SEPTEMBER 1982

TRS-80™ NEWSLETTER

SOUTH BAY - USERS GROUP

COMMUNICATE!

South Bay TRS-80 Users Group

Page - Contents

SBUG meetings are held the 3rd Tuesday of each month in the north east corner of Dysans' building at:

5401 Patrick Henry Drive
Santa Clara, Ca

Time - 7:15 to 10:30 PM

September 21, October 19, November 16

Features: 1) Open Rap Session...

Your Steering Committee follows

		SBUG-80 Username	Phone
Discussion Leaders:			
	Phil Coffman	(CAP)	(408) 997-0247
	Mike McHenry	(MCHENRY)	(408) 245-4704
	Gerry McKee	(BMCKEE)	(408) 926-4063
	Sabri Kawash	(SABRI)	(408) 732-5484
	Bill Richerson	(OPTOBILL)	(408) 257-8267
Treasurer:			
	Larry Gunderson	(TREAS)	(408) 259-5349
Newsletter Editor:			
	Robert Byrd	(EDITOR)	(408) 732-6775
Librarians:			
(disc)	Pete Huntsinger	(PETE)	(408) 227-7125
(tape)	Bob Brown	(SAMBROWN)	(408) 379-2774
(doc.)	Ron Carpenter	(RON)	(415) 726-3487
Host Computer:			
	SBUG-80	(EVERYONE)	(408) 737-SBUG

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

Send Newsletter articles to:

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

Deadline for the October newsletter is: 25 Sep 82.
If at all possible send articles via modem or saved on disc/tap. I will see that your disc or tape is returned to you. Thanks . . .

FRICTION FEED KIT FOR EPSON MX-80 PRINTER

If you have an EPSON MX-80 printer without friction feed, you may be interested in the $Micro-Grip friction feed kit. I purchased my EPSON printer over a year ago at a time when they were scarce and settled for the tractor feed only. After struggling for a year or so with makeshift arrangements to print addresses on envelopes, I decided to try the friction feed kit advertised by Micro-Grip Ltd. in BOMICRO (Micro-Grip Ltd., P. O. Box 4278, Norton AFB, Ca. 92409).

I mailed my order to Micro-Grip on June 14th and heard nothing from them until July 9th, when I received a notice that my VISA account authorization had been denied! Someone apparently entered the account number wrong. I returned the notice and asked them to try again. This time it worked and I finally received my kit in about two more weeks.

The kit consists of a three-inch long one-inch diameter pressure roller, two molded rubber "micro-platens", two coil springs and an Allen wrench (for $39.95 plus tax and shipping). Installation of the kit is straightforward. The three-inch pressure roller mounts on the EPSON paper pressure bar and the "micro-platens" slide on to the square platen bar of the EPSON, after removal of the paper guide. The only possible difficulty is in removing or re-installing the spring clip on the end of the square platen bar. The clip may go sailing across the room if you are not very cautious (and lucky). I was lucky - no problem.

The two coil springs appear to be an afterthought. The instructions say to install them on the paper pressure bar if the springs in your printer are "too weak". Without them the paper will not feed because the pressure between the roller and the micro-platens is inadequate. I suspect that all EPSON printers require the extra springs. In order to use the printer with envelopes or single sheets of paper it is also necessary to disable the "out-of-paper" switch. I did this with a write-protect foil stuck over the switch.

The Micro-Grip friction feed device will never win a prize for elegance or beauty, but it does the job. Printing can start about 3/4-inch from the top of a page. When an envelope is printed, the flap must be turned up to reach the pressure roller when the top edge of the envelope is in line with the print head. Envelopes and paper feed straight and the spacing is accurate. The tractor feed works normally with no interference from the friction feed attachment. The product works as advertised but some may find it - what can I say? - "Mickey Mouse"?

($Trade Mark of Micro-Grip Ltd.)

Vigo Smith
VIGO @ SBUG-80

SKETCHPAD

Sketchpad enables the user to print graphic characters on the screen using single key strokes and to position the cursor using the arrow keys. The user will also be able to save the display to disk, to print the display on a line printer, and to copy one portion of the display to another location.

Sketchpad consists of the following subroutines.

INITIALIZATION	LINES	10-40
ERROR	LINES	40 and 10000
KEY DEFINE	LINES	50-160
SETUP GRAPHIC KEY CODE	LINES	330 and 8000-8060
CURSOR	LINES	750-785
COPY A PORTION OF SCREEN	LINES	4000-4180
PRINT THE SCREEN	LINES	5000-5080
SAVE SCREEN ON DISC	LINES	6000-6310
NORMAL KEYBOARD	LINES	9000-9120

---- ERROR

The error routine is used to trap out undefined lines in the GOSUB commands in lines 130 and 150. Some of the line numbers in lines 130 and 150 are just fillers and the actual lines do not exist . If these ghost lines were addressed, the program would stop and the screen presentation would be lost.

One word of advice, if you type in this program, do not type in the error routine until the program is completely debugged.

---- KEY DEFINITION

The graphics characters that will be placed on the screen (at the cursor position) when one of the letter keys is typed, are displayed on the bottom line of the screen. This bottom line is ignored by the screen to disk and the screen to printer routines.

Key definition is accomplished by the use of the INKEY$, the ASC(STRING$), and the ON n GOSUB LINE1, LINE2,LINE3,....,LINE26 commands. The ON n GOSUB command in line 130 processes the ASCII values from 31 to 64. The same command in line 150 processes the ASCII values from 65 to 94. Two seperate lines are used for these commands so that lines 130 and 150 will not exceed 255 characters in length. If a line should exceed 255 characters in length, you will be cursed with a DIRECT STATEMENT IN FILE ERROR when you try to load and run this program.

The number keys (1,2,3,4,5,and 6) are used to change the graphics characters that are displayed on the screen when the letter keys (A,B,C,D,E,F,G,H,J,K, and L) are typed.

The keyboard is set to its normal character font when the 7 key is pressed and remains in this normal font mode until the CLEAR KEY is typed.

Typing the letter "P" lists the key encoding on the line printer. Table I shows the encoding of the keys. Any undefined keys are available for encoding to select routines that the operator may develop to be use with this program.

---- CURSOR

One problem that had to be overcome was the use of arrow keys by the TRS-80 as backspace, tab, and line-feed.

Enter the following program.

```
10 CLS
20 Z$=INKEY$: IF Z$="" THEN 20
30 PRINT @ 600, Z$" = " ASC(Z$);
40 GOTO 10
```

When this program is run the following ASCII values are displayed when the arrow keys are typed.

```
UP ARROW    =  91
DOWN ARROW  =  10
LEFT ARROW  =   8
RIGHT ARROW=   9
```

These are not the values (for the down arrow, left arrow, and right arrow) that are given in the code table on page C2 of the LEVEL II BASIC REFERENCE MANUAL.

Line 124 adds 64 to the TRS-80 "ASCII VALUES" of the down, left, and right arrows. This translates the arrow keys to their approximate ASCII values, allowing the arrow keys to be processed by line 150.

When the program is initialized, the cursor will increment to the right each time a graphics character is sent to the screen. At times it would be desirable to have the cursor increment to the left. Typing a GREATER THAN (<) character will cause the cursor to increment to the left. Typing the LESS THAN (>) character causes the cursor to increment to the right.

---- COPY A PORTION OF THE SCREEN

To copy one portion of the active screen to another position on the screen is a feature that adds a great deal of flexibility to this program.

To start the move, position the cursor in the space immediately proceeding the starting position of that portion of the screen which is to be moved. Press the "M" key.

Now position the cursor in the space immediately following the last character of the portion screen that is to be moved. Press the "N" key. There will be a slight delay while the program is gathering in that portion of the screen that is to be moved. You might press the

up or down arrow at this time, then when the cursor moves you are ready to continue.

Now position the cursor at the point where the picked up section is to be to be placed. Press the "O" key and the picked up section will appear on the screen. The cursor will now be located at the end of the newly printed display block.

If the CLEAR KEY is pressed the screen will be cleared. Now if the "O" key is pressed the previously picked up section will appear on the screen.

---- PRINTING THE SCREEN

Printing the screen on the line printer occurs when the letter "Q" is typed. The print routine should be modified and/or enhanced to accommodated your printer.

---- SAVE THE SCREEN TO DISK

The save to disk routine in lines 6000 - 6180 is a modified version of a routine found in the book BASIC FASTER AND BETTER AND OTHER MYSTERIES by Lewis Rosenfelder.

Saving the screen to disk is tricky. This routine directs the screen to a disk file and directs a disk file to the screen. In lines 6120 and 6140 the numbers 26318 and 26319 are the memory locations for the DATA CONTROL BLOCKS when using Apparat's NEWDOS 80. If you are using a different DOS then these memory locations will have to be changed.

Table II shows the memory locations data control blocks used by 4 popular disk operating systems (DOS).

DOS	DCB(1)	DCB(2)
NEWDOS 80	26318	26319
TRSDOS 2.0	27718	27719
TRSDOS 1.2	25765	25766
DOSPLUS3.3	28023	28024

TABLE II

Rosenfelder's book contains the information (for the most popular disk operating systems) to enable you to change the data control block memory locations in lines 6120 and 6140 to be compatible with your DOS.

The save to disk routine is self prompting. All answers to prompts are single key strokes.

The number of screens that can be saved is determined by available disk space. The saved screens are directed to drives 0,1,2,3 in lines 6130, 6132, 6134, and 6136 respectively. These lines should be modified to match your system.

Dedicating screen 0 on drive 0 to a table of contents of your screens would be a simple way to keep track of what is where. This table of contents can be created with the normal keyboard font routine.

Charts, graphs, family trees, indexes, the different screens required for animations, and lots of fun are just a few of the uses that I can think of for SKETCHPAD. DRAW!

---- BIBLIOGRAPHY

ROSENFELDER, LEWIS:Basic Faster and Better and Other Mysteries: IJG Inc.: Upland, California (1981) Pages 171, 172 and 274.

RADIO SHACK a division of TANDY CORPORATION: Level II Basic Reference Manual: Tandy Corporation Fort Worth, Texas (1979)

Roger Anderson
ROGERAND @ SBUG-80

```
10 CLS:CLEAR2000:T=65:C=15360:B=63:Q=1:DIMW(1023)
60 GOSUB 8000
110 POKEC,42
115 'error trap...
117 ON ERROR GOTO 10000
120 Z$=INKEY$:IF Z$="" THEN 120
124 IF ASC(Z$)-30<0 THEN Z$=CHR$(ASC(Z$)+84)
125 IF ASC(Z$)>64 THEN 150
130 ON ASC(Z$)-30 GOSUB
10,160,170,180,190,200,210,220,230,240,250,260,270,280,290,300,310,320,330,330,3
30,330,330,330,9010,9010,410,420,430,440,450,460,470,120
140 GOTO 110
150 ON ASC(Z$)-64 GOSUB
490,490,490,490,490,490,490,490,490,490,490,490,4000,4020,4110,3000,5000,660,600
0,680,690,700,710,720,730,740,750,760,770,780
155 GOTO 110
160  C=C+1:POKE C-1,128:RETURN
330 B=63+(ASC(Z$)-49)*12:T=65:GOTO 8000
440 Q=-1:RETURN
460 Q=1:RETURN
470 CLS:T=65:C=15360:Q=1:B=63:GOTO 8000
490 POKEC,ASC(Z$)+B:C=C+Q:IFC>16383 THEN C=15360
495 RETURN
750  C=C-64:POKEC+64,128:IF C<15360 THEN C=C+960 :'[(UP)
755 RETURN
760  C=C-1:POKEC+1,128:IF C<15360 THEN C=16383:'(LEFT)
765 RETURN
770  C=C+1:POKEC-1,128:IF C>16383 THEN C=15360:'(RIGHT)
775 RETURN
780  C=C+64:POKEC-64,128:IF C>16383 THEN C=C-960:'(DOWN)
785 RETURN
3000 G=128:F=4:E=1:'3000-3120(PRINT CODE TO PRINTER)
3010 LPRINTTAB(F)"("E")  ";
3020 F=F+10:E=E+1:IF E>6 THEN 3030 ELSE 3010
3030 LPRINT:LPRINT
3040 L=65:R=0
```

```
3050 FOR F=3 TO 55 STEP 10
3060 LPRINTTAB(F)CHR$(L)"= "CHR$(G+32);
3070 G=G+12
3080 NEXT F
3090 LPRINT:LPRINT
3100 R=R+1:L=L+1:G=G-71:IF R>11 THEN 3120
3110 GOTO 3050
3120 LPRINT:RETURN
4000 J=C:A=C:I=1:'START SCREEN SECTION PICKUP
4010 RETURN
4020 S=C-J:'4020-4090 PICKUP SCREEN SECTION
4030 J=J+64 :IF J>C THEN 4040 ELSE 4030
4040 J=J-64:K=C-J-1
4050 FOR P=0 TO K
4060 W(I)=PEEK(A+P)
4070 I=I+1
4080 NEXT P
4090 A=A+64:IF A>C THEN RETURN
4100 GOTO 4050
4105 'WRITE PICKEDUP SECTION TO SCREEN(4110-4170)
4110 I=1:D=C-1
4120 FOR P=0 TO K
4130 POKED+P,W(I)
4140 I=I+1
4150 NEXT P
4160 D=D+64:IF D>C+S THEN 4170 ELSE 4120
4170 C=C+S:RETURN
5000 'SCREEN DUMP TO PRINTER
5010 FOR X=15360 TO 16256 STEP 64
5020 FOR Y=0 TO 63
5030 A=PEEK(X+Y)
5035 IF A>127 AND A<192 THEN A=A+32
5040 LPRINTCHR$(A);
5050 NEXT Y
5060 LPRINT
5070 NEXT X
5080 GOTO 110
6000 'SAVE SCREEN TO DISC
6005 POKE C,128
6010 PRINT @960,CHR$(255);
6020 PRINT @960,"TYPE R=READ FROM DISC OR W=WRITE TO DISC";
6030 Z$=INKEY$:IF Z$="" THEN 6030
6040 IF ASC(Z$)=31 THEN RETURN
6050 IF Z$="R" OR Z$="W" THEN 6060 ELSE 6020
6060 A$=Z$
6070 PRINT @960,"ENTER SCREEN NUMBER 0 TO 9 OR A TO Z";
6080 Z$=INKEY$:IF Z$="" THEN 6080
6090 IF ASC(Z$)=31 THEN RETURN
6100 IF ASC(Z$)>64 THEN SN%=ASC(Z$)-54:GOTO 6120
6110 SN%=VAL(Z$)+1
6120 PF%=1
6125 PRINT @960,"       ENTER DRIVE NUMBER  (0,1,2, OR 3)";
6126 Z$=INKEY$:IF Z$="" THEN 6126
6127 ON VAL(Z$)+1 GOTO 6130,6132,6134,6136
6129 IF ASC(Z$)=31 THEN RETURN
6130 OPEN "R",PF%,"DISPLAY1/RND:0
```

```
6131 GOTO 6140
6132 OPEN "R",PF%,"DISPLAY1/RND:1
6133 GOTO 6140
6134 OPEN "R",PF%,"DISPLAY1/RND:2
6135 GOTO 6140
6136 OPEN "R",PF%,"DISPLAY1/RND:3
6140 POKE 26318,0:A1%=SN%84-3
6150 FOR A%=0 TO 3
6160 POKE 26319,60+A%
6170 IF A$="W" THEN PUT PF%,A1%+A% ELSE GET PF%,A1%+A%
6180 NEXT A%
6190 CLOSE PF%
6200 Z$="1":T=65:B=63:GOTO 8000
8000 PRINT @960,CHR$(255);
8010 FOR X=960TO1015STEP5
8020 PRINT@X,CHR$(T)"="CHR$(T+B)" ";
8030 T=T+1:IF T+B>191 THEN RETURN
8050 NEXT X
8060 RETURN
9000 'ALPHABET ROUTINE
9010 PRINT @960,"                    ALPHA/NUMERIC                    ";
9020 Z$=INKEY$:IF Z$="" THEN 9020
9030 IF Z$="<" THEN Q=-1:GOTO 9020
9040 IF Z$=">" THEN Q=1: GOTO 9020
9050 IF ASC(Z$)=9 THEN C=C+1:POKE C-1,128:IF C>16363 THEN C=15360
9060 IF ASC(Z$)=8 THEN C=C-1:POKE C+1,128:IF C<15360 THEN C=16383
9070 IF ASC(Z$)=10 THEN C=C+64:POKEC-64,128:IFC>16383 THEN C=C-960
9080 IF ASC(Z$)=91 THEN C=C-64:POKEC+64,128:IF C<15360 THEN C=C+960
9090 IF ASC(Z$)=31 THEN Z$="1":T=65:B=63:GOTO 8000
9100 IF ASC(Z$)>10 AND ASC(Z$)<91 THEN 9110 ELSE 9120
9110 POKE C,ASC(Z$):C=C+Q:POKE C,42:GOTO 9020
9120 POKE C,42:GOTO 9020
10000 RESUME 120:'ERROR TRAP
```

HELPFUL HINT ON TRANSFERING TAPE TO DISK

SOME AMONG US MIGHT HAVE HAD TROUBLE IN TRYING TO PUT GAMES LIKE THE ADVENTURE GAMES SERIES FROM TAPES TO DISK USING LMOFFSET UTILITY. THE PROBLEM IS WE ALWAYS GET THE 'P#' ERROR WHICH MIGHT HAVE BEEN PUT THERE INTENTIONALLY BY THE AUTHOR. FORTUNATELY, THERE IS ONE WAY TO OVERCOME THIS: COMBINING LMOFFSET/CMD AND CMDFILE/CMD.

FIRST YOU HAVE TO USE CMDFILE TO PUT THE PROGRAM FROM TAPE TO DISK, OF COURSE THE LOADED PROGRAM CANNOT BE RUN BECAUSE THE OVERLAPING ON DOS. NOW USE LMOFFSET FOR THE NEW DISK VERSION, CHOOSE THE RIGHT ADDRESS ABOVE THE DOS LOCATION, APPEND THE SMALL APPENDAGE, DISABLE THE DOS AND SAVE THE PROGRAM WITH A NEW FILESPEC. IF ALL GOES WELL YOUR PROGRAM SHOULD EXECUTE FROM DOS.

I BELIEVE CMDFILE/CMD IS AVAILABLE IN ONE OF OUR LIBRARIES AND OF COURSE LMOFFSET/CMD IS A NEWDOS80 UTILITY.

BICH TRAN

SBUG-80 USERS
Alphabetically by LAST NAME

EVERYONE.....Messages to ALL
DICKALLE.....Dick Allen
ANDERSON.....Tom Anderson
ROGERAND.....Roger Anderson
STAN.....Stan Anticouni
BARRETT.....Barry J.R. Barrett
BAS.....Steve Basye
BENNETT.....William Bennett
BLAINE.....Richard A. Blaine
BREWER.....Eric T. Brewer
SUNNY.....Sunny Briese
LNW2MDX.....Sam Brown Jr. (SVALE)
SAMBROWN.....Sam Brown
EDITOR.....Robert Byrd
RON.....Ron Carpenter
CARRE.....Don Carre
TMMCO.....Ken Chaaraya
CHOW.....Kam Chow
CAP.....Phill Coffman
JIMDINK.....James Dinkey
GDIXON.....Gary Dixon
MARTY.....Martin Earley
DAVIDF.....David Fibush
FORBES.....Ross W. Forbes
DAVEFOX.....Dave Fox
ROGERFUR.....Roger Furner
BGEORGE.....Brian George
GILL.....Richard Gill
TREAS.....Larry C. Gunderson
EDHALL.....Edwin Hall
WAYMAN.....Wayman Hall
PABLO.....Paul L. Hanz
K2L.....Kirk Charles Harmon
TOMHER.....Tom Hering
HODGSON.....John Hodgson
HOOVER.....Edward F. Hoover
HUNTLEY.....Wright H. Huntley
HYGH.....David Hygh
DON.....Don Irving
JIBBYTRM.....William K. Jibby
ANTONJOH.....Anton Johnson
PETERK.....Peter Kamas
SABRI.....Sabri Kawash
JEFFLSMN.....Jeff Lasman
GORDON.....Gordon Leak
GILSLEE.....Gil S. Lee
LEE.....Ronald W. Lee
TED.....Ted Lester
LOUDEN.....Bob Louden
MCTUG.....Marin County Users Group
MARRAZZO.....Joe Marrazzo
PATRICIA.....Patricia Marrazzo
TIHULK.....Jerry Martin
MASTERS.....George Masters
MCHENRY.....Mike McHenry
GMCKEE.....Gerald F. McKee
MAC.....David J. McLaughlin
GFMET.....Glenn Metcalf
NIELS.....Niels Mortensen
PHONES.....David E. Newman
NEWT.....Stan Newton
OLE.....Mike Olson
OCTUG.....Orange County Users Group
MIKEVP.....Mike Van Pelt
PERRY.....Randolph Perry
PHELPS.....Sandy Phelps
BILLRAM.....William Ramsey
OPTOBILL.....Bill Richerson
SCHNELL.....Ed Schnell
TOMS.....Tom Shoemaker
DANSHUM.....Daniel T. Shum
LSMITH.....Leland Smith
VIGO.....Vigo Smith
STORMY.....Bob Storaberg
CLONER.....Jeffery D. Thomas
BERNIE.....Bernard Thompson
GLENN.....Glenn Vaughn
LEEVERN.....Lee Verna
WALKER.....John R. Walker
WATKINS.....Ken Watkins
MARV.....Marvin Wiley

SBUG-80 USERS
Alphabetically by USERNAME

ANDERSON.....Tom Anderson
ANTONJOH.....Anton Johnson
BARRETT.....Barry J.R. Barrett
BAS.....Steve Basye
BENNETT.....William Bennett
BERNIE.....Bernard Thompson
BGEORGE.....Brian George
BILLRAM.....William Ramsey
BLAINE.....Richard A. Blaine
BREWER.....Eric T. Brewer
CAP.....Phill Coffman
CARRE.....Don Carre
CHOW.....Kam Chow
CLONER.....Jeffery D. Thomas
DANSHUM.....Daniel T. Shum
DAVEFOX.....Dave Fox
DAVIDF.....David Fibush
DICKALLE.....Dick Allen
DON.....Don Irving
EDHALL.....Edwin Hall
EDITOR.....Robert Byrd
EVERYONE.....Messages to ALL
FORBES.....Ross W. Forbes
GDIXON.....Gary Dixon
GFMET.....Glenn Metcalf
GILL.....Richard Gill
GILSLEE.....Gil S. Lee
GLENN.....Glenn Vaughn
GMCKEE.....Gerald F. McKee
GORDON.....Gordon Leak
HODGSON.....John Hodgson
HOOVER.....Edward F. Hoover
HUNTLEY.....Wright H. Huntley
HYGH.....David Hygh
JEFFLSMN.....Jeff Lasman
JIBBYTRM.....William K. Jibby
JIMDINK.....James Dinkey
K2L.....Kirk Charles Harmon
LEE.....Ronald W. Lee
LEEVERN.....Lee Verna
LNW2MDX.....Sam Brown Jr. (SVALE)
LOUDEN.....Bob Louden
LSMITH.....Leland Smith
MAC.....David J. McLaughlin
MARRAZZO.....Joe Marrazzo
MARTY.....Martin Earley
MARV.....Marvin Wiley
MASTERS.....George Masters
MCHENRY.....Mike McHenry
MCTUG.....Marin County Users Group
MIKEVP.....Mike Van Pelt
NEWT.....Stan Newton
NIELS.....Niels Mortensen
OCTUG.....Orange County Users Group
OLE.....Mike Olson
OPTOBILL.....Bill Richerson
PABLO.....Paul L. Hanz
PATRICIA.....Patricia Marrazzo
PERRY.....Randolph Perry
PETERK.....Peter Kamas
PHELPS.....Sandy Phelps
PHONES.....David E. Newman
ROGERAND.....Roger Anderson
ROGERFUR.....Roger Furner
RON.....Ron Carpenter
SABRI.....Sabri Kawash
SAMBROWN.....Sam Brown
SCHNELL.....Ed Schnell
STAN.....Stan Anticouni
STORMY.....Bob Stormberg
SUNNY.....Sunny Briese
TED.....Ted Lester
TIHULK.....Jerry Martin
TMMCO.....Ken Chaeraya
TOMHER.....Tom Hering
TOMS.....Tom Shoemaker
TREAS.....Larry C. Gunderson
VIGO.....Vigo Smith
WALKER.....John R. Walker
WATKINS.....Ken Watkins
WAYMAN.....Wayman Hall

** DFT PROTOCOL **

This note is for those of you who want to code your own terminal program or are just curious about how DFT works. I don't claim to fully understand the internals of DFT (I haven't even seen a manual) but I am successfully down/up loading files on SBUG-80. I would be happy to share the source code for the DFT commands or my entire terminal program.

RECORD FORMAT

DFT transmits records that are 132 bytes long as follows:

Byte 1 - control byte (SOH or EOT - see below)
Byte 2 - record number, modulus 256 (starting at 1) in hex
Byte 3 - negative sum of bytes 1 and 2 in hex
Bytes 4 to 131 - 128 data bytes (zero filled if short)
Byte 132 - check sum of bytes 4 thru 131 in hex

where the control byte is:

SOH (01H) - start of 132 byte record
EOT (04H) - end of text (1 byte record)

HANDSHAKING

The convention followed by DFT to control transmission is:

NAK (15H) - Negative acknowledgement is sent by receiver to initiate transmission or to request that a record be resent because of error.
ACK (06H) - Positive acknowledgement is sent by receiver to indicate that record was received correctly.
CAN (18H) - Sent by receiver or sender to abort (cancel) the transmission.

TIMING

The sender has approximately 100 seconds to begin transmission before the receiver sends CAN and aborts.

Once transmission has started, the receiver has approximately 10 seconds to acknowledge correct receipt (ACK) or to request that the record be resent. If neither ACK nor NAK is received within 10 seconds (timeout), the record is automatically resent. After 10 tries on a given record the transmission will be aborted (no limit to the total number of resends in a transmission).

In the final exchange, the sender sends EOT (1 byte) and the receiver responds with ACK; this exchange is also subject to 10 timeouts/NAKs.

OPERATION

1. To down load a file from SBUG-80

    a. Enter the command "DFT S filespec" on SBUG-80

DFT will verify that the file exists and send the message: "Ready to send - awaiting initial NAK".

b. Initiate the DFT receive function on your system. The initial NAK will be sent and each record transmitted automatically. At the end of transmission DFT will send the message "File transfer complete" and the file will be written to your disk.

2. To upload a file to SBUG-80

a. Enter the command "DFT R filespec" on SBUG-80. DFT will send the message "Ready to receive - awaiting transmission" and then send NAK's every 10 seconds.

b. Initiate the DFT send function on your system. The file will be loaded into memory and as soon as NAK is received, transmission will begin. At the end of transmission the message "File transfer complete" will be received.

DATA HANDLING

My guess is that DFT (direct file transfer) reads one sector (256 data bytes) from the file and sends it; i.e., the entire file is not read into memory first. My observation is that DFT holds 20 records (10 sectors) in memory prior to writing to disk in receive mode.

The advantage of this approach is that the size of memory does not limit file size. The disadvantage is that the disk motor starts and stops periodically.

Since I have 48K memory with 36K available for the file, my implementation uses memory to hold the file to minimize disk motor wear (and the wear and tear on my nerves).

PROBLEMS

There are two kinds of problems with DFT. The first is due to my implementation while the second is an operating system problem.

1. File size is limited to 36K in my implementation. Prior to up/loading a file, the directory should be checked to verify that the file will fit in memory.

2. My implementation writes 128 byte records, DFT on SBUG-80 writes 256 byte records to disk. Since the original EOF (end of file) byte is not sent with the file, the resulting EOF byte will always be 0 or 128 in my implementation or 0 at the SBUG-80 end.

Normally there is no problem; however, some DOS commands and many user programs may read the trailing zeros as part of the record. BASIC files are terminated by 3 zeros so that the additional trailing zeros will not cause a problem.

The "DO" command is an example of a command that will have a problem. The command will read past the last valid input data,

which is terminated by a 0D). At a minimum DOS will ask WHAT? If the file is taken to be data for the FORMAT or BACKUP utilities, the problem could be more serious.

Other examples are data files and ASCII BASIC files. The trailing hex zeros will cause a problem if EOF(1) is used to determine the end of data. After transmiting this file to SBUG-80 and listing it, a string of periods appears on the screen (DOS converts non-displayable characters to period).

R. A. Blaine
BLAINE @ SBUG-80

SBUG-80 Mail Corner

<Message from EDITOR at SBUG-80 07/25/82 03:01:08>
RE: PASCAL/CMD -- PASCAL 80 --
Read command :
Yes it is here, I have seen it... A standard PASCAL compiler and editor for NEWDOS/80 mod I and III!! 'CMD' files can be made from PASCAL source code eliminating the need of an interpreter like BASIC.

This program does limit the size of the source though, how much I don't presently know. It does look interesting though, maybe a real neat way to program...
Rob...

Read command :
<Message from BREWER at SBUG-80 07/25/82 06:01:46>
RE: 'C' and the TRS-80...
Read command :
Hi all...
I have found a 'C' compiler for the TRS-80... If anyone is interested in going in on a group buy, let me know... Remember, the more people there are, the less it is... If you are wondering what 'C' is, it is a PASCAL like language designed by BELL LABS. It is a lot easier to use than PASCAL and a lot more versatile...
—Eric

Read command :
<Message from LNW2MDX at SBUG-80 07/26/82 23:00:31>
RE: MDX E/I
Read command :
AS I STATED IN AN EARLIER MEMO I AM THIS AREAS DEALER FOR THE MDX E/I SO IF ANYONE IN THIS AREA WOULD LIKE POSSIBLY ONE OF THE BEST E/I'S AROUND FOR THE PRICE AND EASE OF ASSEMBLY, CALL FOR MORE INFO (408) 734-0898 YOU WILL NOT REGRET IT I ASSEMBLED IT AND HAVE NEVER BEEN SORRY.
THANX
SAM (SVALE)
(408)734-0898

Read command :
<Message from DAVEFOX at SBUG-80 07/28/82 12:05:12>
RE: PERCOM DD...

Read command :
Hi all,
Can anybody out there tell me how to convert an old Percom Data Separator into a Doubler board? I found out positively that the board I have is an exact duplicate to the old Doubler board. The Doubler seems to be identical in every way except for a few IC's (and the Double Density 1771 chip, of course). If you have every done this mod before I would like to talk. Please leave a msg for me.

Dave <DAVEFOX>
or 408 255-3270
Thanx!

Read command :
<Message from JIBBYTRM at SBUG-80 08/01/82 19:50:07>
RE: PROBLEM WITH NDBO BACKUP AND USE OF SUPERZAP
Read command :
HELLO AGAIN!!!
SOME OF YOU MAY REMEMBER THE PROBLEM I WAS HAVING WITH THE COPY COMMAND OF NDBO AND LOOSING THE BOOT ON TRACK 0 AND SECTOR 0. I FOUND THE PROBLEM. YOU MAY ALL WANT TO CHECK YOUR (BAT) SECTOR ON THIS ONE. THE DISK WAS PROGRAM ED FOR 40 TRACKS. WHY SHOULD THAT MAKE A DIFFERENCE YOU SAY IN A 35 TRACK DRIVE. I USED (DIRCHECK) AND FOUND THE BAD BYTES IN THE (BAT) SECTOR. AS SOON AS I ZAPPED THE DISC TO WORK ON 35 TRACKS THE COPY WORKED FINE. CHECK YOUR BAT SECTOR FOR THE RIGHT INFORMATION. YOU MAY BE SURPRISED !!!!
REGARDING SUPERZAP!!!!
THOSE OF YOU WHO MAY HAVE A COPY OF NEWDOS+ WILL FIND THAT THE SUPERZAP PROGRAM IS IN BASIC AND CAN BE MODIFIED TO DO A LOT OF THINGS. IF YOU HAVE DISK MYSTERIES THERE ARE SOME SUGGESTIONS ON WORKING WITH SUPERZAP AND DIRCHECK TOGETHER. I HAVE REVISED THE PROGRAM PER THE ARTICLE AND IT WORKS GREAT. ONLY ONE MAJOR DIFFERENCE WITH THE NEW SUPERZAP, THE MODXX WORKS DIFFERENTLY BUT IT IS NOT THAT HARD TO LEARN. IF ANYONE IS INTERESTED DROP ME A LINE.
LEARNING MORE EVERYDAY
JIBBYTRM

Read command :
<Message from ANDERSON at SBUG-80 08/03/82 20:25:38>
RE: Sbug financing
Read command :
WOW!!!!! I just did a listing of the users and found over 80 names. It was a great surprise that so many were on board. This gives some hope that we can really finance this thing. There is something that we should all realize that is missing from the list of contributors. That is the time that Rob, Eric, ETAL have put into making the thing go. I consider that I have gotten off very inexpensively compared to the time that these people have put in. If somehow we could get about 1/10th the equivalent out of each of the users we would have a big winner. Get on board guys. This thing is just getting ready to roll. Tom Anderson

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

FIRST CLASS MAIL