Kings Byte

The Kings County
TRS-80 Users Group

c/o MORTY LIBOWITZ
1063 EAST 84 STREET
BROOKLYN, NY 11236
(212) 763-4233

MAY 1982

KINGS BYTE NEWSLETTER

VOLUME 1 NUMBER 12

KINGS BYTE OFFICERS

President:	Steve Abramowitz
Vice President:	Kal Bergen
Treasurer:	Sid Epstein
Secretary:	Bernie Warren
Communications Officer:	Morty Libowitz

Editor:	B. Warren
Arts Editor:	Al Hartman
Technical Editor:	Kal Bergen

PRINTING: MIDWOOD LEAGUE FOR THE LEARNING DISABLED - A SUBSIDIARY OF VIP FOUNDATION

AREA TRS-80 USER GROUPS

CLUB	TRS-80 GROUP	CONTACT	
METRO	TRUG	BARRY KORNFELD	(212)675-3831
LICA	LITRUG	FRED KEMPLER	(212)229-4040 EXT 6753

AREA TRS-80 BULLETIN BOARDS

BRIAN BOYLE'S B B	(212)935-9459
CONNECTION 80 QUEENS	(212)441-3755
CONNECTION 80 MANHATTAN	(212)991-1664
SISTER STATEN ISLAND	(212)442-3874
CONNECTION 80 BROOKLYN	(212)277-5831
PROGRAMS UNLIMITED	(516)334-3134
INNOVATION 80	(516)482-8491
CONNECTION 80 HQ	(516)588-5836
PHOTO 80	(201)790-6795
BULLET 80 HQ	(203)744-4644
CONN 80 STONY PT	(914)942-2638
BULLET 80	(914)277-0665
WE ARE COMPUTERS	(914)592-5385

KINGS BYTE MEETINGS

Our meetings take place on the first Monday of each month at BAM (B'klyn Academy of Music) located near the Atlantic Ave. and Pacific St. subway stations. It is also near the L.I.R.R station.

Meetings start at 7:00 PM. Assembly language course at 7:30 PM. Lectures and/or demonstrations start at 8:15 PM. For up-to-date information call our Official Contact, Morty Libowitz at (212)763-4233.

Presidents Keyboard

by

Steve Abramowitz

Today has to be a loser day. While in the course of typing a brief for my BIG case in the office, I decided to make a label up for the disk with the brief. Woe is me! The standard one across labels from the Shack have worked just fine so far. Little did I know what grief that one label would cause me.

The procedure I have followed for about 50 disk labels is relatively satisfactory, and using one of ABRAMOWITZ's amendments to Murphy's infamous law, "If it WORKS don't fix it!", I decided to use the same procedure. Another amendment? Sure, "Why do we do it THIS way? Because it is harder."

I have written a program that I call EPSON/SET which does a lot of things for me, and my FT. It will print a letterhead, send a top of form to the EPSON, inquire if I want ITALICS (Why not? I've paid for the Graftrax modifcation and I like the looks on a disk label), and it will send control codes to the printer for text or for a BASIC program listing. The listing routine provides for margins and has an option for indent at each colon in the BASIC program to be listed.

(The other day, I showed this program to Walter Gorkowitz, on his machine, and he asked how he could stop the italics. I told him, "You can't, you have italics forever." Actually, there are two ways, either turn the printer OFF and then ON which resets the defaults, or send an ESCAPE and a "5" to the printer.)

When all this is over, it goes into a Menu of 14 options including calling various versions of SCRIPSIT, looks at directories, and if required will reconfigure my PDRIVE under NEWDOS 80. If it is running under LDOS or MULTIDOS, it can activate their fine printer sPoolers. A graceful exit to DOS or BASIC is also available, as is VFU, (like BFU/TRS but under double density, if you are running under MULTIDOS). Enough "and then I wrote".

Now to the business at hand, writing a disk label. Drats! My 40-track Tandon is on the fritz for the umpteenth time. Well, I still have the other 40 and old reliable MPI 80. Epson Mode is told to send out the control codes to the printer so that it will print in italics, compressed, normal print, and 7/72" line spacing. I then go to my PCHANGE program to change the PDRIVE so it will read a 40-track disk. (When am I going to stop making up data disks so that I don't have to rely on my 80 tracker to run as a forty?). Exit to DOS and issue DIR :2 P.

Oh no! A label has become unstuck and has adheared to the paper holder by the print head. I must get that label off! OK, it's off now, but a little piece has to be removed from the metal between the print head and the roller. I'll have to remove those tiny holddown screws. Lost the two screws.

No label, no TANDON, no EPSON. Murphy, whoever you are, I'll never write an amendment to your law again, never again, I promise!

For the May meeting on May 3 at 8AM, we will have another panel discussion, "Model I and III Disk operating Systems." Kal will continue his Assembly Language course for beginners.

Not so happy computing - Steve

P.S. I just called Bernie, our unsung but outstanding newsletter editor, who told me this story is too late for the April issue. Try this then, does anybody want a story about Morty's latest coffee-keyboard fix? Now that he is running MULTIDOS with the AM Multiplier, he is using TEA instead. As they say in Robot Attack, "Chicken!".

Murphy, I hope you get eaten by your gold bug!

***** ***** ***** ***** *****

THE TRS-80 MODEL I AND THE DISK DRIVE - PART II

by Bernie Warren

TRS-80 DISK FILE MANAGEMENT

GENERAL

The directory, located on track 17, is the key to disk file management. It uses all 10 sectors on this track. All of the information that is needed for the DOS (disk operating system) to perform its file management functions is located on this track.

HOW THE DISK IS ORGANIZED

Since the typical disk drive for the TRS-80 Model I that is now sold is a 40-track drive, I will use this as my model. The magnetic make-up of the disk is as follows:

Tracks	40
Sectors/track	10
Sectors/disk	400
Sectors/granule	5
Bytes/sector	256
Bytes/disk	102,400
Grans/track	2
Grans/disk	80

GAT SECTOR

This is sector 0 of the directory (on track 17). GAT stands for GRANULE ALLOCATION TABLE. This is where the DOS allocates space, in grans, for files. The information is in hex code. Also in this sector is where DOS indicates tracks that were locked out during format. SUPERZAP was used to display the GAT sector below.

```
111000  FFFF FFFF FCFE FCFC FCFC FCFC FCFF FFFF ................
111010  FFFF FFFF FFFF FFFF FCFC FCFC FCFC FCFC ................
111020  FCFC FCFC FCFC FCFC FCFC FFFF FFFF FFFF ................
111030  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
111040  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
111050  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
111060  FCFC FCFC FCFC FCFC FCFC FCFC FCFC FCFC ................
111070  FCFC FCFC FCFC FCFC FCFC FCFC FCFC FCFC ................
111080  FCFC FCFC FCFC FCFC FCFC FFFF FFFF FFFF ................
111090  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
1110A0  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
1110B0  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
1110C0  FFFF FFFF FFFF FFFF FFFF FF82 0000 E042 ...............B
1110D0  4EF4 544E 414D 4544 3030 2F30 302F 3030 NOTNAMED00/00/00
1110E0  0DFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
1110F0  FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF ................
```

Note that at the extreme left of the display, there are six columns of characters that are bunched together. For each row of the six columns, the first character is the drive number, the next two indicate the track number, in HEX, the next character is the relative sector number, and the last two indicate the relative byte position for the first byte of machine code that is just to the right of these six columns. The right-most two charactors of the first row of the six columns are 00. This indicates that the first byte of this sector, just to the right of 00 is in relative byte position 00. The next byte, to the right of the 00 position, is in relative byte position 01, etc. Just below 00 is 10, and this indicates that the byte just to its right is in relative byte position 10, etc. This explanation is valid for any sector of a disk with the NEWDOS+ system, or the TRSDOS system.

The first three rows of code, excluding the six columns just covered, indicate which granules are free and which are occupied with data. Each byte represents one track. (Each byte of code contains two characters.) The status of each track would be shown by the code below:

FC	1st and 2nd granules free
FD	1st granule allocated
FE	2nd granule allocated
FF	1st and 2nd granule allocated

In the dispayed sector, the last FC byte on row 3 is the 42nd byte, starting from byte 00, and following this are six consecutive FF bytes. This indicates that the disk has 42 tracks, and the last six consecutive FF's indicate that there is space here in row 3 for six more tracks, if they can be formatted physically on this disk. If this had been a 35-track disk, there would be 15 consecutive FF bytes at the end of row 3, leaving 35 bytes of code for 35 actual tracks.

Starting at relative byte 60, the first 42 bytes look similar to those indicated above. DOS places the TRACK LOCKED OUT information here. The code used can be one or both of the bytes FC and FF. FC means that the corresponding track is locked out, and FF means that it isn't.

The code at relative bytes CE and CF is what is called HASH CODE, and this gives the DOS the master disk password when it is needed. Starting at relative byte D0 is a full line of code for the disk name and the date that the disk was first used. To the right of this, in the ASCII section is the name and date in ASCII. The next two lines is where the COMMAND FILE for the AUTO function is located if this function is used. If the relative byte at E0 is 0D, then this function is not used.

HIT SECTOR

This is sector 1 of the directory on track 17. HIT stands for HASH INDEX TABLE. This is where a HASH CODE is stored for each file on the disk. From this code, DOS obtains information as to where on the directory track is the file management information for these files. In other words, each non-zero entry is the coded filename for a directory entry, and its location in this sector gives the location of the FPDE entry in the directory (FPDE will be explained later). This HASH CODE is used by DOS to find the location of each directory entry very rapidly. An actual hit sector is shown below, and again this was displayed with SUPERZAP.

```
111100   A2C4 0000 CC00 0000 0000 0000 0000 0000 ................
111110   0000 0000 0000 0000 0000 0000 0000 0000 ................
111120   00E1 0000 0000 0000 0000 0000 0000 0000 ................
111130   0000 0000 0000 0000 0000 0000 0000 0000 ................
111140   0000 00D6 0000 00DC 0000 0000 0000 0000 ................
111150   0000 0000 0000 0000 0000 0000 0000 0000 ................
111160   0000 00DE 0000 0000 0000 0000 0000 0000 ................
111170   0000 0000 0000 0000 0000 0000 0000 0000 ................
111180   0000 0000 0000 0000 0000 0000 0000 0000 ................
111190   0000 0000 0000 0000 0000 0000 0000 0000 ................
1111A0   0089 0000 0000 0E00 0000 0000 0000 0000 ................
1111B0   0000 0000 0000 0000 0000 0000 0000 0000 ................
1111C0   000A 0000 F600 00AB 0000 0000 0000 0000 ................
1111D0   0000 0000 0000 0000 0000 0000 0000 0000 ................
1111E0   4D99 0000 0000 0000 0000 0000 0000 0000 M...............
1111F06 0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Since each byte of code has two characters, there are 16 vertical columns of code, which includes 00's. Any byte except 00 is a HASH CODE. They are bunched in columns of two bytes (four characters) to make it easier to read, so each column has two characters. Only the first 8 columns are used. Note that the second 8 contain all zeros. These first 8 columns represent the remaining 8 sectors of the directory track (relative sectors 2-9). Also, only every other row is used, and the rows that aren't used contain all zeros. For this HIT sector, some of the useable rows are not used. There is room for a maximum of 8 bytes of code on each useable line (row), and with 8 such lines, there could be 64 bytes of codes for a possible 64 files.

Again, the 8 columns of code, indicated previously, represent the 8 remaining sectors of the directory track (relative sectors 2-9). The position of the bytes (called HASH CODE), with respect to the columns they are in, is important. This indicates in which of sectors 2-9 the directory data for this byte's file is in. For example, if a hash code is in column 3, 3 is the number used to determine in which sector the file information is in. (How to find the ACTUAL sector will be shown later.) Each of these bytes represents a specific file on the disk.

Look at the hash codes on the first row (line). There may be 8 or less depending on how many files DOS placed in these positions. Because they are on the line with 00, to the left, the directory information, for the files they represent, starts at relative byte 00 in their respective sectors, and it is said that this code points to the location of the starting byte for the file in its directory sector. Skipping the line of zeros below, look at the second line of hash codes. The directory information, for the files they represent, starts at relative byte 20 in their respective sectors, etc.

Now there will be an explanation as to how to determine the ACTUAL sector that all this file information is in. If a hash code is in the first column, column 1, then the file information is in the first sector AFTER THE HIT SECTOR, which is relative sector 2. The HIT sector is relative sector 1, so relative sector 1 plus 1 equals relative sector 2. If the hash code is in column 5, then the sector is relative sector 6, etc. Just add 1 to the column number. When using this method, remember that the columns of machine code entries start with the numbering of column 1, not 0. To be continued ...

LSCRIPT/CMD

by

Steve Abramowitz

(C) 1982 Reproduction by written permission only.

I am using a new version of SCRIPSIT that comes as a patch with

LDOS 5.1.1. The information below is for the Model I only. This will explain the SCRIPSIT/LC patch called LSCRIPT/FIX. The purpose of the patch or Zap is to permit:

1) Operation with LDOS keyboard (*KI) and its Keystroke Multiplier (KSM) drivers and with Mini Dos.

2) Use of all the other ASCII keys such as []\^_{}| (great to write out math formulas for example).

3) Use of the clear key as control key (not the @) with the numbers as control characters.

4) Send out control codes to the printer within Scripsit.

The New Control Keys

SHFT DWN ARR. Clear key is CTRL

S or	CTRL 1=insert
X or	CTRL 2=line
D or	CTRL 3=delete
	CTRL 4=word
Q or	CTRL 5=block
W or	CTRL 6=window
E or	CTRL 7=exchange
R or	CTRL 8=repeat
A or	CTRL 9=long left arrow
F or	CTRL 0=long right arrow
C or	CTRL :=paragraph
V or	CTRL -=page
Z	End of text or Block
	SHIFT-CLEAR=Mini Dos Enter a letter F,Q,K,T,or C
	SHIFT-ENTER =SPECIAL COMMAND Mode
	SHIFT-left-CLEAR=TAB
	SHIFT-DWN * = Screen Print of Text (JKL)

Special Command Mode

P	Prints with P,I prints invisible control char.
D	(with Drive No.) Displays Directory
F>	Finds text
?I	Paragraph Indent
?M	prints memory size available
?L	Document length
?N	Current file name
?W	Video line width
S	Save current file name on drive no indicated
L	Loads file designated with drive no. use
L,A S,A	For loading and saving file in ASCII format

SHFT DWN ? Displays end block which may be useful with F> (undocumented)

LSCRIPT/CMD should be used with *KI KI (TYPE) enabled, which allows TYPE ahead with KSM, which permits dynamic redefinition of the letters. LSCRIPT/CMD allows all other ASCII letters by typing CLEAR with:

```
                         , . / ;  < > ? + @
Which results in:
                         [ ] \ ^ { } ! ~ '
CLEAR enter =            __Underline________
```

To exit to DOS READY in the Special Command mode, type END. If it happens that you exit or reboot without saving your file, don't worry! I even heard it happened to KAL, once. (He claims it was just once.)

To attempt a restore of a lost LSCRIPT/CMD file and the previous text from memory, while in DOS, don't issue any other prior DOS commands, unless you must (such as enabling double density). Be sure to suppress AUTO, if you have an Auto command on drive 0 and have to reboot. NOW type LSCRIPT/CMD * . It works just like BASIC * .

Printer format commands : >$value,value,value

You must place all format commands immediately following a text boundry. The printer format line can contain either a comma or a space between each decimal value. The printed output ignores these lines, unless P,I is used to print the otherwise invisible commands. The value can be any decimal string from 0 to 255.

***** ***** ***** ***** *****

USEFUL USER INFORMATION

by

Bernie Warren

MX-80 GRAPHICS AND THE TRS-80 MODEL I

In order to print anything on a printer, every character from RAM memory is sent through the printer cable in the form of a code made up of numbers. This code is the ASCII code and the code can be sent either as part of a program, as text, or as a BASIC command in the command mode. However, to print a particular character, the computer must be able to send its code.

With respect to printing graphics characters, they can be printed only if the graphics is built into the printer. The MX-80 can print all the graphics characters of the TRS-80 Model I. In other words, any program with SET and RESET statements can be printed with the MX-80.

The MX-80 graphics are the same as those in the Model I, so the MX-80 prints them. However, the codes for the matching graphics of the MX-80 and the Model I are not the same. Therefore, if Model I graphics codes were used, the graphics printed on paper and screen would be different. However, If 32 is added to each of the Model I graphics codes, then they will be the same as the MX-80 graphics codes.

In practice, this is what is done. Another method is to move switch 2-4 inside the printer to the on position, but this removes many of the fine features of the MX-80.

SCREEN PRINTING

TRS-80 Model I users can print the screen with either NEWDOS PLUS or NEWDOS/80, using the JKL option. Howver, any graphics characters on the screen are printed as dots. Now, even the graphics characters can be printed if the user has an MX-80, and if SUPERZAP is used to change the machine code in the SCREEN PRINT routine in NEWDOS PLUS, NEWDOS/80 version 1.0 and NEWDOS/80 version 2.0. The new code adds 32 to the graphics code for the Model I.

This information appeared in the FAIRFIELD COUNTY TRS-80 USERS GROUP NEWSLETTER for June 1981, and it was provided by Joe Simon as it applies to NEWDOS/80. The same information as it applies to both NEWDOS PLUS and NEWDOS/80 appeared in 80 MICROCOMPUTING, November 1981 issue, and it was provided by Herbert S. DuBois. The information with respect to NEWDOS/80 version 2.0 appeared in 80 MICROCOMPUTING, December 1981 issue and was provided by Burgin L. Howdeshell.

Before any ZAPS are entered, switch 2-4 inside the computer must be in the off position as shown in the MX-80 manual. I have outlined the procedure for both NEWDOS PLUS, NEWDOS/80 version 1.0 and NEWDOS/80 version 2.0 below.

FOR NEWDOS PLUS

1. With the NEWDOS PLUS system disk in drive 0, run SUPERZAP.
2. Use the DD command to bring up track 0, relative sector 6.
3. Use the MOD command to change the following relative bytes:

Relative byte:	7F	80
FROM:	3E	2E
TO:	C6	20

4. press ENTER
5. Answer Y to the prompt. The modification is entered.

FOR NEWDOS/80 VERSION 1.0

1. With the NEWDOS/80 verson 1.0 system disk in drive 0, run SUPERZAP.
2. Use the DD command to bring up track 0, relative sector 7.
3. Use the MOD command to change the following relative bytes.

Relative byte:	B9	BA
FROM:	3E	2E
TO:	C6	20

4. Press ENTER
5. Answer Y to the prompt. The modification is entered.
6. Go into DOS mode and use SYSTEM command:

```
SYSTEM:0 AK=N
```

7. Press ENTER.
8. The TRS-80 graphics option is turned off.

FOR NEWDOS/80 VERSION 2.0

1. With the NEWDOS/80 version 2.0 system disk in drive 0, run SUPERZAP.
2. Use the DFS command to bring up SYS3/SYS, relative sector 4.
3. Use the MOD command to change the machine code starting at relative byte B9.
4. Type the following code:

```
FE 80 38 02 C6 20 47 3A 70 43 B8
78 30 02 3E 2E CD 3B 00 C3 85 51
```

5. Press ENTER.
6. Answer Y to the prompt. The modification is entered.

Now, when the JKL option is used, the graphics characters will be printed also.

Application for membership in KINGS BYTE TRS-80 USERS GROUP

You don't have to live in Brooklyn to become a member.

NAME..

ADDRESS...

CITY........................STATE...........ZIP..........

Enclosed is fifteen dollars ($15) for membership in KINGS BYTE TRS-80 USERS GROUP.

Make check payable to: KINGS BYTE COMPUTER CLUB.

Send this form (or copy) to:

TREASURER
KINGS BYTE TRS-80 USERS GROUP
c/o MORTY LIBOWITZ
1063 EAST 84 STREET
BROOKLYN, NY 11236

PM
5 MAY
1982

Ralph Bunche
USA
20c

KINGS BYTE NEWSLETTER
c/o MORTY LIBOWITZ
1063 EAST 84 STREET
BROOKLYN, NY 11236

FIRST CLASS MAIL

FIRST CLASS MAIL