

SOUTH BAY TRS-80® USERS GROUP



©CONTENTS COPYRIGHT © S.B.U.G. 1983

# DYNAMIC MEMORIES™

\* Tandy Corp./Radio Shack Inc.

MAY 1983



## South Bay TRS-80 Users Group

### Page - Contents

- 2. THE EDITOR'S TWO BYTES
- 3. HOW TO MEASURE A DISKETTE
- 4. PATCHING TRSDOS FOR MODEL III
- 6. SUMMER COMPUTER CLASSES
- 7. THE ELECTRONIC CLOSET EC-1 CHARACTER ROM
- 10. MX-80 GRAPHIC PICTURES
- 13. BASIC PROGRAMMING
- 15. GRAPHIC PROGRAMS AVAILABLE ON SBUG-80

SBUG meetings are held the third Tuesday of each month in the north east corner of Dysan's building at:

Time - 7:15 to 10:30 PM                      5401 Patrick Henry Drive  
Santa Clara, Ca

May 17, June 21, July 19

### Topic of the month:

Computer Media for the TRS-80

Features: The meeting will be chaired by Sabri Kawash.

- 1) Discussion on hard disk and other TRS-80 media.
- 2) Random access of questions and answers.

The June meeting will feature hardware modifications chaired by Eric Brewer. Don't forget the first Wednesday meeting on June 1st to discuss programming techniques, problems and answers. Bring your computer.

=====

### Send Newsletter articles to:

c/o Robert Byrd, Editor  
South Bay TRS-80 User Group  
P.O. Box 60116  
Sunnyvale, Ca 94088

Deadline for the June newsletter is; 25 May 83. Please, send articles via modem, or saved on disc/tape. I will see that your media is returned to you. Thanks . . .

=====

COPYRIGHT (C) 1983 SOUTH BAY TRS-80 USERS GROUP (SBUG).  
WORLD RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY  
BE STORED IN A RETRIEVAL SYSTEM, TRANSMITTED, OR  
REPRODUCED IN ANY WAY, INCLUDING BUT NOT LIMITED TO,  
PHOTOCOPY, PHOTOGRAPH, MAGNETIC OR OTHER RECORD, WITHOUT  
PRIOR WRITTEN PERMISSION OF THE EDITOR.

**\*\* Your Steering Committee \*\***

	Member	SBUG-80 Username	Phone
<b>Discussion Leaders:</b>			
	Mike McHenry	(MCHENRY)	(408) 245-4704
	Eric Brewer	(BREWER)	(408) 252-9332
	Sabri Kawash	(SABRI)	(408) 732-5484
<b>Treasurer:</b>			
	Larry Gunderson	(TREAS)	(408) 259-5349
<b>Newsletter Editor:</b>			
	Robert Byrd	(EDITOR)	(408) 732-6775
<b>Librarians:</b>			
(disc)	Gerry McKee	(GMCKEE)	(408) 926-4063
(tape)	Gary Dixon	(GDIXON)	(408) 262-6937
(doc.)	Bill Richerson		(408) 257-8267

**\*\* Other key individuals \*\***

<b>Assistant Editor:</b>			
	Dave Fox	(DAVEFOX)	(408) 255-3270
<b>Public Relations:</b>			
	Ron Carpenter	(RON)	(415) 726-3487
<b>Host Computer:</b>			
	SBUG-80	(EVERYONE)	(408) 737-SBUG
<b>SBUG-80 Sysops:</b>			
	Robert Byrd	(EDITOR)	(408) 732-6775
	Eric Brewer	(BREWER)	(408) 252-9332

If the need arises, feel free to give one of us a call.

=====

**THE EDITOR'S TWO BYTES**

Now that Radio Shack has come out with two new excellent computers the Model 100 and the Model IV, let's see more articles concerning these machines. There has been much talk about the new machines on SBUG-80, so I am aware that there are many members who have bought either of these machines.

SBUG-80 will soon have direct support for the Model 100 and of course the Model IV is all ready supported due to its ability to act like a Model III. I guess the Model I/III are now obsolete, but my Model I still works very well. And I have yet to learn all its' ability! How can I keep up? Dynamic Memories could help...

Robert Byrd  
EDITOR @ SBUG-80

# HOW TO MEASURE A DISKETTE

The most closely guarded secret in the magnetic media industry is the iron oxide coating formulation covering the surface of your floppies.

How well a WRITE/READ operation works depends a great deal on the ability of the magnetic particles to realign themselves when a WRITE current passes through the disk drive head.

I can't tell why one coating does a better job than another, but a test can be used to give some comparison of the performance of floppy diskettes.

The Achilles' heel of a double sided drive is the drive's ability to WRITE and READ on side 1 of track 39. That's the upper head on the last of a 40 track formatted diskette (0 to 39). This is where a READ or VERIFY error will most likely to occur and where you need maximum performance from both the floppy diskette and the disk drive.

To get on with our floppy test, assorted diskettes were tested using a TANDON TM100-2 drive, a disk drive exerciser, and oscilloscope. A 2F pattern (all ones) is written on side 1, track 39 and the output of the READ amplifier measured and recorded as Eout.

The following data seems to indicate the recording ability of a diskette. Keep in mind, the tests are not conclusive and may have differing results with another sample lot.

MANUFACTURER	PART NUMBER ID	Eout	COMMENTS
3M Scotch #1	745-O-RH	166 mV	
3M Scotch #2	745-O-RH	160	
Radio Shack	26-0305	136	
Maxell	MD2/DD	168	Rated for 96 tpi
Maxell	MD2/D	136	
Dysan	104/2D	144	
Verbatim	MD5500118191	144	
Sentinel #1	no part number	140	Single sided rating
Sentinel #2	no part number	140	Single sided rating
Control Data #1	1241-00	150	See text
Control Data #2	1241-00	156	See text
Generic #1	no name or no.	152	
Generic #2	no name or no.	146	

Notice that the Maxell MD2/DD has the highest output. I would say, the Maxell is rated higher, because it is rated for a more demanding format of 80 tracks on a side. Any of the other manufacturers may have an equally high

May 1983

reading if rated as such. Except as noted, the other diskettes are rated for 48 tpi (tracks per inch). Some of the diskettes were identified as "single side", but they have the READ/WRITE slot on both sides. The Control Data diskettes are identified for use as single side, single density, and 35 track.

As with anything, one measurement doesn't tell the whole story. A quality diskette has many other attributes such as surface smoothness, lubrication, certification, and the jacket as well.

As you can see, the sampling is a grab bag of floppies and perhaps an unfair test for some manufacturers. Given (or loaned) a broad spectrum of diskettes, a more comprehensive test could be performed.

Sandy Phelps  
PHELPS @ SBUG-80

#### PATCHING TRSDOS FOR MODEL III

I don't often use TRSDOS on the Model III, but I know some people do and there are some programs that just plain don't have patches to allow them to run under NEWDOS/80.

In my teaching, students use TRSDOS and therefore I am always on the lookout for useful patches to make TRSDOS more user friendly and to help it over the rough spots AND to keep the users out of the software.

These patches are from several sources. The main ones are the MARIN COUNTY TRS-80 USERS GROUP NEWSLETTER and the PORTLAND AREA TRS-80 USERS GROUP. Some of the patches are ones that I have devised myself after examining the listing of TRSDOS. (If anyone wants a patch to prevent you from using the BACKUP or PATCH command, or information on how to make your files into SYSTEM files so they can't be purged - user proof the system - I can supply them also!)

#### \*\*\* FAST BOOT \*\*\*

These cut the boot time by about 2 seconds. They can automatically set the number of drives. The fancy start up messages and graphics may be bypassed. You don't need to use all the patches. Try them one at a time, booting up in between to see what you have done. See also the patches in the next section; there is some duplication of functions.

PATCH \*0 (ADD=4E11,FIND=21F443,CHG=C3264E)

PATCH \*0 (ADD=4E43,FIND=3A1244,CHG=C3614E)

PATCH \*0 (ADD=4E63,FIND=0604CDD444,CHG=3E03C3744E)

The 03 after the 3E in the CHG part is the number of drives you want active on power up. 01=2, 02=3, 03=4, etc.

PATCH \*0 (ADD=4E9A,FIND=FIND=CD1B02,CHG=000000)

## South Bay TRS-80 Users Group

PATCH \*0 (ADD=5039,FIND=20,CHG=0D)  
PATCH \*0 (ADD=516C,FIND=D0BF9F,CHG=1C1F03)

### \*\*\* DATE, TIME, MISC. \*\*\*

PATCH \*0 (ADD=4EB8,FIND=213B51,CHG=C3394F)-skip date, time  
PATCH \*0 (ADD=4EF7,FIND=06,CHG=36) - skip only time query  
PATCH \*0 (ADD=516C,FIND=D0,CHG=03) - skip picture  
PATCH \*0 (ADD=508A,FIND=28,CHG=03) - skip copyright message  
PATCH \*0 (ADD=4F62,FIND=CD1B02,CHG=000000)-skip AUTO mess.  
PATCH \*1 (ADD=4E78,FIND=FIND=2E,CHG=20)-skip .. after READY  
The following changes the dating of the directory files.  
PATCH \*2 (ADD=4FD2,FIND=1C,CHG=00) - 00/ in file date  
PATCH \*2 (ADD=4FD7,FIND=1A,CHG=00) - /00 in file date  
for 00 - 17=secs, 18=mins, 19=hrs, 1A=yr, 1B=day, 1C=mo.  
PATCH \*0 (ADD=4EB5,FIND=CD1B02,CHG=C3F64E) - allows ENTER  
in response to DATE query

### \*\*\* TO MAKE IT EASIER TO OPERATE \*\*\*

PATCH \*4 (ADD=4E28,FIND=20,CHG=18) - display error messages  
PATCH \*2 (ADD=5135,FIND=2C,CHG=00) - gives access the  
BASIC/CMD type files  
PATCH \*7 (ADD=528E,FIND=C8,CHG=C9) -allows unlimited backup  
PATCH \*6 (ADD=5CC4,FIND=5953,CHG=2020) - use S for SYS  
PATCH \*6 (ADD=5CCC,FIND=4E56,CHG=2020) - use I for INV  
PATCH \*6 (ADD=5CD4,FIND=5254,CHG=2020) - use P for PRT  
PATCH \*2 (ADD=4ED4,FIND=20,CHG=18) - no password checks  
PATCH \*11 (ADD=5647,FIND=216658,CHG=C36E56)-disable p'word  
PATCH \*11 (ADD=5689,FIND=2ACE4D,CHG=C39A56)-only on PURGE  
PATCH \*1 (ADD=5023,FIND=C0,CHG=00) -")" optional in  
commands

The following changes the PATCH command to make it possible  
to use single letter entries: A=ADD, F=FIND, C=CHG.

PATCH \*9 (ADD=56C5,FIND=044144,CHG=02413D) - use A, not ADD  
PATCH \*9 (A=56CA,FIND=054649,CHG=02463D) - use F, not FIND  
PATCH \*9 (A=56D0,F=044348,CHG=02433D) - use C, not CHG

### \*\*\* REPAIRS \*\*\*

PATCH \*9 (ADD=5B9A,FIND=3E1FC309FF,CHG=3A2F44FE80) -  
enables the ROUTE library command  
PATCH \*6 (A=5855,F=FD212D40FDE5C33044,C=CD3044C20944C32D40)  
- fixes the LOAD command to do error checking (Make change  
in PATCH command shown earlier or do this in steps.)  
PATCH \*10 (ADD=4E47,FIND=02,CHG=03) - fixes garbage when  
using BASIC CMD "D:X" - "OFFICIAL R/S PATCH"  
PATCH \*8 (ADD=5655,FIND=72,CHG=66) - fixes typo error -  
"OFFICIAL R/S PATCH"

### \*\*\* CONVENIENCE \*\*\*

PATCH \*5 (ADD=4EDF,FIND=38E6,CHG=0000) - allows DEBUG to  
PATCH \*5 (ADD=4F04,FIND=D0,CHG=C9) - display any  
PATCH \*5 (ADD=506E,FIND=38E3,CHG=0000) - address  
PATCH \*3 (ADD=4FB7,FIND=EDB0,CHG=0000) - stops the erasing

May 1983

of entire directory entry when files killed so you can recover from an accidental killing of files.

\*\*\* USE WITH SLOWER STEPPING RATE DRIVES \*\*\*

X = F, E, D, C FOR 30, 20, 12, 6 MILLISECOND STEP RATES  
Y = B, A, 9, 8 FOR 30, 20, 12, 6 MILLISECOND STEP RATES

PATCH \*0 (ADD=4544,FIND=1C,CHG=1X)  
PATCH \*7 (ADD=580E,FIND=0C,CHG=0X)  
PATCH \*7 (ADD=5841,FIND=0C,CHG=0X)  
PATCH \*7 (ADD=5923,FIND=1C,CHG=1X)  
PATCH \*7 (ADD=5B3C,FIND=58,CHG=5Y)

PLUS: Using superzap, superutility, or similar, change relative byte 0D0H in the boot sector from 1CH to 1FH, 1EH, 1DH, 1CH for 30, 20, 12 or 6 millisecond step rates.

\*\*\* BASIC \*\*\*

PATCH BASIC/CMD (A=593D,F=21A859FE4C28,C=E65F21A859FE)  
PATCH BASIC/CMD (A=5943,F=1521AD59FE45,C=4C28132EADFE)  
PATCH BASIC/CMD (A=5949,F=280E21B959,C=45280D2EB9)

Make changes shown earlier in PATCH command, or do these in steps. Allows Disk BASIC to use upper or lower case for A (a) AUTO, D (d) DELETE, E (e) EDIT, L (l) LIST. Could then use d10 or D10 or DELETE10, etc.

Ian Webb  
IANWEBB @ SBUG-80

S U M M E R   C O M P U T E R   C L A S S E S

This summer, I will be teaching the following courses through the Community Development Program at West Valley College. Each course will be 9 hours long, meeting three different evenings. Classes will be held at West Valley College in Saratoga.

Enrollment will be limited so each student will have exclusive use of a Model III, 2 drive, 48K machine during class hours. There is a \$68 fee for each course, since these classes must be self-supporting. The fee may seem stiff, but is what the school set to cover their costs and pay for the teacher - me. (Didn't think I worked for nothing did you?) Reading the newsletters from other areas, I notice that other users group give classes and typically charge \$8 per instructional hour. (Marin Co. User's Group, for example.)

The classes are aimed at the "novice" user; no background will be assumed. Software will be provided for IN CLASS USE and will NOT be allowed to leave the school. Classes meet from 7-10 PM on the days and dates given.

## South Bay TRS-80 Users Group

INTRODUCTION TO BASIC -	W-6/29, Tu-7/5, Th-7/7
VISICALC -	Tu-5/31, Th-6/2, M-6/6
PROFILE III+ -	Tu-6/21, Th-6/23, M-6/27
SUPERSCRIPSIT/SCRIPSIT -	W-6/8, M-6/13, W-6/15

The BASIC class applies to other machines using versions of Microsoft BASIC. VISICALC is also quite independent of machine. The PROFILE III class uses a version of the PROFILE data base management program that is nearly the same as what is used on the Model II/12/16. The SCRIPSIT class will aim at SUPERSCRIPSIT, but if any enrollee wishes to learn the older, SCRIPSIT, that will be possible. With the exception of PROFILE III, all of these courses are applicable to the user of the Model I.

If you are interested and want to enroll, or get further information, call the Community Development Office at West Valley College at (408) 867-8448.

Ian Webb  
IANWEBB @ SBUG-80

### THE ELECTRONIC CLOSET EC-1 CHARACTER ROM

I have opened up my TRS 80 MOD I so many times that I should feel like the surgeon who is about to perform his 10,000th appendectomy, BUT.... I must keep reminding myself that I am the fumbler who drilled through a stud with my electric drill and struck--not gold but--another 110 volt "hot" wire. The sparks were surely pretty! I am also the guy who thought he could repair his garage door spring, and who spent one month in the hospital, two surgeries, and a lost eye (complete with black patch and all) because I didn't completely (or even generally) understand the problems/opportunities/dangers/possibilities that the unknown held for me. Nevertheless, I still tend to follow Tom Anderson's advice: "when in doubt, diddle." I sometimes forget the second part of his advice: "...if you are sure you can put it back like it was in the first place."

The "guts" of my MOD I still vaguely resemble the original beast, but the operative word is "resemble." I quickly voided my warranty (my machine is now 6 years old) within the first 6 months and added my own memory chips. (Remember when the Shack socked it to you by charging big \$\$ for adding memory?) This was my first experience with both opening the marvels of the TRS-80, as well as opening up the miracle of the marvelous mind of Art Schawlow (SBUG's very own Nobel Prize Winner.) I still remember that rainy day when he suggested that I bring my computer to his office and we would upgrade the 16K to 48K. I wasn't aware of Professor Schawlow's achievements. When he was involved with interruptions during my visit, I took the



time to look at his library. I discovered that he had many books on lasers. After we had successfully installed the 48K of memory, he took me to my car, and I attempted to make conversation. I said "I noticed that you have many books on lasers. Is this one of your interests?" His reply will remain imprinted in my mind forever: "I suppose you could say so, I invented it!" Since that time, I have attempted not to make "small talk" with Stanford Professors!

I later decided to add the lower-case mod. With fear and an unsteady hand, I again opened the box, and began what was for me a first attempt with the soldering iron. In spite of the smell of freshly burned flesh and a funny looking "piggy-backed" chip precariously perched on top of another IC, I was elated when I put the machine back together and it worked! Since that time I have installed the Archbold Reverse Video kit, the Archbold Speed Up kit, the Mumford Speed-up kit, the Percom Data Separator, and the Percom Doubler. By this time I must be expert--but only at seared flesh and tangles of wires.

Tim Worcester of the Electronic Closet sent me a Mod I character generator for evaluation. The EC-1 is a replacement for the MCM6678 character generator ROM which the Shack lovingly "willed" us years ago. Remember that they didn't expect us to want lower-case--even though it was available in the MCM6678. Although my lower-case worked fine, my secretary and students all remarked "Why is the 'a' funny like that, and how come the 'p' 'g' 'j' and 'y' set above the line?" I really had no answer for them except to say that "This is the way it is." I had hoped for many moons that I could replace the character generator, and even called Emmanuel Garcia Associates in Chicago to order the chip, only to learn that the Shack had convinced its supplier that if it was to continue supplying this item to the Shack, it could no longer make it available to the public. I subsequently called National Parts at the Shack and inquired about the new-and-improved character generator. At first they didn't want to sell to me at all, and when they agreed to do so, it was at a greatly inflated price. My decision was to continue with the old generator and ignore the secretary and students who didn't like the way the characters looked.

Electronic Closet has made a lovely little circuit card with an EPROM which does the trick. With a little effort, I now have lovely descenders, and a correct "a." If you have the newer keyboard, you probably will only have to open the unit, remove Z29 and plug in the new EC-1 circuit card. Originally Radio Shack, in its infinite wisdom, determined that they could save a few cents in each MOD I if they did not provide a socket for Z29. Since I have an old keyboard unit I, therefore, had to surgically remove the old chip and install an 18 pin socket. I did my best to get the old chip out by using both a solder sucker

## South Bay TRS-80 Users Group

and solder-up. I am still not very expert at this sort of thing, and I was unsuccessful in salvaging the old chip. "Land-o-Goshen, what if the new EC-1 didn't work???"

It was necessary for me to clip the "legs" off Z29 to remove the IC. After this was done, all that was necessary was to add the socket (available from Radio Shack for \$.49--part #276-1992), plug in the circuit card and put the screws back in the case. It worked!!! The characters are beautiful and the "I" is even a nice shape! The EC-1 board with a standard character set sells for an unbelievably low price of \$11.00!!! For those who desire a custom character set in addition to the standard character set, the price is \$19.00. All you must do is submit the patterns you want along with your order. The character set which you submit must be in a 5x8 matrix.

The instructions are very clear. If you don't feel comfortable installing the EC-1 after you receive it, Electronic Closet states, "you can return the EC-1 for a complete refund." By installing a switch, you may manually select the standard or the custom character set, or it may be selected by a software controlled latch. I had no problems in the installation process (and could have saved myself about 45 minutes if I had first cut the "legs" off the MCM6670 prior to the desoldering process.) I originally thought that I needed to purchase a socket, but upon closer examination, I realized that the PC card comes already inserted into a socket for those whose Mod I lacks the socketed IC.

I guarantee that you will be very pleased with the way your new character set looks! If you are interested in this excellent piece of hardware, write to Electronic Closet, 8187 Blakely Court West, Bainbridge Is., WA. 98110. They are very fast and the product is excellent. Now I have to order another EC-1 for my computer at home (which resides in my hall closet!--that should give Electronic Closet a chuckle.)

PS- I have now installed my third Electronic Closet chip, and all have worked well. I even learned how to remove the old IC without cutting off the legs and thus ruining the old chip. If you have any difficulties after installation, I would suggest that you re-check your earlier lower-case mod to see if any of your previously soldered wires have either come loose or broken. This did happen in one instance with me, and it took quite a search to correct the problem--that which was created by me, not by Electronic Closet.

William Ramsey  
BILLRAM @ SBUG-80

=====

## MX-80 GRAPHICS PICTURES

Many members of SBUG have used the picture programs I have written for the MX-80 (with GRAPTRAX PLUS only). The programs are available on SBUG-80. In this article I will tell the secret of how the pictures were created.

I'm sure all of you have seen the computer pictures made up of letters and spaces that look great from a distance, but not very good close up. One of these pictures on disc is the first item needed to create a neat graphics picture on the MX-80. If you try to use one of the fancier pictures, with the shading coming from the use of various characters, the results will be disappointing.

Once you have decided on the picture to use, you have to convert the each of the letters to a single dot on paper. Sounds simple doesn't it? It actually is once you know the secret. You already have the first clue - the single dot corresponds to a single hammer in the print head of the MX-80 printer. The MX-80 manual says that you can only use 8 of the 9 pins (hammers) in the graphics mode - how convenient - a byte has 8 bits (clue 2).

At this point I will explain the basic concept of the conversion from letters to dots. Each line of print (letters) becomes a line of dots, or every 8 lines of print becomes 1 line of graphics dots (remember the 8 hammers?). This is done by converting each print position to a single bit in a character to be printed in the graphics mode. Before I continue with this, let me describe the byte as it pertains to the graphics mode on the MX-80 printer.

A byte is composed of 8 bits, each bit having a value or either 1 or 0, they are usually numbered from 7 thru 0 (or 8 thru 1 depending on who taught you, for our purpose I will refer to them as 7 to 0). Bit 7 is known as the high order bit and bit 0 as the low order bit. In addition to having a value of either 1 or 0, a bit has an implied value depending on it's position in the byte; bit 0 has an implied value of 1, bit 1 has an implied value of 2, bit 2 has an implied value of 4. Each bit has an implied value double that of the next lower bit (or half the implied value of the next higher bit). For example, the number 143 would be represented in an 8 bit byte as '10001111' (in hexadecimal that is '8F'). If we think of the hammers in the print head of the MX-80 it would look like this:

bit	value	implied value
7	1	(1 x 128) = 128
6	0	(0 x 64) = 0
5	0	(0 x 32) = 0
4	0	(0 x 16) = 0
3	1	(1 x 8) = 8

# South Bay TRS-80 Users Group

2	1	(1 x 4) =	4
1	1	(1 x 2) =	2
0	1	(1 x 1) =	1

Adding the values (128 + 8 + 4 + 2 + 1) gives us 143 - I know 'BIG WHOOP' - but wait - look at the example again. Remember, each bit is a hammer and the MX-80 sees each byte exactly as it is represented above (if the bit is a 1 the hammer strikes the ribbon, making a dot on the paper, if it is a 0, no dot is printed in that position).

One more brief lesson, then we can get down to the nitty gritty. The implied value of each bit position is a power of 2; 2[0 = 1, 2[1 = 2, 2[2 = 4 etc. With this last piece of information, we have all that we need to convert pictures.

Going back to the original premise - each letter is a dot on the paper - each space is a blank spot, we can now start the conversion process. The following will describe each of the steps involved in the process.

First, find the length of the longest print line (this will become the record length for the final data file). The program listed here will do that for you:

```
10 CLS: CLEAR1000
20 INPUT "ENTER NAME OF FILE TO SCAN"; F$
30 IF F$ = "END" THEN CLOSE: END
40 OPEN "I", 1, F$
50 L = 0
60 IF EOF(1) THEN ? "THE LONGEST LINE IS"; L; "CHARACTERS": END
70 LINE INPUT #1, A$
80 A = LEN(A$): IF A > L THEN L = A
90 GOTO 60
```

Write down the number display by the program as it is needed for the next step. You are now ready to write the program to convert the picture file to a graphics data file for the MX-80. In the following BASIC statements the <n> has to be replaced by the number you wrote down.

```
1 REM          PICTURE CONVERSION PROGRAM
10 CLS: CLEAR1000: REC = 1
20 DIM C(<n>): ' Number of characters in a line
30 Z = 7: ' This will be used to set the bits to print
40 INPUT "Enter name of file to convert"; F1$
50 INPUT "Enter name for output file"; F2$
60 OPEN "I", 1, F1$
70 OPEN "R", 2, F2$, "FF", <n>
```

Line 70 tells NEWDOS80 we want to open a random access file with fixed length record that are <n> bytes long.

```
80 GOSUB1000 : REM to clear array to all 0'S
90 Z1 = 2[Z
```

Line 90 converts the bit number (Z) to an implied value for it's position in the byte.

```
100 IF EOF(1) THEN 3000 : REM to see if anything in array
100 LINE INPUT#1,A$
110 GOSUB2000 : REM go convert to bit values
120 Z=Z-1:IF Z<0 THEN Z=7:GOSUB4000 : REM write out line
```

Line 120 decrements the bit number (Z) and checks to see if all 8 bits have been done (Z less than 0). If so, the routine at 4000 is called to write the data to disc. from the array (C(n)).

```
130 GOTO100 : REM go get next line
```

```
1000 REM clear array 'C' to all 0's
1010 FOR X=1 TO <n>
1020 C(X)=0
1030 NEXT
1040 RETURN
```

```
2000 REM convert letters to bit values in array
2010 FOR X=1 TO LEN(A$)
2020 IF MID$(A$,X,1) = " " THEN 2040
2030 C(X)=C(X)+Z1 : REM add in implied value for bit
2040 NEXT
2050 RETURN
```

```
3000 REM check for data not written to disc
3010 IF Z<7 THEN GOSUB4000
3020 CLOSE
3030 END
```

```
4000 'write data to disc as a string with a length of <n>
4010 PL$=""
4020 FOR X=1 TO <n>
4030 PL$=PL$+CHR$(C(X))
4040 NEXT
4050 PUT2,REC,(<n>)PL$
```

Line 4050 is the special "FF" file write format, it writes a record with the length <n> from the variable PL\$.

```
4060 REC=REC+1
4065 GOSUB 1000: 'Clear array to all 0's
4070 RETURN
```

Now we have to write a program to print the picture.

```
1 REM graphics picture print program
10 CLS:CLEAR1000
20 GOSUB1000 : REM go initialize printer
30 OPEN"R",1,"filename","FF",<n>
40 FOR REC = 1 TO LOF(1)
```

## South Bay TRS-80 Users Group

```
50 GET1,REC,(<n>)P$
60 GOSUB2000 : 2000 is the print routine
70 NEXT
80 CLOSE
90 END
```

```
1000 REM initialize the printer for graphics
1010 LPRINT : REM clear the print buffer in the printer
1020 PR$=CHR$(27)+"A"+CHR$(8) : 'set to 8/72's of an inch
1030 GOSUB2020
1040 RETURN
```

```
2000 REM special print routine, used because the regular
    print routine has problems with some hex values
2010 PR$=CHR$(27)+"K"+<nl>+<n2>+PR$
    *** note if n is less than 128 then n1=n and n2=0, if
        n is greater than 128 then n2=1 and n1=(n-128)
        The CHR$(27) is 'ESC', the "K" tells the printer
        to print graphics (480 dots per line). <n1> +
        <n2> x 128 is the number of characters to print.
2020 FOR P = 1 TO LEN(PR$)
```

Use this coding for a MODEL I:

```
2030 A=PEEK(&H37E8) : REM check status
2040 IF A<>63 THEN 2030 : '<> 63 means printer not available
2050 POKE(&H37E8),A
2060 NEXT
2070 RETURN
```

Use this coding for the MODEL III:

```
2030 A=INP(&HF8) : REM CHECK STATUS
2040 IF A <> 63 THEN 2030 : BUSY?
2050 OUT&HF8,A
2060 NEXT
2070 RETURN
```

You now have the basic information on writing programs to print graphic pictures from ASCII character pictures. I hope this has been clear enough for you to create your own picture programs.

Glenn Vaughn  
GLENN @ SBUG-80

## BASIC PROGRAMMING

I have reviewed some of SBUG's older newsletters and discovered some interesting material that I thought many beginners in the club would enjoy. The following article is a reprinted article by Bryan Devendorf from the October 1980 newsletter. Bryan did several articles on basic programming. I plan to reprint articles that I believe many new members in SBUG would enjoy. So, watch for these fine articles by Bryan...

Rob...

This month let's look at a couple of ways to LPRINT material that would ordinarily go to the screen (using PRINT) or is already on the screen. Most of my programs that allow output to the screen or lineprinter would look something like this:

```
1000 PRINT"DO YOU WANT TO OUTPUT TO THE LINEPRINTER?"
1010 A$=INKEY$:IF A$="" THEN GOTO 1010
1020 IF A$="Y" LPRINT .....ELSE PRINT .....
1030 IF A$="Y" LPRINT .....ELSE PRINT .....
1040
1050 (etc)
```

This means quite a lot of typing. For each LPRINT there is a PRINT and vice versa. And what if you want the output both to the screen and the lineprinter. Then you might have a lot of:

```
1000 PRINT .....:LPRINT.....
type statements. Again a lot of extra typing. Why not cut
down on some of this extra work by changing the device
control block for the screen. For example, try this
instead:
```

```
1000 GOSUB 5000
1010 PRINT .....
1020 PRINT .....
1030 (etc)
1040 GOSUB 6000
5000 PRINT"DO YOU WANT TO OUTPUT TO THE LINEPRINTER?"
5010 A$=INKEY$:IF A$="" THEN GOTO 5010
5020 IF A$="Y" THEN POKE 16414,PEEK(16422):POKE 16415,
      PEEK(16423)
5030 RETURN
6000 POKE 16414,88:POKE 16415,4:RETURN
```

The effect of the 5000 subroutine is to put the lineprinter driver routine into the display DCB. Whenever a PRINT is executed the computer is 'fooled' into LPRINTing instead. By using the PEEK(16422) etc. instead of the absolute addresses allows for whatever kind of lineprinter routine you are using, (serial) etc. Subroutine 6000 returns the screen DCB back to it's normal state. Now you say you want both the screen and the lineprinter! OK try this.

```
1000 GOSUB 5000:FOR X=1 TO Z
1010 IF X=2 THEN GOSUB 6000
1020 PRINT .....
1030 PRINT .....
1040 (etc)
1050 NEXT
1060 GOSUB 7000
5000 PRINT"OUTPUT TO      (L)INEPRINTER"
5010 PRINT"                (S)CREEN"
5020 PRINT"                (B)OTH"
5030 PRINT:PRINT"ENTER YOUR SELECTION"
5040 A$=INKEY$:IF A$="" THEN 5040
5050 IF A$="S" THEN Z=1: RETURN ELSE IF A$="L" THEN
      Z=1:GOSUB 6000: RETURN ELSE IF A$="B" THEN
      Z=2:RETURN ELSE GOTO 5040
6000 POKE 16414,PEEK(16422):POKE 16415,PEEK(16423):RETURN
```

## South Bay TRS-80 Users Group

7000 POKE 16414,88:POKE 16415,4:RETURN  
You can also use PRINT USING within this loop and change the string for the lineprinter if needed.

Comprint 912 printer users should take special note when using the 'swap pointer' routines above. If you use the CLS command after poking the lineprinter routine into the screen DCB, the printer will lock up. This is because the CLS command prints a CHR\$(28)--home cursor followed by a CHR\$(31)--clear to end of frame. When these are LPRINTed the Comprint will recognize the CHR\$(31) as a stop print command and will ignore any further characters until reenabled. The simplest fix is simply to execute all necessary CLS's before swapping the pointer.

If you want to dump the screen to the lineprinter after you have outputted to the screen there must be a hundred ways to accomplish this. The 'JKL' routine in NEWDOS is the easiest. If you don't have disk see me at the meeting for a copy of an assembly language program that gives non disk users a 'JKL' command. The following is one of the hundred ways to dump the screen using a basic routine. After printing to the screen, execute this subroutine:

```
8000 PRINT"DO YOU WANT TO DUMP THE SCREEN OUTPUT TO THE  
      LINEPRINTER?";  
8010 A$=INKEY$:IF A$="" THEN GOTO 8010 ELSE IF A$="Y"  
      THEN GOTO 8012 ELSE RETURN  
8012 O$=""  
      REM 64 BLANKS  
8015 Z=PEEK(VARPTR(O$)+2)*256+PEEK(VARPTR(O$)+1)  
8020 FOR X=0 TO 960 STEP 64:FOR Y=0 TO 63  
8030 POKE Z+Y,PEEK (15360)+X+Y:NEXT Y  
8040 LPRINT O$:NEXT X:RETURN
```

If your lineprinter doesn't print graphic characters you may want to check each peek of the screen first and poke it into the string only if it is not a graphic character (less than 128 dec) and poke some other number like 46 (a period) to represent the graphic character instead. Another way is to build each string like O\$=O\$+CHR\$(PEEK(16380)+X\*Y) within a for-next loop and after setting O\$="". However, this approach takes much longer and eats memory like crazy.

Hope this had helped some of you with your output problems.

Bryan Devendorf

## GRAPHIC PROGRAMS AVAILABLE ON SBUG-80

For those members not communicating with SBUG-80, I have spooled from SBUG-80's index all the programs pertaining to graphics. If you are interested in any of



May 1983

these programs, ask a member who is communicating with SBUG-80 if they would download the program for you. I'll bet the member would be happy to help out...

For those members who are communicating with SBUG-80 and wonder where I found this information, it can be found by executing 'DO INDEX' from SBUG-80 READY.

Robert Byrd  
EDITOR @ SBUG-80

\*Program: DRAWR/LNW      \*Date: 09/21/82      \*Donor: DAVEFOX  
\*Reference: BILL HUMMEL      \*GRANS: 1

\*Description:

This program is the equivalent of the SKETCH/BAS program, but is written in Lnw Basic for the LNW80. The program utilizes a non-destructive cursor to draw high-resolution pictures to the screen. The screen can then be saved to disk. A MUST for all LNW80 users. This program requires LNW BASIC.

\*Program: ENTERPRZ/GRF      \*Date: 09/21/82      \*Donor: DAVEFOX  
\*Reference: STAR TREK      \*GRANS: 1

\*Description:

This is a high resolution screen dump to a LNW80. You MUST have LNW Basic to run this dump. In LNW basic:

PCLS  
MODE 0  
PLOAD\*ENTERPRZ/GRF  
CLS  
MODE 1

\*Program: GOTHIC      \*Date: 11/27/82      \*Donor: MSILLS  
\*Reference: SOFTSIDE APRIL 1982      \*GRANS: 1

\*Description:

THIS PROGRAM WILL PRINT YOUR MESSAGE IN LARGE GOTHIC CHARACTERS.

\*Program: HILLS/GRF      \*Date: 09/21/82      \*Donor: DAVEFOX  
\*Reference: 80 MICRO      \*GRANS: 1

\*Description:

This is a high resolution screen dump for the Lnw80. You MUST have LNW Basic to run this dump. In LNW Basic:

PCLS  
MODE 0  
PLOAD\*HILLS/GRF  
CLS  
MODE 1

\*Program: JACKOLAN/BAS      \*Date: 10/05/82      \*Donor: JIBBYTRM  
\*Reference: 80 MICRO      \*GRANS: 1

\*Description:

This is a real Halloween treat for the kids. This program is in Oct 82, 80 Micro. It was not in the LOAD80 so Bill Jibby took the time to key it in for us. The mouth will also move if you talk through the

South Bay TRS-80 Users Group

cassette deck. For Model III Newdos 80 change line 10 CMD"T" to CMD"CLOCK,N". The left and right arrows will move the face from side to side. To hook it up to the cassette deck using the Radio Shack Mini-Amplifier-Speaker just plug a microphone into the input jack and plug the black jack on the cassette cable to the external speaker jack. Run the program and adjust the volume until the mouth moves when you speak into the mike.

\*Program: JUNQUE/BAS      \*Date: 12/01/82      \*Donor: ROGERAND  
\*Reference: ROGER ANDERSON      \*GRANS: 2  
\*Description:  
THIS PROGRAM IS A COLLECTION OF SOME PROGRAMS  
I WROTE WHEN I WAS INVESTIGATING TRS-80 GRAPHICS.  
MOST USE THE SET/RESET COMMANDS.

\*Program: LINE/BAS      \*Date: 09/28/82      \*Donor: DAVEFOX  
\*Reference: 80 MICRO MAY81 PG119      \*GRANS: 1  
\*Description:  
This is a MX80 printer plotter to utilize the Graftrax  
option to create high res line designs. First run  
Line/cim, then in basic--run"line/bas.  
See Aug 82 Newsletter for example of output.

\*Program: LINE/CIM      \*Date: 09/28/82      \*Donor: DAVEFOX  
\*Reference: 80 MICRO MAY81 PG119      \*GRANS: 1  
\*Description:  
This is the machine code for Line/bas. First run this  
program, then in basic--run"line/bas

\*Program: PICTURE/BAS      \*Date: 11/28/82      \*Donor: GLENN  
\*Reference: SEE PICTURE/DAT      \*GRANS: 2  
\*Description:  
This program will print 5 picture on the MX-80 with  
GRAFTRAX PLUS. It requires the data file 'PICTURE/DAT'.  
The program can be executed by typing:  
BASIC,3V,RUN"PICTURE/BAS  
It will only run under NEWDOS80

\*Program: PICTURE/CMD      \*Date: 03/18/83      \*Donor: GLENN  
\*Reference: SEE PICTURE/DAT      \*GRANS: 6  
\*Description:  
THIS IS THE MACHINE LANGUAGE VERSION OF PICTURE/BAS  
IT USES PICTURE/DAT FOR INPUT, AND PRINTS 5 PICTURES  
ON THE MX80 (REQUIRES GRAFTRAX PLUS). THE PICTURES  
ARE 1 - MICKEY MOUSE, 2 - BUTTERFLY, 3 - SEAL WITH BALL  
4 - AIRPLANE AND 5 - PENGUIN. PRINTER MUST BE RESET  
AFTER USE AS IT CHANGES THE LINE SPACING.

\*Program: PLOT/CMD      \*Date: 11/21/82      \*Donor: MCHENRY  
\*Reference: 80 MICRO MAY'81 P.119      \*GRANS: 3  
\*Description:  
This is a machine language subroutine to allow vector  
graphics on the MX-80 printer with graphtrax. It was  
modified by Mike McHenry to run with the MX-80 and with  
Model 1 DOS. It sets HIMEM using 4049H. You will have

to reset HIMEM after you are done. You need 48K to run this and it is configured to run with PLOTFILE/BAS to allow plotting of X-Y graphs with a grid. Also see LINE/CMD on this system that allows plotting of designs.

\*Program: PLOTFILE/BAS \*Date: 11/21/82 \*Donor: MCHENRY

\*Reference: MIKE MCHENRY+80 MICRO MAY81 \*GRANS: 4

**\*Description:**

This is a BASIC program that uses PLOT/CMD as a USR call to plot X-Y data in memory and then dump the plot to an MX-80 printer with graphtrax. It makes a large full page graph with grid, labeled axis, title and plots for as many sets of data as you want with symbols and/or lines for the data points. The data comes from a disk file created with a word processor and saved in ASCII or from another program output such as PRINT#1,X,Y,Z. The program will prompt you for the required input and ask you for the name of the disk file to plot. It will then ask how many columns of data there are and which column is X and which is Y.

Make sure your printer is on when using this program or it will hang. The program writes "<>" to the screen for each line it draws in memory. If you want both symbols (such as triangles) and a line through the points, just plot the same data twice, first with the symbols and next with the line option.

Experiment with it and I am sure you will see how it works. I have found this program to be very useful at work. It can be used with data saved to disk from VISICALC's print feature or from AIDS 3 data base with printer output routed to disk.

\*Program: SKETCH/BAS \*Date: 09/19/82 \*Donor: ROGERAND

\*Reference: FASTER AND BETTER \*GRANS: 2

**\*Description:**

This program "SKETCH/BAS" is used to create screen graphic with single keystrokes. Also you will be able to save the screen on disc and send the screen to the printer. The program "SKETCH/DOC" describes the functions of SKETCH/BAS in detail.

SKETCH/BAS will be found in the September/82 NEWSLETTER.

=====

South Bay TRS-80 Users Group  
P.O. Box 60116  
Sunnyvale, Ca 94088

**FIRST CLASS MAIL**