## Subscription Fees

- ❑ 6 issues USA $35
- ❑ 6 issues Canada/Mexico $42.50
- ❑ 6 issues other countries surface mail
  - ___ Surface mail $40
  - ___ Air mail $52

Outside U.S., pay via postal or international money order; personal checks from non-U.S. banks will be returned. ADDRESS CHANGES: Subscribers who move may have the delivery of their most recent issue(s) delayed unless MICROpendium is notified six weeks in advance of address changes. Please your old address as it appears on your mailing label when making an address change.

Check each item ordered (or list on separate page and enter total amount here

$__________

No sales tax on magazine subscriptions. Texas residents add 8.25% sales tax on other items, including back issues and disk subscriptions.

Name____________________

Address____________________

City____________________

State_____ZIP______________

The set of numbers at the top of your mailing label indicates the cover date of the last issue of your subscription.

Micropendium
P.O. Box 1343
Round Rock, TX 78680

PERIODICALS

667-2241

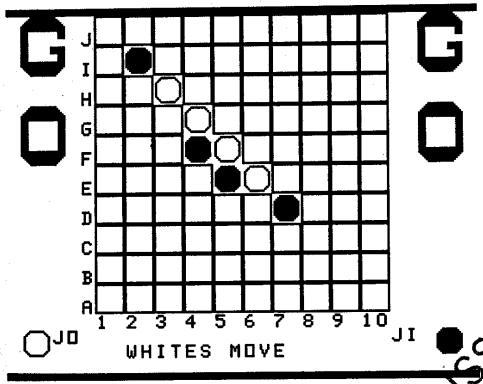
T
A 1/99
CHARLES GOOD
P.O. BOX 647
VENEDOCIA OH 45894

Covering the TI99/4A and Geneve home computers

# MICROpendium

Volume 16 Number 1 January/February 1999 $6

## Want to play?



## TIMUG'99 planned for May in Ohio

LAST ISSUE
RENEW NOW

## Inside

Seagate SCSI drives listed
A colorful experiment in assembly
CATCHAT in XBASIC

## Reviews

TI Bingo, Interest Calculation for MDOS, Disk Manager 2000, Cellgrow, BMP Converter, Backup Bitremover, Bacteria, Copy-C, and Hardware Tests

## CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published bimonthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Periodical postage paid at Round Rock, Texas. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

No information published in the pages of MICROpendium may be used without permission of the publisher, Burns-Koloen Communications Inc. Only computer user groups

While all efforts are directed at providing factual and true information in published articles, the publisher cannot accept responsibility for errors that appear in advertising or text appearing in MICROpendium. The inclusion of brand names in text does not constitute an endorsement of any product by the publisher. Statements published by MICROpendium which reflect erroneously on individuals, products or companies will be corrected upon contacting the publisher.

Unless the author specifies, letters will be treated as unconditionally assigned for publication, copyright purposes and use in any other publication or brochure and are subject to MICROpendium's unrestricted right to edit and comment.

All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680.

Foreign subscriptions are $42.50 (Canada and Mexico); $40 surface mail to other countries; $52 airmail to other countries.

All editions of MICROpendium are mailed from the Round Rock (Texas) Post Office.

Mailing address: P.O. Box 1343, Round Rock, TX 78680.

Telephone & FAX: (512) 255-1512

Internet E-mail: jkoloen@earthlink.net

Home page: http://www.earthlink.net/~jkoloen/

John Koloen Publisher
Laura Burns Editor

## COMMENTS

### We did it again!

It's been brought to my attention that a program we ran last month (RATINFEST) was published in MICROpendium in 1993. Sheesh! I think the copy editor put one over on me again.

This is not the first time this has happened, and, as I said the previous time, I hope it is the last. The problem is that we have a limited amount of time to put MICROpendium together, much of it printing the pages on a laser printer, manually collating them, stapling them, folding them, sealing them with tape, labeling them, sorting them for the post office, etc. You get the picture. It's a very time-consuming process and we often rush through the copy selection and preparation functions in order to allow enough time for the very labor-intensive production functions.

We'll keep our fingers crossed, that is until we start stapling. It's not a good

Continued on page 4

### More photos from the Chicago TI Faire

Lew King, Barry Harmsen, and Bud Mills are glued to the monitor.

Bruce Harrison demonstrated several programs at the Chicago TI Faire.

PHOTOS BY GARY COX

## We did it again!

It's been brought to my attention that a program we ran last month (RATINFEST) was published in MICROpendium in 1993. Sheesh! I think the copy editor put one over on me again.

This is not the first time this has happened, and, as I said the previous time, I hope it is the last. The problem is that we have a limited amount of time to put MICROpendium together, much of it printing the pages on a laser printer, manually collating them, stapling them, folding them, sealing them with tape, labeling them, sorting them for the post office, etc. You get the picture. It's a very time-consuming process and we often rush through the copy selection and preparation functions in order to allow enough time for the very labor-intensive production functions.

We'll keep our fingers crossed, that is until we start stapling. It's not a good

Continued on page 4

## More photos from the Chicago TI Faire

Lew King, Barry Harmsen, and Bud Mills are glued to the monitor.

PHOTOS BY GARY COX

## FEEDBACK

### Milwaukee group goes electronic

Our user group (Milwaukee) is, as is everyone else's, gradually getting smaller. We find that more than half of our membership dues goes towards printing and mailing out our newsletters. So we've decided to discontinue printing up a newsletter. We still will have a newsletter, of sorts, but since just about all our members now have e-mail accounts, it will simply be sent out in that manner. Right now we're looking for at least one member from all other user groups, that we can send it to. So if you're in a user group wishing to receive our monthly newsletter, send me your e-mail address and your user group name.

**Gene Hitz**
genehitz@juno.com
**Milwaukee Area User Group**

### The truth is out there

As is Gene Barret (Feedback, Nov./Dec. 1998), I am hanging in for similar reasons. I am really impressed that you have gone on for so long, and as long as you can keep publishing, I will likely keep subscribing. We (TI users) and you are probably making some kind of history here.

When other things were occupying most of my time I have packed away my old hardware on a number of occasions, only to dig it out again to try out something in one of your articles.

The way this little orphan survives is almost spooky. Maybe they should do an X-Files episode on an antique computer which has spawned a life of its own.

**Geoff Frusher**
**Halifax, Nova Scotia, Canada**

## COMMENTS

Continued from page 3

idea to staple with your fingers crossed. (We've taken extra effort to double-check the content of this issue and with the exception of a User Note or two feel confident that we have not recycled any articles or programs from previous editions.)

### MUG coming up May 15

The Multi-User Group conference scheduled for May 15 in Brookfield, Ohio, will be different from previous MUG's in at least two ways. One is that it isn't hosted by the Lima user group. Second is that it's not organized by Charlie Good. Glenn Bernasek of the host TI-CHIPs group says those who attended previous MUG conferences won't be disappointed. The MUG is a free and informal way for TIers to get together to talk about their TI and Geneve computers and find out what's going on in the TI community.

—JK

## THE ART OF ASSEMBLY 74

# A Colorful Experiment

BY BRUCE HARRISON

As we promised in our last column, today we take up a new subject, that of printing in color from TI-Artist picture files. The development of a program to perform that function was no easy task, partly because we have no color printer here at the Harrison residence on which to test results. Thus we express sincere thanks to Lew King of Industry, Pennsylvania, and to Gary Cox of Memphis, Tennessee, for their help with this project. Lew and Gary were the guinea pigs for this development, and thanks to them we have two versions of the TIACOLOR product, one for use with 24-pin dot matrix color printers, and one for use with Canon Bubble Jet color printers.

The idea started when Lew mentioned that his printer was a Star Micronics 24-pin color unit. He sent along the list of escape codes and their corresponding colors. That happened to match exactly the color control codes from a Panasonic printer manual which we have courtesy of Harley Ryan. Given this match, it occurred to us that this might be common to all Epson-compatible color models. Our first step, then, was to make up a short test program in Extended BASIC, then ship that off to both Lew and Gary. Results were similar, except that Gary's Bubble Jet model makes a better red than the impact model makes.

### How Do They Work?

In the impact case, the ribbon has four colors, these being magenta, cyan, yellow, and black. In the Bubble Jet case, there are four separate inks in one cartridge, with those same three subtractive primaries plus black. Taken with combinations, these four colors result in a total of seven colors available (including black). Red is made by combining magenta and yellow, green is made by combining cyan and yellow, and blue by combining cyan and magenta. These same four colors are used to print color glossy magazines and such. Results of our test with Extended BASIC showed that the red on the impact printer was more like orange, and its blue more like purple. On the Bubble Jet, the "combined" colors red and blue were just about what we'd expect. Apparently Canon uses very high quality inks.

### The Mapping Problem

On its screen, our faithful old TI produces 16 colors. Sixteen is more than seven, so we had to devise a "mapping" scheme so that these 16 colors would reduce gracefully to seven for the printer. Black was of course no problem, and white is made by printing nothing, but the TI has three shades of red and three of green, two of blue and yellow, and so on.

Thus was born a lookup table, whereby the colors on the TI are mated up

Continued on page 6

Continued from page 5

and combined into seven colors for the printer. This of course limits the accuracy of color rendition, so that things which are in different shades of red or blue will appear exactly alike on paper. Also, without getting into halftone rendering, we allowed gray to simply appear as white.

There are two tables used, one of which has the numbers of screen colors which map to each printed color, while the other has the screen color codes for comparison. Since we're going to have to do a lot of comparing and moving of bytes from both the pattern and color tables, our program first copies those tables from VDP RAM into the 32K memory. This is done by two VMBR operations before the actual printing starts.

## Multiple Passes

In order to print a bit-map image in color, we must examine each pixel first for its color, then to see whether that pixel is on or off. The colors for the printer start with black, so that's the first color in our lookup table. We start at the bottom left corner of the image, and get the color byte that corresponds to this byte. Since we're looking at the foreground color in this first pass, we take the color byte into a register and then shift it four bits to the right, so that the left byte of the register contains the foreground color for the byte.

Now we compare that to the first byte in our lookup table (01), and if it matches, then we'll examine the corresponding byte from the pattern table. In other words, if this pixel's foreground color is not a match, we ignore this byte entirely. If the color matches, then we examine the high bit of this byte, and if that's a 1, we save this byte in our temporary storage for printing.

We proceed in like manner until each pixel in the left-hand dot-column has been accounted for, then send out ESC "r" 0 to set the printer for black and print this one dot-column to the paper. Note that after printing this one column in black, we issue a carriage return without line feed to the printer, because we have six more colors to print in this same dot-column.

The next color in the map is magenta, which corresponds to ESC "r" 1 control sequence. As with black, the TI has only one shade of magenta, so this is a one pass process, working on that same dot-column from the screen. As before, each color byte is examined for a match with magenta, then those bytes that have the most significant bit turned on get saved in our temporary buffer. When we've again finished this dot-column, we set the printer for magenta and print anything whose foreground was magenta and left bit 1.

When we reach colors such as blue, there are two such matchings done, one for each shade that we'll print as blue. All of this gets repeated until we've done all seven printer colors for this one dot-column. When that's done, we're ready to repeat the whole process again, but this time examining the background color each time, and printing with the pattern byte inverted, so that we print something only if the background color matches and the left bit was a zero, which became 1 when inverted.

Like the foreground, this background printing cycles through each of the seven colors with just a carriage return sent to the printer on each pass. Finally, after the background colors have been printed for this first dot-column, we're ready to send a special linefeed and start processing the next dot-column of the picture.

To do this, we access the very same bytes from the stashed color and pattern tables, but we shift each byte left by one bit, so that we're processing the second dot-column. We perform the same processing, moving through the colors in both foreground and background. In like manner, we take these same bytes six more times, shifting by two through seven bits, until all eight bits of each byte have been printed. This completes one of the 32-character columns in the picture.

All of the above gets done 32 times, which completes one picture. Then and only then do we send a form feed and reset code to the printer, so it's back in its normal power-up condition, ready for whatever's next from the computer.

## Today's Sidebar Shows...

The sidebar is only a snippet of the source code, plus a portion of the data section. The file is called PRNMODE, and as you may guess, this is the part of the code that takes the picture data from memory and processes it for output to the printer. Although it's not immediately obvious, the bulk of this code is a set of nested loops. The nesting is intricate, to say the least, because various start and end points overlap each other depending whether we're at color, bit, or byte boundaries in the processing.

In this code we have used an instruction that we've never used before in this series, and we learned something about this instruction's use. The instruction is SLA R7,0 just before the label ANDI7 in the source code. This instruction means that the content of register 7 gets shifted left by the number in the low nybble of register 0. What we learned is that this doesn't work if R0 contains 0. Thus when we move the data word SFTQTY into R0, we skip over the SLA instruction if the data word was zero, which happens when we're processing the high order bit in any byte. By incrementing the number at label SFTQTY, we walk through the bits in each byte for the printing process. The data word at SFTQTY gets incremented after each dot-column just after label CKBIT. Thus it ranges from 0 through 7 as the counter BITCNT decrements from 8 to 1. When one character column has been completed, the code loops back to label PRNST2, at which the lookup table indexing restarts and SFTQTY goes back to zero.

The public domain disk containing this software has been released and made

Continued on page 8

Continued from page 7

available through the usual channel, the Lima Users Group library. On the disk are two versions of the program file, one for 24-pin impact color printers, and one for the Bubble Jet type. They are called COLOR24 and COLORBJ, respectively. The source code supplied on the disk is for the 24-pin version only, so that users won't get confused. If any of our readers would like to see the source for the Bubble Jet version, just request that disk from yours truly.

## Caveat Emptor

While we were pecking away at the source code for this product, we got a plea from another TI owner in Pennsylvania, concerning the use of our drawing program. This user had obtained a copy of that public domain disk, and was trying to use it with his new Hewlett-Packard Deskjet printer. He was getting "total nonsense" results, so he sent along a copy of the escape code pages from its manual. These sequences bore no resemblance at all to the sequences we're used to finding on the Epson-compatible printers that most TI owners have. Even the words used to describe the actions of these escape sequences were very different from the descriptions used in the usual printer manuals. What, for example, does "Raster Graphics" mean?

Thus we were unable even to understand what the sequences would accomplish. We advised this user to look carefully in the manual and on the printer to see whether the printer had a DIP Switch setting (as do many printers) to put it into an Epson-compatible mode. No such luck!

The escape sequences used by his Hewlett-Packard printer are in a whole new language called PCL, which HP considers the new standard for printers. It seems that HP decided that all users would have modern PC computers with Windows capability and the ability to install special drivers for this new language, and that thus there was no need for catering to the older crowd (like your author) who still depend on using Epson-type escape sequences to control printers.

Caveat emptor is Latin for "buyer beware," and that certainly applies in this case. If you plan to buy a new printer, we recommend you avoid Hewlett-Packard like the Black Plague. Our fear, though, is that other makers of printers will follow HP's lead in this matter, so we'd better keep our Epson-compatibles in good condition, else give up using a printer with our TI for anything but straight text applications.

When I purchased the Canon Bubble Jet model that I now use, I first checked the description carefully to be sure that Epson emulation was among its features. That's our best advice to anyone buying a new printer.

Our topic for next time is undecided. It will in all likelihood depend on what problems readers throw our way, so if there's something you'd like to see, just ask.

## Sidebar 74

```
* SIDEBAR 74
*
* PRINTER OUTPUT SECTION
* 24 PIN IMPACT COLOR VERSION
*
* CODE BY Bruce Harrison
* PUBLIC DOMAIN
* NOTE: THIS IS NOT A COMPLETE PROGRAM
*
PRNOUT LI   R1,PPABDT    PRINTER PAB DATA
       BL   @FILOP       OPEN THE FILE
       JNE  PRNSET       IF SUCCESS, JUMP
       BL   @CLOSE       ELSE CLOSE
       BL   @BLNK        BLANK SCREEN
       BL   @SETGM       BACK TO GRAPHICS
       BL   @CLS         CLEAR SCREEN
       BL   @UNBLNK      UNBLANK
       LI   R1,PNAMSG    PRINTER NOT AVAILABLE
       BL   @ERRRPT      REPORT ERROR
       B    @SAVE0       THEN BACK TO MAIN PROGRAM
PRNSET CLR  @BGFLG       CLEAR BACKGROUND FLAG
       LI   R0,PRNBUF    PRINT BUFFER
       LI   R1,RSTSTR    RESET STRING (ESC '@')
       BL   @DISSTR      PUT IN VDP BUFFER
       BL   @PRNSND      SEND TO PRINTER
PRNST1 LI   R5,32        32 COLUMNS OF PICTURE
       LI   R12,>1707    START AT LOWER RIGHT CORNER
*
* LABEL PRNST2 IS THE START OF ONE CHARACTER COLUMN
* EACH CHARACTER COLUMN HAS EIGHT DOT COLUMNS
*
PRNST2 LI   R9,MAPLUT    COLOR MAPPING
       LI   R13,QTYLUT   NUMBER OF MAPPED COLORS
       CLR  @SFTQTY      START WITH NO SHIFT
       MOV  @EIGHT,@BITCNT EIGHT BITS PER CHAR COLUMN
*
* LABEL PRNOTL IS THE START POINT FOR ONE DOT COLUMN
```

Continued on page 10

Continued from page 9

```
*
PRNOTL MOV  R12,R3        PUT ADDRESS IN R3
       MOV  R3,@SAV3      STASH AWAY R3
       MOV  @SEVEN,@CLRCNT SEVEN COLORS
       MOVB @PPABDT,@COLSTR+3 START WITH BLACK (27)"r"(0)
*
* LABEL NXTCLR IS THE START POINT FOR ONE PRINTED COLOR
* IN ONE DOT-COLUMN
* THE LOOP AT CLST1 CLEARS THE MEMORY BUFFER
*
NXTCLR MOV  *R13+,@MAPCNT NUMBER OF MAPPED COLORS FOR THIS PRINT
COLOR
       CLR  R14           R14=0
       LI   R0,TEMSTR     TEMPORARY STRING
       LI   R2,192        192 COUNT IN R2
CLTS1  MOVB R14,*R0+      ZERO A BYTE
       DEC  R2            DEC COUNT
       JNE  CLTS1         RPT IF NOT ZERO
*
* LABEL LTEM10 IS THE START POINT FOR ONE SCREEN COLOR
* R9 POINTS AT ONE SCREEN COLOR BYTE IN A LOOKUP TABLE
* R4 COUNTS DOWN THE 24 CHARACTER ROWS TO BE DONE
*
LTEM10 LI   R10,TEMSTR    TEMPORARY STRING
       LI   R4,24         24 CHAR ROWS
       MOV  @SAV3,R3      PUT R3 BACK
*
* LABEL PRNMIL IS THE START A GROUP OF EIGHT DOT-COLUMNS
*
PRNMIL LI   R6,8          8 BYTES
*
* LABEL PRNMIL IS THE START OF PROCESSING FOR ONE PIXEL
*
PRNINL MOVB @PIXBUF+>1800(R3),R7 COLOR BYTE TO R7
       MOV  @BGFLG,R0     BACKGROUND PASS?
       JNE  NOSRL         IF SO, JUMP
       SRL  R7,4          ELSE MOVE FOREGROUND NYBBLE BY 4 BITS
NOSRL  ANDI R7,>0F00      MASK TO ONE NYBBLE
       CB   R7,*R9        COLOR MATCH?
       JEQ  GPB           IF SO, GET BYTE FROM PATTERN
```

```
       JMP  PDEC3         THEN JUMP
GPB    MOVB @PIXBUF(R3),R7 GET A BYTE
*      JEQ  PDEC3         IF ZERO, JUMP
       MOV  @BGFLG,R0     TEST BACKGROUND FLAG
       JEQ  MOVR7         IF ZERO, FOREGROUND PASS
       INV  R7            ELSE INVERT THE 1'S AND 0'S
MOVR7  MOV  @SFTQTY,R0    SHIFT AMOUNT TO REG 0
       JEQ  ANDI7         IF ZERO, SKIP THE SHIFT
       SLA  R7,0          SHIFT LEFT BY THE NUMBER IN R0
ANDI7  ANDI R7,>8000      MASK TO MSB ONLY
       JEQ  PDEC3         IF ZERO, JUMP
       MOVB R7,*R10       ELSE PUT BYTE IN TABLE
       INC  R14           AND INCREMENT R14
PDEC3  INC  R10           ADD ONE TO TABLE INDEX
PDEC3A DEC  R3            BACK ONE BYTE
       DEC  R6            DONE 8?
       JNE  PRNINL        IF NOT, JUMP
       AI   R3,-248       NEXT CHAR IN COLUMN
       DEC  R4            DONE 24?
       JNE  PRNMIL        IF NOT, REPEAT
       INC  R9            NEXT MAPPED COLOR
       DEC  @MAPCNT       USED ALL MAPPED COLORS?
       JNE  LTEM10        IF NOT, REPEAT FOR NEXT MAPPED
*
* R14 INDICATES WHETHER ANY PIXELS WERE FOUND FOR THIS PRINT COLOR
* IF NONE WERE FOUND, THEN WE CAN SKIP SENDING ANYTHING TO PRINTER
*
       MOV  R14,R14       CHECK REG 14
       JEQ  ADDBYT        IF ZERO, NO NEED TO PRINT
       LI   R0,PRNBUF     POINT AT PRINT BUFFER
       LI   R1,COLSTR     COLOR ESCAPE STRING
       BL   @DISSTR       PLACE IN BUFFER
       BL   @PRNSND       SEND TO PRINTER
*
* CODE STARTING AT PLRTS SENDS BIT GRAPHICS DATA TO PRINTER
*
PLRTS  LI   R4,4          FOUR GROUPS
       LI   R0,PRNBUF     POINT AT BUFFER
       LI   R1,BGRSTR     BIT GRAPHICS CONTROL STRING
       BL   @DISSTR       PLACE IN BUFFER
```

Continued on page 12

Continued from page 11

```
       BL   @PRNSND      SEND TO PRINTER
       LI   R10,TEMSTR   POINT AT BYTES FOR 1 COLUMN
PRLBUF LI   R0,PRNBUF    PRINT BUFFER
       LI   R2,48        GROUP BY 48
PRMV1  MOVB *R10+,R1     GET A BYTE
       ANDI R1,>8000     MASK ONLY MSB
       SRA  R1,2         REPLICATE IN THREE MSBS
       BLWP @VSBW        WRITE THAT
       INC  R0           NEXT ADDR
       BLWP @VSBW        WRITE AGAIN
       INC  R0           NEXT ADDR
       BLWP @VSBW        WRITE AGAIN
       INC  R0           NEXT ADDR
       BLWP @VSBW        WRITE AGAIN
       INC  R0           NEXT ADDR
       BLWP @VSBW        WRITE 5TH TIME
       INC  R0           NEXT ADDR
       DEC  R2           DONE 48?
       JNE  PRMV1        IF NOT, REPEAT
       LI   R2,240       240 BYTES IN BUFFER
       BL   @PRNSND      SEND THOSE TO PRINTER
       DEC  R4           DONE 4 GROUPS?
       JNE  PRLBUF       IF NOT, ANOTHER GROUP
*
* CODE AT PRCRLF SENDS ONLY A CARRIAGE RETURN TO PRINTER
*
PRCRLF LI   R0,PRNBUF    PRINT BUFFER
       LI   R1,CROSTR    CR ONLY (NO LINE FEED)
       BL   @DISSTR      PLACE IN BUFFER
       BL   @PRNSND      SEND
*
* ADDBYT ADDS ONE TO PRINTER COLOR ESCAPE CODE
*
ADDBYT AB   @BYTONE,@COLSTR+3 SET NEXT PRINTING COLOR
       DEC  @CLRCNT      DEC COUNT OF COLORS PRINTED
       JNE  NXTCLR       IF NOT ZERO, NEXT LOOP
       LI   R9,MAPLUT    RESET MAP LOOKUP POINTER
       LI   R13,QTYLUT   AND MAPPED COLORS POINTER
       MOV  @BGFLG,R0    CHECK BACKGROUND FLAG
       JNE  CKBIT        IF NOT ZERO, JUMP
```

```
       INC  @BGFLG       ELSE SET BGFLG
       B    @PRNOTL      THEN START BACKGROUND PRINT
CKBIT  CLR  @BGFLG       CLEAR BGFLG FOR NEXT PASS
       INC  @SFTQTY      ADD ONE TO SHIFT QUANTITY
       LI   R0,PRNBUF    POINT AT PRINTER BUFFER
       LI   R1,CRLSTR    CARRIAGE RETURN AND SPECIAL LINE FEED
       BL   @DISSTR      SEND TO BUFFER
       BL   @PRNSND      THEN TO PRINTER
       DEC  @BITCNT      EIGHT DOT COLUMNS DONE?
       JEQ  PRDEC5       IF SO, JUMP AHEAD
       B    @PRNOTL      ELSE BACK FOR NEXT DOT COLUMN
PRDEC5 AI   R12,8        NEXT CHAR COLUMN
       DEC  R5           DONE 32?
       JEQ  PRFF         IF SO, AHEAD TO FORM FEED
       B    @PRNST2      IF NOT, CONTINUE
PRFF   LI   R0,PRNBUF    POINT AT PRINT BUFFER
       LI   R1,FFSTR     FORM FEED/RESET
       BL   @DISSTR      PUT IN BUFFER
       BL   @PRNSND      THEN TO PRINTER
PRNCLS MOV  @PABLOC,R0   PAB LOCATION
       MOVB @ONE,R1      CLOSE OPCODE
       BLWP @VSBW        WRITE THAT
       BL   @FILOP3      CLOSE THE FILE
       B    @SAVE0       BACK TO MAIN PROGRAM
*
* DATA - AN EXCERPT
*
QTYLUT DATA 1,1,1,2,2,3,3
MAPLUT BYTE 1,>0D,>07,>04,>05,>0A,>0B,>06,>08,>09,>02,>03,>0C
BGFLG  DATA 0
COLSTR BYTE 3,27,'r',0
FFSTR  BYTE 3,12,27,'@'
RSTSTR BYTE 2,27,'@'
LSPSTR BYTE 3,27,'+',5
BGRSTR BYTE 4,27,'L',192,3
CRLSTR BYTE 5,13,27,'+',15,10
CROSTR BYTE 1,13
SPLF   BYTE 5,13,27,'+',120,10
MARSTR BYTE 3,27,'@',13
```

# Program helps keep checkbook balanced

BY W. LEONARD TAFFS

*The following article originally appeared in the newsletter of the SouthWest 99ers.—Ed.*

This month's Extended BASIC program listing is again, as is often the case, a program I found I needed. It is a program to read D/V80 files of check numbers.

When checks are cashed, the time they are cleared by the bank can vary, depending upon when the payee cashes the check. In some cases people forget to cash checks for weeks. Nonprofit organizations with limited help often collect two to three months of checks before they catch up on depositing them. It can be a lot of work going over a whole year's bank statements trying to account for when checks were cleared by the bank. This program was written for any printer that has the reverse-paper roll feature.

It will "collate" your year's checks and columnize all checks cleared in each month. Checks do not have to be entered in numerical order. You may wish to sort your D/V80 file if numerical order is desired.

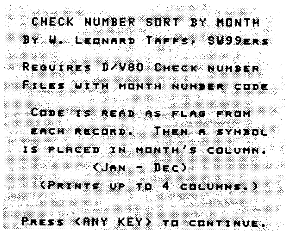
The following program requires the number of each month as a "flag," appended to each check number in the D/V80 file in this format:

```
1234 01
1235 01
1236 02
1240 04
1261 05
1270 07
1237 01
1242 04
```

The first digit of your check number should be in column one and the month code digits start in column six of your file. The program assumes your check numbers do not exceed four digits. Smaller numbers should be prefixed with zeroes (i.e. 0042, 0157).

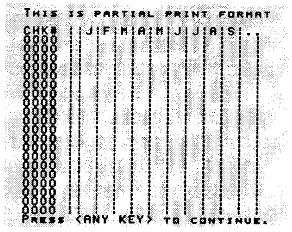
## PRNCOLCK#S

```
1 REM [PRNCOLCK#S][ 2-16-98
by W. Leonard Taffs, SW99ers
2 ! !225
100 GOTO 150 !229
110 A,A$,B$,C,C$,C1,C1$,CK$,
D,D1,D1$,D10,D10$,D11,D11$,D
12,D12$,D13,D13$,D2,D2$,D3,D
3$,D4,D4$,D5,D5$,D6,D6$,D7,D
7$,D8,D8$,D9,D9$,DSC$ !185
120 FN$,HD,HD$,HD1$,HD2$,INC
2,K,K$,L,LN$,LP,OK$,P,PCT,PR
,PR$,R,RC,RCT,S,T,X,YN$ !124
130 CALL CLEAR :: CALL KEY !
164
140 !@@P- !128
150 CALL CLEAR !209
160 DISPLAY AT(2,2):"CHECK N
UMBER SORT BY MONTH": :"By W
. Leonard Taffs, SW99ers": :
 :"Requires D/V80 Check numb
```



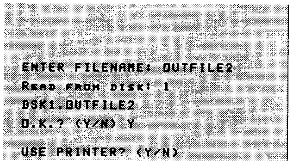
```
  CHECK NUMBER SORT BY MONTH
 By W. Leonard Taffs, SW99ers

 Requires D/V80 Check number
 Files with month number code

  Code is read as flag from
  each record.  Then a symbol
 is placed in month's column.
          (Jan - Dec)
    (Prints up to 4 columns.)


 Press <ANY KEY> to continue.
```

PRNCOLCK#S initial screen



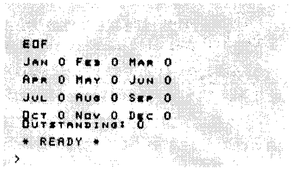
```
 This is partial print format
 CHK#  ||J|F|M|A|M|J|J|A|S|..
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 0000  || | | | | | | | | |
 Press <ANY KEY> to continue.
```

PRNCOLCK#S screen print



```
 ENTER FILENAME: OUTFILE2
 Read from disk: 1
 DSK1.OUTFILE2
 O.K.? (Y/N) Y

 USE PRINTER? (Y/N)
```

PRNCOLCK#S file selection screen



```
 EOF
 Jan 0 Feb 0 Mar 0
 Apr 0 May 0 Jun 0
 Jul 0 Aug 0 Sep 0
 Oct 0 Nov 0 Dec 0
 Outstanding: 0
 * READY *
>
```

PRNCOLCK#S final screen

```
er": :"Files with month numb
er code" !181
170 DISPLAY AT(12,1):" Code
is read as flag from ": :" e
ach record.  Then a symbol":
 :"is placed in month's colu
mn.": :"         (Jan - Dec)"
 !194
180 DISPLAY AT(20,3):"(Print
s up to 4 columns.)" :: DISP
LAY AT(24,1):"Press <ANY KEY
> to continue." !047
190 CALL KEY(0,K,S):: IF S<1
 THEN 190 :: CALL CLEAR !087
200 HD1$="CHK# ||J|F|M|A|M|J
|J|A|S|.." :: HD2$="0000 ||
| | | | | | | | | | " !242
210 FOR A=1 TO 22 !100
220 IF A=1 THEN DISPLAY AT(A
,1):"This is partial print f
ormat" !073
230 IF A>2 THEN DISPLAY AT(A
+1,1):HD2$ ELSE IF A=2 THEN
DISPLAY AT(A+1,1):HD1$ !092
240 NEXT A !215
250 DISPLAY AT(24,1):"Press
<ANY KEY> to continue." :: C
ALL KEY(0,K,S):: IF S<1 THEN
 250 :: CALL CLEAR !215
260 INPUT "ENTER FILENAME: "
:FN$ :: PRINT :: INPUT "Read
 from disk: ":DSC$ :: FN$="D
SK"&DSC$&"."&FN$ :: PRINT :F
N$: :!173
270 INPUT "O.K.? (Y/N) ":YN$
 :: PRINT !159
280 IF (YN$="Y")+(YN$="y")TH
```

Continued on page 16

## EXTENDED BASIC

Continued from page 15

```
EN 290 ELSE 260 !126
290 PRINT :: INPUT "USE PRIN
TER? (Y/N) ":PR$ :: PRINT ::
 IF (PR$="Y")+(PR$="y")THEN
PR=1 !015
300 IF PR THEN INPUT "Lines
per page (60/80) ":LP :: IF
LP=80 THEN L=1 !138
310 CALL CLEAR :: IF PR THEN
 OPEN #2:"PIO" :: T=9 :: IF
L THEN PRINT #2:CHR$(27);CHR
$(48);!095
320 IF PR THEN PRINT #2:CHR$
(27);CHR$(15);:: CLOSE #2 !1
38
330 HD$="CHK# ||J|F|M|A|M|J|
J|A|S|O|N|D||" :: HD=LEN(HD$
):: LN$=RPT$("=",32):: D7$="
Jul" :: D8$="Aug" :: D9$="Se
p" :: D10$="Oct" !126
340 C1$="      || | | | | | |
 | | | | | ||" :: C1=LEN(C1$
):: D1$="Jan" :: D2$="Feb" :
: D3$="Mar" :: D4$="Apr" ::
D5$="May" :: D6$="Jun" !205
350 D11$="Nov" :: D12$="Dec"
 :: D13$="Outstanding:" !015
360 IF PR THEN OPEN #2:"PIO"
,VARIABLE 132 :: PRINT #2:HD
$:LN$ :: PCT=PCT+2 !147
370 OPEN #1:FN$,INPUT !072
380 ON ERROR 930 :: LINPUT #
1:A$ :: RC=RC+1 :: RCT=RCT+1
 :: IF L=0 THEN 480 !064390
! LINPUT #1:A$ :: RC=RC+1 ::
 IF L=0 THEN 260 ! test line
 !021
400 IF POS(A$,"~",1)THEN 930
 !243
410 IF (ASC(A$)=128)+(A$="")
THEN RC=RC-1 :: GOTO 380 !12
5
420 IF RC>320 THEN X=0 :: RC
=1 !199
430 IF (RC>240)*(RC<321)THEN
 X=101 !175
440 IF (RC>160)*(RC<241)THEN
 X=68 !140
450 IF (RC>80)*(RC<161)THEN
X=34 !086
460 IF RC<=80 THEN X=0 !135
470 IF L THEN 530 !093
480 IF RC>240 THEN X=0 :: RC
=1 !200
490 IF (RC>180)*(RC<=240)THE
N X=101 !112
500 IF (RC>120)*(RC<=180)THE
N X=68 !072
510 IF (RC>60)*(RC<=120)THEN
 X=34 !013
520 IF RC<=60 THEN X=0 !133
530 A=LEN(A$):: B$=SEG$(A$,1
,4):: C$=SEG$(A$,6,2):: C=VA
L(C$):: D=C :: CK$=B$ !084
540 DISPLAY AT(13,1):B$;" ";
C;"RC";RC;"RCT";RCT;"X";X !1
99
550 CALL KEY(0,K,S):: IF S<>
1 THEN 580 !076
560 IF (K=81)+(K=113)THEN CL
OSE #1 :: PRINT :" USER TER
MINATED PROGRAM" :: GOTO 950
 !012
570 CALL KEY(0,K,S):: IF S<>
1 THEN 570 !066
```

## EXTENDED BASIC

```
580 IF PCT=82 THEN P=P+1 ::
DISPLAY AT(23,1):"ADVANCE PA
PER!";P !188
590 IF PR THEN IF PCT=82 THE
 PRINT #2: : : :! try 7 !213
600 IF PCT=82 THEN IF P=4 TH
EN 620 ELSE IF PR THEN OPEN
#3:"PIO",VARIABLE 132 :: PRI
NT #2:CHR$(27);CHR$(12);!249
610 IF PR THEN IF PCT=82 THE
 CLOSE #3 !173
620 IF P=1 THEN X=34 ELSE IF
P=2 THEN X=68 ELSE IF P=3 TH
EN X=101 ELSE IF P=4 THEN P,
X=0 !158
630 IF L THEN IF RC=320 THEN
 IF PR THEN PRINT #2: : :!24
0
640 IF PR THEN IF PCT=82 THE
N PRINT #2:TAB(X);HD$:TAB(X)
;LN$ :: PCT=0 :: INC2=0 :: P
CT=PCT+2 !104
650 IF PR THEN PCT=PCT+1 !02
9
660 ! PCT=PCT+1 :: PRINT #2:
TAB(X);RC;" ";RCT :: GOTO 71
5 ! TEST IF PCT=82 !181
670 ON C GOTO 680,690,700,71
0,720,730,740,750,760,770,78
0,790,800 !175
680 IF D=1 THEN D1=D1+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,3)&"*"&SEG$(C1$
,9,C1-T+1)! jan !024
690 IF D=2 THEN D2=D2+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,5)&"*"&SEG$(C1$
,11,C1-T)! feb !128
700 IF D=3 THEN D3=D3+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,7)&"*"&SEG$(C1$
,13,C1-T)! mar !154
710 IF D=4 THEN D4=D4+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,9)&"*"&SEG$(C1$
,15,C1-T)! apr !164
720 IF D=5 THEN D5=D5+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,11)&"*"&SEG$(C1
$,17,C1-T)! may !215
730 IF D=6 THEN D6=D6+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,13)&"*"&SEG$(C1
$,19,C1-T)! jun !228
740 IF D=7 THEN D7=D7+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,15)&"*"&SEG$(C1
$,21,C1-T)! jul !224
750 IF D=8 THEN D8=D8+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,17)&"*"&SEG$(C1
$,23,C1-T)! aug !217
760 IF D=9 THEN D9=D9+1 :: I
F PR THEN PRINT #2:TAB(X);CK
$&SEG$(C1$,5,19)&"*"&SEG$(C1
$,25,C1-T)! sep !235
770 IF D=10 THEN D10=D10+1 :
: IF PR THEN PRINT #2:TAB(X)
;CK$&SEG$(C1$,5,21)&"*"&SEG$
(C1$,27,C1-T)! oct !093
780 IF D=11 THEN D11=D11+1 :
: IF PR THEN PRINT #2:TAB(X)
;CK$&SEG$(C1$,5,23)&"*"&SEG$
(C1$,29,C1-T)! nov !113
790 IF D=12 THEN D12=D12+1 :
```

Continued on page 18

Continued from page 17

```
: IF PR THEN PRINT #2:TAB(X)
;CK$&SEG$(C1$,5,25)&"*"&SEG$
(C1$,31,C1-T)! dec !072
800 IF D=13 THEN D13=D13+1 :
: IF PR THEN PRINT #2:TAB(X)
;CK$&SEG$(C1$,5,25)&" "&SEG$
(C1$,31,C1-6)! cks not retur
ned in Dec statement !019
810 IF L THEN 810 !118
820 IF PCT=62 THEN DISPLAY A
T(23,1):"ADVANCE PAPER!" ::
P=P+1 !182
830 IF PR THEN IF PCT=62 THE
N PRINT #2: :!223
840 IF PCT=62 THEN IF P=4 TH
EN 860 ELSE IF PR THEN OPEN
#3:"PIO",VARIABLE 132 :: PRI
NT #3:CHR$(27);CHR$(12);!233
850 IF PR THEN IF PCT=62 THE
 CLOSE #3 !171
860 IF P=4 THEN P,X=0 !081
870 IF P=1 THEN X=34 ELSE IF
P=2 THEN X=68 ELSE IF P=3 TH
EN X=101 !204
880 IF L=0 THEN IF RC=240 TH
EN IF PR THEN PRINT #2: : :
:!093
890 IF PR THEN IF PCT=62 THE
N PRINT #2:TAB(X);HD$:TAB(X)
;LN$ :: PCT=0 :: INC2=0 :: P
CT=PCT+2 ! header for cols 2
-4 !093
900 C=0 :: IF L THEN IF RC=3
20 THEN RC=0 !047
910 IF L=0 THEN IF RC=240 TH
EN RC=0 !107
920 GOTO 380 !204
930 REM ** ON ERR/EOF ** !08
1
940 ON ERROR 950 :: CLOSE #1
 !219
950 PRINT :"EOF" :: IF PR TH
EN PRINT #2:TAB(102);"EOF" !
181
960 PRINT :D1$;D1;D2$;D2;D3$
;D3 !005
970 PRINT :D4$;D4;D5$;D5;D6$
;D6 !023
980 PRINT :D7$;D7;D8$;D8;D9$
;D9 !041
990 PRINT :D10$;D10;D11$;D11
;D12$;D12:D13$;D13 !002
```

The resulting printout of this program will show up to four columns, as shown in Fig. 1.

The asterisk indicates the monthly statement that the check cleared the bank.

Fig. 1

```
CHK#  ||J|F|M|A|M|J|J|A|S|O|N|D||J|F|M|....
1234  ||*| | | | | | | | | | | || | | |....
1254  || |*| | | | | | | | | | || | | |....
1290  || | | |*| | | | | | | | || | | |....
1370  || | | | | | | |*| | | | || | | |....
```

For those who can make use of it, it is worth having the printout provided. If the checks have been entered in numerical order (or the file has been sorted) you can quickly locate any missing check numbers. The number of checks is totaled at the end of the program (shown on screen only unless you add printer lines to print them out).

You have a choice of 60 lines or 80 lines per page, both options of which will print up to four columns per page. The printer codes used were those similar to the NX10.

Precautionary note: As mentioned previously, this program was designed for a printer with reversible paper-roll capability. See line 580 (for 80 lines per page) or line 820 (for 60 lines per page) if you don't have this reversible form feed feature on your printer. These are lines you can change the DISPLAY AT.. "ADVANCE PAPER!" to INPUTs instead of DISPLAY AT. It will require some editing of print commands besides this and if you are not certain of why (or how), don't hesitate to ask me for a copy of this program that allows you to manually roll your paper back up, which I am happy to do for anyone.

For those with programming interest and a moderate knowledge of programming in Extended BASIC, with a little imagination, this program can be modified to use with other files where you wish to columnize similar parameters.

The heart of the program is its use of the SEG$ function as in line 530, for the D variable in lines 680-800. These latter lines select the column to print the symbol in. When D=13 this will print a column with no symbol. This was reserved to print checks that have not cleared the bank.

The test "LINPUT" line can be unremarked to test your program for errors (see line 390). REMark line 380 to use this and unREMark line 390.

Another test line is 660. If you unREMark this line you will have to insert lines in the program to bypass all the printer command lines using the D variable. This will just print numbers in each column, a test that will save printer ink and wear.

Again, let me say that I am glad to help with any questions or programming changes you might wish to make. Just ask (no charge, of course).

If you do not use or have a TI-Writer-type program to create your D/V80 check files for the above program, here is an Extended BASIC program to make these files:

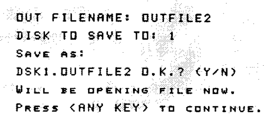
## CKFILE#&MN

```
1 REM [CKFILE#&MN] 2-19-98 !
139
10 CALL CLEAR :: DISPLAY AT(
3,1):"CHECK # AND MONTH FILE
 MAKER" !157
20 INPUT "OUT FILENAME: ":OF
$ :: PRINT :: INPUT "DISK TO
 SAVE TO: ":DSC$ :: PRINT !1
89
30 FN$="DSK"&DSC$&"."&OF$ ::
 PRINT "Save as:": :FN$,"O.K
```

Continued on page 20

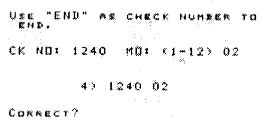
Continued from page 19

```
.? (Y/N) ": : :: INPUT "":NY
$ :: IF (NY$="N")+(NY$="n")T
HEN 20 ELSE OF=1 :: PRINT !0
34
40 DISPLAY AT(22,1):"Will be
 opening file now.": :"Press
 <ANY KEY> to continue." ::
CALL KEY(0,K,S):: IF S<1 THE
N 40 !235
50 IF OF THEN OPEN #1:FN$,OU
TPUT !118
100 CALL CLEAR :: DISPLAY AT
(10,5):"Use ""END"" as check
 number to end." !118
110 DISPLAY AT(15,1):"CK NO:
 " :: ACCEPT AT(15,8):CK$ ::
 CK=LEN(CK$):: IF CK$="END"
THEN 230 !010
120 IF CK>4 THEN 110 !182
```



```
OUT FILENAME: OUTFILE2
DISK TO SAVE TO: 1
Save as:
DSK1.OUTFILE2 O.K.? (Y/N)
Will be opening file now.
Press <ANY KEY> to continue.
```

Initial screen of CKFILE#&MN.



```
Use "END" as check number to
 end.

CK NO: 1240  MO: (1-12) 02

        4) 1240 02

Correct?
```

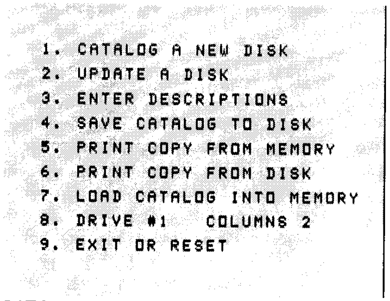
Data entry screen of CKFILE#&MN.

```
130 IF CK=1 THEN CK$="000"&C
K$ !174
140 IF CK=2 THEN CK$="00"&CK
$ !126
150 IF CK=3 THEN CK$="0"&CK$
 !078
160 DISPLAY AT(15,14):"MO: (
1-12) " :: ACCEPT AT(15,25):
MN$ :: MN=LEN(MN$):: IF MN>2
 THEN 160 !028
170 IF MN=1 THEN MN$="0"&MN$
 !115
180 A$=CK$&" "&MN$ :: CT=CT+
1 :: DISPLAY AT(19,9):STR$(C
T)&") ";A$ :: DISPLAY AT(22,
1):"Correct? " :: ACCEPT AT(
22,11):YN$ !024
190 IF (YN$="Y")+(YN$="y")TH
EN 200 ELSE CT=CT-1 :: GOTO
110 !060
200 DISPLAY AT(22,1):"PRINTI
NG TO FILE" !212
210 PRINT #1:A$ !173
220 GOTO 110 !189
230 REM ** END PROGRAM ** !1
77
240 PRINT #1:"~ EOF" !140
250 ON ERROR 260 :: CLOSE #1
 !039
260 PRINT :"YOU ENTERED";CT;
" CHECKS": :"IN FILE: ";FN$
!188
```

Note that this program takes care of prefixing the necessary number of zeroes if your check numbers consist of less than four digits and prefixes single digit months with the necessary zero if it is wished to sort these files by the month field.

# Create informative catalogs of your floppy disks

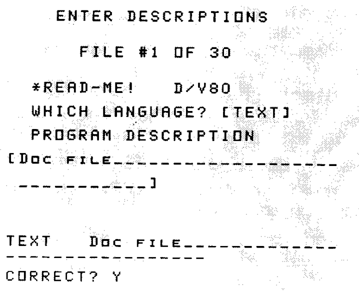
BY STEVE LISONBEE



```
1. CATALOG A NEW DISK
2. UPDATE A DISK
3. ENTER DESCRIPTIONS
4. SAVE CATALOG TO DISK
5. PRINT COPY FROM MEMORY
6. PRINT COPY FROM DISK
7. LOAD CATALOG INTO MEMORY
8. DRIVE #1    COLUMNS 2
9. EXIT OR RESET
```

CATCHAT main menu screen

CATCHAT is a program that I wrote for my own use that I could have a disk cataloger that gave a description of the programs. After awhile I decided to turn it into a finished program and release it to the TI world. It is a program that allows you to catalog a disk and enter a description and the language, or cartridge, needed for each program and print it out, the catalog may also be saved to disk. It is saved under the name CATFILE* and the program looks for this file after the first selection is made, although it is not required for the program to run. This is a simple program that does one disk at a time, it is NOT a library cataloger. If you use John Birdwell's Disk Utilities this program will be of little use as his program does the same thing, only better. What follows is a brief description of how the program works.



```
    ENTER DESCRIPTIONS

      FILE #1 OF 30

  *READ-ME!    D/V80
  WHICH LANGUAGE? [TEXT]
  PROGRAM DESCRIPTION
[Doc FILE___________________
 ___________]


TEXT   Doc FILE_____________
___________________
CORRECT? Y
```

CATCHAT data entry screen

1. CATALOG A NEW DISK — This will check the disk for a file called CATFILE* and set the program flags accordingly, it will then set all catalog variables to null and proceed to catalog the disk and set up space for language and descriptions.

Continued on page 22

Continued from page 21

2. UPDATE A DISK — This will load the file called CATFILE*, if present, and catalog the disk then compare the two. Newly added files will be merged in with the old and deleted files will be dropped. The old descriptions will be retained in the new CATFILE*.

3. ENTER DESCRIPTIONS — This allows you to enter a description of the program, or file, along with a space for the language or cartridge required to use it. You may skip a file by entering ";" as the first character of the language, a "/" will take you back to the menu.

4. SAVE TO DISK — This will save the catalog and descriptions in memory to disk under the name of CATFILE*.

5. PRINT FROM MEMORY — This will make a hard copy of the catalog and descriptions in memory.

6. PRINT FROM DISK — This will check the disk for CATFILE* then load it into memory and make a hard copy of it. The CATFILE* loaded will replace any other catalog in memory, so save before printing if you don't want to loose it.

7. LOAD CATALOG INTO MEMORY — This will load CATFILE* into memory to update or work with. It will also replace any catalog that is in memory.

8. SET DRIVE & COLUMNS — This allows you to choose which drive you wish to work from and choose if you want the catalog printed in 1 (40) or 2 (80) columns.

9. EXIT OR RESET — This is the only way to exit the program. After pressing 9 you will be given the choice of ending the program or rerunning the program from scratch.

Some other notes. Lines 150 and 160 will either ask for the date (150) or open a file called CLOCK (160) and enter the date by reading the Triple Tech card. REM whichever line you don't want to use. The printer default is "PIO" and is in line 500. If you wish to use characters other than ";" & "/" for skip and exit in the descriptions that is in lines 320 and 370. The printer codes are in line 120. UND=underline; SUND=stop underline; CDS=condensed print; POD=ignore paper out detector.

This program is public domain and free to all.

```
=================================================================

NAME=PC99        FILES=30    USED=571    AVAIL=867    DATE=12/28/98

=================================================================

FILENAME   SIZE TYPE   LANG   DESCRIPTION________________________||

*READ-ME!     5 D/V80  TEXT Doc file_____________________________||

BALL/BERT    43 ****** XB__ Game program_________________________||

BOINGY       22 ****** XB__ Game_________________________________||

BOWLING      24 ****** XB__ Game_________________________________||

CAT/DOC      16 D/V80  TEXT Doc file for CATCHAT ________________||

CATCHAT      22 ****** XB__ Disk catalog program_________________||

CATCHAT2     24 TYPE   ---- ---------------------------------------||
```

CATCHAT catalog printout

## CATCHAT

```
10 !************************
 !115
11 !*        CATCHAT         *
 !175
12 !*                        *
 !151
13 !*    A DESCRIPTIVE       *
 !154
14 !*  CATALOGING PROGRAM    *
 !104
15 !*                        *
 !151
16 !*    a public domain     *
 !079
17 !*        program         *
 !175
18 !*  by Steven Lisonbee    *
 !024
19 !*   SALT LAKE SLAVES     *
 !246
20 !*     January 1988       *
 !235
21 !************************
 !115
90 ON BREAK NEXT :: ON WARNI
NG NEXT :: CALL INIT :: CALL
 LOAD(-31806,16)!036
100 GOTO 110 :: CALL SCREEN
:: CALL KEY :: CALL HCHAR ::
 AC,AN,D$,DATE$,E1$,G,G$,G1$
,H,K,LL,LP,LPP,NB,NF,Q$,S !1
33
110 TYPE$(1)="D/F" :: TYPE$(
2)="D/V" :: TYPE$(3)="I/F" :
: TYPE$(4)="I/V" :: TYPE$(5)
="PROG" :: DSK$="DSK1." :: F
LG=0 :: CAT=2 :: CL=2 !068
120 CDS$=CHR$(27)&CHR$(66)&C
HR$(3):: UND$=CHR$(27)&CHR$(
45)&CHR$(1):: SUND$=CHR$(27)
&CHR$(45)&CHR$(0):: POD$=CHR
$(27)&CHR$(56)!097
130 DIM A$(70),AC$(70),AN$(7
0),B(70),BC(70),BN(70),CN(70
),C1(70),CN1(70),C1$(70),CC$
(70),CN$(70),C$(70),E$(70),E
C$(70),EN$(70)!239
140 DIM F$(70),FC$(70),FN$(7
0),C(70):: !@P- !239
150 DISPLAY AT(3,9)ERASE ALL
:"TODAYS DATE?" :: ACCEPT AT
(5,10)BEEP SIZE(10):DATE$ !0
43
160 ! OPEN #1:"CLOCK" :: INP
UT #1:G$,DATE$,G$ :: CLOSE #
1 !051
170 CALL SCREEN(8):: DISPLAY
 AT(4,2)ERASE ALL:"1. CATALO
G A NEW DISK": :TAB(2);"2. U
PDATE A DISK": :TAB(2);"3. E
NTER DESCRIPTIONS" !189
180 DISPLAY AT(10,2):"4. SAV
E CATALOG TO DISK": :TAB(2);
"5. PRINT COPY FROM MEMORY":
 :TAB(2);"6. PRINT COPY FROM
 DISK" !202
190 DISPLAY AT(16,2):"7. LOA
D CATALOG INTO MEMORY": :TAB
(2);"8. DRIVE #"&SEG$(DSK$,4
,1)&"   COLUMNS "&STR$(CL):
:TAB(2);"9. EXIT OR RESET" !
```

Continued on page 24

## CATCHAT

Continued from page 23

```
182
200 IF CAT=2 THEN DISPLAY AT
(23,3):"LOAD DISK AND SET DR
IVE" !057
210 CALL KEY(0,K,S):: IF (S<
1)+(K<49)+(K>57)THEN 210 ELS
E K=K-48 :: IF (K=1)+(K=8)+(
K=9)THEN 220 :: IF CAT=2 THE
N GOSUB 820 !058
220 ON K GOTO 230,600,310,43
0,500,460,460,760,770 !201
230 GOSUB 820 :: DISPLAY AT(
3,8)ERASE ALL:"READING DISK"
 :: OPEN #1:DSK$,INPUT ,RELA
TIVE,INTERNAL :: INPUT #1,RE
C 0:D$,G,G,H :: FLG=1 !123
240 FOR LP=1 TO 70 :: A$(LP)
="" :: NEXT LP :: DISPLAY AT
(6,4):"DISKNAME="&D$: :TAB(4
);"FILE #" !253
250 FOR LP=1 TO 70 !194
260 INPUT #1:A$(LP),C(LP),B(
LP),C1(LP):: IF C1(LP)=0 THE
N C1$(LP)="" ELSE C1$(LP)=ST
R$(C1(LP))!142
270 C$(LP)=TYPE$(ABS(C(LP)))
&C1$(LP):: E$(LP)="____" ::
F$(LP)=RPT$("_",38)!231
280 IF A$(LP)="" THEN 300 EL
SE DISPLAY AT(8,10)SIZE(2):S
TR$(LP)!104
290 NEXT LP !050
300 CLOSE #1 :: LPP=LP-1 ::
GOTO 170 !150
310 IF FLG=0 THEN 580 ELSE D
ISPLAY AT(1,5)ERASE ALL:"ENT
ER DESCRIPTIONS" !156
320 DISPLAY AT(24,2):""";""f
or next  ""/""for exit" !01
1
330 DISPLAY AT(9,3):"WHICH L
ANGUAGE? [    ]": :TAB(3);"P
ROGRAM DESCRIPTION": :"[": :
TAB(13);"]" !240
340 FOR LP=1 TO LPP :: DISPL
AY AT(4,7):"FILE #"&STR$(LP)
&" OF "&STR$(LPP):: DISPLAY
AT(7,3):USING "########## #
#####":A$(LP),C$(LP)!212
350 DISPLAY AT(9,20)SIZE(4):
E$(LP):: DISPLAY AT(13,2):SE
G$(F$(LP),1,27):: DISPLAY AT
(15,2)SIZE(11):SEG$(F$(LP),2
8,11)!207
360 DISPLAY AT(24,2):""";""f
or next  ""/""for exit" !01
1
370 ACCEPT AT(9,20)BEEP SIZE
(-4):E1$ :: IF SEG$(E1$,1,1)
="/" THEN 170 ELSE IF SEG$(E
1$,1,1)=";" THEN 420 ELSE E$
(LP)=E1$ !181
380 DISPLAY AT(24,2):"" !089
390 ACCEPT AT(13,2)BEEP SIZE
(-27):G$ :: ACCEPT AT(15,2)S
IZE(-11):G1$ :: F$(LP)=G$&G1
$ !162
400 DISPLAY AT(20,1):E$(LP)&
"   "&F$(LP): :"CORRECT? Y"
:: ACCEPT AT(23,10)BEEP SIZE
(-1)VALIDATE("ynYN"):Q$ !145
410 IF (Q$="N")+(Q$="n")THEN
 360 !179
```

## CATCHAT

```
420 CALL HCHAR(20,3,32,107):
: NEXT LP :: GOTO 170 !063
430 IF FLG=0 THEN 580 :: DIS
PLAY AT(3,7)ERASE ALL:"SAVIN
G TO DISK" :: OPEN #1:DSK$&"
CATFILE*",RELATIVE,INTERNAL,
OUTPUT !179
440 PRINT #1,REC 0:D$,G,H,LP
P !206
450 FOR LP=1 TO LPP :: PRINT
 #1,REC LP:A$(LP),B(LP),C$(L
P),E$(LP),F$(LP):: NEXT LP :
: CLOSE #1 :: CAT=1 :: GOTO
170 !168
460 IF CAT=0 THEN GOTO 860 !
113
470 DISPLAY AT(3,7)ERASE ALL
:"LOADING CATFILE" :: OPEN #
1:DSK$&"CATFILE*",RELATIVE,I
NTERNAL :: INPUT #1,REC 0:D$
,G,H,LPP !059
480 FOR LP=1 TO 70 :: A$(LP)
="" :: NEXT LP !235
490 FOR LP=1 TO LPP :: INPUT
 #1,REC LP:A$(LP),B(LP),C$(L
P),E$(LP),F$(LP):: NEXT LP :
: CLOSE #1 :: FLG=1 :: IF K<
>6 THEN 170 !022
500 IF FLG=0 THEN 580 :: DIS
PLAY AT(2,8)ERASE ALL:"PRINT
ING FILE" :: OPEN #2:"PIO",V
ARIABLE CL*68+3 !127
510 PRINT #2:CDS$&POD$;:: PR
INT #2:RPT$("=",68*CL)!021
520 PRINT #2,USING 890:D$,ST
R$(LPP),STR$(G-H),STR$(H),DA
TE$;:: PRINT #2:RPT$("=",68*
CL)&UND$;!136
530 FOR LP=1 TO CL :: PRINT
#2,USING 880:"FILENAME","SIZ
E","TYPE","LANG"," DESCRIPT
ION";:: NEXT LP :: PRINT #2:
SUND$ !140
540 FOR LL=0 TO LPP-1 STEP C
L !117
550 FOR LP=1 TO CL :: PRINT
#2,USING 880:A$(LL+LP),B(LL+
LP),C$(LL+LP),E$(LL+LP),F$(L
L+LP);!018
560 NEXT LP :: PRINT #2:"" !
196
570 NEXT LL :: CLOSE #2 :: G
OTO 170 !195
580 DISPLAY AT(3,2)ERASE ALL
:"DISK HAS NOT BEEN CATALOGE
D": :TAB(1);"NO FILES AVAILA
BLE IN MEMORY" !022
590 DISPLAY AT(23,2):"PRESS
ANY KEY TO CONTINUE" :: CALL
 KEY(0,K,S):: IF S<1 THEN 59
0 ELSE 170 !099
600 IF CAT=0 THEN GOTO 860 !
113
610 DISPLAY AT(1,8)ERASE ALL
:"READING CATFILE" :: OPEN #
1:DSK$&"CATFILE*",RELATIVE,I
NTERNAL :: INPUT #1,REC 0:D$
,G,H,LPP !054
620 FOR LP=1 TO LPP :: INPUT
 #1,REC LP:AC$(LP),BC(LP),CC
$(LP),EC$(LP),FC$(LP):: NEXT
 LP :: CLOSE #1 !096
630 DISPLAY AT(1,8):"READING
```

Continued on page 26

## CATCHAT

Continued from page 25

```
 DISK" :: OPEN #1:DSK$,INPUT
 ,RELATIVE,INTERNAL :: INPUT
 #1,REC 0:D$,G,G,H !130
640 FOR LP=1 TO 70 :: AN$(LP
)="" :: NEXT LP !057
650 FOR LP=1 TO 70 :: INPUT
#1:AN$(LP),CN(LP),BN(LP),CN1
(LP):: CN$(LP)="TYPE" :: EN$
(LP)="—" :: FN$(LP)=RPT$(
"-",38)!011
660 IF AN$(LP)="" THEN 680 !
232
670 NEXT LP !050
680 CLOSE #1 :: NF=LP-1 :: N
B=1 :: DISPLAY AT(1,11)ERASE
 ALL:"SORTING" !107
690 FOR AC=1 TO LPP+1 !032
700 FOR AN=NB TO NF !174
710 NB=AN+1 :: IF AC$(AC)<>A
N$(AN)THEN 730 !186
720 A$(AN)=AC$(AC):: B(AN)=B
C(AC):: C$(AN)=CC$(AC):: E$(
AN)=EC$(AC):: F$(AN)=FC$(AC)
:: GOTO 750 !049
730 A$(AN)=AN$(AN):: B(AN)=B
N(AN):: C$(AN)=CN$(AN):: E$(
AN)=EN$(AN):: F$(AN)=FN$(AN)
:: GOTO 740 !149
740 NEXT AN :: FLG=1 :: LPP=
NF :: GOTO 170 !115
750 NEXT AC :: GOTO 740 !209
760 ACCEPT AT(18,12)VALIDATE
(DIGIT)BEEP SIZE(-1):DSK$ ::
 DSK$="DSK"&DSK$&"." :: ACCE
PT AT(18,24)BEEP SIZE(-1)VAL
IDATE("12"):CL :: GOTO 210 !
156
770 CALL SCREEN(10):: DISPLA
Y AT(5,4)ERASE ALL:"(R)eset"
: :TAB(4);"(E)xit" !224
780 CALL KEY(0,K,S)!187
790 IF (K=82)+(K=114)THEN RU
N !031
800 IF (K=69)+(K=101)THEN ST
OP !015
810 GOTO 780 !094
820 OPEN #1:DSK$,INPUT ,RELA
TIVE,INTERNAL :: DISPLAY AT(
23,2):"CHECKING DISK FOR CAT
FILE" !126
830 FOR LP=1 TO 70 :: INPUT
#1:G$,S,S,S :: IF G$="CATFIL
E*" THEN CAT=1 :: GOTO 850 :
: ELSE IF G$="" THEN CAT=0 :
: GOTO 850 :: ELSE 840 !120
840 NEXT LP !050
850 CLOSE #1 :: DISPLAY AT(2
4,1):"" :: RETURN !123
860 DISPLAY AT(3,1)ERASE ALL
:"NO CATALOG FILE ON THIS DI
SK" !115
870 DISPLAY AT(23,2):"PRESS
ANY KEY TO CONTINUE" :: CALL
 KEY(0,K,S):: IF S<1 THEN 87
0 ELSE 170 !124
880 IMAGE ########## #### ##
#### #### #################
###################|| !161
890 IMAGE NAME=##########  F
ILES=###  USED=####  AVAIL=
####  DATE=########## !140
```

## TELECOMMUNICATIONS

# Hitching a ride on the Net with your TI

BY JOHN KOLOEN

You don't need a PC to access the Internet. Much of it is accessible with a TI running Telco, Term-80, or other terminal software. What it takes is an Internet Service Provider that supports shell accounts.

Most ISPs require use of a graphic user interface, such as Windows or Mac OS that run browsers such as Netscape Navigator and Microsoft Internet Explorer. Some relatively large service providers, including CompuServe and Delphi, offer shell accounts that allow you to dial in with a terminal program using Sprintnet or Tymnet. These accounts give you access to much of the Internet, including the web via Gopher servers, newsgroups, and e-mail.

With a shell account, you don't actually install any software on your TI. Once connected to the ISP you access "shell" programs such as Lynx and Pine. Lynx is used to access Gopher space, which is a variant of the web. Pine is used to send and receive e-mail. Also, you can download programs through FTP (File Transfer Protocol).

These shell programs are written in Unix and, for the most part, operate via menus. However, to take full advantage of them you'll need to learn about some of their Unix commands. Fortunately, most of them have on-line help screens.

You should know that the larger services charge a hefty price. For example, Delphi charges $13 per month for six hours of on-line service, $20 for 20 hours.

Smaller, local ISPs may also be a good source for users who need to access the Internet through software that uses a command line interface, including Telco and others. They are likely to have local, direct phone lines, which simplifies the connection process a bit. Additionally, a local provider may offer unlimited hours at a fixed monthly rate. I know that's the policy of a couple of small ISPs in Austin, Texas. Be sure to ask about the minimum modem speed the ISP supports. There's no point in getting an access phone number whose minimum speed is 56K.

Another place to look for a TI-accessible account is a local freenet. Freenets exist in many cities throughout the U.S. Some of them are actually free. There is no charge. Assesses assess a modest fee, such as $20 per year. The fees are used to cover costs. Freenets are often

Continued on page 28

## TELECOMMUNICATIONS

Continued from page 27

associated with public libraries and nonprofit agencies. Some of them offer a limited range of connection possibilities while others offer everything from SLIP/PPP to shell accounts. One way to find out if your community has a freenet is to check with your local library. You can also search the web, using "freenet" as your key word.

In some cases, the only way to use a particular freenet is to visit the library and use one of its public terminals. This gives you access to e-mail, the web, etc. but it doesn't achieve the goal of using your TI on the Net. Also, on-site access is usually time-limited. You can't just sit in front of the computer all afternoon and surf. Others may be waiting.

Obviously, the best way for a TIer to access a freenet is through its "BBS" or dial-up system. Be forewarned, not all freenets offer this capability. But it doesn't hurt to ask.

A typical example of a dial-up freenet system is a library that lets its users access its card catalog remotely. Unless the system places restrictions on its use, the card catalog is a portal to the web. From the catalog you can use shell programs such as Gopher to surf the web.

As you can see, there are ways for TIers to access the Internet using their trusty TIs and Geneves. It just takes a little doing.

## SCSI DRIVES

## Seagate SCSI drives listed

If you own a Western Horizon SCSI card you know that today's multi-gigabyte drives are a waste of money because the SCSI card will format only to 250mb, though it will control up to seven such drives. These smaller drives haven't been manufactured for years but they're still available at swap meets, flea markets, on the Internet, and from businesses that sell used computer equipment, such as Goodwill.

You can use the following specifications to determine suitability of the following Seagate SCSI drives when you come across them. According to WHT, any standard SCSI hard drive that supports 512 bytes per sector is compatible with the SCSI card. It doesn't matter whether the SCSI's were originally intendended for Macintosh or PCs. The following drives are listed here in detail:

| 3.5 Inch Drives | Half Height Drives | Full Height Drives |
|---|---|---|
| ST-138N | ST-225N | ST-4192N |
| ST-157N | ST-251N | |
| | ST-277N | |
| | ST-296N | |

## SCSI DRIVES

### ST - 138N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (26 sectors) | 32.3 MB |
| Actuator Type | Stepper |
| Tracks | 2,452 |
| Cylinders | 613 |
| Heads Data/Servo | 4/0 |
| Discs/Type | 2/Thin Film |
| Recording Method | RLL |
| Transfer Rate mbits/sec | 7.5 |
| Interface | SCSI/RLL |
| TPI (Tracks Per Inch) | 824 |
| BPI (Bits Per Inch) | 23,250 |
| Average Access – ms | 40/28* |
| Single Track Seek – ms | 8 |
| MTBF (hours) | 20,000 |
| Power / +12v Start-up (amps) | 2.0 |
| Power / +12v Typical (amps) | 0.4 |
| Power / +5V Typical (amps) | 1.2 |
| Landing Zone | Auto Park |

* ST138N-0/ST138-1

### ST -157N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (26 sectors) | 48.6 Mb |
| Actuator Type | Stepper |
| Tracks | 3,678 |
| Cylinders | 613 |
| Heads Data/Servo | 6/0 |
| Discs/Type | 3/Thin Film |
| Recording Method | RLL |
| Transfer Rate Mbits/Sec | 7.5 |
| Interface | SCSI/RLL |
| Tpi (Tracks Per Inch) | 824 |
| Bpi (Bits Per Inch) | 23,250 |
| Average Access – ms | 40/28* |
| Single Track Seek – ms | 8 |
| MTBF (hours) | 20,000 |
| Power / +12v Start-up (Amps) | 2.0 |
| Power / +12v Typical (Amps) | 0.4 |
| Power / +5v Typical (Amps) | 1.2 |
| Landing Zone | Auto Park |

* ST157N-0/ST157N-1

### ST-225N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (9,17,32 sectors) | 22.57,21.36,20.13 Mb |
| Actuator Type | Stepper |
| Tracks | 2,460 |
| Cylinders | 615 |
| Heads Data/Servo | 4/0 |
| Discs/Type | 2/Oxide |
| Recording Method | MFM |
| Transfer Rate Mbits/Sec | 5.0 |
| Interface | SCSI/MFM |
| Tpi (Tracks Per Inch) | 588 |
| Bpi (Bits Per Inch) | 9,827 |
| Average Access – ms | 65 |
| Single Track Seek – ms | 20 |
| MTBF (hours | 20,000 |
| Power / +12v Start-up (Amps) | 2.2 |
| Power / +12v Typical (Amps) | 0.9 |
| Power / +5v Typical (Amps) | 1.2 |
| Landing Zone | 670 |

### ST-251N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (13,26 sectors) | 42.7,43.1 Mb |
| Actuator Type | Stepper |
| Tracks | 3,272 |
| Cylinders | 820 |
| Heads Data/Servo | 4/0 |
| Discs/Type | 2/Thin Film |
| Recording Method | RLL |
| Transfer Rate Mbits/Sec | 7.5 |

Continued on page 30

Continued from page 29

| | |
|---|---|
| Interface | SCSI/RLLl |
| Tpi (Tracks Per Inch) | 777 |
| Bpi (Bits Per Inch) | 14,902 |
| Average Access – ms | 40/28* |
| Single Track Seek – ms | 8 |
| MTBF (hours) | 20,000 |
| Power / +12v Start-up (Amps) | 2.0 |
| Power / +12v Typical (Amps) | 0.5 |
| Power / +5v Typical (Amps) | 1.4 |
| Landing Zone | Auto Park |

* ST251N-0/ST251N-1

## ST-277N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (13,26 sectors) | 64.4,64.9 Mb |
| Actuator Type | Stepper |
| Tracks | 4,908 |
| Cylinders | 818 |
| Heads Data/Servo | 6/0 |
| Discs/Type | 3/Thin Film |
| Recording Method | RLL |
| Transfer Rate Mbits/Sec | 7.5 |
| Interface | SCSI/RLL |
| Tpi (Tracks Per Inch) | 777 |
| Bpi (Bits Per Inch) | 14,902 |
| Average Access – ms | 40/28* |
| Single Track Seek – ms | 8 |
| MTBF (hours) | 20,000 |
| Power / +12v Start-up (Amps) | 2.0 |
| Power / +12v Typical (Amps) | 0.5 |
| Power / +5v Typical (Amps) | 1.4 |
| Landing Zone | Auto Park |

* St277n-0/St277n-1

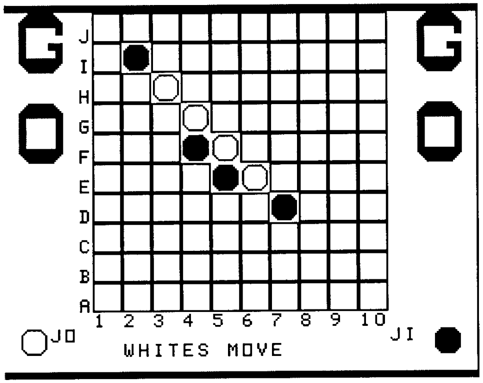
## ST-296N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (26 sectors) | 85.0 Mb |
| Actuator Type | Stepper |
| Tracks | 4,908 |
| Cylinders | 818 |
| Heads Data/Servo | 6/0 |
| Discs/Type | 3/Thin Film |
| Recording Method | SCSI |
| Transfer Rate Mbits/Sec | 10 |
| Interface | SCSI |
| Tpi (Tracks Per Inch) | 777 |
| Bpi (Bits Per Inch | 19,869 |
| Average Access – ms | 28 |
| Single Track Seek – ms | 8 |
| MTBF (hours) | 20,000 |
| Power / +12v Start-up (Amps) | 2.0 |
| Power / +12v Typical (Amps) | 0.5 |
| Power / +5v Typical (Amps) | 1.4 |
| Landing Zone | Auto Park |

## ST-4192N

| | |
|---|---|
| Unformatted | N/A |
| Formatted (36 Sectors) | 168.5 |
| Actuator Type | Voice Coil |
| Tracks | 9,176 |
| Cylinders | 1,147 |
| Heads Data/Servo | 8/1 |
| Discs/Type | 5/Thin Film |
| Recording Method | RLL |
| Transfer Rate Mbits/Sec | 10 |
| Interface | SCSI/RLL |
| Tpi (Tracks Per Inch) | 1,047 |
| Bpi (Bits Per Inch) | 20,078 |
| Average Access – ms | 17 |
| Single Track Seek – ms | 5 |
| MTBF (hours) | 15,000 |
| Power / +12v Start-up (Amps) | 4.0 |
| Power / +12v Typical (Amps) | 1.5 |
| Power / +5v Typical (Amps) | 1.5 |
| Landing Zone | Auto Park |

## Ancient game of Go continues to offer challenge



The game of Go dates back some 3,000 years, originating in China or the Himalayas. Today's rules are virtually identical to the rules followed by the ancients.

Mythology has it that the future of Tibet was once decided over a game of Go, when a Buddhist ruler refused to go into battle. Instead he challenged the aggressor to a game of Go to avoid bloodshed.

Go continues to be a popular game in Asia where the game is played by amateurs and professionals.

Go is a two-player board game in which the players use stones to contain their opponents. One player uses white stones and the opponent uses black stones. In traditional Go, the player with the black stones moves first.

Go has been described as being "like four chess games going on together on the same board." It requires analytical skills and intuition.

The game board consists of a grid of 19 vertical and 19 horizontal lines forming squares. The grid can be thought of as a piece of land shared between two players. In traditional Go the pieces are laid at the intersection of lines. When a stone is surrounded it is removed from the board as a prisoner.

Traditional Go is more complex than this TI version insofar as the object is to accumulate territory and capture prisoners. At the end of the game each player receives one point for each vacant intersection inside his territory and one point for each stone that has been captured. The highest

Continued on page 32

Continued from page 31
score wins.

There are many differences between traditional Go and this TI version. For one thing, white moves first. Also, the stones are placed in the spaces formed by the intersection of the lines, and stones cannot be captured.

The object of the game differs as well. In this version, the object is to create a row of five stones — horizontally, vertically, or diagonally. Of course, your opponent will do what he can to prevent this from happening while trying to do it himself. In this way it resembles Tic-Tac-Toe, which the traditional game doesn't.

This Extended BASIC version of Go was written by Steve Whitehead. While it differs from traditional Go, it offers an easy-to-understand challenge that can test the analytical skills of each player. To make it more challenging, a time limit is set for each move. It's time to Go.

## GO

```
10 LET TIME=0 !115
100 REM GO GRAPHICS !225
110 CALL SCREEN(16)!201
120 CALL CLEAR !209
130 CALL CHAR(100,"0F0F0F0F"
)!170
140 CALL CHAR(103,"070F1F3F7
FFFFFFF")!203
150 CALL CHAR(101,"E0F0F8FCF
EFFFFFF")!252
160 CALL CHAR(102,"FFFFFFFF
FFFFFFF")!060
170 CALL CHAR(104,"F0F0F0F0F
0F0F0F0")!142
180 CALL CHAR(105,"FFFFFF7F3
F1F0F07")!205
190 CALL CHAR(106,"FFFFFFFEF
CF8F0E0")!001
200 CALL CHAR(107,"FFFFFFFF0
F0F0F0F")!233
210 CALL CHAR(108,"0F0F0F0F0
F0F0F0F")!146
220 CALL COLOR(9,5,16)!237
230 CALL COLOR(10,5,16)!022
240 CALL HCHAR(11,14,100)!09
0
250 CALL HCHAR(10,14,101)!09
0
260 CALL HCHAR(10,13,102)!09
0
270 CALL HCHAR(10,12,103)!09
0
280 CALL VCHAR(11,12,104,2)!
024
290 CALL HCHAR(13,12,105)!09
5
300 CALL HCHAR(13,13,102)!09
3
310 CALL HCHAR(13,14,106)!09
8
320 CALL HCHAR(12,14,107)!09
8
330 CALL HCHAR(10,19,101)!09
5
340 CALL HCHAR(10,18,102)!09
5
350 CALL HCHAR(10,17,103)!09
5
360 CALL VCHAR(11,17,104,2)!
029
```

```
370 CALL HCHAR(13,17,105)!10
0
380 CALL HCHAR(13,18,102)!09
8
390 CALL HCHAR(13,19,106)!10
3
400 CALL VCHAR(11,19,108,2)!
035
410 A$="BY STEVE WHITEHEAD"
!241
420 FOR I=1 TO LEN(A$)!229
430 CODE=ASC(SEG$(A$,I,1))!1
17
440 CALL HCHAR(15,7+I,CODE)!
248
450 NEXT I !223
460 GOSUB 4000 !255
480 CALL CLEAR !209
490 INPUT "DO YOU WANT INSTR
UCTIONS Y/N ?":B$ !198
500 IF B$="Y" THEN 940 !241
510 CALL CLEAR !209
520 INPUT "PLAYER #1'S INITI
ALS ":C$ !226
525 IF LEN(C$)<4 THEN 530 !1
18
526 CALL CLEAR !209
527 PRINT "A MAX OF 3 LETTER
S ONLY" !110
528 GOTO 520 !089
530 INPUT "PLAYER #2'S INITI
ALS ":D$ !228
540 CALL CLEAR !209
542 IF LEN(D$)<4 THEN 548 !1
37
544 PRINT "A MAX OF 3 LETTER
S ONLY!" !144
546 GOTO 530 !099
548 CALL CLEAR !209
550 PRINT TAB(9);"START UP I
N" !151
560 PRINT TAB(11);"PROGRESS"
,,,,,,,,,,,,,,,,!012
570 FOR T=1 TO 1000 !214
580 NEXT T !234
590 CALL CLEAR !209
600 CALL SCREEN(5)!150
610 CALL COLOR(11,1,1)!220
620 CALL CHAR(114,"808080808
08080FF")!067
630 CALL CHAR(112,"FF8080808
0808080")!065
640 CALL CHAR(113,"FF0101010
1010101")!017
650 CALL CHAR(115,"010101010
10101FF")!019
660 FOR R=1 TO 19 STEP 2 !04
0
670 FOR I=7 TO 25 STEP 2 !03
4
680 CALL HCHAR(R,I,112)!157
690 NEXT I !223
700 FOR I=8 TO 26 STEP 2 !03
6
710 CALL HCHAR(R,I,113)!158
720 NEXT I !223
730 NEXT R !232
740 FOR R=2 TO 20 STEP 2 !03
3
750 FOR I=7 TO 25 STEP 2 !03
4
760 CALL HCHAR(R,I,114)!159
770 NEXT I !223
780 FOR I=8 TO 26 STEP 2 !03
6
```

Continued on page 34

Continued from page 33

```
790 CALL HCHAR(R,I,115)!160
800 NEXT I !223
810 NEXT R !232
820 CALL COLOR(11,2,16)!020
830 AL$="J I H G F E D C B A
" !032
840 FOR I=1 TO LEN(AL$)!049
850 CODE=ASC(SEG$(AL$,I,1))!
193
860 CALL VCHAR(1+I,6,CODE)!2
07
870 NEXT I !223
880 NU$="1 2 3 4 5 6 7 8 9 1
0" !190
890 FOR I=1 TO LEN(NU$)!071
900 CODE=ASC(SEG$(NU$,I,1))!
215
910 CALL HCHAR(21,6+I,CODE)!
244
920 NEXT I !223
930 GOTO 1190 !249
940 PRINT "G O IS A GAME INV
ENTED IN" !189
950 PRINT "CHINA MANY THOUSA
NDS OF " !225
960 PRINT "YEARS AGO IT IS P
LAYED WITH" !173
970 PRINT "TWO PLAYERS.ONE P
LAYER WITH" !017
980 PRINT "WHITE STONES AND
THE OTHER" !144
990 PRINT "WITH BLACK.EACH P
LAYER" !094
1000 PRINT "TAKES IT IN TURN
 TO PLACE A" !124
1010 PRINT "STONE ON THE BOA
RD,TRYING " !117
1020 PRINT "TO GET 5 OF THEI
R STONES IN" !122
1030 PRINT "A LINE,EITHER VE
RTICAL," !150
1040 PRINT "HORAZONTAL,OR DI
AGONALY.THE" !020
1050 PRINT "WHITE PLAYER STA
RTS FIRST," !192
1060 PRINT "PLACING A STONE
ANYWHERE ON" !198
1070 PRINT "THE BOARD.THE BL
ACK PLAYER" !095
1080 PRINT "MAY THEN BLOCK O
NE OF THE" !245
1090 PRINT "WHITES DIRECTION
S BY PLACING" !064
1100 PRINT "HIS STONE NEXT T
O THE WHITE" !207
1110 PRINT "STONE,OR START E
LSEWHERE." !114
1120 FOR I=1 TO 4000 !206
1130 NEXT I !223
1140 PRINT "PRESS<ENTER>TO S
TART GAME." !187
1150 CALL KEY(0,K,S)!187
1160 IF S=0 THEN 1150 !137
1170 IF K<>13 THEN 1150 !119
1180 GOTO 510 !078
1190 CALL COLOR(1,16,16)!024
1195 GOSUB 5000 !235
1200 CALL COLOR(5,2,16)!230
1210 CALL COLOR(6,2,16)!231
1220 CALL COLOR(7,2,16)!232
1230 CALL COLOR(8,2,16)!233
1240 CALL COLOR(3,2,16)!228
1250 CALL COLOR(4,2,16)!229
1270 CALL COLOR(2,2,16)!227
1280 FOR I=1 TO 3 !058
```

```
1290 CALL HCHAR(21+I,3,35,28
)!239
1300 NEXT I !223
1310 FOR I=1 TO LEN(C$)!231
1320 CODE=ASC(SEG$(C$,I,1))!
119
1330 CALL HCHAR(22,3+I,CODE)
!242
1340 NEXT I !223
1350 FOR I=1 TO LEN(D$)!232
1360 CODE=ASC(SEG$(D$,I,1))!
120
1370 CALL HCHAR(22,26+I,CODE
)!040
1380 NEXT I !223
1390 CALL CHAR(116,"000F1020
40404040")!010
1400 CALL CHAR(117,"40404040
20100F00")!011
1410 CALL CHAR(118,"00F00804
02020202")!013
1420 CALL CHAR(119,"02020202
0408F000")!014
1430 CALL CHAR(120,"000F1F3F
7F7F7F7F")!150
1440 CALL CHAR(121,"7F7F7F7F
3F1F0F00")!151
1450 CALL CHAR(122,"00F0F8FC
FEFEFEFE")!231
1460 CALL CHAR(123,"FEFEFEFE
FCF8F000")!232
1470 CALL COLOR(11,2,16)!020
1480 CALL COLOR(12,2,16)!021
1490 CALL HCHAR(22,2,116)!04
7
1500 CALL HCHAR(22,3,118)!05
0
1510 CALL HCHAR(23,2,117)!04
9
1520 CALL HCHAR(23,3,119)!05
2
1530 CALL HCHAR(22,30,120)!0
92
1540 CALL HCHAR(22,31,122)!0
95
1550 CALL HCHAR(23,30,121)!0
94
1560 CALL HCHAR(23,31,123)!0
97
1570 E$="ENTER CODE TO PLACE
 STONE,EG" !108
1580 F$="A1,PUTS A STONE,BOT
TOM LEFT#" !110
1600 J$="######CONGRATULATIO
NS######" !047
1605 I$="######PRESS <SPACE
BAR>####" !243
1610 FOR I=1 TO LEN(E$)!233
1620 CODE=ASC(SEG$(E$,I,1))!
121
1630 CALL HCHAR(24,I+2,CODE)
!243
1640 NEXT I !223
1650 FOR I=1 TO 500 !158
1660 NEXT I !223
1670 FOR I=1 TO LEN(F$)!234
1680 CODE=ASC(SEG$(F$,I,1))!
122
1690 CALL HCHAR(24,2+I,CODE)
!243
1700 NEXT I !223
1710 FOR I=1 TO 300 !156
1720 NEXT I !223
1740 GOTO 1790 !083
1763 FOR I=1 TO LEN(J$)!238
```

Continued on page 36

Continued from page 35

```
1764 CODE=ASC(SEG$(J$,I,1))!
126
1765 CALL HCHAR(24,I+2,CODE)
!243
1766 NEXT I !223
1768 CALL SOUND(4250,262,0,3
30,0,392,0)!067
1770 FOR I=1 TO 300 !156
1780 NEXT I !223
1782 FOR I=1 TO LEN(I$)!237
1783 CODE=ASC(SEG$(I$,I,1))!
125
1784 CALL HCHAR(24,I+2,CODE)
!243
1785 NEXT I !223
1786 GOTO 1940 !234
1790 G$="WHITES MOVE#####" !
218
1800 H$="BLACKS MOVE#####" !
183
1805 M$="ILLEGAL MOVE####" !
227
1810 CALL HCHAR(24,3,35,27)!
231
1820 LET BW=1 !222
1830 FOR I=1 TO LEN(G$)!235
1840 CODE=ASC(SEG$(G$,I,1))!
123
1850 CALL HCHAR(23,I+8,CODE)
!248
1860 NEXT I !223
1870 GOTO 1940 !234
1880 LET BW=2 !223
1890 FOR I=1 TO LEN(H$)!236
1900 CODE=ASC(SEG$(H$,I,1))!
124
1910 CALL HCHAR(23,I+8,CODE)
!248
1920 NEXT I !223
1922 GOTO 1940 !234
1924 FOR I=1 TO LEN(M$)!241
1926 CODE=ASC(SEG$(M$,I,1))!
129
1928 CALL HCHAR(23,I+8,CODE)
!248
1929 NEXT I !223
1930 FOR I=1 TO 200 !155
1932 NEXT I !223
1934 RETURN !136
1940 CALL KEY(0,K,S)!187
1950 IF S=0 THEN 1940 !162
1960 IF K=32 THEN 4495 !213
1970 IF K<>13 THEN 2190 !139
1980 IF BW=2 THEN 2042 !080
1982 CALL GCHAR(P,T,Z)!160
1984 IF Z<115 THEN 1990 !069
1986 GOSUB 1924 !219
1988 GOTO 1820 !114
1990 CALL HCHAR(P,T,116)!170
2000 CALL HCHAR(P+1,T,117)!1
02
2010 CALL HCHAR(P+1,T+1,119)
!035
2020 CALL HCHAR(P,T+1,118)!1
03
2030 CALL SOUND(100,-3,1)!21
8
2032 LET CH=116 !057
2034 GOSUB 6000 !215
2036 IF TIME=100 THEN 7000 !
185
2040 GOTO 1880 !174
2042 CALL GCHAR(P,T,Z)!160
2044 IF Z<115 THEN 2050 !130
2046 GOSUB 1924 !219
```

```
2048 GOTO 1880 !174
2050 CALL HCHAR(P,T,120)!165
2060 CALL HCHAR(P,T+1,122)!0
98
2070 CALL HCHAR(P+1,T+1,123)
!030
2080 CALL HCHAR(P+1,T,121)!0
97
2090 CALL SOUND(100,-1,1)!21
6
2092 LET CH=120 !052
2094 GOSUB 6000 !215
2096 IF TIME=100 THEN 7000 !
185
2100 GOTO 1820 !114
2110 CALL CLEAR !209
2115 LET TIME=0 !115
2120 INPUT " PLAY AGAIN Y/N
?":Q$ !191
2130 IF Q$="Y" THEN 510 !080
2140 CALL CLEAR !209
2150 PRINT TAB(14);"BYE": :
: : : : : : : : :!089
2160 FOR I=1 TO 100 !154
2170 NEXT I !223
2180 END !139
2190 IF K=65 THEN 2700 !209
2200 IF K=66 THEN 2730 !240
2210 IF K=67 THEN 2760 !015
2220 IF K=68 THEN 2790 !046
2230 IF K=69 THEN 2820 !078
2240 IF K=70 THEN 2850 !100
2250 IF K=71 THEN 2880 !131
2260 IF K=72 THEN 2910 !162
2270 IF K=73 THEN 2940 !193
2280 IF K=74 THEN 2970 !224
2290 IF K=49 THEN 2400 !166
2300 IF K=50 THEN 2430 !188
2310 IF K=51 THEN 2460 !219
2320 IF K=52 THEN 2490 !250
2330 IF K=53 THEN 2520 !025
2340 IF K=54 THEN 2550 !056
2350 IF K=55 THEN 2580 !088
2360 IF K=56 THEN 2610 !119
2370 IF K=57 THEN 2640 !150
2380 IF K=48 THEN 2670 !180
2390 GOTO 1940 !234
2400 LET T=7 !159
2410 CALL HCHAR(23,23,49)!05
6
2420 GOTO 1940 !234
2430 LET T=9 !161
2440 CALL HCHAR(23,23,50)!04
8
2450 GOTO 1940 !234
2460 LET T=11 !203
2470 CALL HCHAR(23,23,51)!04
9
2480 GOTO 1940 !234
2490 LET T=13 !205
2500 CALL HCHAR(23,23,52)!05
0
2510 GOTO 1940 !234
2520 LET T=15 !207
2530 CALL HCHAR(23,23,53)!05
1
2540 GOTO 1940 !234
2550 LET T=17 !209
2560 CALL HCHAR(23,23,54)!05
2
2570 GOTO 1940 !234
2580 LET T=19 !211
2590 CALL HCHAR(23,23,55)!05
3
2600 GOTO 1940 !234
```

Continued on page 38

Continued from page 37

```
2610 LET T=21 !204
2620 CALL HCHAR(23,23,56)!05
4
2630 GOTO 1940 !234
2640 LET T=23 !206
2650 CALL HCHAR(23,23,57)!05
5
2660 GOTO 1940 !234
2670 LET T=25 !208
2680 CALL HCHAR(23,24,48)!05
6
2690 GOTO 1940 !234
2700 LET P=19 !207
2710 CALL HCHAR(23,22,65)!05
3
2720 GOTO 1940 !234
2730 LET P=17 !205
2740 CALL HCHAR(23,22,66)!05
4
2750 GOTO 1940 !234
2760 LET P=15 !203
2770 CALL HCHAR(23,22,67)!05
5
2780 GOTO 1940 !234
2790 LET P=13 !201
2800 CALL HCHAR(23,22,68)!05
6
2810 GOTO 1940 !234
2820 LET P=11 !199
2830 CALL HCHAR(23,22,69)!05
7
2840 GOTO 1940 !234
2850 LET P=9 !157
2860 CALL HCHAR(23,22,70)!04
9
2870 GOTO 1940 !234
2880 LET P=7 !155
2890 CALL HCHAR(23,22,71)!05
0
2900 GOTO 1940 !234
2910 LET P=5 !153
2920 CALL HCHAR(23,22,72)!05
1
2930 GOTO 1940 !234
2940 LET P=3 !151
2950 CALL HCHAR(23,22,73)!05
2
2960 GOTO 1940 !234
2970 LET P=1 !149
2980 CALL HCHAR(23,22,74)!05
3
2990 GOTO 1940 !234
4000 FOR I=1 TO 2 !057
4010 CALL SOUND(500,262,1)!1
33
4020 CALL SOUND(500,262,1)!1
33
4030 CALL SOUND(800,294,1)!1
41
4040 NEXT I !223
4050 CALL SOUND(500,262,1)!1
33
4060 CALL SOUND(500,294,1)!1
38
4070 CALL SOUND(500,311,1)!1
28
4080 CALL SOUND(500,294,1)!1
38
4090 CALL SOUND(500,262,1)!1
33
4100 CALL SOUND(250,294,1)!1
40
4110 CALL SOUND(250,262,1)!1
35
4120 CALL SOUND(800,208,1)!1
```

```
36
4130 CALL SOUND(500,196,1)!1
39
4140 CALL SOUND(500,175,1)!1
36
4145 FOR I=1 TO 12 !107
4150 READ D,N !220
4160 CALL SOUND(D,N,1)!082
4170 DATA 500,196,500,208,80
0,196,250,196,250,175,800,13
9,500,262,500,262,800,294,80
0,262,800,262,1000,294 !070
4175 NEXT I !223
4180 RETURN !136
4495 CALL CLEAR !209
4500 L$="*******************
***" !124
4510 IF BW=1 THEN 4540 !027
4520 LET CO$=D$ !105
4530 GOTO 4550 !038
4540 LET CO$=C$ !104
4550 LET KON$="*CONGRATULATI
ONS "&CO$ !100
4560 FOR I=1 TO LEN(L$)!240
4570 CODE=ASC(SEG$(L$,I,1))!
128
4580 CALL HCHAR(11,4+I,CODE)
!241
4590 NEXT I !223
4600 CALL HCHAR(12,25,42)!04
9
4610 FOR I=1 TO LEN(L$)!240
4620 CODE=ASC(SEG$(L$,I,1))!
128
4630 CALL HCHAR(13,4+I,CODE)
!243
4640 NEXT I !223
4650 FOR I=1 TO LEN(KON$)!14
0
4660 CODE=ASC(SEG$(KON$,I,1)
)!028
4670 CALL HCHAR(12,4+I,CODE)
!242
4680 NEXT I !223
4690 FOR I=1 TO 16 !111
4700 FOR T=1 TO 4 !070
4710 CALL SCREEN(I)!225
4720 NEXT T !234
4730 NEXT I !223
4740 GOTO 2110 !149
5000 FOR I=2 TO 29 STEP 27 !
089
5010 CALL HCHAR(2,I+2,100)!2
55
5020 CALL HCHAR(1,I+2,101)!2
55
5030 CALL HCHAR(1,I+1,102)!2
55
5040 CALL HCHAR(1,I,103)!069
5050 CALL VCHAR(2,I,104,2)!0
03
5060 CALL HCHAR(4,I,105)!074
5070 CALL HCHAR(4,I+1,102)!0
02
5080 CALL HCHAR(4,I+2,106)!0
07
5090 CALL HCHAR(3,I+2,107)!0
07
5100 CALL HCHAR(7,I+2,101)!0
05
5110 CALL HCHAR(7,I+1,102)!0
05
5120 CALL HCHAR(7,I,103)!075
5130 CALL VCHAR(8,I,104,2)!0
09
```

Continued on page 40

Continued from page 39

```
5140 CALL HCHAR(10,I,105)!12
0
5150 CALL HCHAR(10,I+1,102)!
048
5160 CALL HCHAR(10,I+2,106)!
053
5170 CALL VCHAR(8,I+2,108,2)
!201
5180 NEXT I !223
5190 RETURN !136
6000 REM HORAZONTAL R !046
6010 LET R=P !237
6020 LET C=T !226
6030 LET COUNT=1 !206
6040 CALL GCHAR(R,C+2,X)!075
6050 IF X<>CH THEN 6100 !074
6060 LET COUNT=COUNT+1 !024
6070 IF COUNT=5 THEN 1763 !0
43
6080 LET C=C+2 !141
6090 GOTO 6040 !254
6100 REM HORAZONTAL L !040
6110 LET C=T !226
6120 CALL GCHAR(R,C-2,X)!076
6130 IF X<>CH THEN 6180 !155
6140 LET COUNT=COUNT+1 !024
6150 IF COUNT=5 THEN 1763 !0
43
6160 LET C=C-2 !142
6170 GOTO 6120 !078
6180 REM VERTICAL U !137
6190 LET R=P !237
6200 LET C=T !226
6210 LET COUNT=1 !206
6215 IF R-2<1 THEN 6280 !101
6220 CALL GCHAR(R-2,C,X)!076
6230 IF X<>CH THEN 6280 !255
6240 LET COUNT=COUNT+1 !024
6250 IF COUNT=5 THEN 1763 !0
43
6260 LET R=R-2 !172
6270 GOTO 6215 !174
6280 REM VERTICAL D !120
6290 LET R=P !237
6300 LET C=T !226
6310 CALL GCHAR(R+2,C,X)!075
6320 IF X<>CH THEN 6370 !089
6330 LET COUNT=COUNT+1 !024
6340 IF COUNT=5 THEN 1763 !0
43
6350 LET R=R+2 !171
6360 GOTO 6310 !013
6370 REM DIAG R U !182
6380 LET R=P !237
6390 LET C=T !226
6400 LET COUNT=1 !206
6405 IF R-2<1 THEN 6480 !046
6410 CALL GCHAR(R-2,C+2,X)!0
08
6420 IF X<>CH THEN 6480 !200
6430 LET COUNT=COUNT+1 !024
6440 IF COUNT=5 THEN 1763 !0
43
6450 LET R=R-2 !172
6460 LET C=C+2 !141
6470 GOTO 6405 !109
6480 REM DIA L D !088
6490 LET R=P !237
6500 LET C=T !226
6510 CALL GCHAR(R+2,C-2,X)!0
08
6520 IF X<>CH THEN 6580 !044
6530 LET COUNT=COUNT+1 !024
6540 IF COUNT=5 THEN 1763 !0
43
```

```
6550 LET R=R+2 !171
6560 LET C=C-2 !142
6570 GOTO 6510 !214
6580 REM DIA L U !105
6590 LET R=P !237
6600 LET C=T !226
6610 LET COUNT=1 !206
6615 IF R-2<1 THEN 6690 !001
6620 CALL GCHAR(R-2,C-2,X)!0
09
6630 IF X<>CH THEN 6690 !155
6640 LET COUNT=COUNT+1 !024
6650 IF COUNT=5 THEN 1763 !0
43
6660 LET R=R-2 !172
6670 LET C=C-2 !142
6680 GOTO 6615 !063
6690 REM DIA R D !094
6700 LET R=P !237
6710 LET C=T !226
6720 CALL GCHAR(R+2,C+2,X)!0
07
6730 IF X<>CH THEN 6800 !009
6740 LET COUNT=COUNT+1 !024
6750 IF COUNT=5 THEN 1763 !0
43
6760 LET R=R+2 !171
6770 LET C=C+2 !141
6780 GOTO 6720 !169
6800 LET TIME=TIME+1 !100
6810 RETURN !136
7000 CALL CLEAR !209
7010 PRINT "************DRAW
************": : : : : : :
 : :!175
7020 GOTO 2115 !154
```

## TEMP/HUMIDITY

## How cold is it?

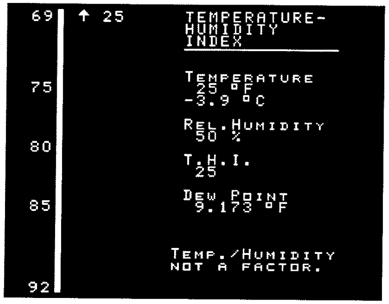
### Extended BASIC helps figure it out

So how cold is it? You can look at the thermometer and figure out the temperature but how cold is it really?

To answer this question you need to know the temperature and the humidity. This gives you the data you need to create a temperature-humidity index.

The following Extend-

Continued on page 42



```
69 ↑ 25       TEMPERATURE-
              HUMIDITY
              INDEX

75            TEMPERATURE
               25 °F
              -3.9 °C

80            REL.HUMIDITY
               50 %

              T.H.I.
               25

85            DEW POINT
               9.173 °F

              TEMP./HUMIDITY
              NOT A FACTOR.
92
```

## TEMP/HUMIDITY

Continued from page 41

ed BASIC program, by K.D. Wentzel, is a good one for determining the temperature-humidity index. Simply type in the temperature and humidity and the program will do the rest.

```
1 ! ********************** !
063
2 ! *                    * !
119
3 ! * TEM/HUMIDITY INDEX * !
081
4 ! *                    * !
119
5 ! * By: K. D. Wentzel  * !
128
6 ! *     Charlotte, NC  * !
090
7 ! *  Copyright(c)1985  * !
187
8 ! *  TI Ext. Basic     * !
245
9 ! ********************** !
063
10 CALL CHAR(124,"3C3C3C3C3C
3C3C3C")!144
11 CALL CHAR(126,"10387CFE10
101010")!055
12 CALL CHAR(92,"101010FEFE7
C3810")!050
15 CALL CHAR(123,"103070FF70
301000")!033
16 CALL CHAR(96,"00F09090F0"
)!207
18 CALL CHAR(95,"0000FF")!24
8
45 CALL INIT !157
49 !if you have a copy of
the screendump program, you
can unrem line 50 and print
the chart by pressing  P !20
3
50 !CALL LOAD("DSK1.SCREENDU
MP") !171
100 CALL CLEAR :: CALL SCREE
N(5):: FOR X=1 TO 12 :: CALL
 COLOR(X,16,5):: NEXT X !026
105 DISPLAY AT(10,1):"TEMPER
ATURE-HUMIDITY INDEX" !237
106 DISPLAY AT(19,14):"K.D.
Wentzel" :: DISPLAY AT(21,14
):"Rev.C.2" !162
107 DISPLAY AT(24,1):"Info :
 (Y/N) N" :: ACCEPT AT(24,14
)VALIDATE("YNyn"):A$ !116
109 IF (A$="Y")+(A$="y")THEN
 GOSUB 1000 !178
110 CALL CLEAR !209
112 DISPLAY AT(1,1):"This pr
ogram will calculate the T.H
.I or Comfort Index." !178
114 DISPLAY AT(6,1):"Please
Input the following  Data :"
 !057
116 DISPLAY AT(10,1):"Temper
ature in Fahrenheit" !015
117 ACCEPT AT(12,5)VALIDATE(
NUMERIC)BEEP:TEMP !011
120 DISPLAY AT(15,1):"Relati
ve Humidity %" !089
122 ACCEPT AT(17,5)VALIDATE(
NUMERIC)BEEP:RH !116
125 CTEMP=5/9*(TEMP-32)!082
126  CTEMP=INT(10*CTEMP+.5)/
1
0 !183
```

## TEMP/HUMIDITY

```
130 XRH=RH/100 !106
131 IF TEMP<58 THEN THI=TEMP
 :: GOTO 140 !150
135 THI=TEMP-(.55-(.55*XRH))
*(TEMP-58)!017
139 ! OK folks lets find the
approx. dew point ... !146
140 !XDP=THI-(.55*TEMP)-17.5
 !255
145 !TDP=XDP/.2 !035
150 DP1=LOG(RH/100)!122
153 DP2=30.1104*DP1 !050
156 DP3=TEMP+DP2 !066
159 DP4=0.113903*DP3 !111
162 DP5=DP4+22.3631 !062
165 DP6=DP5*LOG(RH/100)!011
167 TDP=TEMP+DP6 !103
170 TDP=INT(TDP*1000+.5)/100
0 !089
499 CALL CLEAR !209
500 DISPLAY AT(1,14):"TEMPER
ATURE-" !222
501 DISPLAY AT(2,14):"HUMIDI
TY" !205
502 DISPLAY AT(3,14):"INDEX"
 !214
503 DISPLAY AT(4,14)SIZE(13)
:"_____________" !192
510 CALL VCHAR(1,5,124,24)!2
38
515 DISPLAY AT(1,1)SIZE(2):"
69" !230
517 DISPLAY AT(17,1)SIZE(2):
"85" !028
520 DISPLAY AT(24,1)SIZE(2):
"92" !024
522 DISPLAY AT(12,1)SIZE(2):
"80" !018
524 DISPLAY AT(7,1)SIZE(2):"
75" !233
600 DISPLAY AT(6,14)SIZE(13)
:"Temperature" !123
602 DISPLAY AT(7,14)SIZE(13)
:TEMP;"`F" !117
605 DISPLAY AT(8,14)SIZE(13)
:CTEMP;"`C" !182
610 DISPLAY AT(10,14)SIZE(13
):"Rel.Humidity" !184
611 DISPLAY AT(11,14)SIZE(13
):RH;"%" !131
615 DISPLAY AT(13,14)SIZE(13
):"T.H.I." !134
617 DISPLAY AT(14,14)SIZE(13
):THI !048
620 DISPLAY AT(16,14)SIZE(13
):"Dew Point" !103
625 DISPLAY AT(17,14)SIZE(13
):TDP;"`F" !089
700  ZTHI=INT(THI+.5)!012
701 TTHI=INT(10*THI+.5)/10 !
227
705 IF ZTHI<69 THEN 800 !087
710 IF ZTHI>92 THEN 850 !134
715 ROW=ZTHI-68 !239
720 FOR LOOP=1 TO 15 !095
722 DISPLAY AT(ROW,5)SIZE(6)
:"      " !032
723 DISPLAY AT(ROW,5)SIZE(6)
:"{";TTHI !228
725 NEXT LOOP !208
726 DISPLAY AT(21,13):"THI i
s in the" :: DISPLAY AT(22,1
3):"Discomfort Zone" !192
727 GOTO 870 !184
800 FOR LOOP=1 TO 15 !095
```

Continued on page 44

## TEMP/HUMIDITY

Continued from page 43

```
805 DISPLAY AT(1,5)SIZE(6):"
     " !067
810 DISPLAY AT(1,5)SIZE(6):"
~";TTHI !233
815 NEXT LOOP !208
816 DISPLAY AT(21,13):"Temp.
/Humidity" :: DISPLAY AT(22,
13):"not a factor." !143820
GOTO 870 !184
850 FOR LOOP=1 TO 15 !095
853 DISPLAY AT(24,5)SIZE(7):
"      " !122
855 DISPLAY AT(24,5)SIZE(7):
"\";TTHI !254
857 NEXT LOOP !208
858 DISPLAY AT(21,13):"Acute
" :: DISPLAY AT(22,13):"Disc
omfort!" !210
870 FOR DELAY=1 TO 500 :: NE
XT DELAY :: DISPLAY AT(24,16
):"another? (A)" !074
900 CALL KEY(0,K,S):: IF S=0
 THEN 900 !203
910 IF (K=65)+(K=97)THEN 110
 !135
920 IF (K=80)+(K=112)THEN DI
SPLAY AT(24,16):" " :: CALL
LINK("SCRDMP"):: GOTO 870 !1
20
999 CALL CLEAR :: DISPLAY AT
(12,3):"Program Terminates"
:: END !221
1000 CALL CLEAR :: PRINT " T
EMPERATURE-HUMIDITY INDEX" !
113
1005 PRINT !156
1010 PRINT "Different combin
ations of  heat and moistur
e in the aircause different
sensations  of comfort or di
scomfort in" !245
1015 PRINT "human beings. TH
I is a term used by the Weat
her Bureau  to express what
this        combined Temp./H
umidity" !066
1020 PRINT "effect has on th
e majority  of people, altho
ugh it is   known that indiv
idual" !104
1025 PRINT "reactions vary c
onsiderably from person to p
erson." !209
1030 PRINT "By extensive lab
 tests THI  values have been
 establishedfor each group o
f separate" !183
1035 PRINT "simultaneous air
 temp. and  relative humidit
y readings  which give equiv
alent " !191
1040 PRINT "feelings of comf
ort." !001
1045 PRINT "-press any key t
o continue-" !094
1050 CALL KEY(0,K,S):: IF S=
0 THEN 1050 !098
1070 CALL CLEAR :: PRINT "Th
e THI has a direct        re
lationship with the level of
 comfort experienced by " !0
03
1075 PRINT "most people. Rel
atively few people in summer
 will be   uncomfortable fr
```

## TEMP/HUMIDITY

```
om heat and humidity while t
he THI is" !031
1080 PRINT "70 or below.  Ab
out half of the people will
be          uncomfortable by
 the time   the THI reaches
 75.  Almost" !089
1085 PRINT "everyone will be
            uncomfortable by
 the time   the THI reaches
 79,  and" !236
1090 PRINT "discomfort becom
es more     acute as the ind
ex climbs   still higher." !
217
1095 PRINT "Infrequently in
some parts  of the U.S. the
THI values  reach as high as
  90." !061
1100 PRINT "The highest THI
noted so farin U.S. data is
 92." !228
1105 PRINT " -press any key
to start-" !069
1110 CALL KEY(0,K,S):: IF S=
0 THEN 1110 !158
1115 CALL CLEAR :: RETURN !2
19
```

## NEWSBYTES

### TIMUG'99 planned for May 15

TI-CHIPs will host the 1999 TI99/4A and Geneve Multi User Group Conference (TIMUG'99) May 15 in Brookpark, Ohio.

A pre-conference get-together party will be held from 7 to 10 p.m. May 14 at the Middleburg Heights Recreation Hall, located on Bagley Road (about four miles southeast of Cleveland Hopkins International Airport).

TIMUG'99 will be held May 15 at the Spang Mansion on Kolthoff Road in Brookpark (a southwest suburb 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.

Glenn Bernasek of the users group says maps to the site and motel information will be published.

Schedule for the Saturday conference is setup, 7-9 a.m.; seminars and demonstrations, 9 a.m.-5 p.m.; MUG officers/members conference 1-2 p.m.; Jim Peterson Service Awards, 5-5:15 p.m.; cleanup, 5:15-6 p.m.; and pizza party, 6-7 p.m. Bernasek says on-site food service will be available during the conference.

Like the Lima MUG of former years, the conference will be free to vendors and attendees. However, donations to defray costs will be accepted.

For information or to make reservations, contact Glenn Bernasek, 13246 Harper Rd., Strongsville, OH 44136; phone: (440)846-0865 (after 9:00 p.m. EST); e-

Continued on page 46

## NEWSBYTES

Continued from page 45

mail: gbbasics@aol.com. Bernasek says all messages will be answered.

### Voting starts for Peterson awards

Voting is under way for the 1999 Jim Peterson Achievement Award, to be presented May 15 at TIMUG 99. The TI-Chips user group handles the nomination process. The nominees and their categories are:

Community Service: Rich Polivka, TI Web Page; MICROpendium, TI magazine; SouthWest 99ers, Fest West '98.

TI99/4A Software: Bruce Harrison, Midi for the Super AMS; John Bull, Contract Bridge

TI99/4A Hardware: System 99 user-group, SCSI board modifications; Michael Becker, High Speed GPL card; Don O,Neil, ongoing SCSI work

**Myarc, Geneve 9640**: Tim Tesch, Enhanced the 9640 OS; Don Walden,SCSI for the Geneve

You may vote for one nominee in each category. The deadline is midnight, April 15. Submit votes to Glenn Bernasek, 13246 Harper Road, Strongsville, Ohio 44136; or e-mail: GBBasics@aol.com or dd314@Cleveland.Freenet.Edu.

## BUGS AND BYTES

### What are they doing now?

Bill Gaskill forwarded this update from Archiver author Barry Boone.

Yes, I've kept track of the TI world, even if I haven't had much time to really take part in it. Steve Lamberti is still at Creative Labs, and I was there up until mid-October, when I took the job as head Web infrastructure guy at PennWell Publishing in Tulsa (they're a large magazine publisher, and make the likes of Computer Graphics World, BackOffice magazine, Oil & Gas Journal, and another 40 or so magazines and trade journals.) I'd been doing that same type of work at Creative for over 5 years, and it was time for a change. I still talk to Steve regularly (he's the manager of their customer service department.

I doubt I'll ever lose interest in the TI. It was the machine I learned everything on, and I've never found anything in the PC world to be nearly as much fun to figure out, except possibly the Internet. I really miss the nights of hacking together assembler code to do things that hadn't been done by someone else on the TI. Or porting Infocom games from the PC, or the Tomy OS to the Geneve, or a hundred other things like that.... I still have all sorts of goodies in my "TI closet" and I hope someday to have the time to dig through it and set up a system, but until then, I at least try to read some of what comes through the list, just to see what people are doing with the TI nowadays.

## MICROREVIEWS

# TI Bingo, Interest Calculation for MDOS, Disk Manager 2000, Cellgrow, BMP Converter, Backup Bitremover, Bacteria, Copy-C, and Hardware Tests

BY CHARLES GOOD

### INTEREST CALCULATION for MDOS by Martin Zeddies

These two programs are for Geneve owners. They have an 80-column display, run directly out of MDOS, and have been tested with MDOS 6. There are lots of interest calculation programs for the 99/4A. Many of these execute slowly because they are written in BASIC. Martin's programs are they only interest calculation I know of for the Geneve, and they execute very rapidly.

The first program asks for a starting balance, interest rate in percent, starting year, and number of years. You get a yearly table showing interest earned each year and balance at the end of each year, calculated as interest compounded yearly. This program lets you enter interest as a decimal percent, such as 5.8 percent.

The second program asks for starting balance and a minimum percent interest rate. You get tables showing the ending balance for each of 20 years, interest compounded yearly. The tables show you data for the minimum interest rate you entered and the next five interest rate percents. If you entered a minimum rate of 3 percent then you get 20-year tables for 3, 4, 5, 6, 7, and 8 percent so that you can compare possible yields at all these percents side by side. This program does not calculate decimal interest percent rates.

These programs are public domain. Send me $1 and I will mail them to you on a SSSD disk complete with source code and on disk documentation.

### TI BINGO by Bruce Harrison

A first class commercial bingo game requires lots of preprinted bingo cards as well as a ping-pong ball blower machine to randomly mix the balls with bingo numbers printed on them and pop these balls up one at a tine. You have probably seen such machines on state lottery "live drawing" television shows. Well now anyone can operate a bingo game for fun or profit without any of this hardware. All you need is a 99/4A

Continued on page 48

Continued from page 47

system and the TI Bingo software.

When you first run TI Bingo you get three menu options — print some bingo cards, play a game, and exit. The bingo card option works with any printer, but was specifically designed for 24-pin dot matrix or bubble jet printers. You get your choice of large or small cards. The preferred option is small cards because this will normally print four different bingo cards on one 8.5 x 11-inch sheet of paper. After selecting large or small cards you next pick the number of paper sheets you want to print. Press Enter and your printer grinds out the selected number of bingo cards.

Numbers printed in the five rows under the "B" column are a random combination of numbers 1-15. Under the "I" column you get a random selection of numbers 16-30, etc. Each bingo card will show different combinations, 24 numbers from within the range of 1-75. There is a "free" space in the middle of the card.

Bruce reports that on his Gemini 10X printer cards are printed but the box borders appear as strange characters. With my own ancient 9-pin IBM Graphics Printer I was unable to print multiple bingo cards on a sheet of paper. I could only get this 9-pin printer to produce one card per sheet. The bingo cards thus printed are nicely formatted.

When you choose the play a game option just press a key to announce the next bingo number. The randomly chosen letter/number combination is displayed at the top center of the screen and then spoken, if you have the speech synthesizer. The letter/number combination is also stored at the bottom of the screen. As each additional letter/number is announced it remains displayed at the bottom of the screen.

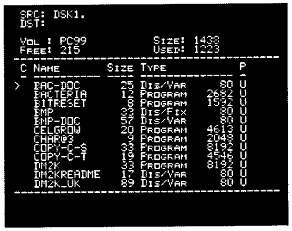
These called combinations accumulate at the bottom of the screen in numerical order in columns corresponding to B-I-N-G-O. When someone claims to have bingo, the game host can verify the winning card against the called numbers displayed on the bottom of the screen.

One of the features I really like is the verbalization of letter/number combinations. Unfortunately the speech of TI Bingo doesn't work on my Geneve. I have a Rave speech card in my Geneve which is normally the only way to add speech to the Geneve. TI Bingo is the only software I have found in which speech works on a 99/4A system but not on my Rave speech card equipped Geneve.

TI Bingo is written in assembly and comes on a SSSD disk. You can load it from any assembly loader or from Extended BASIC. On disk documentation and source code are included. Unlike much of Bruce Harrison's software, TI Bingo is not public domain. It is commercial and costs $5 including shipping. Send your money directly to Bruce at the address below.

## TI SOFTWARE FROM EUROPE by TI Club Errorfree and TI Gebruikersgroep

This DSSD disk was given to those attending the 1998 Chicago TI Faire. It has mostly new software, all of which is free. Copying is encouraged. Many of the programs on the disk have extensive documentation. Unfortunately all documentation is in German or Dutch. I found my German dictionary useful in figuring out how to use some of the software.

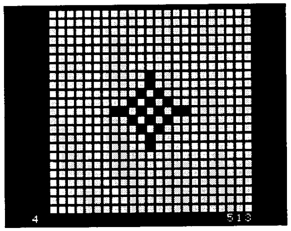


```
 SRC: DSK1.
 DST:

 VOL : PC99           SIZE: 1438
 FREE: 215            USED: 1223
----------------------------------------
 C NAME        SIZE TYPE            P
 - ----------- ---- -------------- -
>  BAC-DOC       25 DIS/VAR       80 U
   BACTERIA      12 PROGRAM     2682 U
   BITRESET       8 PROGRAM     1592 U
   BMP           33 DIS/FIX       80 U
   BMP-DOC       57 DIS/VAR       80 U
   CELGROW       20 PROGRAM     4613 U
   CHAR@3         9 PROGRAM     2048 U
   COPY-C-S      33 PROGRAM     8192 U
   COPY-C-T      19 PROGRAM     4546 U
   DM2K          33 PROGRAM     8192 U
   DM2KREADME    17 DIS/VAR       80 U
   DM2K_UK       89 DIS/VAR       80 U
----------------------------------------
```

DM2K, or Diskmanager 2000 v1.2, is the most significant software on the disk, in my opinion. This was described in Berry Harmsen's article in the Nov/Dec 1998 issue of MICROpendium. It is a must have for 40-column SCSI and HFDC hard drive systems because it is the best software available in 40 columns for creating, deleting, and viewing the contents of directories and subdirectories. You can easily move up and down the directory structure of your hard drive. You can also copy move and view in ASCII or hex individual files.

For Geneve users Clint Pulley's 80-column Directory Manager v1.2 is a better product. This can do everything DM2K can do plus format disks and execute MDOS and EA5 software.

DM2K loads using either the Editor/Assembler module or Funnelweb's loader No. 3 (program E/A). Be aware that DM2K seems to crash frequently with some non-TI equipment. It doesn't recognize the Myarc RAMdisk on my PC99 system and I have reports that it has trouble with Horizon and CorComp RAMdisks.



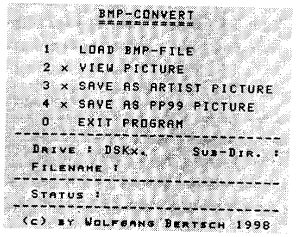
**Cellgrow** is an Extended BASIC program that resembles a fractal image maker. It also resembles a computer simulation generally known as the "game of life". Continuously changing patterns are generated on screen. You can, before you run the program, change the "rules" which govern these patterns so that you get different sorts of

Continued on page 50

## MICROREVIEWS

Continued from page 49

patterns each time you run the program.



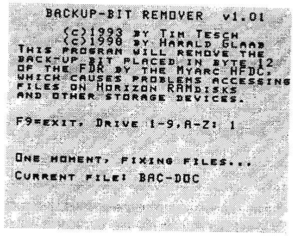
```
          BMP-CONVERT
          ===========

   1   LOAD BMP-FILE
   2 x VIEW PICTURE
   3 x SAVE AS ARTIST PICTURE
   4 x SAVE AS PP99 PICTURE
   0   EXIT PROGRAM
------------------------------
 DRIVE : DSKx.     SUB-DIR. :
 FILENAME :
------------------------------
 STATUS :
------------------------------
 (c) BY WOLFGANG BERTSCH 1998
```

**BMP Converter v1.2** lets you manipulate bitmap images as large as 480 x 792 pixels that are normally stored as *.bmp files on a PC. No 80-column device is needed to do this. The software works on a basic 40-column 99/4A and it also works with a Geneve in GPL mode. After loading an image you can view it on screen. You can also save the bmp image as a TI Artist 16-color disk file or a black and white large Page Pro image. No other software lets you use PC bmp images on a TI system.

There are lots and lots of such images available. For example, a standard installation of Windows 95 or 98 includes over 100 of these graphics. This is the standard format for the Windows "Paint" program.

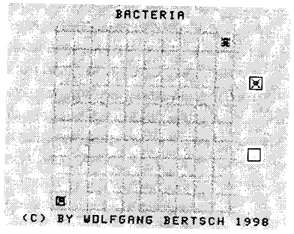
The main problem is how to get bmp files onto a TI disk. Image files can be downloaded with terminal programs and they can also be transferred from a PC disk to a TI disk using PC Transfer and its D/F28 option.

Unfortunately not all PC *.bmp files can be viewed. I put several Windows 95 bmp files on a TI disk. Some gave me a "file too large error." Others gave other error messages including "no hires allowed." Others viewed and saved just fine.



```
    BACKUP-BIT REMOVER  v1.01
      (c)1993 BY TIM TESCH
      (c)1990 BY HARALD GLAAB
THIS PROGRAM WILL REMOVE THE
BACK-UP BIT PLACED IN BYTE 12
OF THE FDR BY THE MYARC HFDC,
WHICH CAUSES PROBLEMS ACCESSING
FILES ON HORIZON RAMDISKS
AND OTHER STORAGE DEVICES.

F9=EXIT, DRIVE 1-9,A-Z: 1


ONE MOMENT, FIXING FILES...
CURRENT FILE: BAC-DOC
```

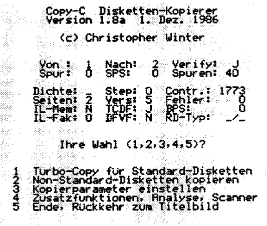
**Backup-Biteremover v1.01** is updated in 1998 from an original written in 1993 by Tim Tesch. It removes the backup bit placed in floppy disks by the Myarc HFDC. This bit has caused problems accessing files on Horizon RAMdisks. I don't own an HFDC, so I haven't been able to test this software.



**Bacteria v1.0** is a two-player game written in Extended BASIC. You use the joysticks or arrow and "Q" keys.

## MICROREVIEWS

Each player starts with one bacterium at opposite corners of a grid. Players alternately add a new bacterium to the matrix immediately adjacent to an existing bacterium. The object is to fill the matrix and prevent your opponent from doing so. There is a law in ecology I teach to my biology students which states that no two species of organisms can successfully live in the same physical area while competing for the same energy or food source. This game resembles a laboratory simulation of this ecological law.



```
      Copy-C  Disketten-Kopierer
      Version 1.8a  1. Dez. 1986

        (c) Christopher Winter

     Von :  1  Nach:  2  Verify:  J
     Spur:  0  SPS:   0  Spuren: 40

    Dichte: _  Step: 0  Contr.: 1773
    Seiten: 2  Vers: 5  Fehler:    0
    IL-Mem: N  TCDF: J  BPS:       0
    IL-Fak: 0  DFVF: N  RD-Typ:  _/_

        Ihre Wahl (1,2,3,4,5)?

  1  Turbo-Copy für Standard-Disketten
  2  Non-Standard-Disketten kopieren
  3  Kopierparameter einstellen
  4  Zusatzfunktionen, Analyse, Scanner
  5  Ende, Rückkehr zum Titelbild
```

**Copy-C v1.8a**, a disk copy program, will fast-copy regular and many "funny" sector protected TI disks, and it will also copy a 360K PC disk onto another 360K PC disk if you have a DSDD disk controller. This version has been altered so that it will work on a Horizon RAMdisk.

**Hardware tests v1.2** is actually two programs. One tests keyboard and joystick functions. This is handy if you have an IBM keyboard interfaced to your 99/4A and it is also useful in calibrating the key delay setting (k value) when using PC99. Many of us already have this keyboard/joystick test program.

The second hardware test program checks the functioning of the 9901 chip. Various register addresses are displayed and you are told if the byte is set for 1 or 0. Pressing some keyboard keys will alter some of these settings.

On the TI Software from Europe disk are several interesting sounding 80-column utilities. Unfortunately none of them work on my Geneve so I can't test them. Apparently you need one of the various 99/4A 80-column cards to make them work. These 80-column utilities include a SCSI sector editor, an updated GIF-99 GIF file viewer now with mouse support, and a SCSI disk manager. The disk manager is supposed to show a split screen with information about the source and destination directories.

If you want the TI Software from Europe disk, send me $1 and I will send it to you. I can also, for free, e-mail it as an attached file in PC99 format.

Access:

Bruce Harrison (for TI Bingo)
5705 40th Place
Hyattsville MD 29781
Phone 301-277-3467
e-mail rottencat1@aol.com

Charles Good (Interest calculation for MDOS, disk of European software)
P.O. Box 647
Venedocia OH 45894
Phone 419-667-3131
e-mail good.6@osu.edu

## Two-column printing

*The following item was written by Gene Bohot and appeared in the newsletter of the TI Orphans of Puget Sound.—Ed.*

This may not be the fastest or slickest way to do a two-column printout, but it shows a little about file handling and is easy to explain. It will only work with a maximum of one page.

First type out the article with Funnelwriter, BA-Writer, or TI-Writer in any format you would like to use. Then make the first line LM4;RM38;FI;AD and the second line IN+5 and save it as usual. Then, in formatter, instead of "PIO" enter "DSK1,TEST1". This will print the article to disk and format it with fill and adjust to a single long column.

Return to the editor, load TEST1 and remove the blank lines at the start and at line 60. These were put in by the formatter. Now you can clean up any lines that did not format properly into 34 columns, add a title at the top of the column, and check the layout. For now ignore the control characters in the file.

Now pad the file at the end if necessary to get an even number of lines and then print it to disk with "PF" and "C DSK1.TEST2". The "C" tells the file printer to delete the control codes. If you look at the file onscreen you should see a single long article that is 34 columns wide and right-justified.

Then run the following program.

```
100 DIM L$(128)
110 OPEN #1:PIO
120 OPEN #2:"DSK1.TEST2"
130 A=A+1
140 LINPUT #2:L$(A)
150 IF EOF(2)=O THEN 130
160 CLOSE #2
170 FOR X=1 TO A/2
180 PRINT #1:L$(X);TAB(41);L
$(X+(A-1)/2)
190 NEXT X
200 CLOSE #1
```

This will read the entire file and put each line into an array numbered from one to the last line number. When it reaches the End Of File marker, it will close the file and print the first line, tab to the center of the page and print the first line of the second half of the article. It will continue until each line is printed, then close the printer. Very simple.

The program can be modified to print a longer article by increasing the DIMension of the array, but you would have to add a line or two to make it skip over the perforation. Try the program with an article you already have typed in to see how well it works.

## FIX ensures privacy

FIX by Ken Woodcock, modified by Jim Peterson, when merged into any program (which does not have a line number less than 4) and RUN, will change each line length byte to 0, making the program unlistable, although it can be saved, loaded and

run.

After running the first time type 1, use FCTN X and FCTN 3 to delete line 1. Do the same with lines 2 and 3 only. Then SAVE.

UNFIX, also by Woodcock, will restore the line length bytes to make the program listable. RESequence the program, MERGE in UNFIX, and RUN. Then DELETE lines 1-6 and 100-110.

Continued on page 54

## USER NOTES

Continued from page 53

### FIX

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4
2 CALL PEEK(X,E,F,G,H):: ADD
=G*256+H-65536 :: @=@+1 :: I
F @<4 THEN 3 :: CALL LOAD(AD
D-1,0)
3 NEXT X :: !@P-
```

### UNFIX

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT "LINE # "
;E*256+F
2 I=1 :: CALL PEEK(ADD-1,V):
: IF V THEN 6
3 CALL PEEK(ADD+I,V,W):: IF
V THEN I=I+1 :: GOTO 3
4 FOR Y=SL TO EL STEP -4 ::
CALL PEEK(Y,E,E,E,F):: IF E*
256+F-65536=ADD+I+2 OR W=0 O
R ADD+I>-3 THEN CALL LOAD(AD
D-1,I+1):: GOTO 6
5 NEXT Y :: I=I+1 :: GOTO 3
6 NEXT X :: STOP :: !@P-
```

### Work-around for text buffer full errors

*The following item was written by Ted Peterson and appeared in his TI Notes column. We saw it in Wordplay, the newsletter of the Portland (Oregon) Users of Ninety-Nines.—Ed.*

Text buffer full! What a nasty thing to find at the top of your screen when you are in a hurry to print a file.

One way to overcome this is to use the print function of DM-1000. You can print the entire file without unwanted things that the TI-Writer formatter would do. In e-mail it is common to find the "at" sign (@), which the formatter interprets as a printer command. The formatter also insists on wasting paper with its paging functions.

Another way is to use the print function of Barry Boone's Archiver program. It works well, similarly to that of DM-1000.

If you print out a long file this way you can actually find out how many lines are used. Then by loading segments of the file you can save it in several parts. When you get near the end of the file you can determine exactly how many lines are used by counting them from the printout made by Archiver or DM-1000. By loading each segment separately you can do necessary editing and re-assemble the finished article.

**Have a subscription problem?**
**Didn't receive a disk?**

You can call us at 512-255-1512 between 9 a.m. and noon Saturdays CT; or fax us anytime at 512-255-1512; or e-mail us at micropendium@yahoo.com.

## USER NOTES

### Eliminate extra spaces

The following routine removes extra spaces from TI-Writer text files that have been filled and adjusted

```
100 DISPLAY AT(3,6)ERASE ALL
:"TIGERCUB UNFILLER":"":" To
 remove extra spaces from":"
a TI-Writer text which has":
"been Filled and Adjusted by
"
101 DISPLAY AT(8,1):"the For
matter, prior to":"reformatt
ing."
110 DISPLAY AT(15,1):"Input
file? DSK" :: ACCEPT AT(15,1
6):IF$ :: OPEN #1:"DSK"&IF$,
INPUT
120 DISPLAY AT(17,1):"Output
 file? DSK" :: ACCEPT AT(17,
17):OF$ :: OPEN #2:"DSK"&OF$
130 LINPUT #1:M$ :: P=1
135 X=POS(M$," ",P):: IF X=P
 THEN P=P+1 :: GOTO 135
140 X=POS(M$,"  ",P):: IF X=
0 THEN PRINT #2:M$ :: GOTO 2
00
150 M$=SEG$(M$,1,X)&SEG$(M$,
X+2,255):: GOTO 140
200 IF EOF(1)<>1 THEN 130 ::
 CLOSE #1 :: CLOSE #2
```

## DISKS/BACK ISSUES

❏ Back Issues,$3.50 each to March 1996, later $6 each. List issues on separate sheet.

No price breaks on sets of back issues. Free shipping USA. Add $1, single issues to Canada/ Mexico. Other foreign shipping 75 cents single issue surface, $2.80 airmail. Write for foreign shipping on multiple copies.

OUT OF STOCK: V1#1-2; V2#1

**GENEVE PUBLIC DOMAIN DISKS**
**(SSSD unless specified)**

These disks consists of public domain programs available from bulletin boards. If ordering DSDD, specify whether Myarc or CorComp.

| | SSSD | DSSD | DSDD |
|---|---|---|---|
| ❏ Series 1 | $9 | $7 | $5 |
| ❏ Series 2 | $9 | $7 | $5 |
| ❏ Series 3 | $9 | $7 | $5 |
| ❏ Series 4 | $9 | $7 | $5 |
| ❏ Series 5 | $9 | $7 | $5 |
| ❏ Series 6 | $9 | $7 | $5 |