



Copyright 1988

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 120 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO JUST \$1.00 EACH!, plus \$1.50 per order for cassette or disk and PP&H. Minimum order of \$10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette.

Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION GAMES, REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCAB-

ULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPIES AND DISPLAYS

NUTS & BOLTS DISKS

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended Basic. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS (No. 1) has 100 subprograms, a tutorial on using them, and 5 pp. documentation. NUTS & BOLTS No. 2 has 108 subprograms, 10 pp of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pp. of documentation. NOW JUST \$15 EACH, POSTPAID.

TIPS FROM THE TIGERCUB

These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions.

TIPS (Vol. 1) contains 50 original programs and files from Tips newsletters No. 1 through No. 14. TIPS VOL. 2 contains over 60 programs and files from Nos. 15 thru 24. TIPS VOL. 3 has another 62 from Nos. 25 through 32. TIPS VOL. 4 has 48 more from issues No. 33 through 41. NOW JUST \$10 EACH, POSTPAID.

```

#####
& NOW READY &
& TIPS FROM TIGERCUB VOL.5 &
& Another 49 programs and &
& files from issues No. 42 &
& through 50. Also $10 ppd &
#####

```

TIGERCUB CARE DISKS #1, #2, #3 and #4. Full disks of text files (printer required). No. 1 contains the Tips newsletters #42 thru #45, etc. Nos. 2 and 3 have articles mostly on Extended Basic

programming. No. 4 contains Tips newsletters Nos. 46-52. These were prepared for user group newsletter editors but are available to anyone else for \$5 each postpaid.

This one should come in handy for bowling league captains and Little League coaches.

```

100 DIM M(29,29),T$(30)
110 GOTO 130
120 M;Q$;J;I;X;P$;S$;K
130 !BP-
140 DISPLAY AT(3,7)ERASE ALL
:"LEAGUE SCHEDULER":;"by th
e Burwells          adapt
ed by Tigercub"
150 DISPLAY AT(8,1):" This p
rogram sets up a":"schedule
for up to 30 teams":"so that
each plays each":"other onc
e and only once."
160 DISPLAY AT(12,1):" If an
odd number of teams":"are s
cheduled, each gets one":"by
e."
170 DISPLAY AT(16,1):"Number
of teams?" :: ACCEPT AT(16,
18)VALIDATE(DIGIT):N :: IF N
>30 THEN DISPLAY AT(18,1):"L
IMIT OF 30!" :: GOTO 170
180 DISPLAY AT(18,1)ERASE AL
L:"Schedule teams by name? Y
" :: ACCEPT AT(18,25)SIZE(-1
)VALIDATE("YN"):Q$ :: IF Q$=
"N" THEN 200
190 FOR J=1 TO N :: DISPLAY
AT(20,1):"Team no.":J;"name?
" :: ACCEPT AT(22,1):T$(J)::
NEXT J :: GOTO 210
200 FOR J=1 TO N :: T$(J)="T
eam No. "&STR$(J):: NEXT J
210 IF N/2<>INT(N/2)THEN N=N
+1 :: T$(N)="bye"
220 DISPLAY AT(23,1):"Schedu
le by day, week, month":"or
what?" :: ACCEPT AT(24,10):S
$ :: FOR J=1 TO N-1 :: M(I,J
)=J+1
230 NEXT J :: FOR J=1 TO N-1
STEP 2 :: GOSUB 260
240 NEXT J :: FOR J=2 TO N-2
STEP 2 :: GOSUB 330
250 NEXT J :: GOSUB 390 :: S
TOP
260 FOR I=1 TO N-2 :: IF M(I
,J)=M THEN 280

```

```

270 M(I+1,J)=M(I,J)+1 :: BU
D 290
280 M(I+1,J)=M(I,J):: GOTO 3
00
290 NEXT I
300 X=I+1 :: FOR I=X TO N-2
:: M(I+1,J)=M(I,J)-1
310 NEXT I
320 RETURN
330 FOR I=1 TO N-2 :: IF M(I
,J)=2 THEN 350
340 M(I+1,J)=M(I,J)-1 :: GOT
D 360
350 M(I+1,J)=M(I,J):: GOTO 3
70
360 NEXT I
370 X=I+1 :: FOR I=X TO N-2
:: M(I+1,J)=M(I,J)+1
380 NEXT I :: RETURN
390 DISPLAY AT(12,1)ERASE AL
L:"Output to - 2":;" (1) Sc
reen":;" (2) Printer" :: ACCE
PT AT(12,13)SIZE(-1)VALIDATE
("12"):K :: IF K=1 THEN 440
400 DISPLAY AT(18,1):"Printe
r? P10" :: ACCEPT AT(18,10)S
IZE(-18):P$ :: OPEN #1:P$ ::
PRINT #1:"LEAGUE SCHEDULE":
: :: FOR I=1 TO N-1 :: PRIN
T #1:S$;" #":I :: PRINT #1:T
$(I);" vs ";T$(M(I,1))
410 FOR J=2 TO N-2 STEP 2 ::
PRINT #1:T$(M(I,J));" vs ";
T$(M(I,J+1))
420 NEXT J :: PRINT #1:"":
430 NEXT I :: RETURN
440 FOR I=1 TO N-1 :: PRINT
TAB(7);"LEAGUE SCHEDULE":
: :: PRINT "WEEK #":I: : :: PR
INT T$(1);" vs ";T$(M(I,1)):
: FOR J=2 TO N-2 STEP 2 :: P
RINT T$(M(I,J));" vs ";T$(M
(I,J+1))
450 NEXT J :: PRINT "": : ::
PRINT "PRESS ANY KEY FOR NE
XT WEEK"
460 CALL KEY(0,K,S):: IF S=0
THEN 460
470 CALL CLEAR
480 NEXT I :: RETURN :: END

```

Some folks seem to think that the subprograms on my Nuts & Bolts disks are just flashy screen displays. Not so! This one will be on the next diskfull, if I ever get it full, which is most unlikely. ACCEPT AT with a negative

size is useful to accept a default string from the screen, but the length of the string is limited to 28 characters; and if you want something other than the default, you must be sure to delete any extra characters. CALL DEFAULT(R,C,M#,R#), where R and C are the row and column to accept at, M# is the default string which can be up to 254 characters long, and R# is the string accepted, will display the default string, accept it if Enter is pressed, or accept any other string without having to blank out the extra characters. Just don't type too fast!

```
100 M#="TESTING" :: CALL CLEAR
110 CALL DEFAULT(12,1,M#,R#)
:: DISPLAY AT(24,1):R# :: GO TO 110
10000 SUB DEFAULT(R,C,M#,R#)
:: R#="" :: X=ASC(M#)
10001 DISPLAY AT(R,C):M#
10002 CALL HCHAR(R,C+2,ASC(SEE$(M#,1,t))):: CALL HCHAR(R,C+2,30)
10003 CALL KEY(O,K,S):: IF S=0 THEN 10002 ELSE IF K=13 THEN RM=M# :: SUBEXIT ELSE DISPLAY AT(R,C):CHR$(K):: ACCEPT AT(R,C+1):R# :: R#=CHR$(K)&R#
10004 SUBEND
```

CALL DEFAULT(R,C,M,RN), with M as the default value and RN as the value accepted, will do the same for numeric input, and will reject any non-numeric input. Errors due to fast typing can be prevented by omitting the DISPLAY AT(R,C):CHR\$(K) in line 1002.

```
100 N=176453.897 :: CALL CLEAR
110 CALL DEFAULTN(12,1,N,RN)
:: DISPLAY AT(24,1):RN :: GO TO 9999
10000 SUB DEFAULTN(R,C,M,RN)
:: DISPLAY AT(R,C):N :: M#S=SEE$(STR$(N),1,1)
```

```
10001 CALL HCHAR(R,C+2,ASC(M$)):: CALL HCHAR(R,C+2,30)
10002 CALL KEY(O,K,S):: IF S=0 THEN 10001 ELSE IF K=13 THEN RM=N :: SUBEXIT ELSE DISPLAY AT(R,C):CHR$(K):: ACCEPT AT(R,C+1):R# :: R#=CHR$(K)&R#
10003 ON ERROR 10004 :: RM=VAL(R#):: GOTO 10005
10004 CALL SOUND(200,110,5,-4,5):: DISPLAY AT(R,C):N :: ON ERROR STOP :: RETURN 10002
10005 SUBEND
```

Ed Machonis discovered an easy way to count the words in a TI-Writer file, using TI-Writer itself. Just put in a line before line 0001, with .LM 0;RM 1;FI;PL nnn with nnn being the sector length of the file multiplied by 40. Save it, go into the Formatter and print it to disk under a different filename. Return to Editor, load the resulting file, page through it with FCTN 4 counting any blank lines, subtract the number of blanks from the last line number, and that's it! The Formatter takes about one minute to count 1000 words. If the resulting file is very large, you may have to load it in two sections.

```
100 M#="POS WILL FIND THE FIRST OCCURRENCE OF A SUBSTRING WITHIN A STRING BUT I OFTEN NEED TO FIND THE LAST OCCURRENCE SO I WROTE THIS SUBPROGRAM"
105 INPUT "SUBSTRING?":L#
110 CALL LAST(M#,L#,P):: IF P=0 THEN PRINT "NOT FOUND" :: GOTO 105 ELSE PRINT SEE$(M#,P,255):: GOTO 105
120 SUB LAST(M#,L#,P):: X=1
130 Y=POS(M#,L#,X):: IF Y=0 THEN P=0 :: SUBEXIT ELSE Z=Y
140 X=Y+1 :: Y=POS(M#,L#,X):: IF Y=0 THEN P=Z :: SUBEXIT ELSE Z=Y :: GOTO 140
150 SUBEND
```

Here's a new way to make music. The algorithm in 110 sets up a 3-octave chromatic scale - note the N(1)=F, I have erroneously omitted it when I previously published that algorithm.

To change the key of the music you have programmed, just change the value of F. Lines 190-220 contain the part of the music that is repeated within the melody. A is the subscript of the melody note, B is the subscript number of the chord. These must be above 13, as the frequency is divided by 2 in the subroutine.

Each beat of the music has a GOSUB, to 230 to play a bass accompaniment with the first note of each bar, to 260 for the other notes of the bar. The chord note is divided by different values to play the three notes of the chord in succession, and multiplied by 3.75 in the 3rd voice to produce a bass note two octaves lower in the -4 noise. The melody note is multiplied by 1.01 in the second voice to give a richer tone.

```
100 DISPLAY AT(12,3)ERASE ALL:"THE MAORI FAREWELL SONG"! programmed by Jim Peterson
110 F=110 :: DIM N(36):: FOR J=1 TO 36 :: N(J)=INT(F*1.059463094^(J-1)):: NEXT J :: N(1)=F :: T=-999
120 GOSUB 190 :: A=30 :: B=23 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: A=32 :: B=28 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: A=28
130 GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: A=30 :: B=23 :: GOSUB 230 :: GOSUB 260 :: A=28 :: GOSUB 260 :: A=27 :: GOSUB 230 :: GOSUB 260
140 A=28 :: GOSUB 260 :: A=30 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: GOSUB 260 :: GOSUB 260 :: GOSUB 260 :: GOSUB 190
150 A=30 :: B=23 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260
```

```
:: A=32 :: B=16 :: GOSUB 230
:: GOSUB 260 :: A=28 :: GOSUB 260
160 A=33 :: B=23 :: GOSUB 230
0 :: GOSUB 260 :: A=32 :: GOSUB 260 :: A=25 :: B=13 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: GOSUB 260
170 A=27 :: B=23 :: GOSUB 230
0 :: GOSUB 260 :: GOSUB 260 :: A=28 :: B=16 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260
180 B=28 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: B=16 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: GOTO 120
190 A=32 :: B=28 :: GOSUB 230
0 :: GOSUB 260 :: GOSUB 260 :: A=28 :: B=16 :: GOSUB 230 :: GOSUB 260 :: A=30 :: GOSUB 260
200 A=32 :: B=28 :: GOSUB 230
0 :: GOSUB 260 :: GOSUB 260 :: B=16 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: GOSUB 230 :: GOSUB 260
210 A=30 :: GOSUB 260 :: A=33 :: B=23 :: GOSUB 230 :: GOSUB 260 :: A=27 :: GOSUB 260 :: A=28 :: B=16 :: GOSUB 230
0 :: GOSUB 260 :: GOSUB 260
220 B=28 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: B=16 :: GOSUB 230 :: GOSUB 260 :: GOSUB 260 :: RETURN
230 CALL SOUND(T,N(A),5,N(B)/1.585,9,N(B)*3.75,30,-4,9):: GOSUB 290
240 CALL SOUND(T,N(A),5,N(B)/1.334,9,N(B)*3.75,30,-4,9):: GOSUB 290
250 CALL SOUND(T,N(A),5,N(B)/2,9,N(B)*3.75,30,-4,9):: GOSUB 290 :: RETURN
260 CALL SOUND(T,N(A),5,N(A)*1.01,5,N(B)/1.585,9):: GOSUB 290
270 CALL SOUND(T,N(A),5,N(A)*1.01,5,N(B)/1.334,9):: GOSUB 290
280 CALL SOUND(T,N(A),5,N(A)*1.01,5,N(B)/2,9)
290 FOR D=1 TO 20 :: NEXT D :: RETURN
```

MEMORY FULL.....

Jim Peterson



Disk Fix

by WESLEY R. RICHARDSON
BLUEGRASS 99 COMPUTER SOCIETY, INC.

When you have a disk with several files that you have been working on and you do a catalog and it comes up DISKETTE IS BLANK, or DISK NOT INITIALIZED, it can be very frustrating. There are times when the sectors used and available get changed to values like 2389 free and 7887 used, but you know you have a single sided, single density (SSSD) disk drive, with a maximum of 360 sectors. It is also possible to have a disk which will not catalog, yet when Extended BASIC is selected, the disk will run the LOAD program and continue without a problem. These have happened to me and I am sure it has happened to others, so I thought I would document a way which may recover your disk for you.

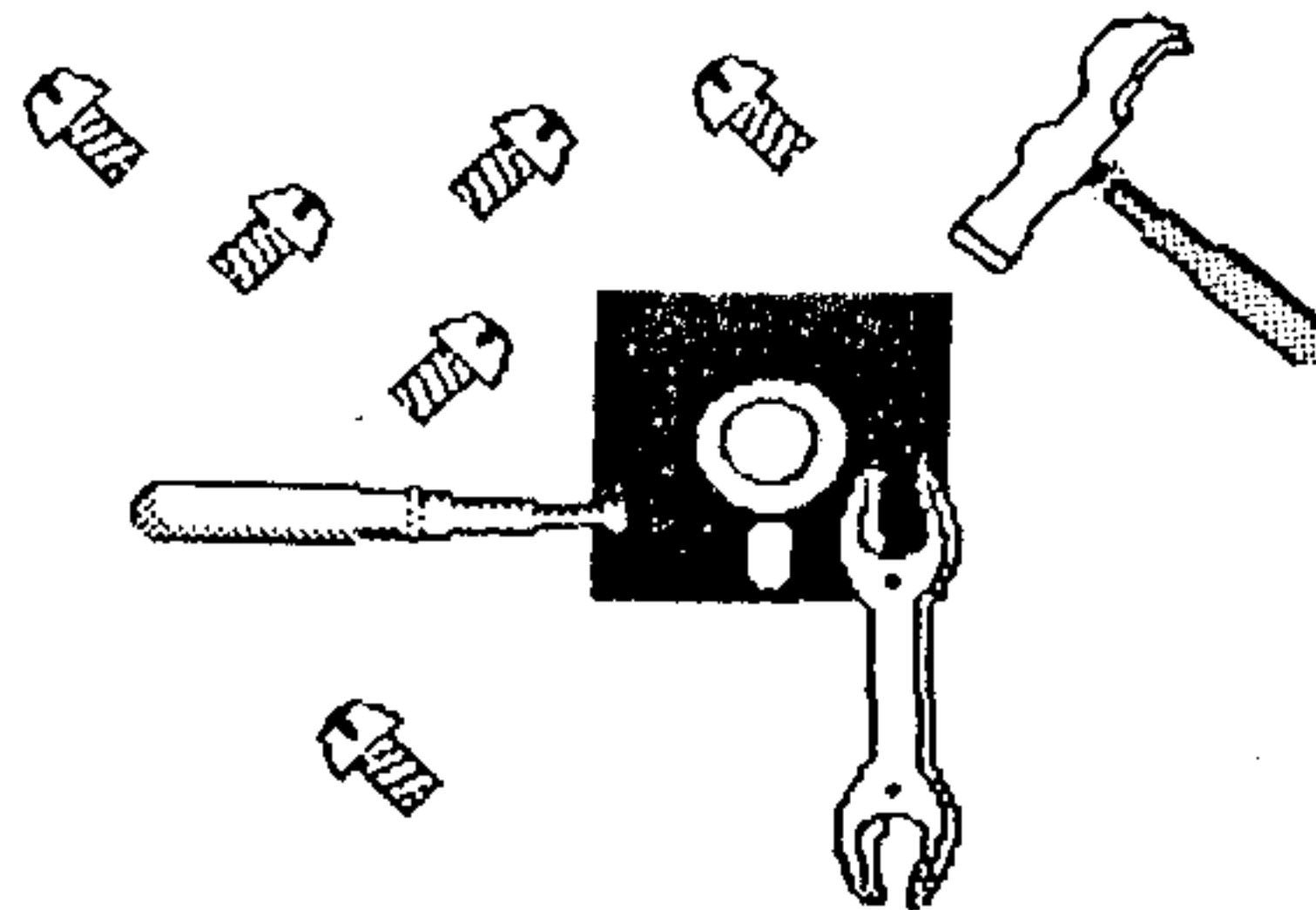
The items which you will need are your blown disk, two blank disks, Disk Manager 1000 v3.5, Disko or Disk Patch, and a sector or track copier program, or the equivalent of any of the above. I will use the Funnelweb v4.10 DISK-PATCH for the sector editor.

1) The first step is to initialize a disk in the format which you believe the blown disk was, for example SSSD. For the disk name, use the name that you want on the blown disk after it is restored.

2) Using the sector copier or track copier, make a copy of the blown disk. If you get a read error in sector 0, just tell the program to ignore the error. If you are unable to copy the disk with the copier programs which you have available, you may still continue the following steps with the original disk, but be advised that you may lose everything on the disk.

3) Load DISK-PATCH or DISKO and then insert the back-up copy of the blown disk in drive 1. Select option 1 for disk sector editor. Then disk 1, and sector 0. The screen should come up with the data from sector 0. Pressing FCTN 2 will change the screen to ASCII and pressing FCTN 1 will change it to HEX. In ASCII, the first ten characters will be the disk name. In HEX, at byte 12h (h=HEXADECIMAL) will be 01 for single sided and 02 for double sided. At byte 13h, will be 01 for single density and 02 for double density.

4) Press FCTN 4 to go to sector 001h. You should



find groups of four digits of HEX numbers such as 0002 0003 0009 0015 and so on. These indicate where the file names and file maps may be found. Write down each of these numbers in the order which they are found when read from left to right and top to bottom on the screen. Note also if the first number is 0000, then the disk will catalog as being blank and no file names will appear.

5) Press FCTN 4 to go to sector 002h. In the first ten ASCII characters you will find a file name. Write this down next to the appropriate four digit number you had in step 4). Do this for each of the numbers from step 4). If there were several files on the disk, you may need to press FCTN 9 and then option 1 again to go directly to the location. While in sector edit mode, pressing FCTN 6 will take you to the next lower numbered sector.

6) You now should have a table similar to the one below with the file name and location of each file on the disk.

0000	A-SECTOR2	000D	PACMAO
0003	CENTIPEDE	0005	PINBALL
0009	DEFENDER	0006	PINBALM
000A	KONG	0007	POLE/POS
000B	KONH	0008	POLE/POT
0004	LOAD	000E	TI/INVADER
000C	PACMAN	000F	TI/INVADES

7) Note in the case that we did find a 0000 but a file was there, as in this case file A-SECTOR2 directory was located at sector 002h, then use the sector editor to view sector 001h. Move the cursor to the first 0000 in HEX and change it to read 0002. Then press CTRL W to write the sector back to the disk, and answer Y to the question RE-WRITE SECTOR?

8) Remove the copy of the blown disk and insert the formatted blank disk in drive 1. Select the sector editor, giving drive 1 and sector 0. After the sector comes up, remove the blank disk and insert the blown disk copy in drive 1. Press CTRL W to rewrite the sector.

9) Load Disk Manager 1000 version 3.5 (DM1000), and then put the blown copy disk back in drive 1. Select option 1, File Utilities. Then select option 2 for Recover file. Give the drive as 1. Enter the first file name on you list and press enter. The program will say SEARCHING DISK, then RE-BUILDING LOST FILE, then FILE RECOVERED. Press enter and then 2 for Recover file. Repeat these steps until all of the files are recovered.

10) Press 1 for Copy/Move/Delete... and give the disk number as 1. Your disk files should now be restored. If the disk free and used does not match up with the sum of the file sizes plus 2 sectors, then go to step 11), otherwise you are done.

11) Do this step only if the disk free is not correct. Place a D in the left column to delete all of the files and a U in the right column to unprotect all of the files. DM1000 will unprotect and then delete all of the files. At this point a catalog should show free 358, used 2 for a SSSD disk. Go back to the recover file section of step 9) and recover each file again.

One other piece of advise, if you have a disk with a bad directory, do not write any files to the disk until you have a chance to fix the directory. If you write a new file, then you are taking the chance that part of another file will be over-written. This can happen because sector 0 may show that a location is free, when in fact it has part of a file in it.

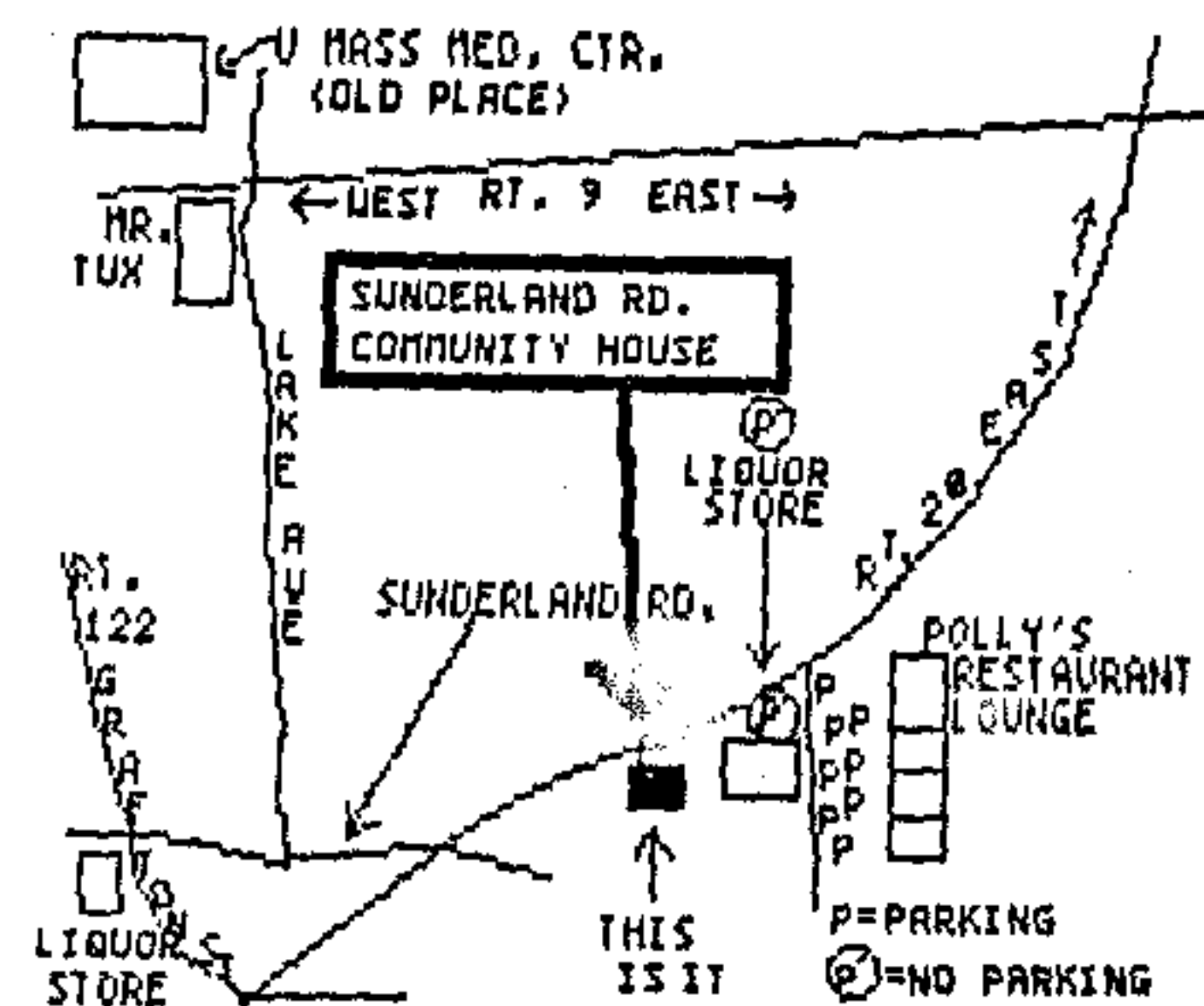
The other advise is to always keep a back-up copy of anything which you do not want to lose. It is a good idea to keep a write protect tab on your master disk and keep it away from your work disk. On documents or programs, save your work to disk every 15 minutes so if the power goes off or your computer locks up, you only lose 15 minutes worth of work. Alternate saving to two disks when you have a large and important program or file.

If you always keep back-ups, I hope you will not need to use DISK-FIX, but if that time comes when the disk is blown, now you have something to try.

FIXING GLITCHED XBASIC PROGRAMS AND DV/80 FILES:

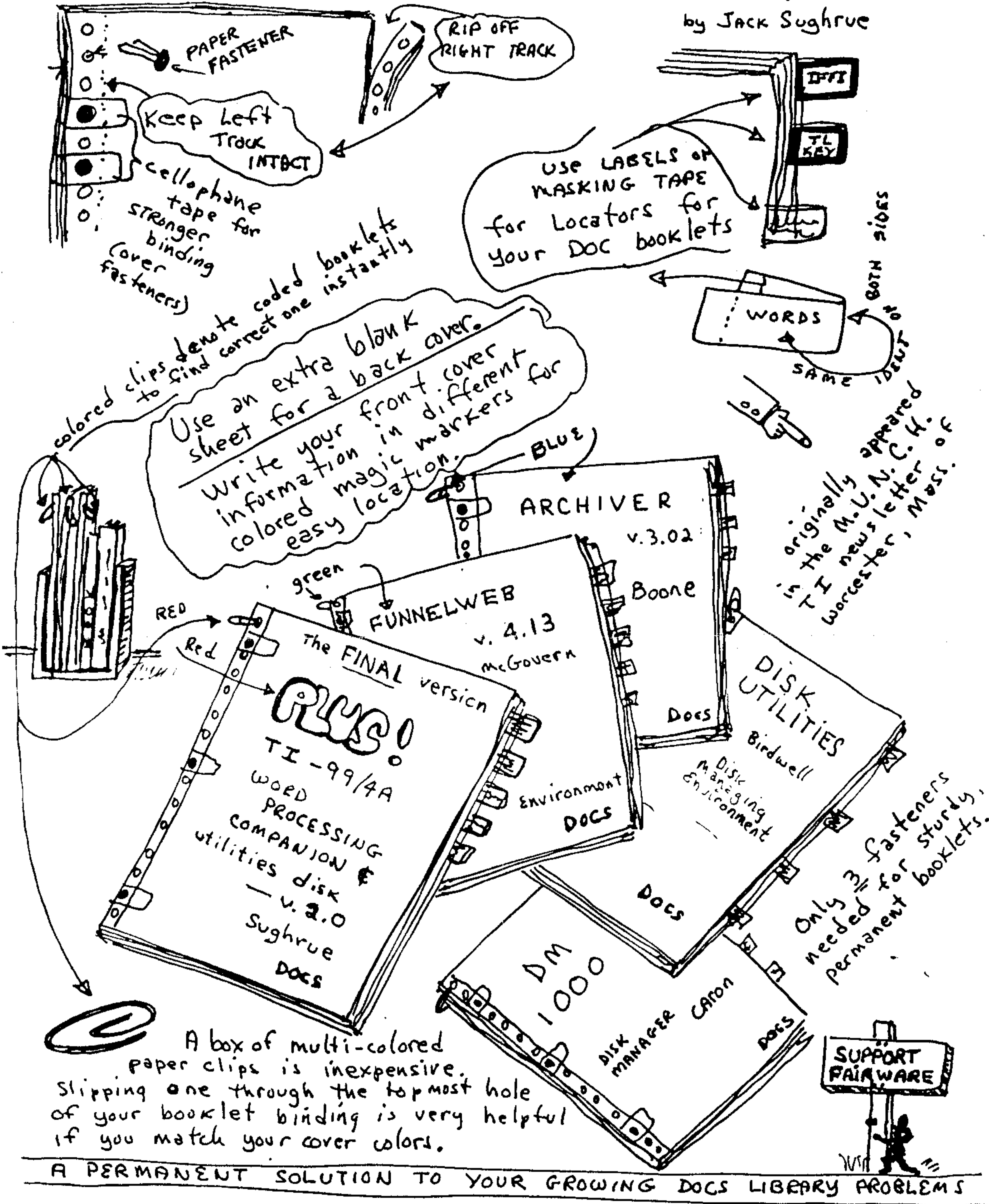
Got an adventure graphics game on disk at our last club meeting. After playing through several screens the next one to load stopped with a syntax error. Listing the program showed several lines of code to be glitched. Trying to edit out the glitched code caused the screen to change from blue to red and then lock up the computer. Not wanting to wait a month for the next club meeting to exchange the disk I decided to experiment. First copied the disk with Jim Schroeder's REDISKIT. The program on the original disk would not even load because of a bad sector. Next saved the program to disk with the command LIST "DSKx.filename". This DV/80 file must next be printed to disk with the TI-Formatter. It will not load into the Editor after listing because the file still has the glitches in it. Next load the formatter file into the Editor and delete the glitched lines and print back to disk with the command "C DSKx.filename" to remove linefeed symbols put in by the formatter. If you are lucky to have a printout of the program before it got glitched it will be easy to add the missing code and then convert it back to program format with a DV/80 to program conversion utility. In my case the next screen to load after this one had identical code except for a few lines that were different, so I added the lines and thus reconstructed the glitched program. If neither of the above options are available you could try guessing at the missing code. Of course if you knew of someone else that bought the same disk and had a modem, he could send a replacement for the bad file to you, but that is not much of a challenge. The above procedure will also work for glitched DV/80 message files from BBS. This is a lot easier as most of the above steps can be eliminated. Sometimes just printing the glitched file from the formatter to printer is all that is necessary if you do not wish to save the file for later use. Have fun...KCS (sorry I don't know who KCS is. I would like to give him credit for this information. ed.)

Thanks to the Great Lakes User Group for this article.Oct 1988



BUILT-IN BOOKLETS with Tractor-Feed Paper

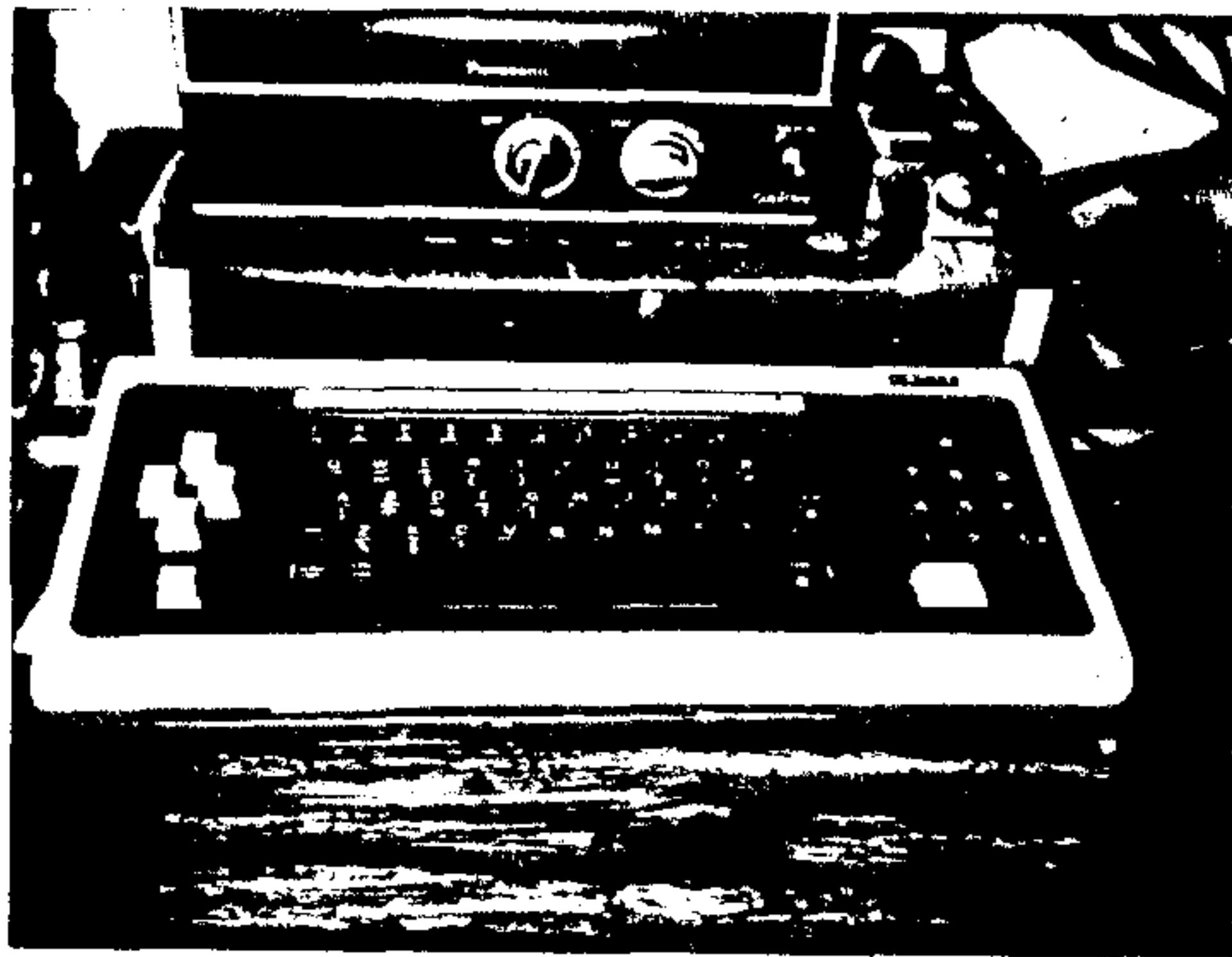
by Jack Sughrue



A PERMANENT SOLUTION TO YOUR GROWING DOCS LIBRARY PROBLEMS

Subject: TI-99/4A Keyboard Wiring Guide
By Glenn Bernasek TI-Chips Cleveland, Ohio

Wire/Pin Number	Key
1	Y , U , I , O , P , ENTER
2	6 , 7 , 8 , 9 , 0
3	A , S , D , F , G , SHIFT
4	H , J , K , L , ; , SPACE
5	N , M , < , > , . , = , /
6	ALPHA LOCK
7	1 , 2 , 3 , 4 , 5 , ALPHA LOCK
8	1 , Q , A , Z , O , P , ; , /
9	5 , 6 , T , Y , G , H , B , N
10	Q , W , E , R , T , CTRL
11	Z , X , C , V , B
12	SHIFT , CTRL , SPACE , FCTN , E
13	2 , W , S , X , 9 , 0 , L , .
14	3 , E , D , C , 8 , I , K , < , >
15	4 , R , F , V , 7 , U , J , M



FROM: U.G.O.C.

THE THEORY OF DARK

By Earl Raguse

(Newt Armstrong, Scientific consultant)

Dark is the natural state of things. To make anything visible to the human eye, you must remove Dark. If you doubt this, go into a windowless dark-tight room, where no Dark can escape. You will not be able to see anything. Now if the sun is up, and you open a small crack in the room, some of the dark will leak out and you will be able to see large objects. If you make the crack large enough, the sun will suck up all the dark and you can then see things clearly.

Many people believe in Light theory; just as they do in Ben Franklin's wrong way theory of electrical current. We however, think that Dark theory is the more reasonable. Rumor is that Al Einstein privately subscribed to Dark theory, but publically used Light theory, because it made it easier to explain his Universal Relativity Theory to skeptical scientists of that day.

Our sun is the largest darksucker in our solar system. That's how it gets the energy to radiate all that heat and ultra-violet rays. The early Greeks had discovered that lighted candles had some dark sucking capacity, but it took Thomas Edison to invent a good artificial darksucker.

Dark travels in straight lines. Dark cannot penetrate solid objects and thus they make good dark shields. To prove this, look under any car, at high noon, with the sun directly overhead, and you will see a patch of dark that the sun is unable to suck up. Stand with your back to the sun, and you will see that you too are a good dark shield. You will have trapped a patch of dark which can't get around your fat frame to one of the universe's best darksuckers, the sun.

Dark travels at 186,000 mi/sec. That's why, when you switch on the artificial darksucker (sometimes erroneously called a light) it sucks up the dark so fast that you can't really time it, after all in a 30 foot square room with the darksucker at center ceiling, its less than 22.67 feet to the most remote corner, and with a velocity of 186,000 mi/sec, it only requires about 23 pico seconds for dark to go from the corner to the darksucker. Most people hardly notice the delay.

Dark is made up of all possible frequencies in the subtractive mode, such that each frequency cancels another, thus the frequency of true dark is zero (ie dc). Pure white is total absence of dark, that takes a powerful darksucker like the sun or a good arclamp. Some artificial darksuckers have a frequency bias, and do not suck up all dark, and

hence leave the illusion of color.

Certain darksuckers, like neon and sodium lamps eg, although apparently, energy efficient, are primitive with respect to frequency purity, and thus appear to be colored. Any color can be created by subtraction of the correct frequencies.

The proliferation of outdoor artificial darksuckers in our cities, eg Los Vegas, make it very difficult for astronomical observation, they pollute the night sky which in its natural state is void of darksucking, thus enabling high powered astronomical telescopes to detect faint points of dark loss caused by far off galaxies of darksuckers like our sun.

One of the most remarkable darksuckers is modern television; the best of which have a normally black (very dark) screen when unpowered. They incorporate electronics modules, and computers in some cases, and have the ability to convert radiated TV signals so as to suck dark from discrete points on the screen in varying amounts and frequency such that a colored picture appears on the normally dark screen.

What will they think of next. We have heard that a new super-computer will use dark logic. This does not require perfection of room temperature superconductivity or the expense of supercooling as is the current state of the art. This new computer will be very fast, extremely energy efficient and so cheap that you can afford one for every program you use, hence saving all that boot up time.

Dark logic is very simple. The basis of which is:

Dark = Not Light

Light = Not Not Not Dark

See how much simpler Dark logic is.

We know that Light theory is a lot of bull, and probably a conspiracy also. Astronomers speak of Black Holes. Now you know this can't be true, how can anything with so much mass be a hole? Actually these things are good Dark Radiators, (light suckers, to the under educated) and are the true source of dark in this universe

Newt Armstrong, a psuedo scientific and computer wizard, has graciously offered to promigate this obscure branch of dark knowledge at the weekly Garage SIG, along with other sagacious information. Call the author for exact time, address and directions for getting there. Like the true academician that he is, Newt will charge no fees before you are hooked on the theory.

This may not be a continuing series.

(Ed note: Some people spend most of their lives in the dark.)

A TECHNICAL NOTE ON ORIGINAL T. I. DRIVES

by Michael Martinko
Lima Ohio User Group

My disk system presently is composed of corcomp's disk drive controller, and TI's original SS/DD disk drive. Over the past year I have been noticing more and more disk errors, particularly in initializing new disks in a double density format. In fact they had become so predominant that I could no longer initialize in DD. In observing the problem there appeared a pattern to the madness, i.e., the bad sectors were in multiples of 18. Mr. Randy Belisle of Belisle Interactive Systems here in Lima has proven to be a great source of computer technical information. He suggested that the TI drives are probably divided into 18 sectors, and that the drives are running slow, hence not leaving enough room for the last sector on each rotation. He also suggested that many drives are belt driven and that over the years the belt may stretch, slowing the operating speed of the drives.

The procedure for examining the drive was simple. After removing the drive from the expansion box and unplugging the connecting wires, the metal housing is removed by straightening the metal tab that protrudes into the drive and sliding off the cover. Please note that if you have previously replaced the shunt that was originally installed with dip switches to determine which drive number your drive will respond to, it will be necessary to first remove the dip switches before the cover can be removed. You will find a series of dots on a white disk on one side of the drive. Using a flourscent light on these dots will show if the drive is slow, fast, or within tolerable range. Merely plug the wires into the drive, proceed to initialize a disk, and shine the light on the drive. Mine was running slow. Noticing that the belt was smooth on the inside and textured on the out I simple reversed the belt. Testing the drive proved my suspecions. The drive functioned perfectly. The cost was \$00.00. I will however order a new belt from TI, not knowing how long this one will last! But at least I have bought enough time, maybe years, before I will need it. Thank you Randy for your advise. I hope others who are having similar problems will find help in this article.

DONE

The Complete TI ADVENTURE COMPENDIUM

Jam-packed disks of all the fairware and public domain adventures ever written for the TI-99/4A and Geneve

THE TI ADVENTURE COMPENIUM DISK SET:

Order directly today

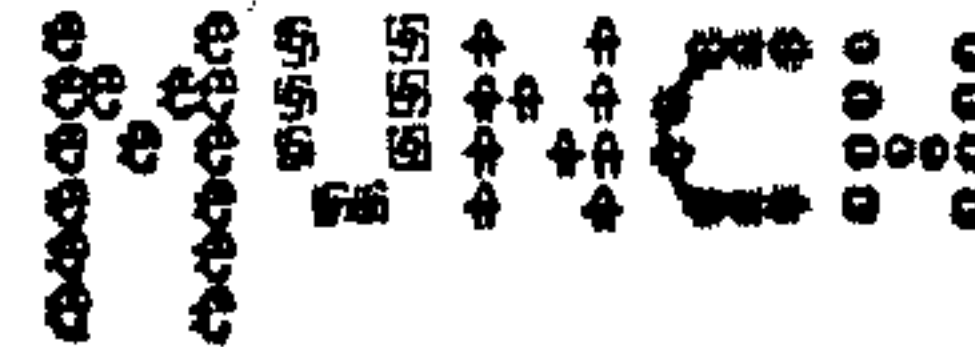
Make out checks to MUNCH and mail to
M.U.N.C.H.
560 Lincoln St.
P.O. Box 7193
Worcester, MA 01605

\$11.95 - 3 full DSDD Disks: \$15.95 - 5 DSDD:
\$19.95 - 10 SSSD

JULY 11, 1989 HAPPY FOURTH OF JULY!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

President/Mail	W.C. Wyman	839-4134
Vice^President	Bruce Willard	852/3250
Secretary	Jim Cox	
Treasurer	Jim Cox	869-2704
Acting Editor	Jim Cox	
Adv.Prog. Chair	Dan Rogers	248-5502
Library	Al/Lisa Cecchini	
Disk Librarian	Lou Holmes	617 965/3584
Tape Librarian	Walter Nowak	413 436/7675
+++++++	Jack Sughrue	476/7630



JUNE MEETING. The high-light of the May meeting was the demo that Lou and Corson did to show how two computers can link up and talk to each-other with the use of a modem. Corson show his hard disk and how to format and use it. He also demoed a number of Dissney programs which were never released to the public. They mixed games with educational subjects and were very interesting. Someone should try to get Disney to release these games/educational tools to the T.I. public at large, I think they would be a hit.

JULY MEETING. The July meeting will feature a suprise demo i.e., I don't know what we will be doing, probably from both Lou and Corson; it is dure to be interesting.

RAFFLE. Each month we have a raffle and the dollar donation per ticket helps to cover the monthly fee to rent the hall. This month's raffle will have a TI programming book, two educational game carts and at least one other item.

MONTHLY SALES. At each meeting you have the opportunity to buy and/or sell new or used hardware, software, books and original programs. Please have prices marked on any items you have to sell.

LIBRARY NOTICE. Please return any items borrowed from our library. If you can not come to a meeting or give these items to someone who will be at the meting, please mail any library items to the group address which is listed on the cover of this newsletter. There are no late fees, we don't care how long they have been out, please return these items.

REPRINTS. Reprints of any items in this newsletter is permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interests you will probably interest other members of the TI community, so please share your ideas and opinions with all of us.

NEWSLETTER EXCHANGE EDITORS. Please note our corrected address on the front cove of this issue.

BEST WISHES TO JACK FOR A SPEEDY RECOVERY.

