(066) 8709

DIGITIZER FOR TI?!!

From the Charlotte TI 99/4A UG Newsletter

An article in the TI Forum on Compuserve, by Steve Langguth, caught my attention this month. In it he describes a new product called the ImageWise Video Digitizer, developed by Steve Ciarcia, of BYTE magazine. The system is composed of a digitizer/transmitter and a receiver/display, which are boards which may be purchased assembled, or bought with the one necessary EPROM and constructed by the buyer for about $150 total. The board receives input from a video camera or VCR or laserdisc player, and stores the picture with abundant detail. With conversion through a program written by Langguth, the image is then downloaded into the TI through the RS232 port, and written to disk. Another simple XB program converts it to TIArtist format.

Langguth did not repeat the source of the board, which is certainly available from BYTE magazine (to which I do not subscribe). But he did give his own address, for more information:

Steve Langguth
2956 South Barnes
Springfield, MO 65804

This is one letter I am GOING to write!

****

KANKAKEE TI USERS GROUP
PO BOX 1945
KANKAKEE, IL.
60901

KANKAKEE, IL
SEP 14
PM
1987
609

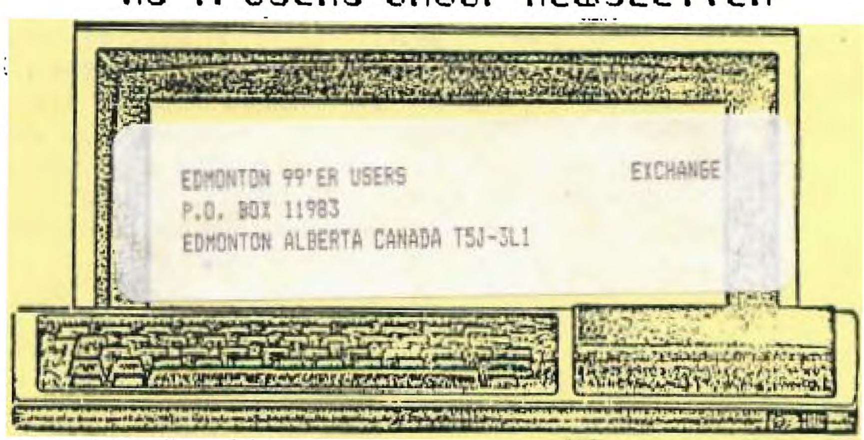
MEETING DATES

SEPT. 19TH
OCT. 17TH

BOURBONNAIS
MUNICIPAL
CENTER

1 pm 'til 4 pm

K3 TI USERS GROUP NEWSLETTER



SNAP-CRACKLE-POP

By Beverly Cook

On June 8, 1987, as my husband, 2 children and I were sitting on a friend's dock fishing, a storm blew up to the south of us. We started gathering our things together just as it started to rain. When finished, we ducked under a roof overhang to keep as dry as possible (it didn't help much). As we waited, the storm got stronger (frequent lightning and heavy rain). A very sharp crack of thunder sent us all ducking. When the rain let up, we started for home, about a block away. The closer we came to our house, the more I noticed that something didn't look right. Needless to say, I was right. Our beautiful 300(approx) year-old oak tree that sat in our front yard had been hit by lightning. Our van was almost invisible under the debris. After a quick once over to make sure there was no more damage outside, I went inside to discover that our power was dead in the living room. Another quick check found a blown circuit breaker. Once reset, I started checking electronics. My computer was on a surge protector. It held- I think. Our television, stereo and VCR and my printer were on another surge supressor. The surge supressor, I am happy to say, is still doing its thing. The television, stereo and VCR were undamaged. I am sorry to say that my computer didn't do so well. The power surge was so strong that the circuit breaker had taken the shock. The surge went into my printer, through the ribbon cable, and had taken out EVERY card in the P-box. The surge supressor held until the circuit breaker blew. Thankfully, our homeowners policy covered the damage, but we waited for almost 6 weeks before things were settled.

Since I have a mixed bag of cards, things weren't as simple as I had hoped they would be. The Flex-cable card and Memory Expansion were shipped back to T.I. after finding out the repair charges. The RS232 and CorComp Controller went to CorComp. I received the Flex-cable in 10 days and Memory in 12. (Why they weren't shipped together, I don't know.) CorComp took 3 weeks. My new printer came in 1 week. After spending the summer without a computer, I was suffering withdrawal symptoms.

The breakdown went as follows: Flex-cable- $25.75+$5.00 S&H -- Memory Expansion -- $44.00+$6.00 S&H. CorComp charges $50.00 per card, that includes shipping. Needless to say, whenever I am not using my computer, EVERYTHING is unplugged. Several others in our User's Group have adopted the same practice. I hope this tale of woe helps someone else. Surge Supressors will NOT stand a powerful surge such as a close lightning strike. We never expected our tree to be hit since we live in a heavily wooded area, but it was. Oh, by the way, the tree had to be cut down the next day before it fell on the house. I guess God loves us.

USE OF COMMAND FILES

By Rick Felzien
West Jax 99er's

Now that we have covered the basics of the TI operating system, we can now go into the use of Advanced Diagnostics by Miller's Graphics (DIAGS).

Command files are made up of a group of DIAGS. commands and key codes which can be executed by the program to accomplish tasks concerning disk files while freeing you of the tedium of executing each and every item. For one-time operations, the Command Files are of little value unless you want to write one for the practice. Where the command files are extremely helpful is in the area of tasks that will be repeated from time to time. With a command file you can do these operations through DIAGS. while you have a cup of coffee or something.

As most of you know, each keystroke such as [FCTN 9] (BACK), which is used in DIAGS. to Exit Sector mode etc. has ASCII code of [15]. These are typed into the file to make the computer execute the keystrokes without our having to do them. Note the brackets around the number 15, they are used by DIAGS. to denote that the value is the code for a keystroke. Another note on brackets, a pair of empty brackets [] is DIAGS. way of detecting the end of a command file, so be sure to end all your command files with a pair of empty brackets.

In our Feb. issue, I listed a command file that I had written to make a DS/DD copy of PILOT99. Let us take this file and go through this process step by step. One of the first things we need is a spare copy of the PILOT99 disk and we also need a blank initialized disk set up for DS/DD. We also need DIAGS., the TI-Writer module and program set, and a disk to store the command file. In the Jan. issue we had a command file for converting the DIAGS. disk to DS/SD, this allows room on the DIAGS. disk to store our command files, which is convenient as it saves disk swapping to some extent. DIAGS. used DV/80 files, so the TI-Writer editor is ideal for writing our command files. A very important thing to remember is that when we save our file, we dont want the margin and tab info so we want to use the Print File rather tahn Save File as this will eliminate saving the unwanted data. Incidently, the reason that I wanted to make a DS/DD was to have room on the PILOT disk to save disk swapping all the time.

Now that we have what we need, the first thing to do is to fire up TI-Writer and select Editor. A couple of things to keep in mind that there are two Command Lines, each 26 characters long and we must write our command files accordingly. Also, when we Print File, the trailing spaces are stripped off, so at the end of a line, we need to use [32](SPACE) so that DIAGS. knows that there should be one at that point. Another important item is that the code [7] which is FCTN3 (erase) erases the command line or if we are in Edit Sector mode, it erases the sector data and zeroes it out so you could wind up writing a blank sector to the disk. Therefore, we must consider very carefully where to place these keystrokes in our file.

I have included the complete command file for you to refer to while reading the following information.

The first line of our file will be [255][7], this sets up high speed execution and clears the command line. Next, type in

Master in DSK1. [32] (places information on the first command line and denotes a space.)

Copy in DSK2. [32] (more info on line 2)

[253][253][253][253][253][7] (gives a 5 second delay and erases the command field.)

Beep PA [13][7] (executes a beep then pauses for you to continue by pressing a key. [13] is the Enter key. [7] erases the command lines.)

SD 2 FD 2 D PILOT99[13][7] (selects drive 2 and tells DIAGS. to format disk 2 for double sided D for double density with the name of PILOT99.)

SD 1 CR 0 36[32] (selects drive one Copy Reads sectors 0-36 then
SD 2 CW 0 36[13] selects drive 2 and Copy Writes the sectors.)

These commands are continued, increasing the sector numbers appropriately until all data is transferred from drive 1 to drive 2. It is not necessary to use [7] every time if the next command is the same length or longer as it will overwrite the previous one anyway. The [254] on the last line of copy commands sets up slow speed execution so you can watch the data being typed in for the changes to sector 0. After the copy commands we are ready to make the changes to sector 0.

Now we type in SD 1 ES 0 [13] which selects drive 1 and does an Edit Sector 0 then <ENTER> to execute. We next need twenty [9] (right arrow key to place the cursor in proper place to type in >05A0 (1440) the number of formatted sectors. Next we need a [10] (down arrow) then sixteen [8] (left arrow) to type in >0202 for double sided double density. Now we need six [10] and six [8] codes to type in the appropriate number of zeroes to indicate the unused sectors for a DS/DD disk. Now we use [15] (FCTN 9) to exit Edit Sector mode, then type in SD 2 WS 0 SD 1 [13][7][] to select drive 2 write sector 0 and go back to drive 1. Note the empty set of brackets to denote the end of our file.

Hopefully this has helped you to understand how to write a command file for the DIAGS. program. Any questions that you may have are welcome and I will try to give you a satisfactory answer.

```
[255][7]
     MASTER IN DSK1.      [32]
     COPY IN DSK2.       [32]
[253][253][253][253][253][7]
Beep PA [13][7]
SD 2 FD 2 D PILOT[13][7]
SD 1 CR 0 36[32]
SD 2 CW 0 36[13]
SD 1 CR 36 36[32]
SD 2 CW 36 36[13]
SD 1 CR 72 36[32]
SD 2 CW 72 36[13]
SD 1 CR 108 36[32]
SD 2 CW 108 36[13]
SD 1 CR 144 36[32]
SD 2 CW 144 36[13]
SD 1 CR 180 36[32]
SD 2 CW 180 36[13]
SD 1 CR 216 36[32]
SD 2 CW 216 36[13]
SD 1 CR 252 36[32]
SD 2 CW S5S 36[13]
SD 1 CR 288 36[32]
SD 2 CW 288 36[13]
SD 1 CR 324 36[32]
SD 2 CW 324 36[13][7][254]
SD 1 ES 0[13][9][9][9][9][9]
[9][9][9][9][9][9][9][9][9]
[9][9][9][9][9][9]05A0[10]
[8][8][8][8][8][8][8][8][8]
[8][8][8][8][8][8][8]0202[10]
[10][10][10][10][10][8][8][8]
[8][8][8]000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
00000000000[15]
SD 2 WS 0 SD 1 [13][7][]
```

32K ON THE 16 BIT BUS

By - John Clulow
Based upon ideas from Mike Ballmann

The following is a step-by-step description of how to add 64K of RAM memory on the 16 bit bus. The present modification uses only 32K. This corresponds to the memory space of the 32K Memory Expansion. The modification yields a speed increase of about 50%.

Mike Ballmann is currently working on a circuit to allow CRU decoding of the remaining 32K. This will open up a whole new area of software, including such possibilities as a real DOS which could be loaded into RAM from disk on power-up. The 32K modification described below can easily be modified for full decoding upon completion of Mike's work.

You will need two Hitachi HM62256LP-12 RAMs. One source of these is Microprocesors Unlimited. They cost around $12. You'll also need a 74LS21 and a 74LS153. These can be obtained from various electronics supply houses. All wiring should be done with wire-wrap wire. You should use a low wattage soldering pencil with a fine, pencil type tip.

The modification is done on the main board of the Black Silver console, and you'll need to refer to the Logic Board Component Location Diagram in the TI-99/4A Console Technical Data book.

1) Remove the board from the console, and identify the two ROMs. They are located between the GROM connector and the 9900 IC. One is parallel to the 9900 and the other is perpendicular to it. They are U610 and U611 on the Component Location Diagram.

2) Bend the pins on the HM62256 IC's closer so they will firmly contact the ROM pins when piggy-backed. One way of doing this is to place the RAM on it's side on a table and then move the body of the IC toward the table to bend the pins uniformly.

3) Bend out the following pins on both HM62256 RAMs: 1 2 20 22 23 26 27 28. These pins will NOT be soldered to anything on the ROMs. Holding the IC with the notch up and looking at the top, pin numbers start with pin 1 on the upper left, go down the left side, then across and up the right side. Pin 28 is opposite pin 1 on the end with the notch.

4) Place one HM62256 over the ROM that is parallel to the 9900. Make sure the notch points toward the 9900 and that the writing on the 9900 and the 62256 can be read from the same direction. Place the RAM such that pins 1 2 27 and 28 extend beyond the end of the ROM. The un-notched end of the RAM should line up with the un-notched end of the ROM. There should be a sort of "spring tension" that clamps the RAM pins onto corresponding ROM pins below it. This will help to insure good solder joints. If the RAM doesn't fit tightly, remove it and bend the pins closer.

5) Solder all RAM pins not bent out to the ROM pins below. Use a low wattage pencil with a fine, pencil type tip. Inspect each solder joint carefully in good light, under magnification.

6) Place the second 62256 on the ROM that is perpendicular to the 9900. The notch on the RAM points away from the 9900 and toward the edge of the board. As above, solder and inspect all pins that were not bent out.

7) Bend out the 74LS21 pins 1 2 4 5 6 8 10 12 14. Note that pins 1 and 14 are across from each other on this 14 pin IC.

8) The 74LS21 will be piggy-backed on the 74LS138 U504. This IC is located adjacent to the end of the board where the edge connector is. There are two 138's next to each other. U504 is the one nearest the end of the board. You will place the 74LS21 so that the UN-NOTCHED end lines up with the un-notched end of the 138 (pointing toward the cassette connector). Pins 1 and 16 of the 138 will extend beyond the notched end of the 74LS21.

9) Before positioning the 74LS21, solder 1/2" lengths of wire-wrap wire to the 138 pins 7 and 9. Then position the 74LS21 on top of the 138 and solder all pins not bent out to the 138 pins below and inspect the connections.

10) Bend out all of the 74LS153 pins EXCEPT 8 and 16.

11) Place the 153 over U613, a 74LS194. The notch will line up with the 194 notch and point toward the edge of the board away from the 9900. Solder pins 8 and 16 of the 153 to pins 8 and 16 of the 194 below.

12) At the end of the 9900 opposite to where the RAM's have been piggy-backed, you will see a line of three ICs. They are a 74LS00, 74LS32, and 74LS04. The 74LS00 is U606 and the 74LS32 is U605. Turn the board upside down so you can see the traces. Find the trace that runs from pin 11 of the 74LS00 (U606) to pin 13 of the 74LS32 (U605). Double check to make sure you're doing the pin numbering correctly. When you've found the trace, cut it with a knife so there is no continuity between the LS00 pin 11 and the LS32 pin 13.

13) Identify the piggy-backed RAM that is perpendicular to the 9900. Solder wire wrap wires connecting every bent out pin on this RAM to the corresponding bent out pin on the RAM that is parallel to the 9900. Pin 1 to pin 1, pin 2 to pin 2, etc. There will be eight wires in all to solder.

14) Solder wire-wrap wires to make the following connections on the RAM that is parallel to the 9900. Pin 1 goes to pin 24 of the 9900 (solder the wire to the 9900 pin on top of the board). Pin 2 goes to the 9900 pin 22. Pin 20 goes to two places. Connect pin 20 of the RAM to pin 22 of the RAM and also to pin 8 (bent out) of the 74LS21. There should be three wires coming off pin 20 of the RAM. Pin 23 of the RAM goes to pin 21 of the 9900. Pin 26 of the RAM goes to 23 of the 9900. Pin 27 of the RAM goes to pin 61 of the 9900 (fourth from the top on the right side). Finally, connect pin 28 of the RAM to pin 20 of the 74LS244 adjacent to the piggy-backed 74LS21.

15) Connect the following 74LS21 pins with a bare wire: 1 2 4 and 14. Connect the short wire from the 138 pin 7 to the LS21 pin 5 (bent out). Connect LS21 pin 6 to LS21 pin 12. Connect LS21 pin 8 (bent out) to the piggy-backed 153 pin 2. Connect the short wire coming from the 138 pin 9 to LS21 pin 10. Finally, connect the 74LS21 pin 14 to the 74LS244 pin 20 that you connected the RAM pin 28 to.

16) OK, we're almost done, so take a break and have a beer.

17) On the 153, connect pin 9 to pin 13 on the 74LS32 (U605). Pin 10 of the 153 goes to pin 14 of the 74LS74 next to it (U607). Also connect pin 10 of the 153 to pins 11 and 13 of the 153. Connect pin 12 of the 153 to pin 15 of the 153, and then connect pin 15 of the 153 to pin 7 of the 74LS00 U612 (next to the 74LS74). Connect pin 14 of the 153 to pin 11 of the 74LS00 U606; that's the one you cut the trace on.

18) That's it! Now have another beer before putting your computer back together. When you try it out, remember that this version isn't compatible with other 32K in the system.

If you have problems with this I can't promise I can help but feel free to give me a call or write EMAIL (419) 874-8838. Ask for John (or Hose-Head.)

DIJIT Systems, the San Diego based company that brought professional quality RGB display to the TI-99/4A, introduced its latest product at the 99/FEST-WEST/87 in Los Angeles: The Advanced Video Processor Card. The AVPC fits into the Peripheral Expansion Box and is compatible with existing TI99/4A software. It features 80 column text and advanced graphics with up to 512 colors. The AVPC also supports Mouse and Light Pen inputs. The DIJIT Systems card contains 192K of video RAM and is designed to work with the "DIJIT-EYEzer", an external Gen-lock and video digitizing accessory. It will allow titling and graphic overlays on home videos as well as computer manipulation of external video images. The DIJIT Systems AVDP gives the TI-99/4A video processing power comparible with the Atari ST and the Amiga. The product is scheduled for release in August for $195.00. DIJIT Systems 4345 Hortensia Street San Diego, CA. 92103 (619) 295-3301

```
100 DIM C$(4)
110 DIM BC(9,4)
120 CALL CLEAR
130 CALL SCREEN(2)
140 PRINT "               CRASH'
EM": : : : :
150 PRINT "             BY JIM B
ECK"
160 PRINT : : : : : : : :
170 PRINT "PRESS ANY KEY TO
START GAME."
180 FOR D=1 TO 14
190 CALL COLOR(D,16,2)
200 NEXT D
210 CALL KEY(0,K,S)
220 IF S=0 THEN 210
230 CALL CLEAR
240 FOR D=1 TO 14
250 CALL COLOR(D,2,2)
260 NEXT D
270 SCR=0
280 LT=0
290 CALL CHAR(120,"101010101
0101010")
300 CALL CHAR(121,"00000000F
F000000")
310 CALL CHAR(122,"00000000F
0101010")
320 CALL CHAR(123,"10101010F
0000000")
330 CALL CHAR(124,"000000001
F101010")
340 CALL CHAR(125,"101010101
F000000")
350 CALL CHAR(126,"000000001
0000000")
360 C$(1)="0010387C7C387C7C"
370 C$(2)="0000367EFE7E3600"
380 C$(3)="007C7C387C7C3810"
390 C$(4)="0000D8FCFEFCD800"
400 FOR D=1 TO 9
410 FOR DE=1 TO 4
420 READ BC(D,DE)
430 NEXT DE
440 NEXT D
450 RESTORE
460 FOR D=1 TO 4
470 CALL CHAR(127+D,C$(D))
480 CALL CHAR(135+D,C$(D))
490 NEXT D
500 FOR D=2 TO 12 STEP 2
510 CALL HCHAR(D,D+3,124)
520 CALL HCHAR(D,D+4,121,25-
(D*2))
530 CALL HCHAR(26-D,D+3,125)
540 CALL HCHAR(26-D,D+4,121,
25-(D*2))
550 CALL VCHAR(D+1,D+3,120,2
5-(D*2))
560 CALL HCHAR(D,29-D,122)
570 CALL VCHAR(D+1,29-D,120,
25-(D*2))
580 CALL VCHAR(26-D,29-D,123
)
590 CALL HCHAR(D+1,D+4,126,2
5-(D*2))
600 CALL VCHAR(D+1,D+4,126,2
5-(D*2))
610 CALL HCHAR(25-D,D+4,126,
25-(D*2))
620 CALL VCHAR(D+1,28-D,126,
25-(D*2))
630 NEXT D
640 FOR D=15 TO 17
650 CALL VCHAR(3,D,32,9)
660 CALL VCHAR(15,D,32,9)
670 NEXT D
680 FOR D=12 TO 14
690 CALL HCHAR(D,6,32,9)
700 CALL HCHAR(D,18,32,9)
710 NEXT D
720 CALL HCHAR(13,16,32)
730 CALL COLOR(12,8,2)
740 CALL COLOR(13,16,2)
750 CALL COLOR(14,10,2)
760 CALL COLOR(3,16,2)
770 CALL COLOR(4,16,2)
780 DIR=3
790 DOT=0
800 LT=LT+1
810 IF LT<10 THEN 830
820 LT=1
830 DC=BC(LT,3)
840 LC=BC(LT,4)
850 CALL HCHAR(13,16,LT+48)
860 RPC=BC(LT,1)
870 CPC=BC(LT,2)
880 ON DC GOSUB 1390,1420,14
50,1480
890 RMC=X1
900 CMC=X2
910 RP=23
920 LEV=5
930 OD=32
940 CP=17
950 DIR=DIR+1
960 ON DIR GOSUB 1390,1420,1
450,1480,1510
970 RM=X1
980 CM=X2
```

```
990 CALL GCHAR(RP+RM,CP+CM,F
R)
1000 IF FR=32 THEN 1070
1010 IF FR=135+DC THEN 2330
1020 IF FR<>126 THEN 950
1030 DOT=DOT+1
1040 SCR=SCR+5
1050 CALL SOUND(-10,330,9)
1060 IF DOT=180 THEN 1830
1070 CALL HCHAR(RP,CP,32)
1080 RP=RP+RM
1090 CP=CP+CM
1100 CALL HCHAR(RP,CP,DIR+12
7)
1110 CALL GCHAR(RPC+RMC,CPC+
CMC,FR)
1120 ND=32
1130 IF FR=127+DIR THEN 2330
1140 IF FR=32 THEN 1170
1150 IF FR<>126 THEN 1250
1160 ND=126
1170 CALL HCHAR(RPC,CPC,OD)
1180 OD=ND
1190 RPC=RPC+RMC
1200 CPC=CPC+CMC
1210 CALL HCHAR(RPC,CPC,135+
DC)
1220 IF RPC=13 THEN 2030
1230 IF CPC=16 THEN 2180
1240 GOTO 1320
1250 DC=DC-1
1260 IF DC>0 THEN 1280
1270 DC=4
1280 ON DC GOSUB 1390,1420,1
450,1480
1290 RMC=X1
1300 CMC=X2
1310 GOTO 1110
1320 IF CM<>0 THEN 1360
1330 IF RP=13 THEN 1530
1340 IF RP=13-RM THEN 1530
1350 GOTO 990
1360 IF CP=16 THEN 1680
1370 IF CP=16-CM THEN 1680
1380 GOTO 990
1390 X1=-1
1400 X2=0
1410 RETURN
1420 X1=0
1430 X2=-1
1440 RETURN
1450 X1=1
1460 X2=0
1470 RETURN
1480 X1=0
1490 X2=1
1500 RETURN
1510 DIR=1
1520 GOTO 1390
1530 CALL JOYST(1,X,Y)
1540 CP=CP+X/2
1550 IF CP=16 THEN 1660
1560 IF CP=4 THEN 1660
1570 IF CP=28 THEN 1660
1580 CALL HCHAR(RP,CP-(X-2),
32)
1590 IF X=0 THEN 1610
1600 CALL SOUND(-40,220,3,22
2,3)
1610 IF CP>16 THEN 1640
1620 LEV=LEV-X/4
1630 GOTO 1650
1640 LEV=LEV+X/4
1650 GOTO 1080
1660 CP=CP-X/2
1670 GOTO 990
1680 CALL JOYST(1,X,Y)
1690 RP=RP-Y/2
1700 IF RP=13 THEN 1810
1710 IF RP=1 THEN 1810
1720 IF RP=25 THEN 1810
1730 CALL HCHAR(RP+(Y/2),CP,
32)
1740 IF Y=0 THEN 1760
1750 CALL SOUND(-40,220,3,22
2,3)
1760 IF RP>13 THEN 1790
1770 LEV=LEV+Y/4
1780 GOTO 1800
1790 LEV=LEV-Y/4
1800 GOTO 1080
1810 RP=RP+Y/2
1820 GOTO 990
1830 CALL CLEAR
1840 CALL SOUND(1000,262,0,3
30,0,392,0)
1850 CALL COLOR(3,2,2)
1860 CALL COLOR(4,2,2)
1870 PRINT "          YOU DID
 IT!!"
1880 PRINT : : : : : :"     L
EVEL";LT;"CLEARED."
1890 PRINT : : : : : :
1900 PRINT " PRESS ANY KEY T
O CONTINUE."
1910 DOT=0
1920 FOR D=1 TO 14
1930 CALL COLOR(D,16,2)
1940 NEXT D
1950 CALL KEY(0,K,S)
```

```
1960 IF S=0 THEN 1950
1970 CALL CLEAR
1980 FOR D=1 TO 14
1990 CALL COLOR(D,2,2)
2000 NEXT D
2010 SCR=SCR+(100*LT)
2020 GOTO 500
2030 IF LEV=LC THEN 1320
2040 IF CPC>16 THEN 2070
2050 CTR=-2
2060 GOTO 2080
2070 CTR=2
2080 CALL SOUND(-40,770,4)
2090 IF LC<LEV THEN 2140
2100 LC=LC-1
2110 CPC=CPC-CTR
2120 CALL HCHAR(RPC,CPC+CTR,
32)
2130 GOTO 1320
2140 CPC=CPC+CTR
2150 LC=LC+1
2160 CALL HCHAR(RPC,CPC-CTR,
32)
2170 GOTO 1320
2180 IF LEV=LC THEN 1320
2190 IF RPC>13 THEN 2220
2200 CTR=-2
2210 GOTO 2230
2220 CTR=2
2230 CALL SOUND(-40,770,4)
2240 IF LC<LEV THEN 2290
2250 LC=LC-1
2260 RPC=RPC-CTR
2270 CALL HCHAR(RPC+CTR,CPC,
32)
2280 GOTO 1320
2290 RPC=RPC+CTR
2300 LC=LC+1
2310 CALL HCHAR(RPC-CTR,CPC,
32)
2320 GOTO 1320
2330 CALL HCHAR(RP,CP,32)
2340 CALL HCHAR(RP+RM,CP+CM,
32)
2350 CALL HCHAR(RPC+RMC,CPC+
CMC,32)
2360 CALL HCHAR(RPC,CPC,32)
2370 CALL SOUND(-1000,-7,0)
2380 FOR D=1 TO 6
2390 FOR DE=1 TO 4
2400 CALL HCHAR(RP,CP,127+DE
)
2410 NEXT DE
2420 NEXT D
2430 CALL HCHAR(RP,CP,32)
2440 CALL CLEAR
2450 PRINT "        C R A S H
 ! !"
2460 PRINT : : : : : : : :
2470 PRINT "    YOUR SCORE I
S ";SCR
2480 PRINT : : : : : : : :
2490 PRINT "PRESS ANY KEY TO
 PLAY AGAIN."
2500 FOR D=1 TO 14
2510 CALL COLOR(D,16,2)
2520 NEXT D
2530 CALL KEY(0,K,S)
2540 IF S=0 THEN 2530
2550 CALL CLEAR
2560 FOR D=1 TO 14
2570 CALL COLOR(D,2,2)
2580 NEXT D
2590 GOTO 120
2600 DATA 23,16,2,5,13,6,1,5
,12,8,1,4,12,10,1,3
2610 DATA 21,15,2,4,15,15,2,
1,17,15,2,2
2620 DATA 9,17,4,2,14,22,3,3
```

BOARD MEETING MINUTES

by GEORGE LEMPEOTIS

SECRETARY

*** Hello everybody, it will be good to see you all back after our long summer vacation. The summer meetings were pretty slow. I hope everybody is ready to get back to their computers after this long hot summer of 1987. Our TI K*I group still has about 25 members and sufficient funds to operate for some time ahead. I will now try to give a brief summary of our June and August board meetings. At the June board meeting it was decided to let any group member participate in the upcoming bimonthly board meeting (which will be held at noon before a main meeting), thus making all group members board members. The members present elected officer as follows. Chairmen- Bev Cook Vice Chairmen- Bruce Shear Secretary- Membership- George Lempeotis Librarian- Treasurer- Wyane Willimgton NewsLetter Editor- Mark Harms All members are welcome to the bimonthly board meeting at noon, and the K*I TI group needs the help and input from everybody. At the August board meeting the members decided on what to demo at the September and October main meeting. The September 19, 1987 meeting will feature a demo of Quality Soft's Basic Converter by Wyane Willimgton. The October 17, 1987 meeting will feature a demo of the Ryte Data's Basic Compiler by Mark Harms and will have a board meeting at noon. These two meeting should show us how to speed up our basic programs by the use of a basic compiler. A basic compiler converts interperted basic to straight run machine code, increasing program runing speed manytimes. We should get a sample of what is avaible and how a compiler works for the TI at the next two meetings. The September and October meeting should be very interesting, do not miss them. Hope to see you at the next meeting. ***

```
100 ! PROGRAM BY CHRIS SCHRAN
110 CALL CLEAR
120 OPEN #1:"DSK1.X"
130 DATA 71,64,72,65,70,75,7
3,70,76,67,66,66,65,68,76,68
,77,68,78,71,77,66,68,66
140 DATA 66,67,74,67,74,77,7
4,68,73,71,64,67,72,68,76,65
,72,68,76,65
150 CALL INIT
160 CALL PEEK(-28672,A)
170 IF A<>96 THEN 290
180 FOR Z=1 TO 11
190 PRINT #1:""
200 FOR X=1 TO 4
210 READ A
220 CALL LOAD(-27648,A)
230 NEXT X
240 CALL LOAD(-27648,64)
250 CALL LOAD(-27648,80)
260 NEXT Z
270 CLOSE #1
271 DELETE "DSK1.X"
280 STOP
290 PRINT "YOU DON'T HAVE A
SPEECH":"SYNTHESIZER ATTACHE
D"
```

****

From the SNUGLETter - April 1987
From the disk of Mike Dodd.

DM1000...has an annoying little bug if you happen to own a CorComp disk controller. When DM1000 formats disks in double density, it puts 16 sectors/track on the header, even though it formats 18 sectors/track, which is all very fine and well if you keep it on a CorComp controller, for the reason that the CorComp controller never even heard of 16 sectors, so it doesn't care what the header says. HOWEVER, if you send the disk to someone who has a MYARC disk controller, then the MYARC controller looks at the header and sees "16 sectors per track." So it reads the disk based on that information. But it's 18 sectors per track! So, the MYARC reports a blank disk. After having several people complain about my "blank" disk, I found a fix for DM1000. For V3.5, edit the first sector of the MGR1 file. At byte 216, you should see (in hex):
10 00 02 D0 00 5A

Change the 10 to 12 . Write the sector back out to disk, and never worry about it again. If you are using another version of DM1000 that has the same problem (I don't know if any others do), search for the above bit of code. It should be very close to the beginning.

It is also possible to change those disks formatted as double density by changing the sectors per track information in the first sector (Sector 0). Once again a sector editor is necessary. Sector 0 is called up for editing and byte 12 (decimal) or C (hex) is changed from the existing 10 to 12 (both in hex).

NOTE: MYARC disk controller users should not reset the sector 0 byte from >10 to >12! If you do, the system will assume it is 18 sectors per track. In that case, you will see the same problems that we CorComp users have seen.