

SBTIUG GENERAL MEETING 7 APRIL, 1988
by Norm Knudsen

The April meeting of the SBTIUG was held at the Saratoga Public Library and was called to order at 7:30 P.M. There was 16 members in attendance.

President Mike Ewell first called for a treasurer's report from treasurer, Kevin Daberkow who reported a current balance of \$438.35 after paying for the Post Office Box and the newsletter.. He also reported three membership renewals.

Don Apte stated that he had several copiers in various states of repair that he might consider donating to the club.

Librarian Helmut Fuchs gave an update on our software library which now includes an unarchived copy of FUN-L-WRITER 4.

Mike Ewell announced that he had found an awful lot of TI99/4A software which can be downloaded from the Source. He requested that members review the list and mark those files which they were interested in having downloaded.

Newsletter editor Bill Schult announced that he had some discount tickets (4 day only) for the San Francisco Computer Faire.

No one was quite sure whether the BBS was up or not.

The business session completed, we then adjourned for a demo on the GBS by Helmut Fuchs.

{TREASURERS REPORT}
by Kevin Daberkow

PLEASE look at your mailing label to see if some color has been added. If your membership expiration date has been high-lighted in RED, this is your last issue until you renew. If you membership expiration date is in YELLOW, then you should renew at the May meeting.

>> THE DUES ARE \$15 PER YEAR <<

NOTE: Your membership expiration date can be found on the last line of your mailing label.

If any of the information on your label needs to be changed, please let me know. Call me at (408) 281-7435 or write to me at the following address:

SBTIUG - Treasurer
P.O. Box 23447
San Jose, CA 95153-3447

There were several renewals during the month of April: Jim Van Scyoc, Lou Weston, Eugene McCabe, and William Ferriera Jr. Thank you for your continued support.

There were no expenses paid during the course of the month.

This left a balance of \$543.35 in our treasury at the end of April.

{APRIL DEMO}

There have been numerous people asking me how to configure the latest version of FUNLWEB (4.0) for their systems. This latest release of the popular package has some nifty new features, so I figured it might be worth doing another FUNLWEB demo. If you have version 4.0, I highly recommend that you try it out prior to coming to the May meeting. It may be a bit frustrating, but you will have a much better idea of what I am talking about, and this will give you a chance to ask some "practical" questions.

I am looking forward to seeing you at the May meeting!

Kevin Daberkow

{HELP - HELP - HELP - HELP}

As some of you may know, Don Apte recently donated a couple of desktop copying machines to the SBTIUG. They are Savin copiers models 755 & 780. The model 755 definitely works and the 780 is questionable. I am interested in finding out if any of the club members knows anything about these copiers, or copy machines in general. As I stated above, the 755 does work, but the copies come out "streaked". We are hoping to get one of these copiers working well enough that the club newsletter can be produced upon it. This will save the growing copying costs. I will bring sample output from the model 755 to the May meeting. Again, if you know anything about these copiers and what it might take to get one of them in good working order, please contact:

Kevin Daberkow
Home Phone: (408) 281-7435
Work Phone: (408) 746-3116

Your help in this matter would be much appreciated.

Thanks Kevin

LET'S TALK RAM DISKS PART II

by John F. Willforth
reprinted from ROCKY MOUNTAIN 99ERS

Last month I attempted to present a foundation for us to spring into the subject of RAM DISKS by describing what they are and what features they generally possess. This is the month that I will begin talking about specific RAM DISKS. The order in which they are presented month by month is not related to my personal preference or their public acceptance.

The HORIZON RAM DISK was not the first RAM DISK on the market, but it was the first with STATIC RAM and low power at that. This enabled the units to save the ROS and the FILES in the event of power outages and normal power-down of the PEB for long periods of time (weeks and perhaps months) with the use of only three "AAA" Ni-Cad batteries. The original capacities of the two versions available from HORIZON COMPUTER LIMITED were 104K (360 sectors) and 192K (720 sectors), the difference, if you calculate four sectors per each one K of memory, is the memory used to store the ROS (RAMDISK Operating System).

There were larger RAM DISKS on the market when the Horizon came out, but none that offered the source code openly to any who wanted to develop new applications for this card. This decision was a factor in making the HORIZON RAMDISK the popular unit that it is today as well as the TERRIFIC ENHANCEMENTS both hardware (Mike Ballman) and software (John Johnson) wise. More on this subject later.

The HORIZON RAM DISK (I'll refer to as HRD) was designed to operate with all of the disk controllers on the market (TI, MYARC, CORCOMP, AND PERCOM) and with most 32K memory expansions (TI, MYARC, FOUNDATION, BOXCAR etc.) The HRD is not yet able to operate in the GENEVE (9640) environment, but expect this to change shortly. With the uncertainty of this market, and the advances made every day, the areas that I talk of here are only one more rung of the ladder higher tomorrow, since we have NOT yet seen the apex of the life cycle of the TI-99/4A and it's family of peripherals.

The HRD supports nearly all software that can be loaded onto a disk drive in the normal fashion, file or disk copy with DISK MANAGER II, DM1000, or the MYARC and CORCOMP disk managers. If the program in question has some special track/sector protection you will probably have trouble installing it on the HRD, or for that matter any RAM DISK.

The HRD has a switch selectable CRU Address, which enables you to select it's CRU address easily. The ROS talked of earlier, is RAM and can thus be altered easily as the need arises (improvements to the operating system or customizing). The HRD has a complete set of "CALLS" and can easily have NEW CALLS written by the user. You don't need an EPROM BURNER to implement them on the card. These CALLS are summoned in BASIC or EXTENDED BASIC. They include:

- 1) CALL DN(n) Set drive number
- 2) CALL NO or WF Set/Clear Write Prot. on the drive
- 3) CALL MS(n) Sets Max Sectors the ROS will recognize.
- 4) CALL CO or CF Turns card on/off
- 5) DELETE "XBCALL" Executed after CALL INIT to move all the CALL routines to the LOW BK of expansion memory. CALL LINKS can now access them.
- 6) CALL EX Used with CALL CO/CF allows the user to link to the ROS from BASIC.
- 7) CALL DM Used to load a Disk Manager when the Disk manager files are on the RAMDISK.
- 8) CALL NF(n) Only useful when CRU isn't <1000 to enable Basic and other programs to access the RAMDISK.
- 9) CALL ? :Users can create their own user-defined subroutines. Documentation incl.

The HRD can be purchased as a 192K unit fully assembled and warranted for 90 days from HORIZON for \$195. or as a BARE BOARD w/DOCS and SOFTWARE for \$45, or just the BARE BOARD for \$38 in any quantity. I talked to Ron Grise and was told that the last units he sent out were set up to be easily upgraded to the HRD+ (uses 32K Byte chips and can be expanded to ONE MEG). They include MENU 7.01 as well as their own operating system. BUD MILLS SERVICES provides the complete KIT assemblies for this unit and the HRD+. More on the HRD+ next month. There is just too much difference to include this month.

Ed. Note: I have been using the HRD for nearly as long as it's been available and have had NO problem with it, evidenced by the fact that ROS rev's have been so seldom in coming.

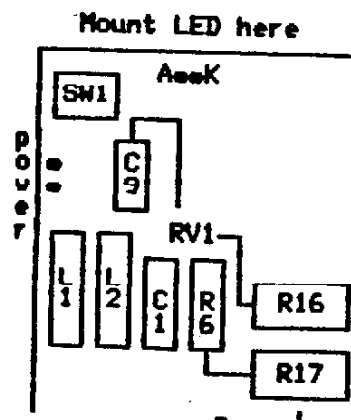
NEW INFO ON POWER SUPPLIES

by BJ Mathis

reprinted from Southwest Ninety Miners

As most members will remember, some of the power supplies we obtained from Radio Shack (cat # 277-1016) had an unsteady 12 volt line, jumping from 9.5-11.5 volts. Those with unstable 12 volt lines had a part number on the board (above the serial number) of 1053214-2, the good ones had a part number of 1053201.

Now, Ed Hallett has discovered by adding a 300 OHM resistor (R6 in diagram) and LED to the "bad supplies" the 12 volt line becomes steady. Now you can use either power supply.



ADDING BOOT-TRACKING TO FAST-TERM

by Chris Schram

Anyone who has used Tony and Will McGovern's FUNNELWEB or John Johnson's BOOT program has been exposed to the concept of Boot-Tracking. Very simply, Boot-Tracking is the ability of a program to know what disk drive it was RUN from. The program then can act more intelligently when it has to access data files or run other programs. The method traditionally used by programmers has been to use the disk name (DSK.MYGAME.HIGHSCORE) when doing file access. A more flexible method is illustrated below. It will run in Console BASIC with the Editor/Assembler or Mini-Memory plugged in. It will also run out of Super Extended BASIC or 6K Extended BASIC.

```
100 REM FOR PROGRAMMING
    ENVIRONMENTS THAT
    SUPPORT "CALL PEEKV"
110 REM
120 CALL INIT
130 CALL PEEKV(16373,DK)
140 PRINT "THIS PROGRAM WAS
    BOOTED FROM";"DSK"&STR$(DK)
```

Try OLDing this program from different drives or RAMdisks (DSK1 - DSK9) and see what happens. Your high score file could now be opened as "DSK"&STR\$(DK)&".HIGHSCORE" without the need to know the disk's name.

Now for a more complicated project. FAST-TERM has been my terminal emulator of choice for some time now. But recently I added a Horizon RAMdisk to my system where I keep the programs and files I use regularly. When FAST-TERM boots up it looks for a CHARA1 file on DSK1 and prompts for a Parameter Filenaae from DSK1. My RAMdisk is currently defined as DSK3, so Boot-Tracking is a definite necessity.

Note: The following procedure will only work on version 1.16/2;ph, the recent enhancement of FAST-TERM by J. Peter Hoddie. The disk I received contains two files that catalog as 34 sectors each and are named MD and ME. Attempting to make these edits to any other version of FAST-TERM is guaranteed to fail.

Copy the MD file to an newly formatted Single-sided Single-density disk. Dig out your sector editor. The first sector of MD should be at disk sector >022 and starts out like this:

```
FFFF 2100 A000 0460 DE26 B310 A00B 04C1
```

Change DE26 to B5FA.

The twenty-third sector of MD should be at disk sector >038 and starts out like this:

```
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 1000 1000 0000
0000 0000 0000 0000 0010 0523 5000 0000
```

Change it to this:

```
02E0 B5FB 0200 3FF5 06C0 D800 8C02 06C0
D800 8C02 1000 D020 8800 0240 0F00 0280
0100 1109 0280 0900 1506 0220 3000 D800
DFB5 D800 DFCD 0460 DE26 1000 1000 0000
0000 0000 0000 0000 0010 0523 5000 0000
```

Check your edits twice and then have someone else check them!

Copy MD, ME, CHARA1, and your parameter file all to the same disk. Try booting from various drives. CHARA1 will be loaded from the boot drive and you will be prompted for a parameter file from the boot drive.

For the programmers out there:

The first operation on the first sector is a branch to the address where the program actually starts. I just changed the destination address to the location of the boot tracking routine which is nested in two workspace registers that are not used until later in the program. This area of memory will, of course, be overwritten as the program progresses. Here is the source code for the beginning of sector >038:

HEX	LABEL	OPCD	OPERAND	COMMENTS
			ADRG >B5FB	
	RE50	DATA	>0000	TEMPORARY WORKSPACE
		ADRG	>B5FA	CLEAR DATA AREA
02E0	DETOUR	LWPI	RE50	ADDR. OF TEMP. WORKSPACE
B5FB				
0200		LI	R0,>3FF5	VDP ADDRESS OF LAST DISK #
3FF5				
06C0		SWPB	R0	LSB FIRST
D800		MOV B	R0,@>8C02	MOVE TO VDP WRITE ADDRESS
8C02				
06C0		SWPB	R0	THEN MSB
D800		MOV B	R0,@>8C02	MOVE TO VDP WRITE ADDRESS
8C02				
1000		NOP		WAIT A MOMENT
D020		MOV B	@>8800,R0	VDP READ DATA
D800				
0240		ANDI	R0,>0F00	\
0F00				
0280		CI	R0,>0100	
0100				
1109		JLT	RESUME	> WITHIN RANGE?
0280		CI	R0,>0900	
0900				
1506		JGT	RESUME	/
0220		AI	R0,>3000	CONVERT TO ASCII
3000				
D800		MOV B	R0,@>DFB5	ADDR. OF DSK1.CHARA1
DFB5				
D800		MOV B	R0,@>DFCD	ADDR. OF DSK#.paramfile
DFCD				
0460	RESUME B		@>DE26	END DETOUR
DE26				
			END	

PILOT LANGUAGE SUMMARY
by Helmut Fuchs

FILE COMMANDS

OF: DSK1.FILENAME OPEN DISK FILE
RF: [n] RESTORE FILE AT RECORD n, 0=DEFAULT, 32767=MAX.
WR: TEXT or #A or #B WRITE TEXT, VAR. #, STRING TO DISK, MAX.= 80 CHAR.
WA: WRITE ANSWER BUFFER TO DISK
RE: #A or #B READ NEXT RECORD FROM DISK TO ANSWER BUFFER
CF: CLOSE DISK FILE

GRAPHICS, BIT MAP MODE: 127 ROWS BY 255 COLUMNS AND 8 TEXT LINES AT BOTTOM

IG: INITIALIZE/CLEAR BITMAP GRAPHICS
GC: FORE, BACK SET GRAPHICS COLORS 1 TO 16
PP: ROW, COL PLOT A POINT
UP: ROW, COL UNPLOT A POINT (DELETE IT)
DL: ROW1, COL1, ROW2, COL2 DRAW A LINE FROM POINT 1 TO 2
DR: ROW1, COL1, ROW2, COL2 DRAW A RECTANGLE FROM 1 TO 2 (UL TO LR CORNER)
DC: ROW, COL, RADIUS DRAW A CIRCLE, WITH CENTER AT R,C OF RADIUS GIVEN
TG: ROW, COL, TEXT TYPE TEXT ON GRAPHIC AREA, START AT ROW, COL
 16 LINES BY 64 CHARACTERS
TC: ROW, COL PLACE TEXT CURSOR AT ROW, COL (ROW 1 TO 5, COL 1-32)

INPUT KEYBOARD COMMANDS

A: ANY TEXT INPUT TO ANSWER BUFFER
A: #A or #B TYPE VARIABLE INTO ANSWER BUFFER
AS: ACCEPT A SINGLE CHARACTER INTO ANSWER BUFFER
M: 1, 2, 3, A, W, F COMPARE ANSWER BUFFER WITH MATCH, SET YES FLAG

JOYSTICK COMMANDS

JS: 1 or 2, #X, #Y PLACE X AND Y POSITION OF JS 1 OR 2 INTO VAR X
 AND Y. VALUE IS 1, 0, -1 FOR UP, CENTER, DOWN
FB: 1 or 2 FIRE BUTTON CHECK, YES FLAG IF PUSHED
 Q AND Y ARE KB FIREBUTTONS

LABELS

*LABEL LABEL FOR JUMP COMMANDS, 10 CHAR. MAX. LENGTH
PR: PROBLEM LABEL, MARKS SECTION OF A PROGRAM
R: TEXT ANY PROGRAM REMARKS

MATH COMMANDS

C: #A <- 55 ASSIGN 55 TO VARIABLE #A
C: #A <- #A + 1 ADD 1 TO #A
C: OPERATIONS ON VARIABLES: + - * / = <> < > <= >= OR AND ATN COS SIN TAN
 INT EXP (e x) LOG (ln x) NEG(#A)

OUTPUT SCREEN COMMANDS

TC: ROW, COL MOVE CURSOR TO ROW, COL (ROW 1-24, COL 1-32)
T: TEXT TYPE ANY TEXT TO SCREEN
TH: TEXT TYPE ANY TEXT ON SCREEN, NO CARRIAGE RETURN
TH: :128: ##12 ##A WORD:: TYPE ASCII, TEXT WITH \$ # : IN IT

PROGRAM CONTROL COMMANDS

J: *LABEL JN: JY: JUMP TO A LABEL, JN or JY ARE CONDITIONAL JUMPS
J: @A JUMP TO LAST A: or AS: COMMAND
J: @M JUMP TO NEXT M: COMMAND
J: @P JUMP TO NEXT PR: COMMAND
JM: *LABEL1[, *LABEL2, ...] JUMP TO LABEL CORRESPONDING TO M: or MJ:
 COUNT (LIKE ON...GOTO)
MJ: 1, 2, A, B JUMP TO NEXT M: or MJ: IF NO MATCH FOUND
BW: BEGIN WHILE LOOP, WH: HAS LOOP BACK CONDITION
WH: #N < 5 GO BACK TO BW: WHILE #N < 5, NEST TO 3 DEEP
LP: 10 BEGIN LOOP 10 TIMES, TO 3 DEEP (- FOR I=1 TO 10)
EL: END OF LOOP (=NEXT I)
U: *LABEL JUMP TO USER SUBROUTINE, GOES TO *LABEL
E: END OF SUBROUTINE OR PROGRAM

PRINTER COMMAND:

TP: TEXT :ASCII: ##12 ##N TEXT:: PRINTER OUTPUT, SPEC'L CHARACTERS DOUBLE

SOUND COMMANDS

S: DUR, FREQ, VOL, VOICE SOUND: DUR 1-255, FREQ 110-32767, VOL 0-28,
 0 IS LOUDEST, VOICE 1-3

SPRITE COMMANDS:

GP: 39, 1898, FF3D, 3C3C, E404 DEFINES ONE OF 42 HEXPATTERNS FOR SPRITES ONLY
SP: 27, 39 ASSIGNS HEXPATTERN 39 TO SPRITE 27
SC: 12, 15 ASSIGN COLOR 15 (1-16) TO SPRITE 12
SS: 4 ASSIGN SPRITE SIZE (1-4) TO ALL SPRITES
SL: 12, ROW, COL ASSIGNS SPRITE LOCATION AT ROW, COL TO SPRITE#12
SM: 12, ROW-VEL, COL-VEL ASSIGNS SPRITE SPEED (-128 TO 127) TO SPRITE#12
SA: SETS YES FLAG IF ANY SPRITES ARE TOUCHING
SH: 12, 14 SETS YES FLAG IF SPRITES 12 AND 14 ARE WITHIN 5
SD: 14 DELETE SPRITE#14
SG: DELETE ALL SPRITES

STRING CONCATENATION:

CS: \$C <- \$A + \$B COMBBINE \$A AND \$B AND ASSIGN TO \$C

TEXT GRAPHICS COMMANDS:

IT: INITIALIZE TEXT GRAPHICS, 24 ROWS BY 32 COL
CH: CLEAR SCREEN, HOME CURSOR TO UPPER LEFT CORNER
SN: 15 SET SCREEN COLOR TO 15
CC: 14,FORE,BACK SET FORE AND BACKGROUND COLORS UP ASCII SET 14
CP: ASCII,1898,FF3D,3C3C,E404 DEFINE AN ASCII CHARACTER (0-255) WITH A NEW
HEX PATTERN: 4 GROUPS OF 4 HEX NUMBERS
HC: ROW,COL,ASCII,REPEAT PLACE ASCII CHAR. AT ROW,COL, REPEAT HORIZ.
VC: ROW,COL,ASCII,REPEAT PLACE ASCII CHAR. AT ROW,COL, REPEAT VERT.

PILOT NOTES:

VARIABLE NAMES: #A..#Z NUMERIC VALUES, FIXED AND FLOATING, NO POWER OF 10
\$A..\$M STRING VARIABLES, UP TO 80 CHARACTERS
ANSWER BUFFER: INPUT PLACED THERE BY KEY INPUT OR DISK, UP TO 80 CHAR.
MATCH: COMPARES ANSWER BUFFER WITH MATCHING EXPECTED ANSWERS
YES FLAG: SET WHEN ANSWER MATCHES WITH MATCH DATA
CONDIT'L JUMPS: JUMPS TO A LABEL IF YES FLAG IS SET OR NOT SET: JY: or JN:
WRITE PROGRAM USING EDIT MODE OF EDITOR/ASSEMBLER AND SAVE UNDER A FILE NAME
RUN PILOT DISK OF EITHER E/A OR XB VERSION, GIVING FIRST DSK1.PILOT, THEN
DSK2.PROGRAMNM FOR SAVED PROGRAM

COLOR CODES: SAME AS BASIC OR XB:

1	TRANSPARENT	6	LIGHT BLUE	11	DARK YELLOW
2	BLACK	7	DARK RED	12	LIGHT YELLOW
3	MEDIUM GREEN	8	CYAN	13	DARK GREEN
4	LIGHT GREEN	9	MEDIUM RED	14	MAGENTA
5	DARK BLUE	10	LIGHT RED	15	GREY
				16	WHITE

ASCII CODES:

THERE ARE 256 ASCII VALUES, 0 TO 255
THESE ARE GROUPED INT 32 SETS OF 8 CHARACTER-SETS, 0 TO 31
ASCII 0-31 ARE CONTROL CHARACTERS: SET 0 TO 3
ASCII 32-47 ARE PUNCTUATION CHARACTERS
ASCII 48-57 ARE NUMBERS 0 TO 9
ASCII 58-63 ARE PUNCTUATION CHARACTERS
ASCII 64-90 ARE UPPER CASE LETTERS A TO Z
ASCII 91-96 ARE PUNCTUATION CHARACTERS
ASCII 97-122 ARE LOWER CASE LETTERS a to z
ASCII 123-127 ARE PUNCTUATION CHARACTERS
ASCII 128-255 ARE FREE FOR ALL