

MANNERS NEWSLETTER

NEWSLETTER OF THE MID ATLANTIC NINETY NINE'ERS

VOL. 3

FORMERLY THE WASHINGTON DC AREA NEWSLETTER

NO. 7

Jim Horn, President

JULY 1984

Richard D. Sturgell
Bill Whitmore Editors

TI CLUB MEETINGS

THE WASHINGTON DC AREA TI HOME COMPUTER USERS GROUP

The Washington DC Area TI Home Computer Users Group meets monthly. The regular meeting night is the second Thursday of each month. The MEETING for JULY will be held on the regular THURSDAY SCHEDULE at 7:00 PM, JULY 12th. This meeting will be held at the FAIRFAX HIGH SCHOOL. For directions or other info Call Frank Jordan at (301) 899-3882 or Jim Horn at (301) 340 - 9617.

TIBUG - THE BALTIMORE USERS GROUP

The Baltimore Group meets the FIRST TUESDAY of each month at the PINEGROVE ELEMENTARY SCHOOL in Parkville, Md. at 7:00PM.

HAGERSTOWN - WILLIAMSPORT TI USERS GROUP

Meetings are held at the WILLIAMSPORT MEMORIAL LIBRARY on the 4th FRIDAY of each month. Also a free INSTRUCTIONAL MEETING is held on the 2nd FRIDAY each month. Meetings start at 7:00PM. For more info call Sam Williams at (301) 223-8014., or Phil Shew at (301) 739-7091.

MONTGOMERY COUNTY TI USERS GROUP

The Group meets at The SLIGO INTERMEDIATE SCHOOL, in the Library at 7:30PM. The regular meeting night is the 4th THURSDAY of each month. For more information call ALLEN MINTON at (301) 493-45026. Either number may be called without toll from Washington DC area. Next meeting THURSDAY, JULY 26th.

BOWIE - CROFTON TI USERS GROUP

The BOWIE - CROFTON Group meets on the THIRD WEDNESDAY of each month. The meetings will be held at the Help Others Help Social Hall in the Bowie Plaza Shopping Center on Rt 197 in Bowie, Md. The meetings start at 7:00PM. For info or directions call Chris Goodman at 262-5570

SEVERNA PARK TI USERS GROUP

The Severna Park TI Users Group's next meeting will be held at the SEVERNA PARK LIBRARY on McMinsey Rd. near Md. Rt. 2, across from the Severna Park Shopping Center at 7:00PM on FRIDAY, AUG 3, 1984. For more info call Randall Rainey at 841-5375.

COMPU-SOURCE CORNER

By Rick Stickle
TI0121
70220,242

Did you ever really read the manual that came with your Terminal Emulator II? If you did you'd find that like the rest of TI's manuals it does very little to let you know the whys and whatever. I'd like to start taking all of you on a trip beyond the TE II book and give you some idea of whats going on when you plug that modem in.

Everything that happens when your TI "talks" to another computer is set down by protocols, these protocols have been set down by The International Standards Organization (ISO). The ISO has developed a model with seven protocol layers to describe communications between different computers. This list which follows starts at the lowest hardware layers and works its way to the highest software layers.

The physical layer. This layer consists of the mechanical, electrical, and functional arrangements necessary for a connection. This includes the telephone lines, and the RS 232C. The RS 232C is the older of two commonly used serial interfaces for on-line communications, the newer is the RS 422. The RS 422's improvements include higher-speeds and longer connecting wires between devices. An example of the speed difference is the RS 232C are generally limited to 19,200 bits per second, the RS 422 has a speed of 230,000 bits per second. In some cases the two types of ports can be mixed (ie. RS 422 port running a RS 232C printer.)

The data-link layer. This electronic protocol is used to convey a unit of information from one node to another. A node is any device - a computer or printer for example - capable of sending and/or receiving information. This layer includes flow control, who can send data and when, and some forms of error control.

The network layer. This layer determines how information from the sender is sent or routed to the correct receiver. This step is usually missing if there are only two devices involved. Examples of this are the electronic mail that we all love to receive.

The transport layer. These are the steps taken to ensure high-quality network service, including confirmation that the information has reached its destination and has been read without error. This layer is not always used; many computers simply send information without knowing where it goes. An example is the error detection and correction features of many communications programs like the auto detection used when downloading a program from Texnet.

The session layer. This is the procedure by which the two

communicating computer devices coordinate action. An example is the sending computer requesting that the receiving computer open a file, accept information, and close the file; the user sitting at the receiving computer does not need to intervene at any time

The presentation layer. This protocol details all the formatting and code conversion necessary to make information from the sending computer intelligible to the receiving computer, this for example is like Texnet supplying graphics information in a form that our TI can display.

The application layer. This layer ensures that the information is sent in a form that can be directly by an application program running on the receiving computer, for example sending a Multiplan file from your TI to someone who has a Apple.

Many computer practices cut across several types of protocol, so things are not as cut and dry as they seem in this listing. Still, we can use the concepts as a guide. For your computer and another to communicate at all, you must agree on the physical and data-link protocol layers. Since these are the most important I will go over them in more detail in the next column, as usual, I am open to suggestions as to what you, the rest of the club, would like to see or read about in this column. If there are any suggestions please get them to me.

Guaranteed Auto Program Loader, Part Two

Or, the continuing quest for a better loader

By C. Bobbitt

The following program, by Mike de Frank, is a lot like the auto program loader found in the April issue of this newsletter, but much better. Unlike that rudimentary version, this is a full featured loader that will catalog diskettes, delete files, and more importantly, allow the user to run any program file on the disk, not just the first twenty-two. Also, unlike that previous listing, every line in this listing may be typed in as is (that and a few other changes allowed me to put "Altered by Chris Bobbitt" in the listing header. Sort of cheap, but legit..

This program is really self explanatory, completely menu and prompt driven, and not only that, it also looks nice on the screen. The program requires the Disk Drive system (of course), and a memory expansion device. A printer, though nice to have, is optional.

One possible alteration to this program, change line 430 to read:

```
430 IF D$="E" THEN CALL CLEAR :: CALL LOAD(-32730,32)
```

This will cause the computer to QUIT back to the Master Title screen when the key "E", for exit, is pressed on the programs function menu. For all of you that receive The Smart Programmer and have the February issue, I didn't get this address out of there, but I discovered how to find it out of there.

```

10 : DISK INDEXER 10
20 : BY MIKE DE FRANK
30 : ALTERED BY C. BOBBITT
40 A$="PIO" :: CALL CLEAR :: CALL SCREEN(2):: CALL COLOR(0,13,1,13,13,1,14,5,1):
: FOR D=1 TO 12 :: CALL COLOR(D,8,1):: NEXT D
50 DIM B$(127),C$(127):: CALL INIT :: GOTO 80 :: CALL ERR :: CALL LOAD :: CALL P
EEK :: CALL VCHAR :: CALL SOUND :: CALL HCHAR :: CALL CHAR
60 D$,E$,F$,G$,H$,I$,J$,K$,L$,M$="" :: A,B,C,D,E,J,K,L,N,F,Q,R,S=0
70 I@P=
80 CALL CHAR(91,"000000000000FF",92,RPT$("0000FF",3),93,RPT$("FF00",4),94,"000B007
E7E0C0800"):: CALL CHAR(95,RPT$("FF00",4))
90 DISPLAY AT(6,9)ERASE ALL:"Disk Indexer" :: CALL HCHAR(4,8,95,18):: CALL VCHAR
(5,25,95,3):: CALL HCHAR(8,8,95,18):: CALL VCHAR(5,8,95,3)
100 DISPLAY AT(11,14):"By":TAB(9):"mike de frank" :: DISPLAY AT(17,5)BEEP:" Dr
ive (1-3): 1"
110 ACCEPT AT(17,21)SIZE(-1)VALIDATE("123"):F$ :: IF F$="" THEN 110 ELSE F$="DSK
"&F$&","
120 DISPLAY AT(17,4):"\ Scanning Drive "&SEG$(F$,4,1)&" \\"
130 DISPLAY AT(21,10):"Filenames":TAB(8);RPT$("\\",14);TAB(8);"\\":TAB(21);"\\":TAB
(8);RPT$("\\",14):: OPEN #1:F$,INPUT ,RELATIVE,INTERNAL
140 INPUT #1:E$,A,B,C :: D=C :: GOSUB 680 :: G$=H$ :: D=B-C :: GOSUB 680
150 I$=H$ :: E$=E$&SEG$(RPT$(" ",17),1,10-LEN(E$))&" Free:"&G$&" Used:"&I$ :: DI
SPLAY AT(11,14):" ":E$ :: E=0
160 INPUT #1:J$,A,B,C :: E=E+1 :: IF E<10 THEN K$="0"&STR$(E)ELSE K$=STR$(E)
170 IF LEN(J$)=0 OR E=128 THEN 270
180 B$(E)=J$ :: DISPLAY AT(23,10)SIZE(10):J$ :: ON ABS(A)GOTO 190,200,210,220,23
0
190 L$="D/F:" :: GOTO 240
200 L$="D/V:" :: GOTO 240
210 L$="I/F:" :: GOTO 240
220 L$="I/V:" :: GOTO 240
230 L$="Program" :: GOTO 250
240 D=C :: GOSUB 680 :: L$=L$&H$
250 IF A<0 THEN G$="Y" ELSE G$="--"
260 D=B :: GOSUB 680 :: I$=H$ :: C$(E)=K$&" "&J$&SEG$(RPT$(" ",17),1,11-LEN(J$))
&I$&" "&L$&" "&G$ :: GOTO 160
270 CLOSE #1 :: E=E-1 :: DISPLAY AT(1,1)ERASE ALL:E$: " Filename Size Type
F"
280 CALL HCHAR(4,2,93,30):: CALL VCHAR(5,2,93,10):: CALL HCHAR(15,2,93,30):: CAL
L VCHAR(5,31,93,10)
290 CALL HCHAR(17,3,95,28):: DISPLAY AT(18,1):" _ D ^ DELETE P ^ PRINT _": "_
R ^ RESCAN E ^ EXIT _": "_ F ^ FORWARD B ^ BACK _"
300 CALL HCHAR(21,3,95,28)
310 R=1
320 P=INT(((E+11)*1.4)/15):: DISPLAY AT(16,9):"PAGE "&STR$(INT(((R+10)*1.5)/15))
&" OF "&STR$(P):: S=R+9 :: IF S>E THEN S=E
330 FOR D=5 TO 14 :: CALL HCHAR(D,3,32,28):: NEXT D :: A=5 :: IF R<1 THEN R=1
340 A=5 :: IF R<1 THEN R=1
350 FOR B=R TO R+9 :: DISPLAY AT(A,1)SIZE(28):C$(B):: IF B>E THEN 370
360 A=A+1 :: NEXT B
370 DISPLAY AT(23,1):"Option (or "&SEG$(C$(R),1,2)&"-"&SEG$(C$(S),1,2)&"):" "
380 ACCEPT AT(23,20)SIZE(-3)VALIDATE("FBDFPRE0123456789")BEEP:D$ :: IF D$="" THEN
380
390 IF D$="F" AND R+10<=E THEN R=R+10 :: GOTO 320 :: ELSE IF D$="F" THEN R=1 ::
GOTO 320
400 IF D$="B" AND R-10>=1 THEN R=R-10 :: GOTO 320 :: ELSE IF D$="B" THEN R=R*10-
9 :: GOTO 320
410 IF D$="D" THEN GOSUB 470 :: GOTO 310
420 IF D$="P" THEN GOSUB 620 :: GOTO 370
430 IF D$="E" THEN CALL CLEAR :: PRINT "TI EXTENDED BASIC" :: END
440 IF D$="R" THEN 90
450 FOR D=1 TO LEN(D$):: IF ASC(SEG$(D$,D,1))<48 OR ASC(SEG$(D$,D,1))>57 THEN 37
0

```

```

460 NEXT D :: C=VAL(D$):: IF C<1 OR C>E THEN 370 ELSE GOSUB 700 :: GOTO 370
470 DISPLAY AT(23,1)BEEP:"Delete Filename:" :: ACCEPT AT(23,17)SIZE(10):D$ :: IF
D$="" THEN GOSUB 570 :: RETURN :: ELSE D$=D$&" "
480 FOR K=1 TO E :: J=POS(D$(K)," ",4):: J=J-3 :: IF J=0 THEN 500
490 M$=SEG$(D$(K),4,J):: IF M$=D$ THEN 510
500 NEXT K :: K$=" [[ invalid filename [[ " :: GOSUB 670 :: GOTO 470
510 IF SEG$(D$(K),28,1)="Y" THEN K$=" [[ protected file [[ " :: GOSUB 670 :: G
OTO 470
520 A=VAL(SEG$(D$(K),15,3)):: C=A+VAL(SEG$(E$,17,3)):: B=VAL(SEG$(E$,26,3))-A
530 D=C :: GOSUB 680 :: G$=H$ :: D=B :: GOSUB 680 :: I$=H$ :: ON ERROR 560 :: DE
LETE F$&SEG$(D$,1,LEN(D$)-1)
540 D$(K)="" :: Q=Q+1 :: E$=SEG$(E$,1,16)&G$&SEG$(E$,20,6)&I$
550 DISPLAY AT(1,1):E$ :: K$=RPT$(" ",ABS(9-LEN(D$)))&"[[ "&D$&"DELETED [[ " :: G
OSUB 670 :: GOTO 470
560 CALL ERR(J,K):: K$=" [[ disk error - code "&STR$(J)&" [[ " :: GOSUB 670 :: RE
TURN 470
570 IF D=0 THEN RETURN ELSE Q=0 :: DISPLAY AT(23,1)BEEP:" Status: Sorting File
s"
580 K=0 :: FOR L=1 TO E :: IF LEN(D$(L))=0 THEN 610
590 K=K+1 :: IF K<10 THEN K$="0"&STR$(K)ELSE K$=STR$(K)
600 D$(K)=K$&" "&SEG$(D$(L),4,28):: A=POS(D$(K)," ",4):: IF A>0 THEN B$(K)=SEG$(
D$(K),4,A-4)
610 NEXT L :: FOR L=K+1 TO E :: D$(L)="" :: NEXT L :: E=K :: RETURN
620 DISPLAY AT(23,1)BEEP:"Device Name:">"&A$ :: ACCEPT AT(24,2)SIZE(-27):D$ ::
IF D$="" THEN RETURN
630 ON ERROR 660 :: OPEN #2:D$ :: PRINT #2:" ":" "
640 PRINT #2:" "&E$&CHR$(10):: PRINT #2:" "&" FILENAME SIZE TYPE P"
650 PRINT #2:RPT$("-",30):: FOR K=1 TO E :: PRINT #2:" "&D$(K):: NEXT K :: PRINT
#2:RPT$("-",30)&CHR$(10):: PRINT #2:" ":" ":" " :: CLOSE #2 :: RETURN
660 K$=" [[ invalid device name [[ " :: GOSUB 670 :: RETURN 620
670 FOR D=1 TO 4 :: CALL SOUND(500,1880,15):: DISPLAY AT(24,1):K$ :: CALL SOUND(
500,440,30):: DISPLAY AT(24,1)SIZE(28):" " :: NEXT D :: RETURN
680 H$=STR$(D):: N=LEN(H$):: IF N=1 THEN H$="00"&H$ ELSE IF N=2 THEN H$="0"&H$
690 RETURN
700 D$=SEG$(D$(D),20,7):: IF D$="Program" OR D$="I/V:254" THEN 710 ELSE K$=" [[
cannot run that file [[ " :: GOSUB 670 :: RETURN
710 CALL PEEK(-31952,A,B):: CALL PEEK(A*256+B-65534,A,B):: D=A*256+B-65534 :: E$
=F$&B$(C):: CALL LOAD(D,LEN(E$))
720 FOR K=1 TO LEN(E$):: CALL LOAD(D+K,ASC(SEG$(E$,K,1))):NEXT K :: CALL LOAD(
D+K,0)
730 !@P+
740 CALL CLEAR :: CALL CHARSET :: CALL SCREEN(8)
750 DISPLAY AT(6,9):"Disk Indexer" :: CALL HCHAR(4,9,95,16):: CALL VCHAR(5,25,95
,3):: CALL HCHAR(8,8,95,18):: CALL VCHAR(5,8,95,3)
760 DISPLAY AT(13,7):"Loading: "&B$(C):TAB(10):"Size: "&SEG$(D$(D),15,3):: DIS
PLAY AT(24,1):"one moment please..."
770 RUN "DSKX.1234567890"

```



TIGERCUB
SOFTWARE

TIGERCUB TIPS #12

COPYRIGHT 1984 TIGERCUB SOFTWARE
156 COLLINGWOOD AVE., COLUMBUS OHIO
43213

DISTRIBUTED TO USERS' GROUPS FOR
PROMOTIONAL PURPOSES, AND MAY BE RE-
PRINTED WITH CREDIT TO TIGERCUB
SOFTWARE.

TIGERCUB SOFTWARE SPECIALIZES IN
LOW-COST QUALITY SOFTWARE FOR THE
TI-99/4A COMPUTER. I HAVE OVER 130
ABSOLUTELY ORIGINAL PROGRAMS FOR
ONLY \$3.00 EACH, CATALOG \$1.00, RE-
FUNDED ON YOUR FIRST ORDER.

NEW PROGRAMS AVAILABLE THIS MONTH
ARE WHITE KNIGHT, A FUN GAME FOR THE
KIDS AVAILABLE IN BASIC OR XBASIC,
AND BARS AND BALLS, A STRATEGY GAME.

IF YOU HAVE TAKEN A COURSE IN COMPU-
TER PROGRAMMING, ONE OF YOUR HOMEWORK
ASSIGNMENTS WAS PROBABLY TO WRITE A
PROGRAM THAT WOULD FIND ALL THE POSS-
IBLE COMBINATIONS OF LETTERS IN A 5-
LETTER WORD. THE FOLLOWING VERSION
CAN HANDLE WORDS OF 3 TO 6 LETTERS,
LISTS THE COMBINATIONS ALPHABETICALLY,
ELIMINATES DUPLICATES (WHEN THE WORD
HAS TWO OF THE SAME LETTER), DOES NOT
REQUIRE A DIM STATEMENT, AND IS FAST.
IT ALSO WORKS WITH NUMBERS. IF YOU
WORK THOSE SCRAMBLED-WORD PUZZLES IN
THE NEWSPAPERS, YOU'LL FIND IT HANDY.

```
100 CALL CLEAR :: PRINT TAB(
5);"TIGERCUB ANAGRAMMER": :!
BY JIM PETERSON
110 INPUT "TYPE A 3-,4-,5- O
R 6-LETTER WORD ":A$: W=L
EN(A$):: IF (W<3)+(W>6)THEN
110
120 PRINT :: FOR J=1 TO W ::
B$(J)=SEG$(A$,J,1):: NEXT J
:: FOR J=2 TO W :: IF B$(J)
>B$(J-1)THEN 160
130 T$=B$(J):: FOR L=J-1 TO
1 STEP -1 :: B$(L+1)=B$(L)
140 IF B$(L-1)>T$ THEN 150
:: B$(L)=T$ :: GOTO 160
150 NEXT L
160 NEXT J
```

```
170 FOR A=1 TO W :: FOR B=1
TO W :: IF B=A THEN 340
180 FOR C=1 TO W :: IF (C=A)
+(C=B)THEN 330
190 IF W=3 THEN 250
200 FOR D=1 TO W :: IF (D=A)
+(D=B)+(D=C)THEN 320
210 IF W=4 THEN 260
220 FOR E=1 TO W :: IF (E=A)
+(E=B)+(E=C)+(E=D)THEN 310
230 IF W=5 THEN 270
240 FOR F=1 TO W :: IF (F=A)
+(F=B)+(F=C)+(F=D)+(F=E)THEN
300 ELSE 280
250 W$=B$(A)&B$(B)&B$(C):: I
F W$<=V$ THEN 330 ELSE 290
260 W$=B$(A)&B$(B)&B$(C)&B$(
D):: IF W$<=V$ THEN 320 ELSE
290
270 W$=B$(A)&B$(B)&B$(C)&B$(
D)&B$(E):: IF W$<=V$ THEN 31
0 ELSE 290
280 W$=B$(A)&B$(B)&B$(C)&B$(
D)&B$(E)&B$(F):: IF W$<=V$ T
HEN 310
290 PRINT W$&" " :: G=G+1 ::
V$=W$ :: ON W-2 GOTO 330,32
0,310,300
300 NEXT F
310 NEXT E
320 NEXT D
330 NEXT C
340 NEXT B
350 NEXT A
360 PRINT : : " ";G;"TOTAL C
OMBINATIONS." : : : G=0 :: V
$="" :: GOTO 110
```

AND STILL ANOTHER AUTOMATIC MUS-
IC MAKER. THIS ONE DOODLES AROUND
THE KEYBOARD IN THE KEY OF A, WITH
AUTOMATIC BASS ACCOMPANIMENT.

```
100 RANDOMIZE
110 DIM N(30)
120 F=220
130 FOR J=0 TO 36
140 X=X+1+(X=12)*12
150 IF (X=2)+(X=5)+(X=7)+(X=
10)+(X=12)THEN 180
160 Y=Y+1
170 N(Y)=INT(F*1.059463094^J
)
180 NEXT J
190 K=8
200 K=K-INT(5*RND+1)+INT(5*R
ND+1)+(K>21)*2-(K<1)*2
210 IF (K<1)+(K>21)THEN 200
220 CALL SOUND(-999,N(K),0,N
(K)*2,0,N(K)*3.75,30,-4,5)
230 GOTO 200
```

```

100 CALL CLEAR
110 REM - PROGRAMMED BY JIM
PETERSON MAY 20, 1984
120 PRINT "TIGERCUB MAGIC SQ
UARE MAKER": ; " A MAGIC SQUA
RE IS A CONSECUTIVE SERI
ES OF NUMBERS": "ARRANGED IN
A SQUARE IN SUCH"
130 PRINT "A WAY THAT EACH H
ORIZONTAL": "ROW, VERTICAL RO
W, AND LONG": "DIAGONAL ROW W
ILL ADD UP TO": "THE SAME TOT
AL.": ;
140 PRINT " THIS LITTLE PROG
RAM WILL": "CREATE AN ODD-ORD
ER MAGIC": "SQUARE OF ANY DES
IRED SIZE,": "STARTING WITH A
NY DESIRED": "NUMBER.": ;
150 PRINT " SQUARES OF 3,5,7
OR 9 SIZE": "WILL BE PRINTED
ON THE": "SCREEN. THE PROGRA
M CAN BE": "MODIFIED TO OUTPUT
LARGER"
160 PRINT "SIZES TO A PRINTE
R.": ;
170 INPUT "SIZE OF SQUARE?(O
DD NUMBER) ":S
180 IF (S<3)+(S/2=INT(S/2))T
HEN 170
190 INPUT "STARTING NUMBER?
":SN
200 N=SN-1
210 CALL CLEAR
220 DIM G(31,31)
230 R=1
240 C=INT(S/2)+1
250 N=N+1
260 IF N=S^2+SN THEN 450
270 G(R,C)=N
280 IF (R-1=0)+(C+1>S)THEN 3
50
290 IF G(R-1,C+1)<>0 THEN 33
0
300 R=R-1
310 C=C+1
320 GOTO 250
330 R=R+1
340 GOTO 250
350 IF (R=1)*(C=S)THEN 400
360 IF (R>1)*(C=S)THEN 420
370 R=S
380 C=C+1
390 GOTO 250
400 R=2
410 GOTO 250
420 R=R-1
430 C=1
440 GOTO 250

```

```

450 IF (LEN(STR$(SN+S^2))+1)
*S>28 THEN 530
460 FOR R=1 TO S
470 FOR C=1 TO S
480 PRINT STR$(G(R,C));" ";
490 NEXT C
500 PRINT : ;
510 NEXT R
520 GOTO 550
530 PRINT "TOO LARGE FOR SCR
EEN."
540 REM - ADD PRINTER ROUTIN
E HERE
550 PRINT : ; "PRESS ANY KEY
TO CHECK"
560 CALL KEY(0,K,ST)
570 IF ST=0 THEN 560
580 FOR R=1 TO S
590 FOR C=1 TO S
600 X=X+G(R,C)
610 NEXT C
620 PRINT "ROW #":STR$(R);"
=";X
630 X=0
640 NEXT R
650 FOR C=1 TO S
660 FOR R=1 TO S
670 X=X+G(R,C)
680 NEXT R
690 PRINT "COLUMN #":STR$(C)
;"=";X
700 X=0
710 NEXT C
720 R=1
730 C=1
740 FOR J=1 TO S
750 X=X+G(R,C)
760 R=R+1
770 C=C+1
780 NEXT J
790 PRINT "RIGHT DIAGONAL=";
X
800 X=0
810 R=1
820 C=S
830 FOR J=1 TO S
840 X=X+G(R,C)
850 R=R+1
860 C=C-1
870 NEXT J
880 PRINT "LEFT DIAGONAL=";X
890 END

```

ALMOST OUT OF MEMORY, SO
HAPPY HACKIN'

JIM PETERSON

ADDING A SECOND DISK DRIVE

A Product review of the CompuAdd Tandon TM 100-2 DSDD
Disk Drive combined the Tex-Comp Disk Drive Kit

By: Dave Manion

Sure, having a Disk Drive was a quantum leap over the cassette storage media for the TI-99, but only 90K of storage...? I found that I was filling Diskettes quickly, and having to jerk them in and out of the drive when copying data or programs and when working with TI-WRITER and MULTIPLAN. Paying for these numerous Diskettes was no pleasure either. In all, the Disk Drive only made my wait a bit shorter. What I really needed was the fast response of a Disk Drive with more storage capability.

At first glance, adding a second Disk Drive wasn't going to be easy. Most other "Dual" Drive owners that I knew were using two slim (Half Height) Disk Drives mounted in the Expansion Box. Now, this certainly looked more attractive, but from a Financial perspective (for me), it wasn't all that wonderful. Besides, what would I do with my old TI PHP1250 Disk Drive...Throw it away? All I really wanted was additional (fast) storage. ...Then came the long-awaited "Home Computer Magazine."

Tex-Comp offered a stand-alone Disk Drive Kit ((Chassis) Volume 4 No. 1 (page 174)) for \$99.95, to allow TI users to add up to TWO additional Drives (If you wanted to purchase Half-height Drives... The Disk Drive Chassis is equipped with connectors for two drives...). All you have to do is slip in either one Full-height Drive or two Half-height drives to begin operating with increased storage capability.

I then purchased a Full-height Tandon TM 100-2 Double-Sided, Double-density Disk Drive from CompuAdd (HCM vol 4 no.1, page 80) for \$229 (\$10 off by the time I purchased it). Both, when configured together, came with absolutely everything necessary (** including documentation **) to mate the two together, connect the assembly to my TI Disk Controller card, and begin running my system with two Disk Drives.

```
*****
*
* It should be noted that the second drive in this new
* chassis should also be set-up to DEVICE #1(DSK1). the
* controller knows that it is device #1 BEYOND the Ex-
* pansion Box' Drive #1. Also, the resistor pack sup-
* plied with the chassis MUST be inserted in the new drive
* for it to work. The Tandon Disk Drive is the exact
* replacement for the drives found in the IBM PC.
*
*****
```

Adding this assembly to my system gives me 270K of on-line disk storage! This allows me to put MULTIPLAN or TI-WRITER in Drive #1, and my data files in Drive #2.. Now, when running PASCAL, I can keep the (complete) Operating System

on Drive #2 and place my Source and Object code on Drive #1. -- No more Disk Swapping -- ...it was worth the \$329.00! I am still using Single-Sided Diskettes in my new Double-Sided Drive (they work just fine!) and have DOUBLED the amount of data or programs per diskette, when I am using Drive #2. The Disk Drive Kit (Chasis) is very close in color to the Expansion Box and looks good with the system peripherals. It is easy to assemble and add to the system.

If you really want, Two Double-Sided Double-Density Half-height Drives could be added to this expansion chasis, giving the owner 3 Drives to chose when storing and retrieving programs. Adding the "rumored" CorComp DOUBLE-DENSITY Controller could then provide from 450K (as in my present system) up to 800K of on-line storage by the addition of this chasis with a single (Full) or twin (Half-height) Double-Sided Double-Density Drives. I believe that this alternative should be considered if you really enjoy your computing and want your system to grow beyond the basic TI-supplied Expansion system.

Call me at (301) 672-2254 if you have any questions about adding on with this assembly, have any trouble with the connections, or just need help getting started.

J C M D I S C O U N T P R I C E S

<p>COMPUTER TI 99/4A Console.....\$ 69 (w/purch of 3 sw) ----- SOFTWARE Ti Writer.....\$ 69 Home Fin Decision.....\$ 19 Pers Rec Keeping.....\$ 19 Household Budget.....\$ 19 Star Maze.....\$ 20 Add & Sub 1.....\$ 20 Multiplication.....\$ 20 Read fun.....\$ 20 Reading Roundup.....\$ 20 Blackjack & Poker.....\$ 15 Zero Zap.....\$ 15 Adventure.....\$ 20 Attack.....\$ 11 Facemaker.....\$ 25 Amazing.....\$ 11</p>	<p>HARDWARE TI Expansion Box.....\$189 32k Memory Crd.....\$117 Disk Drive.....\$189 RS232 card.....Call RS232 Standalone(Par).\$ 85 RS232 Standalone(Ser).\$ 95 Panasonic Mon C/BW....\$259 Cable Monitor.....\$ 15 Panasonic Recorder....\$ 49 Panasonic Printer.....\$289 Gemini 10X Printer....\$269 Cable Par.....\$ 24 Modem Volls w/cable...\$ 69 ----- SOFTWARE Alpiner.....\$ 20 Farsec.....\$ 20 Blast.....\$ 10 Hustle.....\$ 11</p>	<p>MEMORY EXPANSION SPECIAL \$499 TI Expansion Box, 32K Mem Crd, Disk drive, Disk Contr Crd. Plus FREE software: Yu/Can Business Pack Inventory Control, Order Entry, Sales Prog, Cust Mail Lst TI Software Per Fin Aid, Chk Bk, Teach Yourself Basic, Beg Basic Tutor, Teach Yourself Ev Basic ----- ACCESSORIES Ribbons Gemini,Epson....Call Paper 1.4K White.....\$ 18 Diskette Webash (10)....\$ 20 Diskette 3M (10).....\$ 23 Diskette Verbatim(10)...\$ 24</p>
--	--	--

JCM NOW CAN GET YOU VCRs and COLOR TVs AT ROCK BOTTOM DISCOUNTED PRICES.
 Call for complete product price list

Md. Residents add 5% sales tax. Shipping & Handling(add \$2.00 per software)
 Min chg \$2.00. Call for Hardware shipping chgs. Allow 2 weeks for delivery.
 Order by phone or mail: 301/972-5675
 Enclose Money Order, Cert. or personal check payable to:
 JCM Computer Products P.O. Box 537 Germantown Md 20874

I read the article by Duane Shie on uploading to mainframe computers using TE II (May 1984). But, I thought, why should one have to swap cartridges to upload text to a BBS or a mainframe? I experimented with several versions of his program and came up with the following.

```
100 OPTION BASE 1
110 DIM A$(110)
120 PRINT "* READY TO SEND FILE *"
130 INPUT "----- ":FILES$
140 OPEN #1:"DSK1."&FILES$,INPUT,SEQUENTIAL,INTERNAL VARIABLE 254
150 OPEN #2:"RS232/2.PA=E.CH.LF",OUTPUT,VARIABLE 80,DISPLAY
160 INPUT #1:X
170 CALL CLEAR
180 FOR I=1 TO X
190 INPUT #1:A$(I)
200 PRINT A$(I)
210 PRINT #2:A$(I)
220 FOR J=1 TO 100
230 IF J=100 THEN 300 ELSE 240
240 NEXT J
250 NEXT I
260 CLOSE #1
270 CLOSE #2
280 END
300 J=1
310 GOTO 250
```

The procedure for uploading text works like this.

1. Using TE II, you logon to your mainframe or BBS and go through the appropriate steps to set up the system to receive text at line one.
2. Then, before you hit enter, you push the FCTN and QUIT keys to exit to the main TI screen. Then go into TI BASIC. At this point, load and run the above program, which takes any text file created with TI WRITER or TEXSCRIBE and loads the file in one line at a time. It also displays each line on the screen for you to see it go out to the mainframe, etc. The J=100 is a delay loop to allow the mainframe editor to setup the next line number to receive the next line of text.
3. After your program has entered your text, you then type "NEW" and "BYE" to exit to the TI main screen.
4. Next, go back into the TE II menu and you are back into your editor of your mainframe.
5. Type "LIST" and you will see the text you have just sent.

The above experimental program was tried using an IBM 370 and "WYLBUR", a text editor.

I am sure this program will be improved upon to include a downloading feature. If someone would like to help, please call me after 7 pm on 631-1744 or leave a message on our BBS.

By Pete Eddy

TI 99/4A COMPUTER EQUIPMENT & ACCESSORIES

Mr. Carroll (Pete) Eddy
 4624 Dixie Hill Road
 Fairfax, VA 22030
 Ph. (703) 631-1744
 (7-10 pm)

(Partial list)

		RETAIL	SALES
Cor-Comp	Disk Controller Card	199.95	165.00
Cor-Comp	SAU 232 (Stand/ alone RS232)	149.95	135.00
Cor-Comp	RS-232 Card	99.95	88.00
Cor-Comp	32K Card	135.95	112.00
PHP2700	Program Recorder (includes cable)	49.95	40.00

Texas Instruments Software

PHM3013	Personnel Record Keeping	24.95	20.00
PHM3026	Extended BASIC	99.95	78.00
PHM3035	Terminal Emulator II	24.95	20.00
PHM3043	Reading Fun	39.95	32.00
PHM3047	Reading Roundup	39.95	32.00
PHM3048	Reading Rally	39.95	32.00
PHM3058	Mini-Memory	99.95	78.00
PHM3064	Touch Typing Tutor	39.95	32.00
PHM3109	TI LOGO II	99.95	78.00
PHM3111	TI Writer	99.95	78.00
PHM3113	Microsoft Multiplan	99.95	78.00

***** SPECIAL DEAL WHILE THEY LAST *****

TI EXPANSION BOX, DISK CONTROLLER CARD, DISK DRIVE, AND 32K CARD

ALL FOR JUST \$390.00

ORDER TODAY BEFORE THIS DEAL ENDS (ALL ORIGINAL TI EQUIPMENT)

Amdex	13" Color I Plus (with cable)		290.00	
Panasonic	13" RGB Color Monitor DT-D1300D w/cable		350.00	
Percom	Disk Controller/Disk Drive		350.00	
Lexicon	Lex-12 Direct/Acoustic Modem w/cable		85.00	
Anchor	Volksmodem		70.00	NEW PRICE
Axiom	GP-550TI Printer w/cable	399.00	356.00	NEW ITEM
Axiom	GP-100TI Printer w/cable	309.00	275.00	NEW ITEM
Axiom	Parallax TI Expansion Interface	99.00	87.00	NEW ITEM
Gemini	10X Printer	399.00	290.00	
Gemini	15X Printer	459.00	420.00	
Okidata	82a Printer		330.00	
Okidata	83a Printer		550.00	
Okidata	M92 Printer/cable		460.00	
Okidata	M93 Printer		710.00	
Panasonic	KX-P1090 Printer w/cable		335.00	
Panasonic	KX-P1091 Printer w/cable		350.00	NEW ITEM
Flip n File	Diskette Holder (holds 50)	26.95	20.00	
WICO	Joystick & Adapter		28.00	NEW PRICE
Compac	TI PAR, Printer cable, 5 ft.	28.95	24.00	NEW ITEM
Compac	TI PAR, Printer cable, 10 ft.	32.95	28.00	NEW ITEM

Quick Address Reference List

by D. Bobbitt

Ever since the Smart Programming Guide to Sprites from Miller's Graphics came out, I have sort of collected a list of interesting and useful addresses. Some of these, obtained from books, magazines, and locl, are not too useful, but have been included because they may well be useful, but their use hasn't been discovered yet. Anyway, Craig Miller deserves the credit for finding most of these, but quite a few are from other sources.

This list is by no means complete. Much more competent people than I are finding new ones weekly (if not daily). It has become an interesting challenge to some, like probing a dark cave with a long pole looking for treasure (or some equally inane analogy). For the few of you that have the COMPLETE version of this list, you may notice that several addresses have been left out because they are either potentially destructive, or are unfair to the hundreds of 3rd party-manufacturers, on whose support we all depend. I am ashamed to say that many people now make a living making devices and programs to prevent software piracy. Anyway, here's the list. I hope it is as useful to all of you as it has been to me.

PEEKs and LOADs

CALL PEEK(-28672,X) - BFOHRD Equate
if X=0 then the speech synthesizer isn't attached
if X=96 then the speech synthesizer is attached.

CALL PEEK(-31899,X)
X6= ... the amount of memory used

CALL LOAD(-31898,63,255)
Shuts off the disk drive and free's buffer RAM.

CALL LOAD(-31886,55,215)
Turns on disk drive and reopens buffer.

CALL PEEK(-31880,X)
Random number generator from 0 to 99.

CALL PEEK(-31879,X) - VDP Interrupt timer
Counts from 1 to 100 every 4.25 seconds.

CALL LOAD(-31878,0)
Halts all sprite motion instantly.

CALL PEEK(-31897,X) - VDP Status register
Contains screen status:
BIT 0 = the 60HZ Interrupt (VDP)
BIT 1 = (64) if more than 4 sprites are in a row.
BIT 2 = (32) if there is a sprite coincidence.
BITS 3-7 = Contain the hex number of the fifth sprite in a row when
bit 1 is on.

CALL PEEK(-31808,X,Y)
A double random number generator, from 0 to 255.

CALL LOAD(-31806,000) - Interrupt Flag
 128 = all below off
 64 = stops all sprite motion.
 32 = disables sound chip.
 16 = disables <QUIT> key.
 0 = enables all above.
 48 = disables sound chip and <QUIT> key.
 20 = stops sprite motion and disables <QUIT> key.
 96 = stops sprite motion and disables sound chip.

CALL LOAD(-33730,33.
 (-31804,0,36)
 Quits from Extended BASIC back to master title screen.

CALL LOAD(-31923,255)
 Causes a system reset in Extended BASIC.

CALL LOAD(-31748,X): X=ANY NUMBER
 This will cause an increase in the rate the cursor blinks as well as a change in the speed of the sound chip.

CALL LOAD(-31866,X): X=any number between 0 and 159
 Contrary to the Article published in the June issue of Micropendium, this address DOES NOT allow you to access the full 32K in the memory expansion, it just makes you think you have it all. I believe it simply alters the data that the SIZE statement reads. To prove this simply try it in a program (it doesn't work), or try it and then load a program (the memory size will be normal, minus the memory the program takes). It does trick the computer into allowing you to allocate larger DIMs though, whatever use that serves.

CALL LOAD(-31744,X): Sound Equate (>2400)
 (-30721,X)
 (-30700,X)

This particular address, simply put, allows you to turn on the sound chip and leave it "running", while the computer performs other tasks. This can be demonstrated with the CALL SOUND statement by simply calling a sound with a duration greater than 1000 or so. The computer will continue giving the sound yet the cursor (if done in the immediate mode) will be blinking away. The sound initiated by using this address, as explained later, will continue until it is turned off by one of three actions. (1) It is turned off by loading a particular number into this address, (2) Another sound is called using a CALL SOUND statement, or (3), it is used in program and the program is BREAKed (broken?) while running. The only drawback to using this address is that only 4 different sounds can be created, with up to 4 going at once. Before any sounds at all can be made, though, the address must 1st be "initialized". To initialize the address, you must first understand how the data effects. Each note has a particular range of numbers which correspond to the 15 levels of volume that note can be turned on as. These are as follows:

1-15 = The "BREAK" sound
 144-159 = A higher sound
 176-191 = A still higher sound
 240-255 = A noise (similar to -5)

In each of these cases, the lower the number, the greater the volume, and vice-versa. Before any of these sounds can be turned on a number corresponding to the number before the highest volume of the sound to be turned on must be LOADED in (IE, to turn on the noise LOAD in a 239 first.

then in a separate CALL LOAD statement LOAD in the number corresponding to the volume desired of the noise). This address allows multiple sounds of up to 3 notes and 1 sound (that's all you have anyway) to be created. Simply load in the first number after the address has been initialized for that note, but DON'T initialize it for the preceding notes. Each sound must be LOADED in using a separate CALL LOAD statement. To turn off the sounds use either of the last two methods above, or if in a program, simply LOAD in a number corresponding to the number after the lowest volume of the note used (IE. for the example, LOAD in a 256). These numbers given are by no means the only numbers available to make these sounds, but there are only 4 sounds available. (For an interesting effect type in the following mini-program in immediate mode: CALL INIT :: FOR I=1 TO 10000 :: CALL LOAD(-31744.I) :: NEXT I). This load may not work on all consoles, but it has worked on the three I have tested it on, including an old "4".

1. Sprites are user-defined graphics, which are able to appear anywhere on the screen, and move fast and smoothly in high resolution. Once set in motion, they move continuously without any additional programming effort until they are deleted or until the program terminates.
2. Motion is initiated by the user by simply selecting the starting point of the sprite on the screen (row -X- and column -Y- coordinates) and by specifying the direction of travel, as well as the speed (velocity) by which the sprite will move.
3. The electronic circuit, responsible for sprite generation, is called "TMS9918A Video Display Processor (VDP)". Unlike other computers, the VDP allows the creation of any possible pattern by the user through a relatively simple process. It allows 28 sprites to be present on the screen, and monitors all of their attributes constantly, such as direction, coincidence, etc.
4. The VDP is also capable of accepting external video (such as inputs from a VCR or a VRP) and produce "mixed" sprites (sometimes called Movable Object Blocks - MOB's). The TI99/4A has no provision for such applications at the present time, but the potential for educational and game applications are great, and a number of "hackers" and "third party" manufacturers and programmers are working seriously in that direction. This procedure of external video input is called "simultaneous video mixing."
5. The VDP considers the screen as a 32-level (plane) display, and numbers these planes 0-31. The first two of these planes are the "multi-color and patterns" and the "backdrop." The first is used for textual and fixed graphic images, while the second is slightly larger, forming a rectangular frame around the other elements of the display.
6. Sprites can be generated on any of these planes (28 maximum). Although there may be a number of sprites generated, only 4 can be visible on the same horizontal row at the same time. The others are there, and will become gradually visible as one of the "visible" sprites starts moving away from that row. If this attribute of the VDP is programmed properly, it can create "shadows," as well as many other "real life" images.

7. The TMS9918A VDP operates in 4 modes:

a. Pattern Mode (Graphics 1)

In this mode, the screen is divided into a 24-row (X-coordinate) by a 32-column (Y-coordinate) grid. Each grid square is an 8 x 8 matrix of dots called "pixels," and each matrix can have any 2 of the standard 15 colors (the background color for sprites is always 1 - transparent). When 2 sprites move into the same screen coordinates, the sprite on the "lower" plane will progressively cover the sprite on the "higher" plane, thus creating the illusion of passing in front of it, generating a three-dimensional effect.

b. Bit Map Mode (Graphics 2)

In this mode, each of the 8 horizontal rows within the 8 x 8 matrix can have 2 unique colors.

c. Text Mode

In this mode, the screen is a 2-color single plane, with no sprites allowed. There are 24 rows by 40 columns of 8 x 8 grid matrices (squares). ASCII characters are formed in a 5 x 7 grid, with 2 pixels between rows and characters.

d. Multicolor Mode

This mode divides the pattern plane into an unrestricted 48-row by 64-column color square display, comprised of 4 x 4 pixel grids. Each of these grids is allowed to take any of the 15 standard colors, or be made transparent.

8. Sprites are available only under Extended Basic or Assembly Language. The latter is, by far, more versatile. The sprites can move a lot faster with smaller memory requirements.

B. SPRITE GENERATION

1. Each character is an 8 x 8 pattern stored in memory (64 pixels). A sprite is just a character with added capabilities. It can be generated as a single 8 x 8 pixel character, or it can be made larger in 3 different ways:

a. A 16 x 16 pixel pattern, called "double-sized unmagnified."

b. By magnifying the existing sprite by a factor of 4.

c. By utilizing both the above methods combined, to produce a sprite 16 times its normal size.

2. This variable size feature allows the sprites to "grow" or "shrink" at will, with minimum programming effort and great display effects.

3. Extended Basic provides a large number of sub-programs that directly or indirectly affect sprites. Following is a short description of the function of these sub-programs. A programmer should study them at length, and experiment with the short programs provided in the Extended Basic Manual, before attempting any serious programming applications.
 - a. CALL CHAR used to form (plot) any standard (character) or non-standard (graphic) pattern.
 - b. CALL CHARPAT returns the hexadecimal code that represents the character pattern.
 - c. CALL CHARSET restores the standard character pattern to the character code that was redefined.
 - d. CALL CLEAR clears the screen from all characters/patterns.
 - e. CALL COINC returns a value of -1 if there is a coincidence between 2 or more sprites.
 - f. CALL COLOR assigns the desired color(s) to the specified character group.
 - g. CALL DELSPRITE removes one or more sprites from the screen.
 - h. CALL LOCATE moves the specified sprite to a new set of coordinates.
 - i. CALL MAGNIFY sets the size and magnification factor of the sprite(s).
 - j. CALL MOTION changes the motion of the sprite(s).
 - k. CALL PATTERN changes the pattern of a sprite to a pattern of another sprite (or character). If the sprites involved have similar patterns, this is a great way to produce the illusion of "secondary animation" like turning wheels, etc.
 - l. CALL POSITION returns the screen coordinates that a sprite currently occupies.
 - m. CALL SCREEN sets the screen background color.
 - n. CALL SPRITE specifies the existence of the sprite(s) and assigns characteristics.

THIS ARTICLE
SUBMITTED BY

Nick C. Santorineos
WDCTIUG
Montgomery County Chapter

4. The CALL SPRITE sub-program has the following attributes:

CALL SPRITE(#NUM,ASCII,COLOR,HOR,VER,H-VEL,V-VEL)

- a. #NUM - This is a numeric expression (1-28) that defines the
---- sprite number. It can be in the form of a number or
a variable. If #NUM was previously assigned to
another sprite, the old sprite is deleted and re-
placed by the new one. The parameters specifying the
new sprite may be omitted, in which case, the para-
meters for the old sprite remain in effect, contribu-
ting to the easiness of "secondary animation motion"
generation. The #NUM establishes the plane of the
sprite as it appears on the screen among others. If
two sprites move into the same screen location, the
sprite with the "lower" #NUM will cover all or a part
of the sprite with the "higher" #NUM, thus creating
the illusion that it passes in front of it. Similar-
ly, when more than four sprites are generated on the
same row, the ones with the "higher" #NUM will grad-
ually disappear.
- b. ASCII - This is an interger value, 32-143, representing the
----- bit pattern assigned to the sprite. This pattern was
previously generated with the CALL CHAR sub-program,
and can be changed with the CALL PATTERN. The sprite
is defined as the single character code given unless
the CALL MAGNIFY is used, in which case it will also
occupy the next 3 characters.
- c. COLOR - May be a numeric expression 1-16, denoting the fore-
----- ground color of the sprite.
- d. HOR - Specifies the dot of the horizontal plane that the
--- sprite will appear on the screen. The dots are num-
bered from 1 (upper left screen corner) to 192 (lower
right screen corner). HOR could actually be defined
1-256, but positions 193-256 are off the bottom of the
screen. The position of the sprite is the upper left
corner of the character that defines it. Many sub-
programs affect sprite positioning, and should be ref-
erenced individually.
- e. VER - Specifies the dot of the vertical plane of the sreen.
--- Function is identical with HOR. Dots are numbered
1-256, with 1 being the upper left screen corner.
- f. H-VEL - Optional entry. If used, it specifies the velocity
----- of sprite horizontal motion, or if combined with the
V-VEL, diagonal motion. Values range from -128 to
127. The values closer to 0 are the slowest, and
increase as they go higher. A positive value moves
the sprite DOWN, while a negative value moves it UP.
Velocity may be changed with the CALL MOTION sub-
program.

CONT ON PAGE 23

3. Extended Basic provides a large number of sub-programs that directly or indirectly affect sprites. Following is a short description of the function of these sub-programs. A programmer should study them at length, and experiment with the short programs provided in the Extended Basic Manual, before attempting any serious programming applications.
 - a. CALL CHAR used to form (plot) any standard (character) or non-standard (graphic) pattern.
 - b. CALL CHARPAT returns the hexadecimal code that represents the character pattern.
 - c. CALL CHARSET restores the standard character pattern to the character code that was redefined.
 - d. CALL CLEAR clears the screen from all characters/patterns.
 - e. CALL COINC returns a value of -1 if there is a coincidence between 2 or more sprites.
 - f. CALL COLOR assigns the desired color(s) to the specified character group.
 - g. CALL DELSPRITE removes one or more sprites from the screen.
 - h. CALL LOCATE moves the specified sprite to a new set of coordinates.
 - i. CALL MAGNIFY sets the size and magnification factor of the sprite(s).
 - j. CALL MOTION changes the motion of the sprite(s).
 - k. CALL PATTERN changes the pattern of a sprite to a pattern of another sprite (or character). If the sprites involved have similar patterns, this is a great way to produce the illusion of "secondary animation" like turning wheels, etc.
 - l. CALL POSITION returns the screen coordinates that a sprite currently occupies.
 - m. CALL SCREEN sets the screen background color.
 - n. CALL SPRITE specifies the existence of the sprite(s) and assigns characteristics.

THIS ARTICLE
SUBMITTED BY

Nick C. Santorineos
WDCTIUG
Montgomery County Chapter

4. The CALL SPRITE sub-program has the following attributes:

```
CALL SPRITE(#NUM,ASCII,COLOR,HOR,VER,H-VEL,V-VEL)
```

- a. #NUM - This is a numeric expression (1-28) that defines the
---- sprite number. It can be in the form of a number or
a variable. If #NUM was previously assigned to
another sprite, the old sprite is deleted and re-
placed by the new one. The parameters specifying the
new sprite may be omitted, in which case, the para-
meters for the old sprite remain in effect, contribu-
ting to the easiness of "secondary animation motion"
generation. The #NUM establishes the plane of the
sprite as it appears on the screen among others. If
two sprites move into the same screen location, the
sprite with the "lower" #NUM will cover all or a part
of the sprite with the "higher" #NUM, thus creating
the illusion that it passes in front of it. Similar-
ly, when more than four sprites are generated on the
same row, the ones with the "higher" #NUM will grad-
ually disappear.
- b. ASCII - This is an interger value, 32-143, representing the
----- bit pattern assigned to the sprite. This pattern was
previously generated with the CALL CHAR sub-program,
and can be changed with the CALL PATTERN. The sprite
is defined as the single character code given unless
the CALL MAGNIFY is used, in which case it will also
occupy the next 3 characters.
- c. COLOR - May be a numeric expression 1-16, denoting the fore-
----- ground color of the sprite.
- d. HOR - Specifies the dot of the horizontal plane that the
--- sprite will appear on the screen. The dots are num-
bered from 1 (upper left screen corner) to 192 (lower
right screen corner). HOR could actually be defined
1-256, but positions 193-256 are off the bottom of the
screen. The position of the sprite is the upper left
corner of the character that defines it. Many sub-
programs affect sprite positioning, and should be ref-
erenced individually.
- e. VER - Specifies the dot of the vertical plane of the sreen.
--- Function is identical with HOR. Dots are numbered
1-256, with 1 being the upper left screen corner.
- f. H-VEL - Optional entry. If used, it specifies the velocity
----- of sprite horizontal motion, or if combined with the
V-VEL, diagonal motion. Values range from -128 to
127. The values closer to 0 are the slowest, and
increase as they go higher. A positive value moves
the sprite DOWN, while a negative value moves it UP.
Velocity may be changed with the CALL MOTION sub-
program.

CONT ON PAGE 23

THE FASTEST SORT MAY NOT BE A SORT
by D.L.Shie, Asst. Prof., UDC

This article and program again addresses the problem of keeping track of one's growing library of disk files. The problem is keeping track of them all and locating them. An index file of some sort is the thing. Maybe in alphabetical order by name. Many programs and articles have addressed this. I wrote a crude program using a bubble sort, then discovered Larry Hughes' Library donation in XBASIC on Tape#15, a forerunner of his assembler version SUPER CATALOGER.

My aim was to have a custom alphabetical file of all my disk file names with their disks that I could view with TI-WRITER and even upload to the UDC system so I could have it printed and pick it up at work the next day. So modifying Larry's XBASIC version, I was dismayed that even its Shell sort took about 10 minutes and I had more than 300 files with all my own stuff plus our user's group collection so I couldn't even do it all. Timewise I see what all the excitement is about his Assembly version.

But what bothered me about SORTING the directories read from the disks is that the built-in ROM routine in the Disk Controller ALREADY has the individual disk directory sorted and stored that way on our disk. Copying them in one disk at a time and appending them to the huge file immediately undoes all the work the controller already has done.

So how about another algorithm entirely that preserves the work already done: 1. Read in the first list, it is sorted. 2. Then read in the next disk's list, and MERGE it with the previous so that we again have one sorted list. Then repeat 2. till either done or all memory space is used up. Ideally this one pass merge would finish while you wait for the red light to go out on the drive and switch disks. To speed up the XBASIC further I used a structure called a LINKED LIST ... with a set of subscript pointers used to point to the next one alphabetically, the computer never needs to physically interchange name strings.

After days of experimenting to exceed 300 filenames, blowing storage first in my 12 K console stack when I made all but my one set of subscripts into strings; then I blew the 24 K program space with too many numeric variables and subscripts; finally, I make a viable compromise of switching a few of the numbers to short strings, then it can handle 500 program names quickly with the rest of the information I wanted to see with TI Writer with 40 columns across. Only near the end can I beat the merge time with my disk switch time, and that difference is never long. And at the end, I got the whole print#1 loop on one numbered line, giving it the speed to write my disk file as fast as the drive can operate. I also built several more error traps after losing several half hours of disk switching time when loading a bad disk and even finding garbage on my disk's directory and branching around that.

```

100 ! XTCART2JUL   Tape#15           FROM VS 6/11
110 DIM N$(500),S(500),T$(500),DK(500),P(500),TYPE$(5),D$(99),FS(99)
120 CALL INIT :: CALL LOAD(-31878,0):: CALL SCREEN(15)
130 ON ERROR 680
140 N$(0),D$(0)="!!!!!!!!!" :: N$(1),D$(1)="~~~~~" :: P(0)=1 :: P(1)=0 ::
OC=1 :: DK(0)=0 :: DK(1)=1 !init the linked head & tail
150 PRINT :,,:" Modification of total cat program by L.Hughes on group tape #1
5 mod by D.Shie      makes file for TI-WRITER"
160 PRINT :,:"IF WANT TO DO OVER 400 FILES FIRST DO: CALL FILES(1)::NEW...THEN LC
AD AND RUN PROGRAM"
170 INPUT "Do you want to catalog the LOAD files?(Y/N):":LOAD$ :: IF LOAD$<>"Y"
AND LOAD$<>"y" AND LOAD$<>"N" AND LOAD$<>"n" THEN 170
180 PRINT :,:"PLACE FIRST DISK INTO DSK1": " THEN PRESS ENTER TO START"
190 CALL KEY(0,K,ST):: IF ST=0 THEN 190
200 TYPE$(J)="DF" :: TYPE$(2)="DF" :: TYPE$(3)="IF" :: TYPE$(4)="IV" :: TYPE$(T)
="P"
210 D=1 ! initialize disk ct      return below each disk:
220 D=D+1 :: OPEN #2:"DSK1.",INPUT ,RELATIVE,INTERNAL
230 INPUT #2:D$(D),J,J,FS(D)
240 PRINT :,,D$(D); " AT";J-FS(D); " FREE";FS(D)
250 C,N=OC+1
260 ON ERROR 340 !      start of file read loop
270 INPUT #2:N$(C),I,S(C),RL
280 IF LEN(N$(C))=0 THEN P(C-1)=0 :: C=C-1 :: GOTO 350 ! last read last file on
disk
290 IF N$(C)="LOAD" AND (LOAD$="N" OR LOAD$="n") THEN 270 ! skips cataloging load
300 PRINT C;TAB(6);N$(C);TAB(17);S(C);TAB(21);TYPE$(ABS(T));RL
310 T$(C)=TYPE$(ABS(T)):: DK(C)=D !point to dskname
320 IF RL>0 THEN K=INT(RL/100):: J=INT(RL/10-10*K):: T$(C)=T$(C)&CHR$(K+48)&CHR$
(J+48)&CHR$(INT(RL-100*K-10*J)+48)
330 IF C>499.5 THEN 350 ELSE P(C)=C+1 :: C=C+1 :: GOTO 260 !bottom file read loc
p
340 CALL ERR(E1,E2,E3,E4):: PRINT "CATERR";E1;"SKIP:";A$;"..ERR LINE: ";E4 :: C=
C-1 :: RETURN 330 !...skip file with bad catalog entry
350 ON ERROR 680 :: CLOSE #2
360 PRINT :,,C;" FILENAMES (MAX.=500)"
370 IF C<500.5 THEN 410
380 PRINT "500 FILENAMES NOW TO SORT": :
390 C=500 :: P(500)=0 :: GOTO 420
400 ON ERROR 680
410 PRINT :,:"REMOVE DISK": : " INSERT NEXT DISK IF ANY":,,
420 PRINT "MERGING..."
430 !SORTING ROUTINE:(MERGE)
440 IF C>OC THEN O=0 :: L=P(0)ELSE 500 !jump if disk empty
450 IF N$(L)<N$(N) THEN O=L :: L=P(L):: GOTO 450 ! check new against next old
460 IF N$(L)=N$(N) AND D$(DK(L))<=D$(DK(N)) THEN O=L :: L=P(L):: GOTO 450 ! again
ok by disk name, as in 562
470 !here if new name insert
480 IF P(N)<>0 THEN SS=P(N):: P(N)=L :: P(O)=N :: O=N :: N=SS :: GOTO 450 ! ins
ert new by pointers changed.
490 P(N)=L :: P(O)=N :: O=N :: OC=C ! inserted last new, done
500 CALL SOUND(200,900,10)
510 PRINT : " PRESS 'R' to Read next disk": " PRESS 'W' to Write totcat" :: INPUT
AN$
520 IF AN$="R" OR AN$="r" THEN 220 ELSE IF AN$<>"W" AND AN$<>"w" THEN 510
530 PRINT :,:"READY TO WRITE TOTAL CATALOG"
540 PRINT :,:" WHICH DISK AND FILE NAME?":,,
550 ACCEPT AT(23,2)SIZE(-20)BEEP:DEV$
560 PRINT :,:"ENTER": " DATE": " -----" :: ACCEPT AT(23,7)SIZE(
-18)BEEP:A$

```

MINUTES OF GENERAL MEETING OF
MAY 10, 1984 AT FAIRFAX H.S.

For a change we met in the cafeteria instead of in the auditorium. President Jim Horn began with announcements, rumors, and hints. Again Foundation is swearing their RAM disk software is nearly ready. Mike Lambert is taking names for a TI Pascal Users group with P-System interests. We had a break for talking and checking sales and trades while getting set to douse the lights for Rev. Koch's slide show. It was well into the slides till the school maintenance crew got the last light forceably unscrewed.

Rev. Keith Koch showed his 99/2 and mentioned he had obtained one of the defunct Milton Bradley MBX systems with voice recognition and was quite pleased with it and with the games Baseball and I'm Hiding. He then proceeded to show the slides he had taken and first shown to us at the December 82 meeting. (I could understand the tour better myself this time as I recently opened my oldest one up to clean the cartridge port since it had been going crazy with the X-BASIC inserted but now it works fine!) That power supply that converts the low voltage AC from the exterior transformer into such highly stable, highly filtered DC power that we don't really need the add ons that are sold for many systems. But we then get more waste heat at the eternally hot spot below the cartridge port. The power supply is on a seperate module inside the case only connected to the processor board by a four wire connector. (So I see how easy it is for the hobbieist to remove that module to a seperate box to take the major heat source outside the main case.) The keyboard component also unplugs easily from the processor board. (While I had mine out I cleaned it well with a dry toothbrush, it now looks new again!) Keith pointed out that the joystick ports can be used as OUTPUT lines by software commands and it also has external video IN capability -- he has been able to superimpose his video camera images with the TI's graphics -- with assembly language the camera image can be turned off and on and synchronized. We have an 105 K memory (RAM + ROM) machine when full blown. The design has a place for a hard disk controller and up to FOUR DS DD floppy drives (TI's controller only accesses three SD drives but CORCOMP claims their controller will do it all.) and some military micro versions with the ti9900 have a 16 M byte memory map.

Not-Polyoptics then discussed some points on FORTH with us. FORTH gives TOTAL control over every function of our TI99 ... so it is as powerful as machine or assembly, but it uses WORDS. It may be the best piece of software TI came up with and the irony is that they gave it to us FREE. He is forming a FORTH interest group to write new Forth words ... it is nice to delegate tasks and Forth does need further refinement. The club will try to keep the public domain manual in print for \$8.00 for over 100 pages. We have distributed over 150 copies so far with 50 more coming. Miller Graphics claims it can't be modified to run with the X-basic cartridge instead of the ED/Assembler cartridge. Nor has anyone yet released a version for the mini-memory cartridge. There exists a third party version on cassette tape for sale.

Regarding the cpm card being advertized, one member cautioned that it will only read the CPM programs that have been copied to TI/99 formatted disks, and so far very few programs have been so converted. programs might be downloaded via rs232 hookups to other CPM systems and then run if they contain none of the special graphics from the host system. (And if I remember right from my use of CPM several years back, the BASIC programs could be edited easily with the screen editors like Wordstar then loaded nicely into BASIC so the BASICS under CPM must NOT be tokenized like TI's BASICS which kills the easiest downloadings for us.)

One enterprising member asked whether anyone had an estimate of how many TI disk controllers had been sold so he could then deduce how many TI/99 owners had expanded to disk systems: Answer: TI NEVER will tell how many of anything it's sold!

Under our group purchase of at least 10 copies, the price is \$14 in advance for the Best of '99'er, \$5 for best of our Users Group Newsletter if more copies are made. The International 99/4 Users-Group has grown to almost 100,000 and for the time being the \$16 annual membership fee to P.O. Box 67, Bethany, OK 73008, still gets you the very nice Enthusiast '99 magazine hopefully every other month ... I got my March 84 issue on May 11 ... plus some intermediate newsletters. The word was that Vol.4 No.2 (the undated "Feb" issue) was in the mail for the second issue of the Home Computer Magazine reincarnation of the '99'er (mine did arrive about a week later).

FROM PAGE 19

```

570 OPEN #9:DEV#,OUTPUT
580 PRINT #9:TAB(8);"STATUS DISKS": "  DISK  FREE-SPACE": "-----  ----"
590 FOR I=2 TO D :: PRINT #9:D$(I);TAB(12);FS(I):: NEXT I
600 PRINT #9: ",,," "      TOTAL CATALOG": "      AS OF ";A$: :
610 PRINT #9: " #      FILENAME      DISKNAME SEC TYP/r1"
620 PRINT #9: " -----" : : I=P(0)
630 FOR J=1 TO C-1 :: PRINT #9:J;TAB(6);N$(I);TAB(17);D$(DK(I));TAB(27);S(I);TAB
(31);T$(I):: I=P(I):: NEXT J
640 PRINT #9: "-----"
650 PRINT #9:C-1;" FILENAMES"
660 CLOSE #9
670 STOP
680 CALL ERR(E1,E2,E3,E4)
690 PRINT "* ERROR";E1;" IN LINE";E4
700 RETURN 400
710 END

```

NOTICE TO MEMBERS OF THE WASHINGTON DC AREA TI HOME COMPUTER GROUP. Please check your mail label for the renewal date of your Membership. If your date is JULY, AUG or LATER in 1984 the date will be circled in RED. We are now trying to group the expiration dates so that the mail list will be easier to maintain. Your dues will be prorated. Thank you for your cooperation. If your date is circled in red please remit JUL \$12.00, to extend your membership until JULY 1985, for the labels marked later please remit amount marked in RED, to renew until JULY 1985. Remember the DUES ARE TO BE INCREASED on JULY 1, to \$16.00 so please SAVE at TODAY'S RENEWAL RATE.

INFORMATION RESOURCES

This list of members consists of individuals who have volunteered to be pioneers in developing our "infrastructure" as it might have been called in the sixties. Networking is more popular today. Our members skills are a primary resource of our clubs. If you see categories of interest not covered, please call Dave Hill, 941-6876. Resource volunteers should also call Dave with your replacement's name and telephone number. More than one resource volunteer is OK, because of driving time and long distance calling.

SUGGESTIONS ARE CHEAPER THAN WORK

Volunteers enjoy what they are doing, but, like you, have more ideas for new projects than they have people to do them. Resource volunteers have only agreed to put you in contact with others of similar interest, essentially by giving you the names and phone numbers of persons who have called with similar interests. Your help will be welcome along with your suggestions. If you choose to get in the thick of things and become a telephone Resource volunteer, you can personally gain a great deal.

PAUL PEYSER ASSEMBLY LANGUAGE
649-9870

DAVE HILL RESOURCE EDITOR
941-6876 MULTIPLAN
TI WRITER

KEN CHANDLER NOVICES
752-9774 SOFTWARE REVIEWS
BOB DURGAN

RICK STICKLE EDUCATION, CAI,
301-725-6979 LOGO, MODEMS,
TELECOMMUNICATIONS

GARRY CRADDOCK VIDEO DISK
703-494-4952 TECHNOLOGY

DICK STURGELL YOUTH GROUP
NOVICE INFORMATION

CHRIS GOODMAN CP/M
262-5570

ROBINS Inc.



**THE SMALL COMPUTER
SUPPLIES STORE
SHOP BY PHONE**

WE SHIP BY UPS

**PRICES SLASHED
ON RIBBONS!**

[B=brand C=compatible]

- Epson FX/MX/RX 80 [C] ... 3.95
- Epson MX/FX 100 [C] ... 9.95
- Prowriter/NEC 8023 [C] ... 3.95
- Diablo Hytype 2 Film [C] ... 3.50
- Diablo Hytype 2 Film [B] ... 4.25
- QUME Film [C] ... 1.90
- QUME Film [B] ... 2.80
- QUME IV Film [C] ... 3.45
- NEC 5500/7700 Film [C] ... 3.30
- NEC 3500/2000 Film [B] ... 11.90

RELOAD YOUR OWN!

- Diablo, QUME, NEC 7700
Film ... 1.50
- Epson 80, NEC 8023,
Itoh ... 1.95
- Epson 100 ... 3.95

- PAPER**
- Clean Edge 3000 Shts ... ~~28.95~~
 - Labels 3-1/2x15/16 5k ... 16.00

Hours: Monday thru Friday 9-5, Saturday 10-2

6 Outlet Surge and Spike
Suppressor ... 54.95

PRINT WHEELS

- Diablo Plastic ... \$6.40
- Diablo Metal ... \$41.50/49.95
- Diablo 620 ... \$11.40
- QUME ... \$6.40/7.90
- Smith-Corona TP-1 ... \$4.90
- Olivetti ... \$15.50
- Olympia EC ... \$26.50
- Brother ... \$24.95
- Daisywheel II ... \$27.50
- Head Cleaning Kit ... \$15.00

FLOPPY DISKS

- 5-1/4 ss/dd
- Sentinel ... \$22.50
- Kyba, 3M, Verbatim ... \$26.50
- Brown ds/dd ... \$29.95
- Brown ds/(96 tpi) ... \$39.95

**Come Visit Us—Just
3 Minutes off the Beltway**

BULK DISKS \$1.50

560-5900

8304 Hilltop Road

Fairfax, Virginia 22031

(Merrifield Area)

MIKE LAMBERT
920-3955

PASCAL
"C"
FORTH, GPL, PILOT

RON BURKTON
703-368-7187

HARDWARE HACKERS

DAVE ROTHMAN
654-5584

HACKING SINCLAIR
COMPUTERS

BILL WHITMORE
DICK STURGELL

PUBLISHING COMMITTEE

PHIL SIMERLY
301-434-0117

COMPUTER BULLETIN
BOARD

301-645-1432

MEETING INFORMATION
RECORDING

ALLEN MINTON
493-4502

HOUSHOLD PROCESS
CONTROL, MATH,
MUSIC
PRESIDENT MONTGOMERY
COUNTY TI USERS GRP

GEORGE SMYTH
703-533-8710

ADVENTURE, ROLE
GAMES (ETC)

- g. V-VEL - Optional entry, specifies vertical velocity. Same as ----- H-VEL.. Positive values move the sprite right.
5. A successful utilization of sprites combines all of their attributes, as well as all the capabilities of the TI99/4A Extended Basic sub-programs. The WDCTIUG program library includes a number of sprite demos that can be extremely informative to a novice programmer.

I would like to thank Mr. W. K. Balthrop, whose programs and tutorials, as published in numerous issues of the 99er Home Computer Magazine, offered viable sources of information, in my research for this presentation. This text will be available to all club members and their friends and will be submitted for publication to the Newsletter.

DISK DRIVES

*		*
*	MPI M. B52 DS/DD (RECOND) \$145.00	*
*	MPI M. B52 SS/DD (RECOND) \$125.00	*
*	SHUGART SA400 SS/DD 35 TRK. \$105.00	*
*	TEC M. 501 SS/DD 1/2 HT \$145.00	*
*	PANASONIC DS/DD 1/2 HT \$185.00	*
*	TEAC 55-B DS/DD 1/2 HT \$195.00	*
*	APPLE COMPATIBLE DRIVE \$195.00	*
*	DISK DRIVE CABINET WITH P/S \$45.00	*
*	3' DISK DATA CABLE FOR TI \$25.00	*
*	DUAL POWER CABLE FOR TWO 1/2 HT DISK DRIVES. \$10.00	*
*	5 1/4" DISK DRIVE ALIGNMENT \$30.00	*
*	5 1/4" DISK DRIVE REPAIR \$55.00	*

SALMAGUNDI LTD.
PHONE (703) 476-6752

SMART PROGRAMMING GUIDETM FOR SPRITES 5⁹⁵


Plus Shipping

This guide will show you some of our professional programming secrets on how to. Use CALL PEEK • Get sprites to pick up objects eat dots and lay down a trail • Shoot sprites without missing a coincidence • Make one sprite chase another • Easily convert sprite rows and columns into graphic rows and columns and visa versa • Generate moving sprite patterns • Use 3 different CALL KEY or CALL JOYST examples for moving sprites. • Write a GENERAL BAR GRAPHING program (to one pixel accuracy) that shows you sprites aren't just for games

Full of fast running and Byte saving examples that you can use in your existing programs or combine together to write your own programs. Each example program is fully documented in a step by step method that is easy to understand. A TI 99/4 or 99/4A computer and the extended basic command module are required.

Sorry, no C.O.D.'s or credit card orders. Foreign orders payable in U.S. currency. CA residents add applicable sales tax. Shipping and handling U.S. Canada and Mexico 1.50. All other countries 3.50.

15 day money back guarantee


MILLERS GRAPHICS
 1475 W. CYPRESS AVE., E-1 • SANDIMAS, CA 91773
 (714) 599-1431 FREE CATALOG AVAILABLE

SOFTWARE SPECIALS

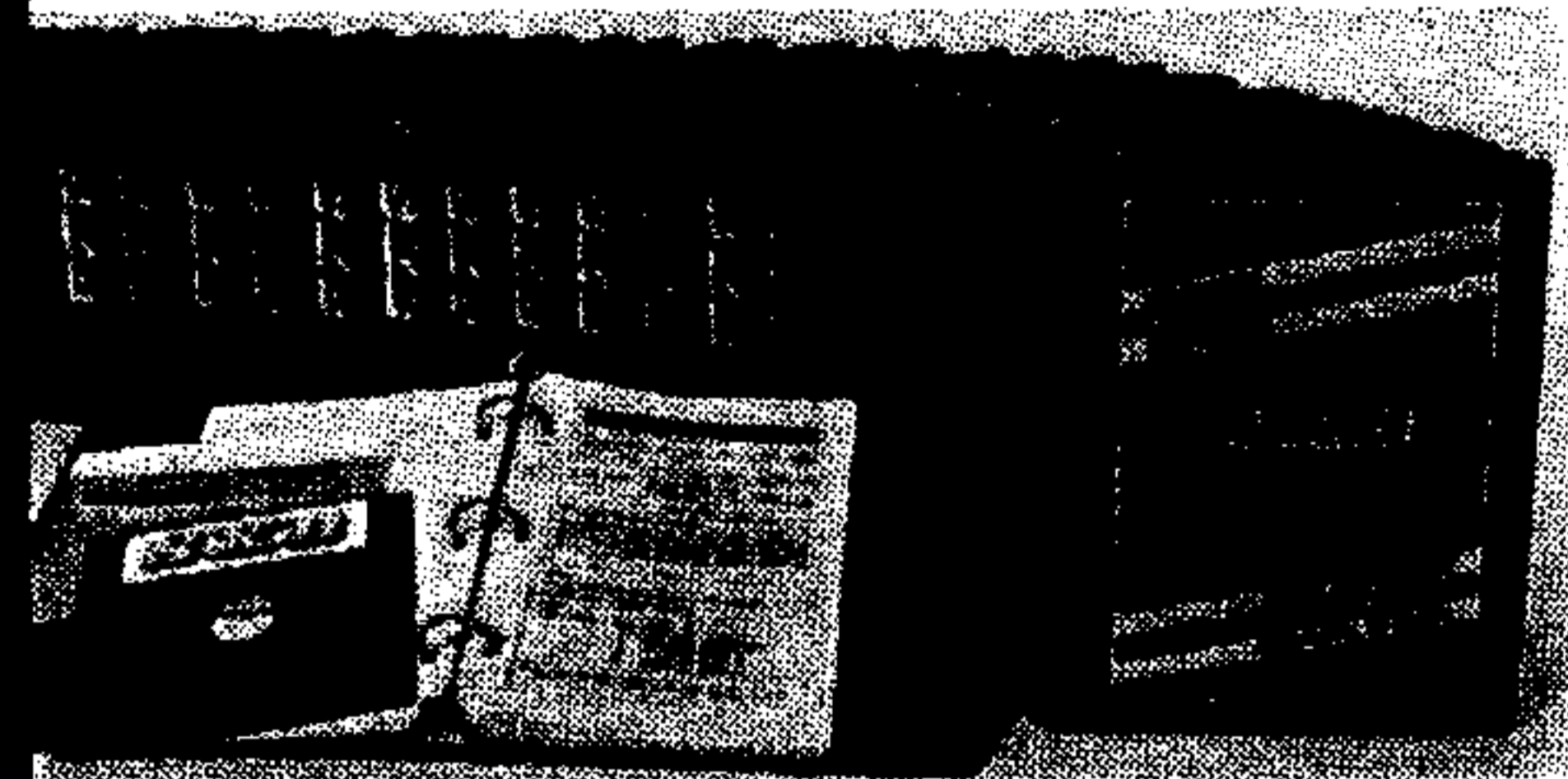
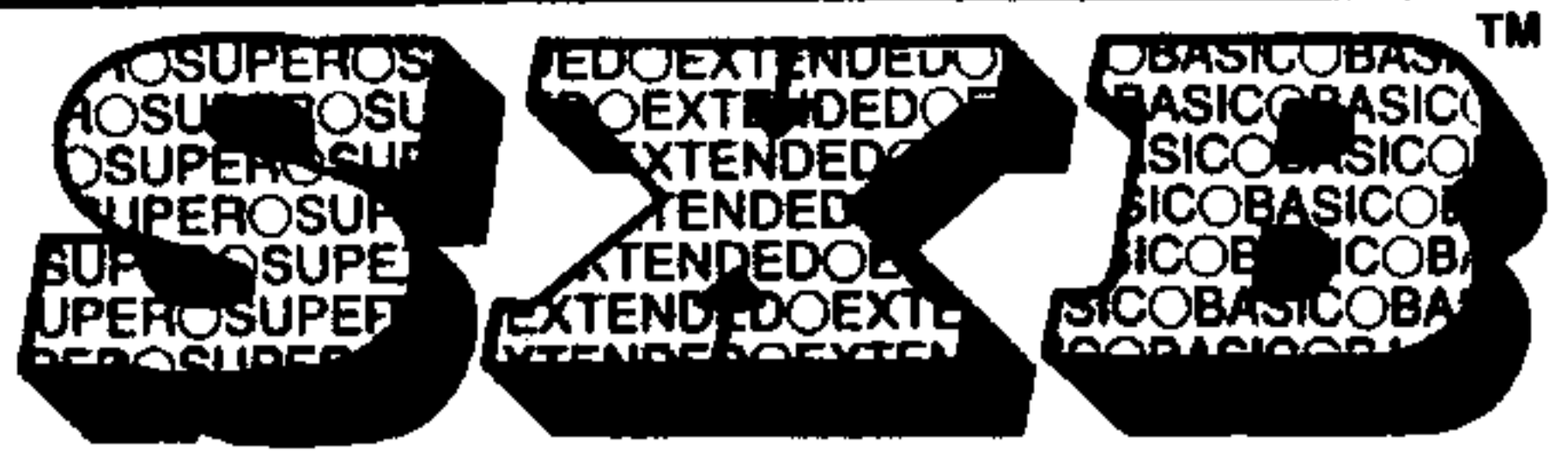
PRODUCT	RETAIL	SALES
Super Extended Basic Over 100 assembly language Subroutines -- XB + XPI + Disk	99.95	70.00
Multi-Disk Informer Master Catalog Program: Stand-Alone -- XB + XPI + Disk	29.95	22.00
SNB Version -- XB + XPI + Disk	19.95	15.00
Video Titles I Produce Custom Titles for your Video Recordings -- Tape or Disk	29.95	22.00
Video Titles II Produce Sources of Custom Titles for Video Recordings or In-Store Advertising -- XB + Disk	49.95	36.00
Video Titles III Produce Custom Titles for your own programs -- Disk or Tape	24.95	20.00

Above programs by J & K H Software

ALSO AVAILABLE
 TEXAS INSTRUMENTS SOFTWARE & HARDWARE
 COR-COMP HARDWARE
 PLUS MONITORS, PRINTERS & MODEMS

SEND ORDERS TO:

Mr. Carroll (Pets) Eddy
 4824 Dixie Hill Road
 Fairfax, VA 22038
 Phone (703) 631-1744, 7-18 p.m.



A POWERFUL GROUP OF MORE THAN 100 TMS9900
 ASSEMBLY LANGUAGE SUBROUTINES WHICH, IN
 CONJUNCTION WITH EXTENDED BASIC & MEMORY
 EXPANSION, SUBSTANTIALLY EXPAND THE
 PROGRAMMING CAPABILITIES OF THE TI-99/4A

\$99.95

FAR EXCEEDS THE WILDEST IMAGINATION
 OF A BASIC PROGRAMMER.
The New York Times PCUG

PRICE INCLUDES FIRST
 SIX ISSUES OF SXBRIEF
 INFORMATION UPDATE
 NEWSLETTER

J&KH SOFTWARE
 2820 S. ABINGDON ST.
 ARLINGTON, VA 22206

Phone: (703) 820-4131
 Dealer Inquiries Invited

Send 1st Class Stamp for FREE CATALOG

MEMBERSHIPS: Memberships are available to the general public. Cost for dues depend on the group you join. Membership information for each group is: WASHINGTON D. C. AREA TI USER'S GROUP--Contact Frank Jordan at (301) 899-3707 or mail your dues to Frank at 4100 Canterbury Way, Temple Hills, Md. 20748; TI BALTIMORE USER'S GROUP (TIBUG)--Send \$15 to Membership Chairman TIBUG PO BOX 3, Perry Hall, MD. 21128; HAGERSTOWN-WILLIAMSPORT TI USER'S GROUP--Write Sam Williams at PO BOX 376, Williamsport, MD 21795 or call (301) 223-8014; MONTGOMERY COUNTY TI USER'S GROUP--Call Allen Minton at (301) 949-4085.

TI NEWS
 BOX 267
 LEESBURG, VA
 22075

TIME SENSITIVE MATERIAL

BULK RATE
 U. S. POSTAGE
 PAID
 LEESBURG, VA.
 PERMIT NO. 68

WASHINGTON D. C. AREA
 TI 99/4 USER'S GROUP
 MAIL TO: F. Jordan
 4100 Canterbury Way
 Temple Hills, MD 20748
 & TI Membership

NAME _____
 ADDRESS _____
 CITY _____ STATE _____ ZIP CODE _____
 TELEPY _____