2:"DSK1.CT"&F$&"_
"&G$,APPEND !194
1005 LINPUT #1:A$ :: IF SEG$
(A$,1,13)="<RETURN>, <S>" TH
EN 1005 !103
1006 PRINT #2:A$ :: PRINT A$
!049
1008 IF EOF(1)THEN CLOSE #1
:: CLOSE #2 :: END !035
1010 IF A$=" ————" T
HEN CLOSE #2 :: RETURN ELSE
```

```
1005 !067
3000 F$=SEG$(A$,10,POS(A$,",
",1)-10)!079
3002 F=POS(A$,"Topic",1)+6 :
: G$=SEG$(A$,F,LEN(A$)-F+1)!
005
3004 RETURN !136
```

GEABCVRT

```
89 ! ABASIC VERSION 9640 !15
9
90 CALL SCREEN(4)!149
100 OPEN #1:"DSK1.CAT4" !104
101 T1=1 :: T2=99 !063
102 !LINPUT #1:A$ :: IF A$="
Category 4,  Topic 13" THEN
120 ELSE 102 !058
110 LINPUT #1:A$ !187
120 IF SEG$(A$,1,8)="Categor
y" THEN GOSUB 1000 !069
130 IF EOF(1)THEN CLOSE #1 :
: END ELSE 110 !193
1000 GOSUB 3000 !019
1002 IF VAL(G$)<T1 THEN RETU
RN !178
1003 IF VAL(G$)>T2 THEN CLOS
E #1 :: END !208
1004 OPEN #2:"DSK1.CT"&F$&"_
"&G$,DISPLAY ,VARIABLE 80,AP
PEND !239
1005 LINPUT #1:A$ :: IF SEG$
(A$,1,13)="<RETURN>, <S>" TH
EN 1005 !103
1006 PRINT #2:A$ :: PRINT A$
!049
1008 IF EOF(1)THEN CLOSE #1
:: CLOSE #2 :: END !035
```

EXTENDED BASIC

```
1010 IF A$=" ————" T
HEN CLOSE #2 ELSE 1005 !057
1012 RETURN !136
3000 F=POS(A$,",",1):: F$=SE
G$(A$,10,F-10)!027
```

```
3002 F=POS(A$,"Topic",1)+6 :
: G$=SEG$(A$,F,LEN(A$)-F+1)!
005
3004 RETURN !136
```

RATINFEST

You've got to outsmart the rats to get the cheese

BY JEFF AUTOR

Imagine you are in charge of a deep African safari's food supply. Little do you know that giant rats nightly invade the area. If they should eat your food supply, the members of the safari and yourself would die. Knowing this, you attack the rats every night using only a swift karate kick. But for some reason, one night a mutant rat joins the pack. This special rat glows a bright red and can kill you in three bites; and has taught the others how to cast a spell of teleportation. The only way to escape this quick mutant is to hide in a hut near the food. Now night is falling, and you know the rats will come again....

THE GAME

After starting this Extended BASIC game, the computer will ask you if you want to use joysticks. A simple "Y" or "N" will do fine. Then the colorful title screen will appear. On this screen, the highest score (which is initially set at 1,000), and the score of the last game played is displayed.

To start the game, push the button on joystick No. 1 or any key on the console. The screen will turn black, and the rats will march in, with the mutant commanding from the rear.

You begin each night inside your hut (probably reading National Geographic) in the upper left corner of the screen. Once the rats begin to move, the food supply (represented by a chunk of cheese and a glass of wine) will appear on the screen. At this time, you may begin fighting off the rat forces.

To move your man around the playing field, use joystick No. 1 or the arrow keys, depending on your choice at the beginning. If you move off one edge of the screen, you will reappear at the opposite side.

To attack a rat, you must sneak up on it. If it sees you, the rat will use its teleportation spell; and you will be teleported somewhere on the screen. However, if you successfully sneak up on the rat, you can kill the rat by pressing the fire button or the "Q"

RATINFEST

Continued from page 37 key, depending on your choice at the beginning. When attacking a rat, you do not have to be facing the rat, but you must be very close. If you are close enough, you will kick the rat. It will then scream and disappear from sight. Think carefully about your attacks, since the mutant will hear the scream and will investigate.

As you move about the screen, the rats will slowly begin to devour your food supply. At the beginning of each night, this food supply is replenished with four sips of wine and four bites of cheese. If a rat gets a good chance, it will eat the cheese or drink the wine. When the supply is gone, the game is over.

The mutant rat will appear every night along with the normal rats. Unlike the others, this mutant is red, and is no threat to your food supply, since it does not eat. However, it poses a different threat to you. With its radioactive bites, it can kill you with only three hits. The mutant will chase after you if you come too close. The mutant rat is faster than you are and will eventually catch up. The best way to lose it is to run to your hut. If you are touching part of the hut, the mutant can't pounce on you.

If the mutant rat touches you, play will stop. On the screen your current score and number of bites you can still take without dying will be displayed. To continue playing, press the fire button or any key on the console.

When you are bitten for the final time, the computer will check your
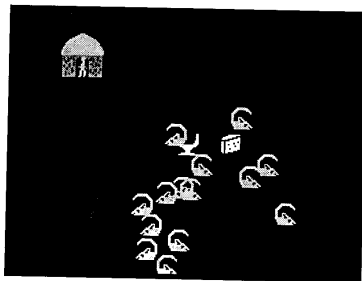
score against the highest score. (The highest recorded score is 79,800.) You will then hop into your grave and listen to "Taps" playing in the background. To return to the title screen and see your final score, press any key on the console.

SPECIAL FEATURES

When you start the game, each rat is worth 50 points. Each piece of the food supply (eight total) is worth 100 points. The points for the food supply are given immediately, while the points for the remaining food supply are given when the night is over. Each night that you survive, the points for kicking a rat goes up 50 points, and the amount of points for each food piece increases 100 points.

After every three nights of attack, you gain enough experience and resistance to withstand another bite

RATINFEST

from the mutant without dying.

If you manage to kill all the rats that attack, play will stop, and your score and the number of bites you can take without dying is displayed on the screen. At this time the bonus for the remaining pieces of food are given. When you are ready to continue, press any key on the keyboard.

RATINFEST

```
100 ! ************** !239
110 ! * RAT INFEST * !007
120 ! ************** !239
130 ! COPYRIGHT 1986 !084
140 !!131
150 ! BY JEFF AUTOR !036
160 !!131
170 ! VERSION 0.0.0 !213
180 ! TI EXTENDED BASIC !051
190 !!131
200 CALL CLEAR :: CALL SCREE
N(2):: CALL MAGNIFY(3):: CAL
L COLOR(3,15,1,4,15,1,5,15,1
,6,15,1,7,15,1):: DIM M(15,3
)!087
210 HS=1000 :: L(1)=5 :: L(2
)=7 :: L(3)=11 :: L(4)=14 !0
17
220 CALL CHAR(33,"0000000003
070F1F030F3FFFFFFFFFFC0F0FC
FFFFFFFFFF00000000C0E0F0F8")
!173
230 CALL CHAR(37,"1F3F7F7"&R
PT$("F",26)&"8FCFEFEFFFFFFFFF
")!219
240 CALL CHAR(40,"7F5E5A5B6B
6F7D5DFFFCB0A0E0A0A0A0FF3F0D
0507070407FEEE6E6A7ADED656")
!204
250 CALL CHAR(44,"5D77777F6F
6D6F7FE060602060E0E0E0070705
05050707077EFA9ADE767EDEFE")
!062
260 CALL CHAR(60,"0000000002
010001437F3C0000000001000018
3C3C78E4C0C0A080C0C0C0C08")!
220
270 CALL CHAR(88,"E0D0D8389C
DEC6F0E0C09030B0B898D8D8C8E8
E0C0000000000000001C4F616C")
!003
280 CALL CHAR(92,"0000000000
80E06000C0E0E0C",81,"0044442
81010101")!060
290 CALL CHAR(96,"0F1F306040
00000001030E1F337BDF3FF0F81C
0E070303E3A3637FFFFEFEFCFC")
!067
300 CALL CHAR(100,"0F1F3870E
0C0C0C7C5C6FEFF7F7F3F3FF0F81
C060200000080C070F8CCDEFBFC"
)!194
310 CALL CHAR(104,"0000051B3
76F6F686A696A6F6F6F6F6F0000F
0F8FCFEFEA2AAA2AEFEFEFEFEFE"
)!055
320 CALL CHAR(108,"030F3F7C0
16F6F6D6F6F6E6F6F2C0000FEF08
63EFEFADEFEEEFEFEF8C")!184
330 CALL CHAR(112,"C0C0C0C0C
0E0E0F07F3F0F0703031F7F03030
3030307070FFEFCF0E0C0C0F8FE"
)!090
340 CALL CHAR(116,"000303030
```

RATINFEST

Continued from page 39

```
10103030F0303060606060C0080C
08080808080C0A080C0C0C0C08")
!251
350 CALL CHAR(120,"000103010
10101010305010303030301000C0C
0C08080C0C0F0C0C060606060B")
!207
360 CALL CHAR(124,"3F3F3F3F3
F1F1F0F00",126,"FCFCFCFCFCF8
F8F00",128,"00003F3F3F1F1F0F
0",130,"0000FCFCFCF8F8F00")!
241
370 CALL CHAR(132,"000000003
F1F1F0F0",134,"00000000FCF8F
8F00",136,"00000000001F1F0F0
",138,"0000000000F8F8F00")!2
47
380 CALL CHAR(140,"0000183C3
C1E2703030C01030303030100000
00040800080C2FE3C000000008")
:: DISPLAY AT(11,3):"DO QOU
HAVE JOQSTICKS" !126
390 CALL KEY(2,K,S):: IF K=1
8 THEN J=1 ELSE IF K<>15 THE
N 390 !176
400 U=88 :: V=112 :: NS=16 :
: MS=6 :: NT=1 :: YS=1.25 ::
WI=124 :: RP=50 :: MN,BT=3
:: CB=4 :: RA=100 :: CALL CL
EAR !068
410 FOR X=1 TO 32 STEP 2 ::
CALL HCHAR(1,X,U,2):: CALL H
CHAR(24,X,V,2)!084
420 IF X<23 THEN CALL VCHAR(
X+1,1,V,2):: CALL VCHAR(X+1,
32,U,2)!180
430 U=U+8 :: V=V-8 :: IF U=1
20 THEN U=88 :: V=112 !066
440 NEXT X :: DISPLAY AT(6,9
):"WELCOME TO" :: DISPLAY AT
(9,9):"RAT INFEST" !006
450 RESTORE 470 :: FOR X=1 T
O 7 :: READ B,D,E :: CALL SP
RITE(#X,B,2,D,E):: NEXT X !2
13
460 CALL COLOR(#1,7,#2,6,#3,
15,#4,15,#5,15,#6,15,#7,12)!
047
470 DATA 100,130,50,116,140,
80,96,117,195,96,125,160,96,
83,205,96,144,190,108,130,10
0 !100
480 DISPLAY AT(12,4):"HIGH S
CORE:";HS :: DISPLAY AT(14,4
):"LAST SCORE:";LS :: D=1 ::
RESTORE 540 !103
490 FOR E=1 TO 23 :: READ A,
B,C :: CALL SOUND(-A*500,B,7
,B+1,7,C,7)!106
500 FOR X=8 TO 12 :: CALL CO
LOR(X,L(D),L(D)):: D=D+1 ::
IF D>4 THEN D=1 !071
510 NEXT X !238
520 CALL KEY(1,K,S):: CALL K
EY(0,K,R):: IF S OR R THEN 5
70 !018
530 NEXT E :: RESTORE 540 ::
GOTO 490 !148
540 DATA 2,233,117,2,220,250
00,2,175,117,1,175,25000,1,1
75,25000,1,196,156,1,220,156
,1,233,25000,1,196,25000 !13
8
550 DATA 1,175,117,1,147,117
,2,147,25000,1,131,131,1,131
,25000,1,196,131,1,196,25000
,1,175,175,1,175,25000 !040
560 DATA 1,262,175,1,294,250
00,2,231,117,2,231,156,3,231
,117,1,231,117 !104
570 CALL SOUND(-1,999,30)::
CALL CLEAR :: CALL DELSPRITE
(ALL)!109
580 DISPLAY AT(12,11):"NIGHT
";NT:"" :: IF NT>1 THEN 610
!023
590 DISPLAY AT(24,6):"CHECK
ALPHA LOCK" :: CALL COLOR(1,
2,1,2,2,1)!208
600 DISPLAY AT(3,4):"!"&CHR$
(34)&"#$":" %&&'":" ()*+
":" ,-./" !218
610 FOR X=1 TO 15 :: RANDOMI
ZE :: CALL PEEK(-31808,E,D)!
147
620 IF E>125 THEN M(X,3)=100
:: M(X,2)=2 :: GOTO 640 !19
7
630 M(X,3)=96 :: M(X,2)=-2 !
117
640 IF D>125 THEN M(X,1)=2 E
LSE M(X,1)=-2 !206
650 NEXT X :: DISPLAY AT(24,
6):"" !205
660 CALL SOUND(2000,262,7,33
0,7,392,7):: CALL COLOR(1,11
,1):: DISPLAY AT(12,6):" O
N QOUR MARK" !071
670 CALL SOUND(2000,262,7,31
1,7,440,7):: CALL COLOR(2,13
,1):: DISPLAY AT(12,9):" GE
T READQ" !064
680 CALL SOUND(2000,262,7,34
9,7,440,7):: CALL SPRITE(#1,
116,8,33,50):: DISPLAY AT(12
,11):" GET SET" !039
690 CALL SOUND(2000,262,7,33
0,7,523,7):: DISPLAY AT(12,1
2):" GO" :: FOR X=1 TO 300
:: NEXT X :: DISPLAY AT(12,1
1):"" !213
700 A=80 :: FOR X=3 TO 16 ::
CALL SPRITE(#X,100,15,A,1,0
,3):: CALL SOUND(200,-1,25)!
057
710 A=A+20 :: IF (X=7)+(X=12
)THEN A=80 !008
720 IF X=14 THEN A=A+20 !166
730 FOR B=1 TO 50 :: NEXT B
:: NEXT X !048
740 CALL SPRITE(#2,100,7,120
,1,0,3):: CALL SOUND(400,-2,
23)!021
750 CALL COINC(#3,80,175,10,
E):: IF E=0 THEN 770 !146
760 CALL SOUND(300,-2,18)::
FOR X=2 TO 16 :: CALL MOTION
(#X,0,0):: NEXT X :: GOTO 79
0 !076
770 IF Z>7 THEN CALL SOUND(2
00,-3,15):: Z=0 !242
780 Z=Z+1 :: GOTO 750 !238
790 FOR X=1 TO 15 :: A=M(X,1
):: B=M(X,2):: C=M(X,3):: E=
X+2 :: IF X=15 THEN E=2 !214
800 CALL MOTION(#E,A,B):: CA
LL PATTERN(#E,C):: CALL SOUN
D(100,-1,19):: NEXT X !142
810 CALL COLOR(11,16,1,8,12,
1,10,12,1):: DISPLAY AT(12,1
```

Continued from page 41

```
5):"pr  ln" :: DISPLAY AT(13
,15):"qs  mo" !226
820 CALL SPRITE(#17,124,9,89
,129):: V=116 :: H=M(15,1)::
 G=M(15,2):: CH=M(15,3)!075
830 CALL MOVE(O,YS,J):: IF O
=18 THEN 1200 !043
840 CALL COINC(#1,#2,70,B)::
 IF B THEN 990 !096
850 IF Z=1 THEN CALL MOTION(
#2,H,G):: CALL PATTERN(#2,CH
):: Z=2 !037
860 FOR X=3 TO NS :: IF WI=1
40 THEN 890 !170
870 CALL COINC(#X,#17,12,D):
: IF D THEN 1630 !120
880 IF CB=0 THEN 900 !192
890 CALL COINC(#X,89,161,12,
E):: IF E THEN 1660 !186
900 IF (X=6)+(X=10)+(X=14)TH
EN CALL MOVE(O,YS,J):: IF O=
18 THEN 1200 !195
910 NEXT X !238
920 IF RM<21-NS THEN RM=RM+1
 :: GOTO 830 !235
930 CALL POSITION(#MN,U,V)!2
48
940 IF U<93 THEN K=2 ELSE K=
-2 !201
950 IF V<144 THEN S=2 :: A=1
00 ELSE S=-2 :: A=96 !159
960 CALL PATTERN(#MN,A):: CA
LL MOTION(#MN,K,S)!079
970 RM=1 :: MN=MN+1 :: IF MN
>=NS THEN MN=3 !077
980 GOTO 830 !144
990 CALL POSITION(#1,K,S)!07
```

```
4
1000 IF (K>12)*(K<48)*(S>34)
*(S<66)THEN 830 !063
1010 CALL COINC(#1,#2,10,X):
: IF X THEN 1060 !205
1020 CALL POSITION(#2,D,E)!0
54
1030 IF K>D THEN C=MS ELSE C
=-MS !008
1040 IF S>E THEN D=MS :: B=1
00 ELSE D=-MS :: B=96 !172
1050 CALL PATTERN(#2,B):: CA
LL MOTION(#2,C,D):: Z=1 :: G
OTO 830 !159
1060 CALL SCREEN(16):: BT=BT
-1 :: IF BT=0 THEN 1120 !033
1070 CALL SCREEN(2):: CALL M
OTION(#1,0,0,#2,0,0)!021
1080 FOR X=6 TO 21 STEP 3 ::
 CALL SOUND(-50,-6,X):: NEXT
 X !239
1090 DISPLAY AT(7,11):"SCORE
 :";SC :: DISPLAY AT(9,11):"
BITES LEFT :";BT :: DISPLAY
AT(17,9):"PRESS ANQ KEQ" !08
0
1100 CALL KEY(1,K,S):: CALL
KEY(0,K,R):: IF (S=0)*(R=0)T
HEN 1100 !119
1110 DISPLAY AT(7,11):"":"":
"" :: DISPLAY AT(17,7):"" ::
 CALL LOCATE(#2,185,10,#1,33
,50):: CALL MOTION(#2,H,G)::
 GOTO 830 !214
1120 CALL SOUND(-1,25000,30)
:: FOR X=6 TO 21 STEP 3 :: C
ALL SOUND(-50,-6,X):: NEXT X
 :: CALL SCREEN(2):: CALL MO
```

```
TION(#1,0,0)!120
1130 CALL COLOR(#1,15):: CAL
L PATTERN(#1,104):: IF SC>HS
 THEN HS=SC !073
1140 RESTORE 1170 :: DISPLAY
 AT(8,9):"SCORE :";SC :: DIS
PLAY AT(16,11):"GAME OVER" :
: DISPLAY AT(22,9):"PRESS AN
Q KEQ" !212
1150 FOR X=1 TO 24 :: READ A
,B :: CALL SOUND(A*150,B,10,
B+1,10,B+2,10):: CALL KEY(1,
K,S):: CALL KEY(0,K,R):: IF
S OR R THEN 1190 !005
1160 NEXT X !238
1170 DATA 3,262,1,262,12,349
,3,262,1,349,12,440,3,262,1,
349,4,440,3,262,1,349,4,440,
3,262 !105
1180 DATA 1,349,12,440,3,349
,1,440,8,523,4,440,4,349,12,
262,3,262,1,262,18,349 !025
1190 CALL DELSPRITE(ALL):: C
ALL COLOR(8,2,1,9,2,1,10,2,1
,11,2,1):: LS=SC :: SC=0 ::
GOTO 400 !208
1200 CALL MOTION(#1,0,0):: O
=0 :: FOR X=3 TO NS !247
1210 CALL COINC(#1,#X,16,R):
: IF R THEN 1230 !206
1220 NEXT X :: O=0 :: GOTO 8
40 !146
1230 RANDOMIZE :: CALL PEEK(
-31880,K):: IF K>85 THEN 133
0 !155
1240 CALL POSITION(#X,U,V)::
 CALL POSITION(#1,T,E)!124
1250 IF E<V THEN CALL PATTER
```

```
N(#1,140)ELSE CALL PATTERN(#
1,60)!225
1260 CALL POSITION(#NS,A,B):
: CALL LOCATE(#X,A,B):: CALL
 COLOR(#NS,1):: CALL DELSPRI
TE(#NS):: NS=NS-1 !046
1270 CALL SOUND(400,-2,7)::
SC=SC+RP :: IF NS=2 THEN 140
0 !007
1280 CALL PATTERN(#1,116)::
IF Z=1 THEN 840 !224
1290 CALL POSITION(#2,A,B)!0
48
1300 IF T>A THEN H=2 ELSE H=
-2 !206
1310 IF E>B THEN G=2 :: CH=1
00 ELSE G=-2 :: CH=96 !233
1320 CALL MOTION(#2,H,G):: C
ALL PATTERN(#2,CH):: GOTO 84
0 !102
1330 X=3 :: U=110 !004
1340 CALL COLOR(#1,X):: CALL
 SOUND(-100,U,15):: X=X+1 ::
 U=U+50 !200
1350 IF X>16 THEN X=3 !143
1360 IF U<800 THEN 1340 !181
1370 RANDOMIZE :: CALL PEEK(
-31808,B,R):: IF B>180 THEN
B=180 !031
1380 IF R>220 THEN R=220 !02
0
1390 CALL LOCATE(#1,B+1,R+1)
:: CALL COLOR(#1,8):: GOTO 8
30 !140
1400 CALL POSITION(#2,A,B)::
 CALL PATTERN(#2,96,#1,116):
: IF B<16 THEN CALL MOTION(#
```

Continued from page 43

```
2,0,0):: CALL SOUND(300,-3,1
0):: GOTO 1420 !036
1410 CALL MOTION(#2,0,(16-B)
*.49):: CALL SOUND(300,-3,10
)!095
1420 CALL SOUND(1,-3,10):: C
ALL DELSPRITE(#1,#2)!084
1430 DISPLAY AT(6,9):"SCORE
:";SC !086
1440 DISPLAY AT(8,9):"BITES
LEFT :";BT !163
1450 FOR X=1 TO 300 :: NEXT
X :: IF CB=0 THEN 1520 !203
1460 CB=CB-1 :: ON CB+1 GOSU
B 1480,1490,1500,1510 !021
1470 CALL SOUND(380,-7,13)::
SC=SC+RA :: DISPLAY AT(6,16
):SC :: IF CB=0 THEN 1520 EL
SE 1460 !062
1480 CALL HCHAR(12,21,32,2):
: CALL HCHAR(13,21,32,2):: R
ETURN !065
1490 CALL HCHAR(12,21,91)::
CALL HCHAR(12,22,92):: CALL
HCHAR(13,22,93):: RETURN !16
7
1500 CALL HCHAR(12,22,89)::
CALL HCHAR(13,22,90):: RETUR
N !247
1510 CALL HCHAR(12,22,88)::
RETURN !066
1520 IF WI=140 THEN 1550 ELS
E WI=WI+4 !012
1530 SC=SC+RA :: DISPLAY AT(
6,16):SC :: FOR X=500 TO 150
STEP -50 :: CALL SOUND(-100
,X,13):: NEXT X !130
```

```
1540 IF WI=140 THEN CALL DEL
SPRITE(#17)ELSE CALL PATTERN
(#17,WI):: GOTO 1520 !104
1550 IF NT/3<>INT(NT/3)THEN
1590 !184
1560 FOR X=1 TO 300 :: NEXT
X !027
1570 DISPLAY AT(20,9):"BONUS
 BITE" :: CALL SOUND(600,131
,7,132,7,133,7):: CALL SOUND
(150,262,7,263,7,264,7)!112
1580 CALL SOUND(2000,523,7,5
24,7,525,7):: BT=BT+1 :: DIS
PLAY AT(8,21):BT !038
1590 DISPLAY AT(17,9):"PRESS
 ANQ KEQ" :: CALL KEY(0,K,S)
:: CALL KEY(1,K,X)!125
1595 DISPLAY AT(17,9):"" ::
IF (S=0)*(X=0)THEN 1590 !211
1600 CALL DELSPRITE(ALL):: D
ISPLAY AT(6,9):"" :: :"":"" ::
DISPLAY AT(16,9):"" :: NT=N
T+1 :: RA=RA+100 :: RP=RP+50
!234
1610 CALL COLOR(1,2,1,2,2,1)
:: IF NT<5 THEN YS=YS+.25 ::
MS=MS+1 !059
1620 CB=4 :: WI=124 :: NS=16
:: GOTO 580 !171
1630 WI=WI+4 :: FOR X=500 TO
150 STEP -50 :: CALL SOUND(
-100,X,13):: NEXT X !0811640
IF WI<140 THEN CALL PATTERN
(#17,WI):: GOTO 830 !126
1650 CALL DELSPRITE(#17):: I
F CB=0 THEN 1120 ELSE 830 !0
96
1660 CALL SOUND(-380,-7,13):
```

```
: CB=CB-1 :: IF CB>0 THEN 16
80 !055
1670 CALL HCHAR(12,21,32,2):
: CALL HCHAR(13,21,32,2):: I
F WI=140 THEN 1120 ELSE 830
!099
1680 ON CB GOTO 1690,1700,17
10 !101
1690 CALL HCHAR(12,21,91)::
CALL HCHAR(12,22,92):: CALL
HCHAR(13,22,93):: GOTO 830 !
175
1700 CALL HCHAR(12,22,89)::
CALL HCHAR(13,22,90):: GOTO
830 !255
1710 CALL HCHAR(12,22,88)::
GOTO 830 !074
1720 SUB MOVE(O,YS,J)!188
1730 CALL JOYST(1,A,B):: CAL
L KEY(1,O,S):: IF O=18 THEN
CALL SOUND(-200,-5,9):: SUBE
XIT !091
```

```
1740 IF J THEN 1810 ELSE IF
S=0 THEN 1840 !031
1750 A,B=0 :: ON O+1 GOTO 17
70,1840,1780,1790,1840,1800
!058
1760 GOTO 1840 !134
1770 B=-4 :: GOTO 1820 !179
1780 A=-4 :: GOTO 1820 !178
1790 A=4 :: GOTO 1820 !240
1800 B=4 :: GOTO 1820 !241
1810 IF (A=0)*(B=0)THEN 1840
!194
1820 IF A>0 THEN CALL PATTER
N(#1,120)ELSE IF A<0 THEN CA
LL PATTERN(#1,116)!223
1830 CALL MOTION(#1,-B*YS,A*
YS):: CALL SOUND(-3000,-4,5)
:: SUBEXIT !173
1840 CALL SOUND(-1,-4,30)::
CALL MOTION(#1,0,0):: SUBEND
!023
```

Menu program designed for text files

The following Extended BASIC program was written by J. Peter Hoddie and Barry Traver. It is a menu program that allows the user to view text files and to change screen and text colors. An option to select a lowercase character set also appears.

The user inputs a list of filenames in a DATA statement at line 110. These statements are read when the program is run and appear as a

selection list. When selected, the file appears onscreen. The user can scroll through the document using the "E" and "X" to move up and down and the "5" key to window across if it is larger than 32 columns. To stop reading a file, enter "Q." You then have the option of reading another file or exiting the program.

Line 270 is set up to run with a

MENU1

LOAD program. If you don't want to use this, REM out or delete the lines.

MENU1

```
10 CALL INIT :: CALL LOAD(81
96,63,248):: CALL LOAD(16376
,65,32,32,32,32,32,255,48)::
CALL LINK("A")!235
100 ! MENU1 COPYRIGHT (C) 19
88 by J. Peter Hoddie and Ba
rry Traver !183
105 DISPLAY ERASE ALL :: CAL
L DELSPRITE(ALL)!114
110 DATA INDEX,FILE1,FILE2,F
ILE3,FILE4,FILE5,FILE6,FILE7
!164
120 DIM A$(11):: FOR A=1 TO
8 :: READ A$(A):: NEXT A ::
DISPLAY AT(1,1)ERASE ALL:"ME
NU:":" A program by JPH and
BAT":"from the TRAVeLER disk
azine":" COPYRIGHT (C) 1987,
1988" !162
130 DISPLAY AT(6,1):"Colors:
":" 2 Black        10 Lt. Red"
:" 3 Med. Green 11 Dark Yell
ow":" 4 Lt. Green  12 Lt. Ye
llow" !158
140 DISPLAY AT(10,2):"5 Dark
 Blue  13 Dark Green":" 6 Lt
. Blue   14 Magenta":" 7 Dar
k Red    15 Gray":" 8 Cyan
    16 White" !193
150 DISPLAY AT(14,2):"9 Med.
 Red" :: DISPLAY AT(16,1):"S
uggested: 16/5, 2/8, 2/12,":
" or 16/2." :: DISPLAY AT(19
,1):"Foreground (2-16)?:  16
```

```
MENU:
A PROGRAM BY JPH AND BAT
FROM THE TRAVELER DISKAZINE
COPYRIGHT (C) 1987, 1988

Colors:
2 Black         10 Lt. Red
3 Med. Green     11 Dark Yellow
4 Lt. Green      12 Lt. Yellow
5 Dark Blue      13 Dark Green
6 Lt. Blue       14 Magenta
7 Dark Red       15 Gray
8 Cyan           16 White
9 Med. Red

Suggested: 16/5, 2/8, 2/12,
or 16/2.

Foreground (2-16)?:  16
Background (2-16)?:   5
Want large letters and true
lower case?  Y
```

```
" :: ACCEPT AT(19,22)SIZE(-2
):B !200
160 DISPLAY AT(21,1):"Backgr
ound (2-16)?:   5" :: ACCEPT
AT(21,22)SIZE(-2):C :: CALL
LINK("COLOR",B,C):: DISPLAY
AT(23,1):"Want large letter
s and true" !026
170 DISPLAY AT(24,2):"lower
case?  Y" :: ACCEPT AT(24,15
)SIZE(-1)VALIDATE("YN"):B$ :
: IF B$="Y" THEN CALL LINK("
NEWSET")ELSE CALL CHARSET ::
CALL LINK("CHRSET"):: CALL
LINK("COLOR",B,C)!053
180 DISPLAY AT(2,1)ERASE ALL
:"A. Read ";A$(1):"B. Read "
;A$(2):"C. Read ";A$(3):"D.
Read ";A$(4):"E. Read ";A$(5
):"F. Read ";A$(6)!186
190 DISPLAY AT(8,1):"G. Read
";A$(7):"H. Read ";A$(8)!03
7
195 DISPLAY AT(10,1):"I. Rea
d ";A$(9):"J. Read ";A$(10):
"K. READ ";A$(11)!138
210 CALL KEY(3,D,E):: CALL L
INK("KEYR",B$):: IF B$<"A" O
```

MENU1

```
R B$>"K" THEN 210 !192
220 F=ASC(B$)-64 :: C$="DSK1
."&A$(F)!045
270 CALL LINK("TXTCOL",B,C):
: CALL LINK("ARRAY",C$):: CA
LL LINK("COLOR",B,C):: DISPL
```

```
AY AT(12,1)ERASE ALL:"Anothe
r? (Y/N)   Y" :: ACCEPT AT(1
2,18)SIZE(-1)VALIDATE("YN"):
B$ :: IF B$="N" THEN RUN "DS
K1.LOAD" :: END !140
280 GOTO 180 !003
```

NEWSBYTES

TI-Chips to host 1999 MUG

TI-Chips will host the 1999 TI99/4A and Geneve 9640 M.U.G. conference (TIMUG'99). TIMUG'99 will be held Saturday, May 15, at the Spang Mansion on Kolthoff Road in Brook Park, Ohio (a suburb southwest of Cleveland). Kolthoff road is 3/4 mile south of Cleveland Hopkins International Airport, off of State Route 237, and directly south of the I-X center.)

Here is the interim conference schedule:

Time	Event
7-9 a.m.	set-up tables and displays
9 a.m.-5 p.m.	seminars and conferences
5-6 p.m.	clean up
6-7 p.m.	pizza party

TIMUG'99 will be set up much like the successful Lima User Group M.U.G. conferences. And like the conferences in Lima, the 1999 conference will be free. There will be no admission charge or set-up fee. However, TI-Chips will accept donations to help cover costs.

For more information, contact Glenn Bernasek at 13246 Harper Road, Strongsville, Ohio 44136; phone: (440)846-0865 (after 9 p.m. EST); e-mail: GBBasics@aol.com.

Mid-South users get new address

The Mid-South TI99 User Group has a new mailing address, according to their October newsletter.

New address is c/o Gary Cox, 1910 Wall St. #6, Memphis, TN 38134

Harrison and partner have 2 web pages

Bruce Harrison's web presence has, you could say, doubled.

"My own page keeps growing, most recently with more recipes, and now Lory has her own web site. Hers is oriented toward the musician, just touching on the stuff she did on the TI, both sound chip and MIDI," Harrison said.

Hers is at: http://members.aol.com/flatpickin/loryspage.html

Mine, with four new recipes, is still growing and changing almost daily at: http://members.aol.com/rottencat1/homepage.html

MDOS 6.0

MDOS 6.0 and CYA show improvements

BY GAMBIT

This article originally appeared in the newsletter of the Southern California Computer Group.—Ed.

It has been 11 years since the Geneve was first made, about 5 years since Myarc went out of business (although Cecure bought-out the remainder of the company), yet we still have updates to the Myarc Disk Operating System (MDOS), thanks to the efforts of Tim Tesch. Tim has been upgrading MDOS for some time now.

The most recent version of MDOS is 6.0. There are some very good changes to the new version of MDOS, which makes it absolutely necessary to have the "fast-memory" (32K of SRAM) upgraded to a minimum of 64K. The first 32K of memory is necessary to use MDOS, which has been the case since version 4.0, as it contains additional code support for SCSI devices, which loads into the added fast-memory area.

With 6.0, we now have an operating system that is compliant with the year 2000 and beyond, or at least, the next 60 years. I am still using MDOS 2.21, because it uses less system memory, the "bug" fixes since then have been minor, and I don't have a SCSI card.

However, in addition to the year 2000 fix, Tim has added an IBMGRAPH(ics) character set (128-255) to MDOS, which can be called by the command "IBMGRF on/off." Both of these changes were a result of conversations that I had with Tim through e-mail. I won't know the full-list of other changes that have been made, until I get a copy of the new version (*See the September/October 1998 MICROpendium—Ed.*)

In addition to a new version of MDOS, there's a new version of CYA. CYA stands for: "Configure-Your-Autoexec." Tim developed and wrote this program. When it was first released, it was sold only as a commercial program. Now it is fairware.

CYA allows Geneve users to customize MDOS, so that many of the commands that are reloaded every time you do a re-boot, will be permanently stored in M-DOS itself.

One of the new features of this new version is that you can permanently replace the default character set of MDOS. You simply load your favorite CHARA1 file (usually, the 0-127 set) into CYA, and it will embed the characters into MDOS.

MICROREVIEWS

TI Splash Screens and Theme

BY CHARLES GOOD

TI SPLASH SCREENS and THEME for PCs by Addatex software

As you are working on your Windows 95 or 98 computer do you miss your good old TI99/4A? Do you miss the good old TI title screen, those familiar beep and uh-oh warning sounds, and that good old cursor always flashing on the lower left of your monitor? Well I have just the cure for what ails you, and you can download it for free from the Internet or have me mail it to you.

Addatex Software has created two software packages they refer to as "funware" (translate as "freeware" or "free") which give the look and feel of a 99/4A to Windows 95 and 98.

The first of these is called Splash Screens. When installed, this software will replace all the normal Microsoft Windows "white cloud" pictures with a graphic of the 99/4A color bar title screen. When you turn on your PC you will briefly see the 99/4A title screen until your desktop appears. When your desktop appears, you will have the good old 99/4A title screen as wallpaper with all your normal Windows icons easily visible. When you shut down your computer ("Start," then "Shut Down") you will first see the 99/4A title screen with the words "Wait for shutdown" just above the lower colorbar. Then, after your computer closes all of its files, if your computer is not one of those that automatically shuts itself off you will see the 99/4A title screen with the words "Please shut down the computer."

The Splash Screens software creates a program window (click on "Start," then "Programs," then "TI99") and one of the options in this window is to turn on and off all these TI title screen graphics. If you turn them off you revert back to the Windows cloud pictures, but the TI graphics are still on your hard drive and you can turn them on again by clicking the appropriate icon in the TI99 program window.

Another thing you can do from the TI99 program window is open an MSDOS window and have it look just like a session of TI BASIC in your choice of either 40- or 80-column display. The effect is very realistic! You see * TI READY * and a flashing rectangular cursor below in the lower left side of the screen with black TI letters on a light blue background. The 40-column display looks just like a real 99/4A in TI BASIC command mode. The 80-column display resembles a Geneve running MDOS or a 99/4A with an 80-column display.

When you type, the font you see is identical to the 99/4A's with lower-case letters that look like small uppercase letters. From this simulat-

Continued from page 49
ed TI BASIC MSDOS window you can execute any DOS commands such as cd, dir, and copy. When you are finished just enter "exit" to return to Windows.

The second Addatex Software product adds 99/4A sounds and icons to Windows. This is a 99/4A theme and requires Microsoft Plus in order to run. Plus is an extra cost addition to Windows 95 or 98 that adds a bunch of cosmetic features, has a great pinball game, and for Windows 95 users includes a greatly enhanced disk compression utility.

Once you purchase and load Plus then you can load up your free Addatex 99/4A theme. The theme adds 14 different 99/4A sounds that sound off during various windows operations. There is the famous Parsec "alert," an "uh-oh" if something goes wrong, the TI "beep," and other familiar and not so familiar sounds. Most of these sounds are from TI game cartridges but not necessarily from well-known cartridges. I think one sound is from the Manacala cartridge which very few folks have.

The 99/4A theme also gives you lots of icons. For mouse pointers you get an animated (yes moving) TI title screen for the "working in background" symbol and an animated Parsec fighter ship for the "busy" indicator. You also get the Donkey Kong monkey for the recycle bin icon. He has a smile on his face and a fat stomach if there is something in

the recycle bin and is thin with a frown if there is nothing in the recycle bin. Network neighborhood also has a new graphic.

The "My computer" icon is a very cute green ball with a face that comes from the Jaw Breaker cartridge game. Within My Computer you get your choice of any of three types of hard drive icons. The default is Munchman, with jaws about to close on a graphic of a hard drive. You can also switch to Robopod drive icons, which are based on an Addatex game that is free for the downloading from the Addatex web site. You can also choose "Lettered" icons in which each hard drive is indicated with a letter that comes from the 99/4A character set. Using the TI99 program window or using Plus, the various icons and sounds can be switched in or out at will, alternating with the default windows equivalents or with others available from the Plus software.

In addition to sounds and icons, the TI99/4A theme changes the overall appearance of your desktop. Theme gives you a 99/4A wallpaper on your desktop. The default background color is light blue, just like TI BASIC command mode, and this background color carries over into many windows applications. The top command bar and the up/down left/right cursor arrow bars are usually green as in TI BASIC run mode. A lot of command bar text is either white on green or green on white and some of the text is rather blocky and resembles TI BASIC ASCII charac-

ters.

The effect of all this, the icons the sounds and the windows bar displays, is that no matter which windows applications you run you can almost always hear and see on screen aspects of the 99/4A. You will never forget your 99/4A computing roots when you run this addatex software on a modern windows capable computer.

Both Splash Screens and 99/4A Theme require Windows 95 or higher. They won't run on Windows 3.11. I haven't tested them on Windows NT, but they will probably work on that platform. Splash Screens does not require additional software, so all 99/4A enthusiasts should set this software up on your windows PC. Theme requires Plus, which can be purchased at most stores where Windows 98 and other Microsoft software is sold. I recommend Plus users download and install both Splash Screens and Theme. The only overlap in the two products is that they both give you a 99/4A title screen desktop wallpaper.

Both TI Splash Screens and Theme can be downloaded for free from the Addatex web site located at http://ourworld.compuserve.com/homepages/addatex/. Click on "download freebies" and then download the second and third freebie listed.

If you don't have access to the web then send me $1 and I will mail both the software packages to you on a single 3.5-inch IBM formatted floppy disk. Whether you download or get

the software from me by mail you will get zip files which have to be unzipped. A good Windows product that does this is Winzip, which is shareware. I will put a trial version of Winzip on the 3.5-inch disks I mail. Internet users can download Winzip from http://www.winzip.com.

To install the Addatex software properly: First install Winzip on your hard drive. Second, click one or both of the Addatex zip files which will start the process of Winzip unpacking the files to a subdirectory under c:\unpacked and will open a window to this subdirectory on your monitor.

Third, leaving the c:\unpacked\... subdirectory window open go to "My Computer" and click on the C: drive icon to open the root directory window of the C: drive.Fourth, find the "9t9" file folder in the open c:\unpacked\... window and drag this file folder to the root directory of the C: drive. If you have downloaded and unpacked both Splash Screens and Theme then there will be two 9t9 folders in different subdirectories of c:\unzipped. Drag them both to the C: root directory window and put one 9t9 folder on top of the other, having the second 9t9 directory overwrite the first as needed.These 9t9 folders with all the Addatex software will only work properly if placed in the root directory of drive C: and named 9t9. You are now ready to go! Click on "start" then "programs" then "ti99" and begin to enjoy

USER NOTES

CONV128 Update

The following comes from Bruce Harrison:

Some time ago, I wrote and distributed (public domain) a program called CONV128, which is designed to convert text files from D/F128 into D/V80. This works most of the time, but now and then we find a D/F128 file in which the last record, instead of being "padded out" with spaces to 128 characters, is padded with >1A characters.

I plan to fix this problem and get a revised edition of CONV128 out to Charlie Good and Lew King in the near future. In the meantime, such problem files can be fixed by using Birdwell's DSKU on the D/F128 file. Here's how:

Use File Utilities, Find String, and have the DSKU replace all >1A characters with >20. Don't forget then to rewrite the changed sector(s) back to disk.

If you need help, call me at (301) 277-3467 and I'll talk you through

the process. (Harrison can also be reached via e-mail at Rottencat1@aol.com.)

Vendor takes offers on used TI products

IEC, a Commerce City, Colo., business, has a web site dedicated to the TI. The company lists a variety of used TI products, including hardware, software, books, and newsletters. Prices aren't fixed. Buyers are asked to submit offers.

For more information, you can reach the site at http://www.iec.net/ti99/ti99.html, or e-mail the company at iecorp@aol.com.

Use LINEWRITER to format DATA statements

LINEWRITER, by Jim Peterson, makes it easy to enter DATA statements into BASIC and Extended BASIC programs. You just type words followed by commas as you would with a normal DATA statement. When you are finished, enter @@@ on a separate line and the statements will be saved in merge format as a file called LINEFILE. Line numbering for the data statements starts at 30000. New line numbers are automatically issued every four screen lines, though you won't see them until you load the LINEFILE file. A comma is automatically issued every time you hit the enter key. Which means if you are lazy you can enter a word followed by a return four times in a row and

MICROREVIEWS

your new modern Pentium-powered super fast PC-TI hybrid.Access:

Addatex Aoftware

On the web at http://ourworld.compuserve.com/homepages/addatex

Charles Good can be reached at P.O. Box 647; Venedocia, OH 45894; phone: 419-667-3131; email: good.6@osu.edu.

USER NOTES

create a DATA statement containing four items, each followed by a comma.

```
100 !LINEWRITER to aid in fo
rmatting screen text into 28
-column format and saving it
```

as DATA program lines in ME RGE format - by Jim Peterson !025

```
110 !strings containing comm
as and quotation marks will
```

MICROpendium Disks for Sale

❑ Series 1998-1999 (May/June 1998-Jan/Feb. 1999, 6 disks, mailed bimonthly) $25.00
❑ Series 1997-1998 (May/June 1997-Jan/Feb. 1998, 6 disks)...$25.00
❑ Series 1996-1997 (May/June 1996-Jan/Feb. 1997, 6 disks)...$25.00
❑ Series 1995-1996 (April 1995-Mar. 1996, 6 disks) $25.00
❑ Series 1994-1995 (April 1994-Mar 1994, 6 disks) $25.00
❑ Series 1993-1994 (April 1993-Mar 1994, 6 disks)................$25.00
❑ Series 1992-1993 (Apr 1992-Mar 1993, 6 disks) $25.00
❑ Series 1991-1992 (Apr 1991-Mar 1992, 6 disks) $25.00
❑ Series 1990-1991 (Apr 1990-Mar 1991, 6 disks) $25.00
❑ Series 1989-1990 (Apr 1989-Mar 1991, 6 disks) $25.00
❑ Series 1988-1989 (Apr 1988-Mar 1989, 6 disks) $25.00
❑ 110 Subprograms (Jerry Stern's collection of 110 XB subprograms, 1 disk) $6.00
❑ TI-Forth (2 disks, req. 32K, E/A, no docs) $6.00
❑ TI-Forth Docs (2 disks, D/V80 files) $6.00
❑ 1988 updates of TI-Writer, Multiplan & SBUG (2 disks) $6.00
❑ Disk of programs from any one issue of MICROpendium between April 1988 and present $5.00
❑ CHECKSUM and CHECK $4.00

Name_____

Address _____

City _____ State _____ ZIP _____

Texas residents add 8.25% sales tax.. .Check box for each item ordered and enter total amount here: $_____