

# TRSTimes

Volume 5. No. 6. - Nov/Dec 1992 - \$4.00



HAPPY HOLIDAYS



# LITTLE ORPHAN EIGHTY



The issue you are looking at is TRSTimes #30. In the very beginning, October 1987, if someone would have told me that my crazy TRSTimes idea would last this long, frankly, I wouldn't have believed them. But, indeed, five years it has been, and it has been a good five years. In the process of publishing TRSTimes, I have made a lot of good friends, and I have learned a lot about my favorite computer. Yes, my Model 4 is still my

chosen machine. I have a fast 386 IBM clone with all the toys, and I have played with Macs and Amigas, but I'm happiest when I get to turn on the Model 4 for a programming session. But I don't need to convince you how great the TRS-80 computers are; you wouldn't be reading this if you didn't agree.

## TRSTimes in 1993

Now let's get on to some business. This issue marks the completion of all 1992 subscriptions. Yes Virginia, your subscription has expired. But not to worry, TRSTimes will continue in 1993 with another six big issues, filled with information, fun and frolic for your TRS-80 Model I/III & 4. So, stay with us, and please get your subscription renewal in early so we can get the unpleasant business of book-keeping out of the way.

Assuming that the U.S. Post Office is not planning a rate hike, we have chosen to keep the subscription prices at the 1992 level; that is, \$20.00 for U.S. and Canada, \$24.00 for surface mail to Europe, Central and South America (\$31.00 for airmail), and \$26.00 for surface mail to Asia, Australia and New Zealand (\$34.00 for air mail).

## TRSTimes on DISK

TRSTimes on DISK #10 is now available. Get your copy now, and don't forget that the previous nine TTOD's are also still available. See the ad on page 32 for details.

## Back Issues

During a search of the TRSTimes vault, we found several sets of 1989, 1990 and 1991 back issues. We also found a few copies of issues 1,2,4 and 5 from 1988. Chose any combination of six issues at the above subscription rates, or \$4.00 per issue for U.S. and Canada; \$6.00 per issue anywhere else. Do yourself a favor, get the issues you're are missing; there's a lot of information in those issues. And do me a favor - get them off of my shelf!

## The new BECK-BBS for TRS-80

And now here is another bit of news. Roy Beck is undertaking the operation of a BBS catering to all us TRS types. The phone number is (213) 664-5056, and should be in operation as you read this. The maximum baud rate is 2400 with 8-N-1 format. Initial testing operation will be at 1200 baud, but 2400 should be on the air almost immediately, so give that a whirl.

The BBS software is FAST PLUS II by Mel Patrick, and is a well-proven system. Any ASCII terminal program will be able to function with it. Mel Patrick offers FASTTERM for use by TRS types, but any good software will function, including XT4. If some other machine than a TRS is used, then of course whatever terminal program is available for that machine should be used.

There are at least two sub-boards in place, one for Model 4 programs, the other for Models I and III. The division in this fashion is simply to help users keep straight which is which. There should be a respectable amount of software available for downloading, but please don't forget this has to be a two-way street. Roy asks that you upload your favorites for others to share. There is, of course, message space for questions and comments by users, with provisions for public and confidential messages. A password system is in use, but this is usual for such BBS', and should not be onerous. Announcements of TRS material wanted and for sale are expected.

Roy has not operated a BBS before, but he is aware that an ongoing problem with all BBS's is checking for and removing material (and people) which do not belong on them. There seems to be a number of people around who delight in "crashing" boards, just for the hell of it. There are also those who post offensive and illegal messages and announcements. Of course, Roy expects you to "keep it clean", as he does not wish to spend a lot of time censoring material. Just remember you are transmitting a form of mail which is open to reading by the public, like postcards, and there is no reason to embarrass any other user.

We hope there will be a significant number of users, and that this will be another useful channel of intercommunication between all of us TRS types. Since I expect to check into the board on a regular basis, this will also be another way for you to communicate with TRSTimes. If you can upload articles, comments, letters, etc to the board, this will save me the time I would otherwise have to spend in retyping from paper copies. Help me, help TRSTimes, and ultimately help yourself!

And now..... Welcome to TRSTimes 5.6



# TRSTimes magazine

Volume 5. No. 6. - Nov/Dec 1992

**PUBLISHER-EDITOR**  
Lance Wolstrup  
**CONTRIBUTING EDITORS**

Roy Beck  
Dr. Allen Jacobs  
Dr. Michael W. Ecker  
**TECHNICAL ASSISTANCE**  
San Gabriel Tandy Users Group  
Valley TRS-80 Users Group  
Valley Hackers' TRS-80 Users  
Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91364. (818) 716-7154.

Publication months are January, March, May, July, September and November.

Entire contents [c] copyright 1992 by TRSTimes publications.

No part of this publication may be reprinted or reproduced by any means without the prior written permission from the publishers.

All programs are published for personal use only. All rights reserved.

1992 subscription rates (6 issues):  
**UNITED STATES & CANADA:**  
\$20.00 (U.S. currency)

**EUROPE, CENTRAL & SOUTH AMERICA:** \$24.00 for surface mail or \$31.00 for air mail.

(U.S. currency only)

**ASIA, AUSTRALIA & NEW ZEALAND:** \$26.00 for surface mail or \$34.00 for air mail.

(U.S. currency only)

Article submissions from our readers are welcomed and encouraged. Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which format is used.

**LITTLE ORPHAN EIGHTY . . . . . 2**  
**Editorial**

**THE MAIL ROOM . . . . . 4**  
**Reader mail**

**A DATE WITH MODEL 4 . . . . . 5**  
**Lance Wolstrup**

**SUMMARY OF RADIO SHACK  
HARD DRIVE PARAMETERS . . . . . 9**  
**Roy T. Beck**

**HINTS & TIPS . . . . . 13**  
**Mohr, TRSTimes vault, McDowell**

**A HACKER'S GLOSSARY . . . . . 18**  
**Jim King**

**LIST UTILITY . . . . . 18**  
**Lance Wolstrup**

**HUNTING FOR BURIED TREASURE . . . . . 25**  
**Lance Wolstrup**

**WHAT THE HECK IS A 26-1138 . . . . . 29**  
**Roy T. Beck**



# THE MAIL ROOM



## RANDOM NUMBERS IN VISICALC

I need a random number while in VISICALC. Do you know of an easy way or hard. To get this number, I would even move to another spreadsheet to get it.

Mike Lingo  
P.O. Box 8074  
Orange, CA 92664

*I am afraid that I am not much of a spreadsheet man, so I can't offer a solution. I did bring this up at a couple of the local TRS-80 user group meetings, but - alas - without success. Can anyone out in TRS-80 land come up with an answer to this?*

*If nothing else works, you could write a program, either in assembler or Basic, that would generate the random numbers you need and save them in a file. This file could then be accessed from VISICALC. I know this is not a particularly graceful solution, but it might work.*

Ed.

## PROGRAM LISTINGS

This is the first opportunity for me to express my appreciation for your magazine. I like the whole magazine, but the articles with the type-in programming are the most informative. They are interesting to type in, and also give me incentive and direction to investigate my computer's systems, and easier ways to do things with it.

John Willoughby  
Hutchinson, KS

*Thank you for the nice words. Program listings will always be a major focus of TRSTimes. I have seen the PC magazines shy completely away from any kind of programming information and, because of it, I have cancelled my subscriptions to them. TRSTimes will keep things the way they used to be back in the good ol' eighties.*

Ed.

## DATE & TIME STAMPING

I program in Basic on a fairly regular basis and continuously update one program or another. To keep track of

my latest version, I insert a line at the beginning of the program with the current date and time. I am doing this manually and sometimes forget. Is there a way to do this automatically?

John C. Demers  
Greenfield, PA

*What you are looking for is a technique called string packing. While it is not your everyday straight-forward Basic programming, it is not particularly difficult! What you want is a line in your program that will update itself to contain the date and time of the latest revision - the following short routine does exactly that!*

```
10 UD$="Last update = dd/mm/yy hh:mm:ss"
11 DT$=DATE$+" "+TIME$
12 X=PEEK(VARPTR(UD$)+1)+
  PEEK(VARPTR(UD$)+2)*256)
13 FOR Y=0 TO 16:
  POKE X+Y+14,ASC(MID$(DT$,Y+1,1)):
NEXT
15 GOTO 100
100 'this is the beginning of your program
```

*This routine is for Model 4 only. For Model I & III, change line 11 to read:*

```
11 DT$=TIME$
```

*You can include this code in whatever program you are working on. Each time the program is RUN, line 10 is changed to the current data and time. If you save the program, and I assume you do after making changes, the modification to line 10 becomes permanent. Also, any LLIST will reflect the changes to line 10.*

*Line 10 is very important. Make sure that UD\$ is exactly 31 characters long. Everything after the space to the right of the equals sign will be changed.*

*Line 11 stores the day and time in DT\$.*

*Line 12 - VARPTR is a powerful tool that allows you to find the address of any string in memory. VARPTR + 1 contains the LSB of the address, and VARPTR + 2 contains the MSB of the address; thus by multiplying what we find at VARPTR + 2 by 256 and then adding what we find at VARPTR + 1, we find the address of the very first character of the string - in this case UD\$.*

*Line 13 - Now that we know where in memory to find UD\$, we set up a loop to POKE the current day and time into UD\$, beginning at the 15th character (UD\$ + 14).*

*Line 15 jumps over any subroutines you might have, to the beginning of your program that would start in line 100.*

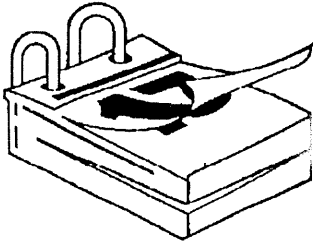
*Good luck with your programming. Let us see some of your efforts.*

Ed.

# A DATE WITH MODEL 4

## Model 4 - Basic

By Lance Wolstrup



A while ago, my wife asked if I had a program that would print a monthly calendar. I immediately put down my autographed copy of 'The Way Things Ought to Be' and began a search through my collection of software. Some hours later I had found several public domain disks con-

taining assorted date and calendar programs. Unfortunately, each proved to be of no value - some just crashed; some worked - but not correctly; some worked fine - but not the way my wife had in mind.

The long and the short of this story is that I sat down on my Model 4 and wrote TRSCAL/BAS to her rather modest specifications. All she wanted was the capability to print a month-by-month calendar for 1993. Being a programmer, however, I couldn't leave well enough alone; now, rather than just printing out the months of 1993, the program can print any month from year 1800 to 3000. This, I trust, should be sufficient for anyone's social agenda!

TRSCAL/BAS begins by prompting the user for the desired month. This is entered as a one or two digit number (1-12).

Next, the program prompts for the year. This can be any year from 1800 to 3000, and is entered as a four-digit number (YYYY).

At this point the program takes over and, using a super algorithm presented by Chris Fara in his 'Basic Days of the Week' article from our Sep/Oct 1992 issue, it displays the chosen month in calendar format on the screen.

The user is then given three options:

(C) another Calendar - pressing C brings the program back to the 'month' prompt.

(P) Print calendar - pressing P print the calendar on your printer. Make sure that the printer is ready before this option is chosen - TRSCAL/BAS does not check for printer ready condition.

(Q) Quit - pressing Q clears the screen and returns to the 'Ready' prompt.

While TRSCAL/BAS does the job it was written to do, I am sure that many modifications can be made. For anyone wishing to play with the program, below is an explanation of the code.

Line 5 - this is the data for the calendar; M\$(1-12) hold the name of the month; ML(1-12) hold the length of the month; W\$(1-7) hold the name of the day.

Line 10 - since all non-string variables are integers, I use DEFINT B-Y. Then the screen width SW is set to 80. Finally, the arrays M\$, ML and W\$ are dimensioned.

Line 12 - the data from line 5 are read into array variables M\$(1-12), ML(1-12) and W\$(1-7).

Line 15 - skips over the subroutines by jumping directly to line 100.

Line 20 - this is part of a series of screen display subroutines that I seem to use constantly. Line 20 must be entered with variable V set as the vertical position of the cursor; variable A\$ must contain the text to display. The subroutine sets variable H (horizontal position of the cursor) to 0 to cause left justified text, and jumps to line 23.

Line 21 - as in line 20, this routine must be entered with V containing the vertical position of the cursor, and A\$ must contain the text to display. Variable H is computed to contain the horizontal position of the cursor to cause centered text.

Line 23 - the routines in lines 20 and 21 both end up here. The cursor position is figured, and the text in A\$ is displayed on the screen. Note that line 23 can also be an independent subroutine. If entered directly, V must hold the vertical position of the cursor; H must hold the horizontal position of the cursor, and A\$ must hold the text to display.

Lines 40 - 44. This is the routine that allows finding the day of the week of any day in any month in any year. Many thanks to Chris Fara for this routine; without it - no TRSCAL/BAS. The only additions to the routine are in line 40, where the name of the chosen month is transferred to M\$, and in line 44, where the length of the month is transferred to ML.

Line 50 - subroutine creating the graphic top line of the week screen display. The first day is done - and is the replicated 6 times.

Line 60 - subroutine creating the graphic top line of the week for the printer.

Line 100 - this is the actual beginning of TRSCAL/BAS. Turn off the cursor, erase the screen, display name and copyright of the program, and draw a graphic line underneath.

Line 110 - turn off cursor (done because program may return here with cursor on), set cursor on line 5, and erase to end of screen, display the month prompt (with flashing cursor).

Line 120 - allow user response to the month prompt; error trap the input - don't allow input length larger than 2, and keep input in range of 1 to 12; if anything wrong, go back to line 110 and do it again.

Line 125 - 126. Make sure that input contains only numbers. If something other than numbers are input, FL is set to 1, and line 126 sends the routine back to line 110 for a better input.

Line 130 - turn cursor off; position cursor on line 7 and erase to end of screen; display 'year' prompt with flashing cursor.

Line 140 - allow user to input the year; turn cursor off; convert user input string to a number; if input is not four characters in length, or input is either smaller than 1800 or larger than 3000 - go back to line 130 and get a better input.

Line 145 - loop through the four character input and set FL to 1 if a non-numeric is found.

Line 146 - if FL = 1 then a non-numeric was found in the input - go back to line 130 and get a better input from the user.

Line 148 - month and year valid, so jump to the subroutine in line 40 to determine the weekday of the first day of the chosen month in the chosen year. The routine returns with D = number of the weekday of the first day of the desired month (1 = Sunday, 2 = Monday, etc). Also returned is ML = length of chosen month.

Line 170 - position cursor at line 0; erase to end of screen (just another way of CLS); display month and year centered on line 0; build the top graphic line of the calendar in A\$; display it on line 1, column 5.

Line 180 - build the day of the week graphic line in A\$ and display it on line 2, column 5.

Line 181 - if month is a normal february and first day of the month is a Sunday we only need four graphic rows - set CA = 12 and jump over line 182.

Line 182 - if the chosen month has 31 days and the first day of the month is a Friday, or the month has 30 at least

30 days and the first day of the month is a Saturday, then we will need to create six graphic rows - set CA = 18; if not, set CA = 15.

Line 185-190 - Loop to create the screen graphics for the calendar outline; display calendar outline on the screen.

Line 200-220 - Loop to display the dates at their correct positions in the calendar outline; begin on line 4; transfer number of first day of the month to DD; loop through all the days in the month (ML); find horizontal position of cursor (H); display the date at V,H (the date is the value of X); increment the day of the week (DD); if DD is larger than 7 (8) then set the vertical position to the next vertical box, and reset DD = 1.

Line 230 - on line 22, display the available commands, 'C', 'P', and 'Q' and prompt with flashing cursor.

Line 240 - wait for keystroke; if upper or lower case Q was pressed - erase screen and end program by returning to Basic Ready.

Line 250 - if upper or lower case C was pressed, program is send back to line 100 so the user can select a new month and year.

Line 260 - if upper or lower case P was pressed, we go to the printer routine in line 300.

Line 270 - invalid key was pressed - program is sent back to line 240 for another keypress.

Line 300 - turn cursor off; erase the 'C', 'P', 'Q' prompt on line 22.

Line 310 - send 2 blank lines to printer; print name of month and year centered; send 2 more blank lines to printer.

Line 320 - uses the subroutine in line 60 to print the top line of the calendar graphic.

Line 330 - loops to build the days of the week in A\$; lprint A\$ at tab(5); set X = 1 and FL = 0 as we will need these in the loop starting in line 340.

Line 340 - uses subroutine in line 60 to lprint the top graphic line of the weekly boxes. If FL = 1 (FL is set when the last weekly box has been lprinted) the program jumps to line 400.

Line 350 - build A\$ with the second-line graphics for the weekly boxes.

Line 360 - if date is larger than end of month ( $X > ML$ ), then if day is not Sunday ( $D < > 1$ ) then lprint the graphic

line containing the dates; set FL = 1 to indicate end of the lprint loop; jump to line 380. On the other hand, if the end of the month is reached (X > ML) and the day is Sunday, then end the loop by jumping directly to line 400. However, if we have not reached the end of the month, then insert the date in the appropriate place in A\$.

Line 370 - increment date (X = X + 1); if day of week is Saturday (D = 7) then lprint graphic line containing the dates for the week. If day is not Saturday, increment day of week (D = D + 1) and jump to line 360.

Line 380 - build third graphic-line in A\$ - and lprint it as the third, fourth, fifth, and sixth lines.

Line 390 - jump back to line 340 and repeat procedure for the remaining weeks.

Line 400 - lprint three blank lines; lprint 'Notes: '; perform top of form and jump back to the 'C', 'P', 'Q' prompt in line 230.

---

### TRSCAL/BAS

---

```

5 DATA JANUARY,31,FEBRUARY,28,MARCH,31,
APRIL,30,MAY,31,JUNE,30,JULY,31,AUGUST,31,
SEPTEMBER,30,OCTOBER,31,NOVEMBER,30,
DECEMBER,31,SUN,MON,TUE,WED,THU,FRI,SAT
10 DEFINT B-Y:
SW = 80:
DIM M$(12),ML(12),W$(7)
12 FOR X = 1 TO 12:
READ M$(X):
READ ML(X):
NEXT:
FOR X = 1 TO 7:
READ D$(X):
NEXT
15 GOTO 100
20 H = 0:
GOTO 23
21 H = INT((SW-LEN(A$))/2)
23 PRINT@SW*V+H,A$;:
RETURN
40 M$ = M$(MM):
IF MM = 2 AND YY/4 = INT(YY/4) AND
YY/400 < > INT(YY/400) THEN ML(2) = 29
41 X = YY + (MM < 3):
YY = YY - 1 + FIX(X/4) - FIX(X/100) + FIX(X/400)
42 MM = (MM - 1 + (MM > 2)) * 2 + FIX((MM - (MM > 7))/2)
43 D = YY + MM + 1:
D = D - FIX(D/7) * 7:
D = D + 1:
ML = ML(VAL(MM$))
44 RETURN
45 A$ = CHR$(191) + STRING$(8,131) + CHR$(191):
FOR X = 1 TO 6:

```

```

A$ = A$ + STRING$(9,131) + CHR$(191):
NEXT:
GOTO 23
50 A$ = CHR$(191) + STRING$(8,32) + CHR$(191):
FOR X = 1 TO 6:
A$ = A$ + STRING$(9,32) + CHR$(191):
NEXT:
RETURN
60 LPRINT TAB(5);STRING$(SW-10,95):
RETURN
100 PRINT CHR$(15):
CLS:
A$ = "TRS-80 MONTHLY CALENDAR":
V = 0:
GOSUB 21:
A$ = "Copyright (c) 1992 by Lance Wolstrup":
V = V + 1:
GOSUB 21:
V = V + 1:
A$ = STRING$(SW,140):
GOSUB 20
110 PRINT CHR$(15);:
V = 5:A$ = CHR$(31):
GOSUB 20:
A$ = "Enter month (MM): " + CHR$(14):
GOSUB 20
120 INPUT MM$:MM = VAL(MM$):
IF LEN(MM$) > 2 OR MM < 1 OR MM > 12 THEN 110
125 FL = 0:
FOR X = 1 TO LEN(MM$):
IF MID$(MM$,X,1) < "0" OR MID$(MM$,X,1) > "9" THEN
FL = 1
126 NEXT:
IF FL THEN 110
130 PRINT CHR$(15);:
V = 7:
A$ = CHR$(31):
GOSUB 20:
A$ = "Enter year (YYYY): " + CHR$(14):
GOSUB 20
140 INPUT YY$:
PRINT CHR$(15);:
YY = VAL(YY$):
IF YY < > INT(YY) OR LEN(YY$) < > 4 OR YY < 1800
OR YY > 3000 THEN 130
145 FL = 0:
FOR X = 1 TO 4:
IF MID$(YY$,X,1) < "0" OR MID$(YY$,X,1) > "9" THEN
FL = 1:X = 4
146 NEXT:
IF FL THEN 130
148 GOSUB 40
170 V = 0:
A$ = CHR$(31):
GOSUB 20:
A$ = M$ + " " + YY$:
GOSUB 21:
V = 1:

```

```

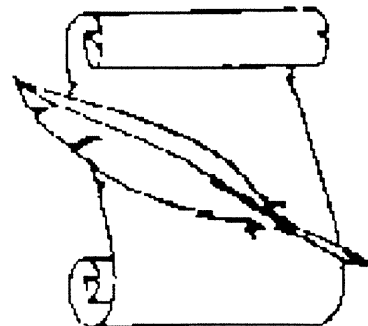
H=5:
A$=CHR$(191)+STRING$(8,131)+CHR$(191):
FOR X=1 TO 6:
A$=A$+STRING$(9,131)+CHR$(191):
NEXT:
GOSUB 23
180 V=2:
H=5:
A$=CHR$(191)+" "+D$(1)+STRING$(4,32)+
CHR$(191):
FOR X=2 TO 7:
A$=A$+" "+D$(X)+STRING$(5,32)+CHR$(191):
NEXT:
GOSUB 23
181 IF ML=28 AND D=1 THEN CA=12:
GOTO 185
182 IF ML=31 AND D=6 OR ML>29 AND D=7
THEN CA=18
ELSE CA=15
185 FOR V1=3 TO CA STEP 3:
V=V1:
GOSUB 45:
FOR Y=1 TO 2:
V=V1+Y:
GOSUB 50:
GOSUB 23:
NEXT
190 NEXT:
V=V+1:
A$=STRING$(SW-10,131):
GOSUB 23
200 V=4:
DD=D:
FOR X=1 TO ML
210 H=DD*10-4:
A$=STR$(X):
GOSUB 23:
DD=DD+1:
IF DD/8=INT(DD/8) THEN V=V+3:
DD=1
220 NEXT
230 V=22:
A$="C = another Calendar, P = Print calendar, Q =
Quit "+CHR$(14):
GOSUB 21
240 I$=INKEY$:
IF I$="Q" OR I$="q" THEN CLS:
END
250 IF I$="C" OR I$="c" THEN 100
260 IF I$="P" OR I$="p" THEN 300
270 GOTO 240
300 PRINT CHR$(15):
V=22:
A$=CHR$(30):
GOSUB 20
310 LPRINT:
LPRINT:
A$=M$+" "+YY$:

```

```

H=INT((80-LEN(A$))/2):
LPRINT TAB(H);A$:
LPRINT:
LPRINT
320 GOSUB 60
330 A$="":
FOR X=1 TO 7:
A$=A$+"!" +D$(X)+STRING$(5,32):
NEXT:
A$=A$+"!":
LPRINT TAB(5);A$:
X=1:
FL=0
340 GOSUB 60:
IF FL GOTO 400
350 A$="":
FOR Y=1 TO 7:
A$=A$+"!" +STRING$(9,32):
NEXT:
A$=A$+"!"
360 IF X>ML THEN
IF D<>1 THEN LPRINT TAB(5);A$:
FL=1:
GOTO 380
ELSE 400
ELSE MID$(A$,D*10-8,LEN(STR$(X)))=STR$(X)
370 X=X+1:
IF D=7 THEN LPRINT TAB(5);A$:
D=1
ELSE D=D+1:
GOTO 360
380 A$="":
FOR Y=1 TO 7:
A$=A$+"!" +STRING$(9,32):
NEXT:
A$=A$+"!":
FOR Y=1 TO 4:
LPRINT TAB(5);A$:
NEXT
390 GOTO 340
400 FOR Y=1 TO 3:
LPRINT:
NEXT:
LPRINT TAB(5);"NOTES:":
LPRINT CHR$(12):
GOTO 230

```





# Summary of Radio Shack Hard Drive Parameters.

by Roy T. Beck

## INTRODUCTION

Recently I received a call from a man who needed to know how many tracks and heads were on a Radio Shack 70 Meg hard drive. Such a simple request, and completely necessary in order to partition and format a hard drive. But where do you find the information when you need it at 1 AM? The answer should be in the Radio Shack hard drive manuals, but it is only presented in a fragmentary and incomplete fashion, unless you have the Service Manual for that particular drive. If you are doing a drive swap, like hot-rodders used to swap engines, forget it. Radio Shack doesn't even want to talk to you!

This article will provide a quick summary of the pertinent factors for the various drives Radio Shack has used in our TRS hard drive packages. I will exclude the 8.4 Meg drive, as that drive is not suitable for the Model I, III and 4 family.

## MODEL I, III and 4 DRIVES

I will include a few other drives which you may run into. To begin with, all drives suitable for use with Radio Shack controllers are categorized as MFM, (meaning modified frequency modulation), and have the same interface as the Seagate ST-412 or Seagate ST-506 drives. This interface physically consists of two card-edge connectors, one with 20 conductors, the other with 34, plus a 4 wire male power cable connector. Usable drives in this category range from 5 to 70 Megs. The maximum head count and cylinder count which can be used by the Radio Shack hard drive controllers (HDC) is 8 and 1024, respectively. All of these drives are full height unless noted to the contrary. Incidentally, the bare hard drive is also commonly referred to as the "bubble", and I occasionally use that term to distinguish the hard drive from the overall package, which includes the HDC and the power supply, all in a case. The bubble is called that because the heads and platters are located inside a sealed dusttight chamber in the drive. Don't ever open the sealed chamber; that should only be done in a Clean Room.

Table 1

Hard Drive (Bubble) Characteristics

Mfg.	Model	Size	Hds	Cyls	Step Rate	Notes	Access
Tandon	TM-602S	5	4	153	10 usec	1	99 ms
Tandon	TM-501	5	2	306	10 usec	2	85
Tandon	TM-502	10	4	306	10 usec	3	85
Tandon	TM-503	15	6	306	10 usec	4	85
Tandon	TM-603S	12	6	230	10 usec	5	99
Seagate	ST-225	20	4	615	10 usec	6	65
Seagate	ST-412	10	4	306	10 usec	7	99
Seagate	ST-506	5	4	153	10 usec	8	
Quantum	Q-540	35	8	512	10 usec	9	45
Micropolis	1325T	70	8	1024	10 usec	10	28

## Notes for Table 1

1. This drive was used in most Radio Shack 5 Meg boxes, Cat No 26-1130. Some of these drives require the step rate to be set at 6 or even more. This corresponds to 3 milliseconds. Try the default value of 10 microseconds. If the step rate is too fast, verification will report numerous bad tracks. If this occurs, repeat the partitioning with 6, which should work.

2. Apparently some Radio Shack 5 Meg boxes had this drive.

3. This drive was not used by Radio Shack, but is mentioned in some of their Service manuals.

4. This drive was used in all the Radio Shack 15 Meg boxes, Cat No 26-4155.

5. This drive was used in all the Radio Shack 12 Meg boxes, Cat No 26-4152.

6. This drive is widely available and works well in Radio Shack boxes. It is a half-height drive, but is a drop-in fit in place of a full height drive.

7. This drive was not used by Radio Shack, but is listed here because it is one of the "generic" MFM drives frequently mentioned.

8. This drive was not used by Radio Shack, but is listed here because it is the other "generic" MFM drive frequently mentioned.

9. This drive was used in the Radio Shack 35 Meg boxes, Cat No 26-4171. It has auto-parking built in; it parks itself at the highest cylinder on loss of power.

10. This drive was used in the Radio Shack 70 Meg boxes, Cat No 26-4173.



## THE INFAMOUS "THREE WIRES"

The "three wires" I am referring to are three wires which run from a harness in the hard drive case to specific points on the bubble, varying with the actual bubble used in the system. Radio Shack soldered these three wires directly to the PC board of the bubble, which could easily be done on the production line. It does pose a problem to us users, especially when we wish to swap bubbles.

It may be useful to explain the purpose of the three wires. The wire colors are as found in a master drive; the colors are different in a slave drive.

The yellow wire responds to the drive select signal. The system can electrically accommodate up to 4 drives, one master and three slaves. There is a drive select line for each of them. The selected drive has 0 volts on its select line, and the other three stand at +5 volts. When the master drive is selected, the yellow wire will be at 0 volts, and the green light may be on. The reason for saying "may" is that the white wire also enters into the picture.

The white wire senses "seek complete", which is a signal returned by the drive to the HDC. While the drive is actively stepping, this line will be at +5 volts, signifying seek is not complete. When seek is complete, the line goes low, and the white wire will be at 0 volts. The white and yellow wires are NORed to turn on the green light when both are low. If the drive is stepping or not selected, the green light goes dark. Thus the steady green light means the drive is selected and is not stepping, and a flickering green light means either the drive is stepping or is momentarily not selected as the DOS checks on a floppy drive, or does something else. Normally the green light is lit on the master drive. If you have a slave connected, its green light will normally be dark.

The orange wire is part of the Write Protect circuit, and sends +5 or 0 volts to the HDC, thereby informing the logic whether the WP switch on the front of the case is depressed. When the switch is depressed and the red light is on, the orange wire is at 0 volts. When the WP light is off, the orange wire is at +5 volts.

A cautionary note on a quirk of the write protect circuit is appropriate. The red lamp is active, in a sense, even when it is dark. When the lamp is dark, +5 volts is passed through it to the remaining logic in the HDC. The quirk is that if the lamp burns out, or fails to make good contact in its socket, then the +5 volts does not get to the logic, and the HDC sees 0 volts, which it interprets as a write protected condition. The result is the DOS cannot write on a drive even though the write protect was not deliberately engaged. With a burned-out lamp, the drive is continuously write protected! The lamp, by the way, is rated 5 volts, 55 milliamp. The current draw is not critical; anything from one milliamp to 100 milliamp will work. The problem

is availability of this special lamp. I have now found a source of these lamps, see my ad elsewhere in this issue. An emergency solution is to replace the burned out lamp with the one from the green ACTIVE socket. The lamps are the same, and the system doesn't care if the green lamp works or not.

I have chased out the original wiring of all the Radio Shack drives, either personally or through other helpful persons. I especially want to thank Art McAninch of Borger, TX and Fred Oberding of Sausalito, CA for their kind efforts and communications.

Table 2 shows the original hookup plus an alternate (and simpler) generic arrangement for master drives, especially where some other bubble is being installed. See the sections headed "generic" for the simpler connections. The advantage of the generic connection is that you can thereafter remove or exchange drives without having to unsolder the three wires as they are now attached to the HDC instead of the bubble.

**Table 2**

### "Three Wires" Connections

#### Early Masters with large HDC's, 8X300 chip type

Wire	Orange	Yellow	White
Function	Write Protect	Drive Select	Seek Complete

#### ORIGINAL CONNECTIONS

5/12/15 Meg	TP-18	U21-8	U7-5
-------------	-------	-------	------

#### GENERIC (SIMPLIFIED) CONNECTIONS

All MFM Drives	HDC resistor pack R27 pin 6 Note 1	HDC J6 pin 26 Note 3	HDC J6 pin 6 Note 3
----------------	------------------------------------	----------------------	---------------------

#### Late Masters with small HDC's, WD1010 chip type

Wire	Orange	Yellow	White
Function	Write Protect	Drive Select	Seek Complete

#### ORIGINAL CONNECTIONS

12/15 Meg	TP-18	U21-8	TP-8
35 Meg	J2-5	U1-5	U15-16
70 Meg	Pad R97 Note 4	Outboard pin, J11	TP-1

#### GENERIC (SIMPLIFIED) CONNECTIONS

All MFM Drives	HDC R54, end nearest J10 Note 2	HDC R23, end nearest U23	HDC U31 pin 8
----------------	---------------------------------	--------------------------	---------------



## Notes for Table 2

1. Solder wire to feedthrough near pin 6. Cut trace to pin 5 of J1 to prevent feeding 5 V into the 20/c cable.

2. Cut trace to pin 5 of J6 to prevent feeding 5 V into the 20/c cable.

3. Use empty pin hole of J6 instead of J5.

4. The pad is marked R97, but no resistor is actually installed there.

All of the drives I deliver to customers are connected in the generic fashion described above. This simplifies matters for both me and the customer, and of course the logic works the same as always, as the generic connection connects the three wires at the destination (the HDC) instead of at the source (the bubble).

Over the last couple of years, I have connected many different drives into our Radio Shack boxes, up to and including the 70 Meg drive, and all work well. Be aware that the 12 Meg, 35 Meg and 70 Meg drives all suffer from a bit of puffery. They were advertised and sold by Radio Shack with those labels, but the labels are only strictly true when installed on machines of the Model II family because those machines formatted more bytes per track than LS-DOS and LDOS can do. The real limits on those drives when used in a Model I, III or 4 system are 11,304,960 bytes, 33,554,432 bytes and 67,108,864 bytes respectively, based upon 8 heads and 230, 512 or 1024 cylinders. Following the usual advertising practice, you could more properly call these 11, 34 and 67 Meg drives. It is not possible to use any drive larger than 67/70 Megs because the HDC cannot deal with more than 8 heads and 1024 cylinders, and the DOS cannot handle other than 32 sectors of 256 bytes each. No matter, the 67 Meg is a LARGE drive, and you are unlikely to fill it.

## SLAVE DRIVES

I have omitted discussing slave drives in this article; The whole point of swapping drives is to get more capacity, and my feeling is that you are better off to put a big drive in the master and file the slave in the closet. You can go as large as 70 Meg in your master, so why fool around with the slave drive?

Just in passing, I will remark that the "three wires" are actually four in number in a slave box, the fourth wire bringing 12 volts to a power relay (which obviates the need for a power switch in a slave), and the wire colors differ from those in a master box.

## MISCELLANEOUS COMMENTS

A few other factors of interest are the sector interleave, the cylinder to reduce write current, the cylinder to begin

precompensation, and the average access time. The sector interleave is predetermined within the formatter, and unless you are good at machine code, is not adjustable. The gurus who designed the Radio Shack hard drive system set this for us. I have not made any attempt to "tune" the interleave, having not had the time (nor the interest) to tackle this area.

The cylinder at which write current should be reduced is specifically stated by the drive manufacturers, but our formatters simply assume a value and go ahead without asking us. If you review the drive specs, the recommended value is typically about half the total number of cylinders, and I believe this is what the formatter programs assume. Furthermore, the newer drives take care of this function in hardware, and so regardless of what the software and controller say, the later drives do their own thing. Don't worry about it, it is not critical.

The cylinder to begin write precompensation varies considerably. Most older drives set it at about 1/2 of the total cylinders, but some of the later ones don't want any precompensation. Again, this value appears not to be critical, and is taken care of in the driver software. I have received one (only) report from a user who said he had to patch his code to suppress precompensation in order to make his drive work, but I lack any further details. It seems not to be a factor of concern, generally.

Finally, the average access time is a bragging point among IBM users, but for us TRS types, hard drives access so much quicker than floppies that the whole process seems like magic anyway. For the drives we are using, the access times are as shown in Table 1, and (except for the 70 Meg drive), are SLOW by IBM standards. Even so, the change from floppies is dramatic, and you will enjoy Radio Shack hard drives anyway.

Most drives have buffered seek, which allows the controller to send stepping instructions at the 10 microsecond rate. The drive actually just holds the count in an internal register until the controller stops sending. The drive then moves its heads at its own best rate, and stops on the desired cylinder. A few very old (mostly 5 Meg) drives lacked this feature, and with these you must tell the controller to send the stepping pulses at 3 millisecond or greater intervals. Always try the 10 microsecond step rate first, and if the formatter chokes, (reports most cylinders bad) then try slower stepping rates until you find one that works.

Here is a warning to NEWDOS-80 V2.5 users. The hard drive version of NEWDOS-80 is V2.5, and it works fine on the old, large HDC boards. However, there is a bug in NEWDOS' formatter which prevents operation on the new, smaller HDC's. Fortunately, this bug was squashed by an Aussie, and a friend of his sent the cure along to me.



The patch to HDFMTAPP, the NEWDOS formatter is as follows, using SuperZap:

DFS of file HDFMTAPP/CMD

FRS1

MOD D1

Find: AF 32 CB 00

Change to: 3E 0F D3 CB

All of the drives in Table 1 are physically and electrically interchangeable. There are actually many more floating around; they just have to be MFM types. The MISOSYS RSHARD5/6 drivers and the PowerSoft Supreme HD Drivers (Series RS) both will work with all of them. Both drivers are available from MISOSYS; The RSHARD5/6 drivers are

easier to install, but the PowerSoft Supreme Drivers allow greater flexibility when you want maximum control over placement of partitions. Both work well, but MISOSYS charges \$5.00 more for the PowerSoft drivers.

If you want to know about a drive not listed in Table 1, write (SASE, please) or call me, and I'll look up the drive in my "Hard Drive Bible" published by Corporate System Center, which lists data for a very large number of drives. If you write, be sure to include the exact Manufacturer's name and Model number.

Roy T. Beck

2153 Cedarhurst Dr., Los Angeles, CA 90027

213-664-5059

## ITEMS OF INTEREST

### FOR SALE

**LIMITED GOLD EDITION - SCOTT ADAMS ADVENTURE SERIES** - Model I/III 16K tape - containing all the famous Scott Adams adventures: Adventureland, Pirate Adventure, Mission Impossible, Voodoo Castle, The Count, Strange Odyssey, Mystery Fun House, Pyramid of Doom, Ghost Town, Savage Island part 1, Savage Island part 2 & Golden Voyage. This is a collectors item \$45.00

**MASTERING THE SAT.** Model III. 4 disks and manual \$15.00

**SERIES 1 EDITOR/ASSEMBLER.** Model I/III tape. RS 26-2011 \$ 8.00

**BUDGET MANAGEMENT.** Model I/III tape. RS 26-1603 \$ 8.00

**EHNBAS** from Cornsoft. Model I/III. Disk & manual \$15.00

**MULTIPLAN.** Model 4. Disk & manual. RS 26-1530 \$20.00

**POWERTOOL.** Model I/III. Disk & manual \$15.00

**GIANT CP/M SPECIAL.** Montezuma Micro CP/M 2.2 for Model 4. Disk & manual, **MONTE'S TOOLKIT, MONTE'S WINDOW, MONTE'S BASCON & 16 Montezuma Micro Public Domain disks.** Also, **CP/M USER GUIDE** from Osborne/McGraw-Hill, **CP/M BIBLE & SOUL OF CP/M** from the Waite Group. Get all for only \$60.00

**MODEL 4/4P TECHNICAL REFERENCE MANUAL.** RS 26-2119 \$10.00

**PROGRAMMER'S SPECIAL. LEARNING TRS-80 MODEL 4/4P BASIC** by David A. Lien, **FAST BASIC** by George A. GRATZER & **PRACTICAL PASCAL.** \$15.00

**ALLWRITE.** Model I. Disk & manual. \$10.00

U.S & Canada please add \$2.00 shipping for each item ordered. Overseas please add \$4.00 per item ordered.

**TRSTimes magazine**  
**5721 Topanga Canyon Blvd., Suite 4**  
**Woodland Hills, CA 91367**



# HINTS & TIPS

## POWER SURGES

By Karl Mohr

A SPIKE is defined as a sudden disturbance of 6,000 volts or greater lasting less than 100 microseconds. In some cases, spikes can cause transistors to fail and/or erasure of solid state memory. Components continually subjected to spikes will suffer from a shortened life span. A surge is a burst of 3,000 volts lasting longer than 100 microseconds.

You can expect more than 100 such spikes and surges each month. And you will probably get a surge of at least 1,000 volts at least once a day. In fact, spikes and surges account for 88.5% of all disturbances.

There's not much we can do in the way of preventing spikes and/or power surges, but as a matter of interest, there is an accompanying surge every time a motor in your home, such as washers, dryers, dishwashers, etc. start and stop.

A small surge is generated when the motor reaches running speed just after it is started, but a larger surge is created when the motor is stopped. This was most evident to me when someone at work connected a spotlight to a bench grinder in parallel with the motor on the MOTOR side of the switch. When the machine was started, the light flared briefly just as the motor reached normal RPM., but the light flared up considerably when the motor was stopped!

Without going into the theories of this, I would suggest that if you aren't using a spike protector on your computer system, then it may be a good time to do so. They are designed so that a SURGE or SPIKE is sent to the neutral power line when the voltage increases to about 200 volts, and not go into your computer system! This is a small investment for the protection it gives, I don't sell them and I won't even recommend a brand type, but they are available for about \$15.00.

## VISICALC TIPS

**#1** -- Reserve the top left corner, the first few Rows and two Columns for documentation, instructions or comments. For example:

```
FILENAME/VC ENDS @N67
PRINT  LOAD DIF FILE CURSOR AT A48
FROM   SAVE AS DIF FILE K48...K67
B4 = = = = >
```

**#2** -- It is possible sometimes to add Rows in a model right at the beginning or end of an existing @SUM range, forgetting that the new line will not be included. Use as the first and last identifiers in the range the label or title at the

top of the column and underline or other separator from TOTAL at the bottom. Then, when you come back and add lines, they will always be included.

**#3** -- One of the more common functions to be performed in accounting applications is to determine whether 'the books are in balance', i.e. whether debits equals credits. If the application is set up so that the debits are entered into one column and the credits in another, this is of course quite easy. If, however, as is more likely to be the case for ease of entering the data, debits and credits appear in the same column with one (usually credits) entered as a negative value, this becomes more complicated.

The problem, very simply stated, is "how do we determine separate sums for the positive numbers and the negative numbers in a single column". Needless to say, the answer to this problem can be useful in non-accounting applications as well.

The secret lies in using the @ABS function, setting up a work column that contains the absolute value of each number in a target column, and using the @ABS function replicated down the column. Use the @SUM function on both the work column and the target column and you have the two partial sums you need to calculate.

To illustrate, if the target column is A and column B contains the simple formula @ABS(A<sub>n</sub>) in each row, then the Visicalc formula to sum the positives is:

**(@SUM(A1...A<sub>n</sub>) + @SUM(B...B<sub>n</sub>))/2**

The formula for calculating the sum of all the negatives in the column is very similar:

**(@SUM(B1...B<sub>n</sub>) - @SUM(A1...A<sub>n</sub>))/2**

If you set up a VC accounting system as outlined above, that puts the transaction amount in one column, the use a work column. You can then create a check location (at the bottom of the model) that keeps a running check for you. Insert into the location the following formula (set here at Row 100 to simulate being at the bottom of the entry area):

**(B100-A100),0,@ERROR**

If the column balances, and debits duly equal credits, a zero will result. If they are out of balance, an appropriate ERROR will appear.. Using a split window that keeps this location on the bottom of the screen, you will have a good running check of your work, and can take the right action if the ERROR shows up.

Of course, since the WORK column is probably not required for the model, it can be anywhere out of the way.



Below is a simple illustration that gives a quick graphic example of this useful utility. By entering a series of digits as positive and negatives, we got an easy-to-throw-out-of-balance column. The formulae are as indicated - in the WORK column it is merely @ABS(TARGET), each column is @SUMed, and in the CHECK = = > location the formula is as above.

TARGET WORK		TARGET WORK	
-123	123	-123	123
123	123	213	213
345	345	345	345
-345	345	-345	345
667	667	667	667
-667	667	-667	667
-346	346	-346	346
346	346	346	346
457	457	457	457
-457	457	-457	457
-546	546	-546	546
546	546	546	546
99	99	99	99
-99	99	-99	99
<hr/>		<hr/>	
0	5166	90	5256
CHECK = = > 0		CHECK = = > ERROR	

**#4 --** If you wish to guarantee that an @SUM location will always include data which might be added later in the range above it, use this simple procedure as a habit: Insert a row of hyphens directly beneath the column headings and another immediately before the @SUM. Use these two row numbers as the beginning and end of the range, and you will always have a formula that correctly adjusts for insertions and deletions.

#### #5 - homebudget template

M26: /F\$ + M11/M3  
 L26: /F\$ + L11/L3  
 K26: /F\$ + K11/K3  
 J26: /F\$ + J11/J3  
 I26: /F\$ + I11/I3  
 H26: /F\$ + H11/H3  
 G26: /F\$ + G11/G3  
 F26: /F\$ + F11/F3  
 E26: /F\$ + E11/E3  
 D26: /F\$ + D11/D3  
 C26: /F\$ + C11/C3  
 B26: /F\$ + B11/B3  
 A26: "savings  
 M25: /F\$ + M10/M3  
 L25: /F\$ + L10/L3

K25: /F\$ + K10/K3  
 J25: /F\$ + J10/J3  
 I25: /F\$ + I10/I3  
 H25: /F\$ + H10/H3  
 G25: /F\$ + G10/G3  
 F25: /F\$ + F10/F3  
 E25: /F\$ + E10/E3  
 D25: /F\$ + D10/D3  
 C25: /F\$ + C10/C3  
 B25: /F\$ + B10/B3  
 A25: "clothing  
 M24: /F\$ + M9/M3  
 L24: /F\$ + L9/L3  
 K24: /F\$ + K9/K3  
 J24: /F\$ + J9/J3  
 I24: /F\$ + I9/I3  
 H24: /F\$ + H9/H3  
 G24: /F\$ + G9/G3  
 F24: /F\$ + F9/F3  
 E24: /F\$ + E9/E3  
 D24: /F\$ + D9/D3  
 C24: /F\$ + C9/C3  
 B24: /F\$ + B9/B3  
 A24: "food  
 M23: /F\$ + M8/M3  
 L23: /F\$ + L8/L3  
 K23: /F\$ + K8/K3  
 J23: /F\$ + J8/J3  
 I23: /F\$ + I8/I3  
 H23: /F\$ + H8/H3  
 G23: /F\$ + G8/G3  
 F23: /F\$ + F8/F3  
 E23: /F\$ + E8/E3  
 D23: /F\$ + D8/D3  
 C23: /F\$ + C8/C3  
 B23: /F\$ + B8/B3  
 A23: "life ins  
 M22: /F\$ + M7/M3  
 L22: /F\$ + L7/L3  
 K22: /F\$ + K7/K3  
 J22: /F\$ + J7/J3  
 I22: /F\$ + I7/I3  
 H22: /F\$ + H7/H3  
 G22: /F\$ + G7/G3  
 F22: /F\$ + F7/F3  
 E22: /F\$ + E7/E3  
 D22: /F\$ + D7/D3  
 C22: /F\$ + C7/C3  
 B22: /F\$ + B7/B3  
 A22: "telephone  
 M21: /F\$ + M6/M3  
 L21: /F\$ + L6/L3  
 K21: /F\$ + K6/K3  
 J21: /F\$ + J6/J3  
 I21: /F\$ + I6/I3  
 H21: /F\$ + H6/H3  
 G21: /F\$ + G6/G3  
 F21: /F\$ + F6/F3



E21: /F\$ + E6/E3  
 D21: /F\$ + D6/D3  
 C21: /F\$ + C6/C3  
 B21: /F\$ + B6/B3  
 A21: "utilities  
 M20: /F\$ + M5/M3  
 L20: /F\$ + L5/L3  
 K20: /F\$ + K5/K3  
 J20: /F\$ + J5/J3  
 I20: /F\$ + I5/I3  
 H20: /F\$ + H5/H3  
 G20: /F\$ + G5/G3  
 F20: /F\$ + F5/F3  
 E20: /F\$ + E5/E3  
 D20: /F\$ + D5/D3  
 C20: /F\$ + C5/C3  
 B20: /F\$ + B5/B3  
 A20: "mortgage  
 N17: /FI + M17 + N13  
 M17: /F\$ + L17 + M16 + M11-M15  
 L17: /F\$ + K17 + L16 + L11-L15  
 K17: /F\$ + J17 + K16 + K11-K15  
 J17: /F\$ + I17 + J16 + J11-J15  
 I17: /F\$ + H17 + I16 + I11-I15  
 H17: /F\$ + G17 + H16 + H11-H15  
 G17: /F\$ + F17 + G16 + G11-G15  
 F17: /F\$ + E17 + F16 + F11-F15  
 E17: /F\$ + D17 + E16 + E11-E15  
 D17: /F\$ + C17 + D16 + D11-D15  
 C17: /F\$ + B17 + C16 + C11-C15  
 B17: /F\$ + A17 + B16 + B11-B15  
 A17: /F\$100  
 M16: /F\$ + B14/12\*L17  
 L16: /F\$ + B14/12\*K17  
 K16: /F\$ + B14/12\*J17  
 J16: /F\$ + B14/12\*I17  
 I16: /F\$ + B14/12\*H17  
 H16: /F\$ + B14/12\*G17  
 G16: /F\$ + B14/12\*F17  
 F16: /F\$ + B14/12\*E17  
 E16: /F\$ + B14/12\*D17  
 D16: /F\$ + B14/12\*C17  
 C16: /F\$ + B14/12\*B17  
 B16: /F\$ + B14/12\*A17  
 A16: "interest  
 O15: /F\$ + N15/N3  
 N15: @SUM(B15...M15  
 H15: 160  
 B15: 160  
 A15: "car insurance  
 B14: /F\$.05  
 A14: "sav acct  
 O13: /F\$ + N13/N3  
 N13: @SUM(B13...M13  
 M13: + M3-@SUM(M5...M11)  
 L13: + L3-@SUM(L5...L11)  
 K13: + K3-@SUM(K5...K11)  
 J13: + J3-@SUM(J5...J11)

I13: + I3-@SUM(I5...I11)  
 H13: + H3-@SUM(H5...H11)  
 G13: + G3-@SUM(G5...G11)  
 F13: + F3-@SUM(F5...F11)  
 E13: + E3-@SUM(E5...E11)  
 D13: + D3-@SUM(D5...D11)  
 C13: + C3-@SUM(C5...C11)  
 B13: /FI + B3-@SUM(B5...B11)  
 A13: "leisure  
 O12: /--  
 N12: /--  
 M12: /--  
 L12: /--  
 K12: /--  
 J12: /--  
 I12: /--  
 H12: /--  
 G12: /--  
 F12: /--  
 E12: /--  
 D12: /--  
 C12: /--  
 B12:/--  
 A12: /--  
 O11: /F\$ + N11/N3  
 N11: @SUM(B11...M11  
 M11: + L11  
 L11: + K11  
 K11: + J11  
 J11: + I11  
 I11: + H11  
 H11: + G11  
 G11: + F11  
 F11: + E11  
 E11: + D11  
 D11: + C11  
 C11: + B11  
 B11: /F\$150 + (B15/6)  
 A11: "savings  
 O10: /F\$ + N10/N3  
 N10: @SUM(B10...M10  
 M10: + L10  
 L10: + K10  
 K10: + J10  
 J10: + I10  
 I10: + H10  
 H10: + G10  
 G10: + F10  
 F10: + E10  
 E10: + D10  
 D10: + C10  
 C10: + B10  
 B10: 120  
 A10: "clothing  
 O9: /F\$ + N9/N3  
 N9: @SUM(B9...M9  
 M9: + L9  
 L9: + K9



K9: + J9  
 J9: + I9  
 I9: + H9  
 H9: + G9  
 G9: + F9  
 F9: + E9  
 E9: + D9  
 D9: + C9  
 C9: + B9  
 B9: 350  
 A9: "food  
 O8: /F\$ + N8/N3  
 N8: @SUM(B8...M8  
 M8: + L8  
 L8: + K8  
 K8: + J8  
 J8: + I8  
 I8: + H8  
 H8: + G8  
 G8: + F8  
 F8: + E8  
 E8: + D8  
 D8: + C8  
 C8: + B8  
 B8: 115  
 A8: "life ins  
 O7: /F\$ + N7/N3  
 N7: @SUM(B7...M7  
 M7: + L7  
 L7: + K7  
 K7: + J7  
 J7: + I7  
 I7: + H7  
 H7: + G7  
 G7: + F7  
 F7: + E7  
 E7: + D7  
 D7: + C7  
 C7: + B7  
 B7: 75  
 A7: "telephone  
 O6: /F\$ + N6/N3  
 N6: @SUM(B6...M6  
 M6: 140  
 L6: 100  
 K6: 50  
 J6: 50  
 I6: 85  
 H6: 85  
 G6: 40  
 F6: 40  
 E6: 80  
 D6: 80  
 C6: 140  
 B6: 140  
 A6: "utilities  
 O5: /F\$ + N5/N3  
 N5: @SUM(B5...M5

M5: + L5  
 L5: + K5  
 K5: + J5  
 J5: + I5  
 I5: + H5  
 H5: + G5  
 G5: + F5  
 F5: + E5  
 E5: + D5  
 D5: + C5  
 C5: + B5  
 B5: 600  
 A5: "mortgage  
 O4: /--  
 N4: /--  
 M4: /--  
 L4: /--  
 K4: /--  
 J4: /--  
 I4: /--  
 H4: /--  
 G4: /--  
 F4: /--  
 E4: /--  
 D4: /--  
 C4: /--  
 B4: /--  
 A4: /--  
 N3: @SUM(B3...M3  
 M3: + L3  
 L3: + K3  
 K3: + J3  
 J3: + I3  
 I3: + H3  
 H3: + G3  
 G3: 2000  
 F3: + E3  
 E3: + D3  
 D3: + C3  
 C3: + B3  
 B3: 1900  
 A3: "income  
 O2: /FR"percent  
 N2: /FR"total  
 M2: /FR"dec  
 L2: /FR"nov  
 K2: /FR"oct  
 J2: /FR"sept  
 I2: /FR"aug  
 H2: /FR"july  
 G2: /FR"june  
 F2: /FR"may  
 E2: /FR"april  
 D2: /FR"mar  
 C2: /FR"feb  
 B2: /FR"jan  
 A2: "month  
 O1: /FR"percent



```

N1: /FR"total
M1: /FI1 + L1
L1: /FI1 + K1
K1: /FI1 + J1
J1: /FI1 + I1
I1: /FI1 + H1
H1: /FI1 + G1
G1: /FI1 + F1
F1: /FI1 + E1
E1: /FI1 + D1
D1: /FI1 + C1
C1: /FI1 + B1
B1: /FI1
A1: "period
    /W1
    /GOC
    /GRA
    /XH18
    /GFI
    /GC9
    /XA1:A2:/TH
    A2:/TV
    /XA1:A1:/GFI
    /GC9
    /XA18:B18:/WS

```

## THE BIG QUESTION

From Paul B. Carroll, writing in the Wall Street Journal;  
 "The joke is told about a theologian who programmed the most powerful supercomputer in the world so he could ask it, 'Is there a God?'"

The computer responded that it lacked the processing power to know. It asked to be connected to all the other supercomputers in the world. Still not enough power. So the computer was additionally hooked up to all the main frames in the world, then all the minicomputers, then all the personal computers. When that still didn't work, the computer asked for a link to all the remaining computer chips - in cars, microwaves, VCR's, digital watches, etc. The theologian asked one final time, 'Is there a God?' The computer responded: 'There is now.'

## MYSTERY for Model I/III

By Bruce McDowell

Here is a reasonably short program for Model I and III - just in time for the upcoming holidays. I won't tell what it does - you'll just have to type it in to find out. Have fun with it.

10 CLS

20 X1 = 63:X2 = X1:Y = 2:

```

MS$ = "20019320820821716020020720420119619321721
1"
30 FOR I = X1 TO X2:SET(I,Y):NEXT
40 X = X + 1:X1 = X1-1:X2 = X2 + 1:Y = Y + 1
50 IF X < > 39 THEN 30
60 READ X1,X2,Y1,Y2
70 IF X1 = -1 THEN 110
80 FOR I = X1 TO X2:SET(I,Y1):NEXT
90 Y1 = Y1 + 1
100 IF Y1 > Y2 THEN 60 ELSE 80
110 READ X1,X2,Y
120 IF X1 < > -1 THEN 140
130 PRINT@0,"";:FOR I = 1 TO 42 STEP 3:
MS = VAL(MID$(MS$,I,3)) AND 127:PRINT CHR$(MS)::
NEXT:GOTO 160
140 FOR I = X1 TO X2:RESET(I,Y):NEXT
150 GOTO 110
160 READ X1,X2,Y
170 IF X1 = -1 THEN 210
180 FOR I = X1 TO X2:RESET(I,Y):NEXT
190 Y = Y + 1
200 IF Y < = 47 THEN 180 ELSE 160
210 SET(53,42):SET(54,43)
220 READ X,Y
230 IF X = -1 THEN 270
240 IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)
250 FOR I = 1 TO 250:NEXT
260 GOTO 220
270 RESTORE
280 READ D
290 IF D = -2 THEN 220 ELSE 280
300 DATA 32,47,41,47,50,57,43,47,60,66,39,47,69,95,
42,47
310 DATA -1,-1,-1,-1,32,47,44,69,95,45,-1,-1,-1,39,40,41
320 DATA 86,87,42,-1,-1,-2,61,7,81,38,74,31,58,20,63,14
330 DATA 87,29,51,37,94,35,45,24,51,17,71,12,33,34,
58,35
340 DATA 30,37,80,27,79,20,66,9,56,11,71,36,48,36,
42,32
350 DATA 40,29,69,24,73,17,53,27,63,29,82,33,98,38,
66,9
360 DATA 87,29,61,7,81,38,74,31,58,20,63,14,87,29,
51,37
370 DATA 94,35,45,24,51,17,71,12,33,34,58,35,30,37,
80,27
380 DATA 79,20,66,9,56,11,71,36,42,36,48,32,40,29,
69,24
390 DATA
73,17,53,27,63,29,82,33,98,38,66,9,87,29,63,2,
-1,-1

```





# A HACKER'S GLOSSARY

Compiled by Jim King

It cannot be said that hackers have failed to make a contribution to the English language. This guide to the jargon of this subculture is selected from a much larger file maintained at MIT by Guy L. Steele, Jr. and at Stanford by Raphael Finkel, Donald Woods, and Mark Crispin.

**BLETCH** 1. interj, Term of disgust. 2. **BLETCHEROUS** adj. Disgusting in design or function. "This keyboard is bletcherous!" Usage: slightly comic.

**BUG** (from telephone terminology, bugs in a telephone cable blamed for noisy lines) n. An unwanted and unintended property of a program.

**CRASH** 1. n. A sudden, usually drastic failure, most often said of the system, sometimes of magnetic disks. "Three 'users' lost their files in last night's disk crash." (See **USER**) 2. v. To fail suddenly. "Has the system just crashed?". Sometimes said of people.

**CROCK** n. An awkward feature or programming technique that ought to be made cleaner.

**CRUFTY** (from cruddy) adj. 1. Poorly built, possibly overly complex. "This is standard old crufty XYZ software." 2. Unpleasant, especially to the touch, often with encrusted junk, like spilled coffee and smeared with peanut butter and ketchup. Hence **CRUFT** n. a disgusting mess.

**DOWN** adj. Not working. "The computer is down." "The up escalator is down" See **UP**.

**FLAME** v. To speak incessantly and/or rabidly on some relatively uninteresting subject, or to speak with a patently ridiculous attitude.

**FROBNICATE** (**FROB**) v. to manipulate or adjust, to tweak. Usage: Frob, Twiddle, and Tweak sometimes connote aimless manipulation; twiddle connotes gross manipulation, often a coarse search for a proper setting; tweak connotes fine-tuning. Someone carefully adjusting a knob on an oscilloscope is probably tweaking it; if he is turning it but looking at the screen he is probably twiddling it; but if he is just turning the knob because it is fun, he's frobbing it.

**GRONK OUT** v. to cease functioning. Of people, to go home and sleep.

**HACKER** (originally someone who made furniture with an axe) n. 1. A person who enjoys learning the details of programming systems and how to stretch their capabilities,

as opposed to most 'users', who prefer to learn only the minimum necessary. 2. One who programs enthusiastically, or who enjoys programming rather than just theorizing about programming.

**LOSE** v. 1. To fail. 2. **LOSER** n. An exceptionally unesthetic person. Generally connotes one who is obnoxious or unusually stupid (as opposed to ignorant).

**MUNG** (recursive acronym for Mung Until No Good) v. 1. To make changes to a file, often large-scale, usually irrevocable. 2. To destroy, usually accidentally, occasionally maliciously.

**PHASE** n. The phase of one's waking-sleeping schedule with respect to the 24 hour cycle. "What's your phase?" "I've been getting in about 8 PM lately, but I'm going to work around to the day schedule by Friday." A person who is roughly 12 hours out of phase is sometimes said to be in **NIGHT MODE**.

**PHASE OF THE MOON** n. Used humorously as a random parameter on which something is said to depend.

**POM** n. Phase Of the Moon. Usually used in the phrase "POM dependent," which means flakey.

**REAL WORLD, THE** n. 1. To programmers, the location of nonprogramming and activities not related to programming. 2. A universe in which the standard dress is coat and tie and in which a person's working hours are defined as 9 to 5. 3. The location of the status quo. "Poor fellow, he's left MIT and gone into the real world." Used pejoratively by those not in residence there. In conversations, talking to someone who has entered the real world is not unlike talking about a deceased person. (the departed?)

**UP** adj. Working, in order. "The down escalator is up." See **DOWN**.

**USER** n. A programmer who will believe anything you tell him. One who asks questions. Identified at MIT with 'loser' by spelling 'luser'.

**WIN** 1. v. To succeed. A program wins if no unexpected conditions arise. 2. **BIG WIN** n. Serendipity.

**WIZARD** n. A person who knows how a complex piece of software or hardware works; someone who can find and fix his bugs in an emergency. Rarely used at MIT where **HACKER** is the preferred term.



# LIST UTILITY

## for Model I & III

By Lance Wolstrup



My wife, Sylvia, is a writer with many magazine articles and books to her credit. She is currently collaborating with Harriet Bronson on a TV-screenplay about the life of famous ex-husband Charlie.

Because there are always many interruptions during the writing sessions, Sylvia brings a Model 100 used to jot down miscellaneous pieces of information dur-

ing the idle time. Her notes range from the latest changes to the screenplay manuscript to names and attributes of the characters of the next book, to simple grocery-shopping lists., to a list of her appointments for the next few days.

At the end of the day, she has usually created 8-10 lists which she wants printed on paper for future reference. As these lists are of various length and written with absolutely no format, other than maybe a comma or a carriage return delimiter, it becomes necessary to go through each file and insert tabs to separate the related fields before being sent to the printer. Sylvia used to do this manually, and it drove me crazy - it was time consuming - so, the next time the Model 100 was free, I wrote a simple program to do the task for her.

Because of the memory constraint of the Model 100, the program was bare-boned - but it did the job! The translation for the Model I & III is called TRSLIST/BAS and is presented below. It has a few more features than the original, and will, hopefully, be of interest to the readers.

TRSLIST/BAS begins by prompting for the name of the file to be loaded. Pressing <ENTER> without a filename, or pressing <ESC> (<SHIFT> <UP-ARROW>) exits the program to Basic. Entering a filename will cause the program to search the drives for a file by that name. If the file is not found, an error message will be displayed and the program flows back to the filename prompt.

If the file is found, TRSLIST/BAS will attempt to load it into memory. If the file is a TRSLIST file, it will load properly and a menu of available options will be displayed. If the file is not a TRSLIST file, the program will, without a doubt, crash.

The menu of available options looks like this:

List to (S)creen or (P)rinter, (N)ew file or (Q)uit

Pressing <ESC> sends the program to the filename prompt.

Choosing S sends the list to the screen - 10 records at a time, and formatted according to the users wishes. Do keep in mind that the program is geared to generate a neat print-out on the printer, so, if the tab settings are set beyond the capabilities of the screen, the video display may turn out messy.

Choosing P prompts the user to 'Ready printer - then press <ENTER>'. Pressing <ESC> here sends the program to the previous prompt - pressing <ENTER> begins the print-out. Each page is headed with the name of the file and the page number.

Choosing N sends the program flow back to the filename prompt.

Choosing Q erases the screen and exits to Basic.

To demonstrate TRSLIST/BAS I have included a couple of TRSLIST files which I am sure you will find interesting and fun. The first, called CELEB1/LST, is a list in two columns consisting of names of celebrities and their original names. Check out Diana Dors, and you'll see why she changed her name, and who is William Henry Pratt?

The second list is called CELEB2/LST. It is in three-column format and contains the names of famous singers/musicians, their place of birth, as well as their date of birth. Now, who is a 52 year old sex symbol, born in Pontypridd?

A TRSLIST file is an unformatted ASCII file. It can be created by any text editor or word-processor, and need only follow a few rules. The very first piece of data must be a number, indicating the number of records in the file; the second piece of data must be a number indicating the number of columns you wish to print; the next pieces of data are also numbers, indicating the tab positions for each column to print; following these numbers is the actual list.

For the readers who are interested in the programming techniques used in TRSLIST/BAS, below is a line-by-line explanation.

Line 10 clears string space, sets screen width (SW) to 64, and jumps over the subroutines to line 100.



Line 20 is the routine to display the contents of A\$ on the screen flush left; this is done by setting variable H = 0, and then jumping to line 23.

Line 21 is the routine to display the contents of A\$ on the screen centered; variable H is calculated and the routine jumps to line 23.

Line 22 is the routine to display the contents of A\$ on the screen flush right; to do this, variable H is calculated to reflect the starting position of the cursor.

Line 23 is where the left, center, and right justify routines end up for the actual printing to the screen. The cursor is set according to H (horizontal) and V (vertical), and the contents of A\$ is printed.

Note that line 23 can be used as a separate print routine, one that prints anywhere on the screen. Simply store whatever text you wish to display in A\$, store the horizontal position of the cursor in H, store the vertical position of the cursor in V, and then GOSUB 23. Sure beats manually figuring out the screen positions!

Lines 30-38 contain the multiple keystroke INKEY\$ routine. I use this type of routine in most of my programs because I am not crazy about the INPUT statement. This routine is entered with three parameters passed from the caller; the vertical and horizontal position of the cursor is stored in V and H, and the number of maximum allowable user keystrokes is stored in ML.

Line 30 displays the maximum allowable keystrokes as periods beginning at the cursor position; the cursor is then made visible; the flag (FL) is set to 0, as is the length of the current user keystrokes (L); A\$, which is used to build the user input, is nilled out.

Line 31 polls for keystroke. If none is found, the program goes back to line 31 and polls again.

Line 32 test if <ENTER> or <ESC> has been pressed. If <ENTER> was pressed then the cursor is made invisible and program flow is returned to the caller. If <ESC> was pressed, the flag (FL) is set to 1; the cursor is made invisible and program flow is returned to the caller. Note that this is the only way to get out of this routine; the calling routine will know how the subroutine was terminated: if FL = 0 the routine was terminated with <ENTER>; if FL = 1 then the routine was terminated with <ESC>.

Line 33 checks if the LEFT ARROW key (chr\$(8)) was pressed at a time when there is no input (L = 0). This, of course, is not allowed, and program flow is directed back to line 31.

Line 34 - we again check if the LEFT ARROW (chr\$(8)) pressed. If it was, we delete the character to the left. The cursor is turned off (chr\$(15)); the horizontal position of

the cursor is decremented; the character to be deleted is overwritten by a period; the cursor (invisible) is positioned on top of the just written period; the length of the user input is decremented (L = L - 1); the user buffer (A\$) is decremented; the cursor is made visible (chr\$(14)), and the program flow is returned to line 31.

Line 35 checks if a character with a value less than the SPACEBAR was pressed. As these are mostly control characters, they are not needed or wanted in this program; thus, if one was pressed, the program flow returns to line 31.

Line 36 makes sure that we do not exceed the allowable maximum length of user keystrokes. If we have reached the max (L = ML) then program flow is returned to line 31.

Line 37 converts the user keystrokes to uppercase, if needed. If the character is lowercase (96 to 122), the routine turns off bit 5 by ANDing the ASC value with all bits on except bit 5 (223).

Line 38 displays the keystroke (I\$); the horizontal position of the cursor is incremented (H = H + 1); the input is added to the input buffer (A\$ = A\$ + I\$); the length of the input is incremented, and program flow is directed back to line 31.

Line 100 is the actual beginning of the program. The screen is erased and the special character set is invoked (POKE 16420,1); the program name and copyrights are displayed on lines 0,1, and 2. Note that Model I should delete POKE 16420,1 and the references to CHR\$(143) + CHR\$(244) + CHR\$(245) + CHR\$(246); these are the special characters, forming the pointing finger, available only in Model III and 4.

Line 110 erases from line 4 to the end of the screen.

Line 120 prompts for the filename of the list, and then uses the INKEY\$ routine in line 30 to get the information.

Line 130 checks if FL is set (ESC was pressed) or A\$ = "" (ENTER pressed without data); if either of these conditions are found to be true, the program restores space compression mode (POKE 16420,0 - Model I skip this statement), erases the screen and ends the program.

Line 140 set up the error trap. The only error the program checks for is the user types a filename that does not exist.

Line 150 open the requested file for input.

Line 160 turns off the error trap, as the file obviously was found. The error trap could be expanded here to trap other error conditions - but you'll have to do that yourself.



Line 170 erases from line 4 to the end of the screen; notify user that the file is being read.

Line 180 read number of record into variable R; read number of columns into variable C.

Line 190 DIMensions the R\$ array and the TB\$ array.

Line 200 reads the tab positions into the TB\$ array.

Line 210 reads the list into the R\$ array.

Line 220 closes the file.

Line 230 erases from line 4 to the end of the screen.

Line 240 displays the available options on line 4.

Line 250 uses the INKEY subroutine to get the user response to the prompt. If ESC was pressed (FL=1) program flow is directed back to the filename prompt. If a new list is requested (N) then restart program (using RUN keeps garbage collection to a minimum). If the user elected to quit (Q) we null out A\$ and use the exit routine in line 130. If a print-out (P) was chosen, we branch to the printer routine in line 340. If the user decided to view the list on the screen (S), we jump to line 260. If none of these options were chosen, then we surmise that a bad input took place and we send the program back to line 230 for another chance at the options.

Line 260 is the screen display routine. We erase from line 4 to the end of the screen.

Line 270 sets up a loop to step through the records in the list.

Line 280 sets up a loop to print the columns of the current record in the list. Since each column is printed using the ; (semicolon) delimiter, we need a PRINT statement at the end to cause a carriage return and a linefeed.

Line 290 makes sure that only 10 records are displayed at any one time. After displaying any record whose number is evenly divided by 10, the routine branches to line 300. This is skipped when the record number is not divisible by 10.

Line 300 is a simple INKEY routine. Press any key and the screen is erased from line 4 on.

Line 310 causes the NEXT record to be chosen.

Line 320 displays the "Press ENTER" prompt on line 15.

Line 330 uses the INKEY\$ routine in line 30. Pressing ENTER branches the program back to line 230 for a repeat of the option prompt. If ENTER was not pressed, the

program loops back to line 320. This ends the screen display routine.

Line 340 is the printer routine. Erase from line 4 to the end of the screen.

Line 350 displays the "Ready printer" prompt.

Line 360 uses the INKEY\$ routine in line 30. If ESC was pressed, the program jumps back to the options prompt.

Line 370 - set line count LC = 0; set page number PG = 1

Line 380 sets up loop to step through the records.

Line 390 prints empty line and increments the line counter to reflect this.

Line 400 prints the page headings if the line count (LC) is 1; LC is incremented; an empty line is printed, and LC is incremented again.

Line 410 - if the line counter is larger than 60 then we need to advance the paper to the top of the next page. To do this, we set up a loop that prints blank lines until we reach the top of the next page; the line counter is reset to 0, the page counter (PG) is incremented, and the program goes back to line 390 print the page header.

Line 420 prints the columns of the current record.

Line 430 goes back to line 380 for another record.

Line 440 is the end of the printer routine. We get here when the last record has been printed and we need to advance the paper to top of form - unless, we are already there. The TOF is done by setting up a loop to print blank lines until we get to the start of the new page.

Line 450 returns the program to the options prompt.

Line 460 is error trap routine. We erase from line 4 to the end of the screen, and then display the "Unable to find file" message.

Line 470 displays the "Press ENTER to continue" prompt.

Line 480 uses the INKEY\$ routine in line 30 to get the user response. If ENTER was not pressed exclusively, the program goes back to line 470 for another try.

Line 490 exits the error trap. I always exit an error trap by jumping to the next line, rather than exiting back into the main body of the program. This way I have some flexibility in handling other errors; now, it isn't implemented here, but I will use this same routine in future programs - so stay tuned!



Line 500 turns off the error trap.

Line 510 goes back to line 110 to prompt for the filename.

---

### TRSLIST/BAS

---

```
1 'TRSLIST/BAS
2 'Copyright 1992 TRSTimes & Lance Wolstrup
3 'All rights reserved
4 '
10 CLEAR 15000:SW = 64:GOTO 100
20 H = 0:GOTO 23
21 H = INT((SW-LEN(A$))/2):GOTO 23
22 H = SW-LEN(A$)
23 PRINT@V*SW + H,A$;:RETURN
30 A$ = STRING$(ML,46):GOSUB 23:
A$ = CHR$(14):GOSUB 23:FL = 0:L = 0:A$ = ""
31 I$ = INKEY$:IF I$ = "" THEN 31
32 IF I$ = CHR$(13) THEN PRINT CHR$(15);:RETURN
ELSE IF I$ = CHR$(27) THEN FL = 1:PRINT CHR$(15);:
RETURN
33 IF I$ = CHR$(8) AND L = 0 THEN 31
34 IF I$ = CHR$(8) THEN PRINT CHR$(15);:H = H-1:
PRINT@V*SW + H,CHR$(46);:PRINT@V*SW + H,"";:
L = L-1:A$ = LEFT$(A$,L):PRINT CHR$(14);:GOTO 31
35 IF I$ < CHR$(32) THEN 31
36 IF L = ML THEN 31
37 IF ASC(I$) > 96 AND ASC(I$) < 123 THEN
I$ = CHR$(ASC(I$)AND 223)
38 PRINT I$;:H = H + 1:A$ = A$ + I$:L = L + 1:GOTO 31
100 CLS:POKE 16420,1:V = 0:
A$ = "TRSTimes Presents: " + CHR$(143) + CHR$(244) +
CHR$(245) + CHR$(246) + "  T R S L I S T":GOSUB 20:
V = 1:A$ = "Copyright (c) 1992 by Lance Wolstrup - All
rights reserved":GOSUB 21:V = 2:
A$ = STRING$(SW,131):GOSUB 20
110 V = 4:A$ = CHR$(31):GOSUB 20
120 PRINT"Enter name of file to list: ";:
H = 28:ML = 12:GOSUB 30
130 IF FL OR A$ = "" THEN POKE 16420,0:CLS:END
ELSE F$ = A$
140 ON ERROR GOTO 460
150 OPEN"I",1,F$
160 ON ERROR GOTO 0
170 V = 4:A$ = CHR$(31):GOSUB 20:
A$ = "Reading " + F$:GOSUB 21
180 INPUT#1,R:INPUT#1,C
190 DIM R$(R,C),TB(C)
200 FOR X = 1 TO C:INPUT#1,TB(X):NEXT
210 FOR X = 1 TO R:FOR Y = 1 TO C:
INPUT#1,R$(X,Y):NEXT:NEXT
220 CLOSE
230 V = 4:A$ = CHR$(31):GOSUB 20
240 A$ = "List to (S)creen or (P)rinter, (N)ew file or
(Q)uit ":GOSUB 21:H = H + LEN(A$)
```

```
250 ML = 1:GOSUB 30:IF FL THEN 110 ELSE IF
A$ = "N" THEN RUN ELSE IF A$ = "Q" THEN
A$ = "":GOTO 130 ELSE IF A$ = "P" THEN 340 ELSE IF
A$ = "S" THEN 260 ELSE 230
260 V = 4:A$ = CHR$(31):GOSUB 20
270 FOR X = 1 TO R
280 FOR Y = 1 TO C:PRINT TAB(TB(Y))R$(X,Y);:NEXT:
PRINT
290 IF X/10 = INT(X/10) THEN 300 ELSE 310
300 IF INKEY$ = "" THEN 300 ELSE V = 4:
A$ = CHR$(31):GOSUB 20
310 NEXT
320 V = 15:A$ = "Press < ENTER > to continue ":
GOSUB 21:H = H + LEN(A$)
330 ML = 1:GOSUB 30:IF A$ < > "" THEN 320 ELSE 230
340 V = 4:A$ = CHR$(31):GOSUB 20
350 A$ = "Ready printer - then press < ENTER > ":
GOSUB 21:H = H + LEN(A$)
360 ML = 1:GOSUB 30:IF FL THEN 230
370 LC = 0:PG = 1
380 FOR X = 1 TO R
390 LPRINT:LC = LC + 1
400 IF LC = 1 THEN LPRINT TAB((80-LEN(F$))/2)F$;
TAB(70)"Page";PG:LC = LC + 1:LPRINT:LC = LC + 1
410 IF LC > 60 THEN FOR Z = LC + 1 TO 66:
LPRINT:NEXT:LC = 0:PG = PG + 1:GOTO 390
420 FOR Y = 1 TO C:LPRINT TAB(TB(Y))R$(X,Y);:NEXT
430 NEXT
440 IF LC < > 0 THEN FOR Z = LC + 1 TO 66:
LPRINT:NEXT
450 GOTO 230
460 V = 4:A$ = CHR$(31):GOSUB 20:
A$ = "Unable to find " + F$:GOSUB 21*
470 V = 6:A$ = "Press < ENTER > to continue ":
GOSUB 21:H = H + LEN(A$)
480 ML = 1:GOSUB 30:IF A$ < > "" THEN 470
490 RESUME 500
500 ON ERROR GOTO 0
510 GOTO 110
```

---

### CELEB1/LST

---

```
112
2,5,35
Alan Alda,Alphonso D'Abruzzo
Woody Allen,Allen Konigsberg
Julie Andrews,Julia Wells
Fred Astaire,Frederick Austerlitz
Lauren Bacall,Betty Joan Perske
Brigitte Bardot,Camille Javal
Pat Benatar,Patricia Andrejewski
Tony Bennett,Anthony Benedetto
Jack Benny,Benjamin Kubelsky
Joey Bishop,Joseph Gottlieb
Robert Blake,Michael Gubitosi
Victor Borge,Borge Rosenbaum
Morgan Brittany,Suzanne Cupito
Charles Bronson,Charles Buchinski
```



Mel Brooks, Melvin Kaminsky  
 George Burns, Nathan Birnbaum  
 Michael Caine, Maurice Micklewhite  
 Cher, Cherilyn Sarkasian  
 Michael Connors, Kreker Ohanian  
 Howard Cosell, Howard Cohen  
 Lou Costello, Louis Cristillo  
 Joan Crawford, Lucille Le Sueur  
 Tom Cruise, Thomas Mopother  
 Tony Curtis, Bernard Schwartz  
 Rodney Dangerfield, Jacob Cohen  
 Bobby Darin, Walden Waldo Cassotto  
 Doris Day, Doris von Kappelhoff  
 Yvonne De Carlo, Peggy Middleton  
 Sandra Dee, Alexandra Zuck  
 John Denver, Henry John Deutschendorf Jr.  
 Bo Derek, Cathleen Collins  
 Diana Dors, Diana Fluck  
 Bob Dylan, Robert Zimmerman  
 Sheena Easton, Sheena Shirley Orr  
 Barbara Eden, Barbara Huffman  
 Chad Everett, Raymond Cramton  
 Morgan Fairchild, Patsy McClenney  
 Stepin Fetchit, Lincoln Perry  
 W.C. Fields, William Claude Dukenfield  
 John Forsythe, John Freund  
 Redd Foxx, John Sanford  
 Connie Francis, Concetta Franconero  
 Greta Garbo, Greta Gustafson  
 Judy Garland, Frances Gumm  
 James Garner, James Bumgarner  
 Stewart Granger, James Stewart  
 Cary Grant, Archibald Leach  
 Lee Grant, Lyova Rosenthal  
 Susan Hayward, Edythe Marriner  
 Rita Hayworth, Margarita Cansino  
 Pee-Wee Herman, Paul Rubenfeld  
 Harry Houdini, Ehrich Weiss  
 Engelbert Humberdinck, Arnold Dorsey  
 David Janssen, David Meyer  
 Elton John, Reginald Dwight  
 Don Johnson, Donald Wayne  
 Tom Jones, Thomas Woodward  
 Boris Karloff, William Henry Pratt  
 Danny Kaye, David Kaminsky  
 Diane Keaton, Diane Hall  
 Michael Keaton, Michael Douglas  
 Chaka Khan, Yvette Stevens  
 Cheryl Ladd, Cheryl Stoppelmoor  
 Veronica Lake, Constance Ockleman  
 Dorothy Lamour, Mary Kaumeyer  
 Michael Landon, Eugene Orowitz  
 Mario Lanza, Alfredo Coccozza  
 Stan Laurel, Arthur Jefferson  
 Brenda Lee, Brenda Mae Tarpley  
 Bruce Lee, Lee Yuen Kam  
 Gypsy Rose Lee, Rose Louise Hovick  
 Peggy Lee, Norma Egstrom

Jerry Lewis, Joseph Levitch  
 Hal Linden, Harold Lipshitz  
 Sophia Loren, Sophia Scicoloni  
 Peter Lorre, Laszio Lowenstein  
 Bela Lugosi, Bela Ferenc Blasko  
 Moms Mabley, Loretta Mary Aitken  
 Shirley MacLaine, Shirley Beaty  
 Madonna, Madonna Louise Ciccone  
 Lee Majors, Harvey Lee Yearly 2nd  
 Jayne Mansfield, Vera Jane Palmer  
 Dean Martin, Dino Crocetti  
 Marilyn Monroe, Norma Jean Mortenson/Baker  
 Demi Moore, Demi Guynes  
 Rita Moreno, Rosita Alverio  
 Harry Morgan, Harry Bratsburg  
 Hugh O'Brian, Hugh Krampke  
 Patti Page, Clara Ann Fowler  
 Jack Palance, Walter Palanuik  
 Bert Parks, Bert Jacobson  
 Minnie Pearl, Sarah Ophelia Cannon  
 Martha Raye, Margaret O'Reed  
 Donna Reed, Donna Belle Mullenger  
 Joan Rivers, Joan Sandra Molinsky  
 Edward G. Robinson, Emmanuel Goldenberg  
 Ginger Rogers, Virginia McMath  
 Roy Rogers, Leonard Slye  
 Mickey Rooney, Joe Yule Jr.  
 Soupy Sales, Milton Hines  
 Jane Seymour, Joyce Frankenberg  
 Omar Shariff, Michael Shalhoub  
 Martin Sheen, Ramon Estevez  
 Connie Stevens, Concetta Ingolia  
 Danny Thomas, Amos Jacobs  
 Tina Turner, Annie Mae Bullock  
 Conway Twitty, Harold Lloyd Jenkins  
 Frankie Valli, Frank Castelluccio  
 John Wayne, Marion Morrison  
 Raquel Welch, Raquel Tejeda  
 Gene Wilder, Jerome Silberman  
 Natalie Wood, Natasha Gurdin  
 Jane Wyman, Sarah Jane Fulks

---

## CELEB2/LST

---

69  
 3,0,22,45  
 Steve Allen, New York NY, 12/26/21  
 Gregg Allman, Nashville TN, 12/07/47  
 Herb Alpert, Los Angeles CA, 03/31/35  
 Paul Anka, Ottawa ONT, 07/30/41  
 Chet Atkins, Luttrell TN, 06/20/24  
 Pat Benatar, Brooklyn NY, 01/10/53  
 Chuck Berry, St. Louis MO, 10/18/26  
 Sonny Bono, Detroit MI, 02/16/35  
 Pat Boone, Jacksonville FL, 06/01/34  
 Victor Borge, Copenhagen Denmark, 01/03/09  
 James Brown, Pulaski TN, 06/17/28  
 Ray Charles, Albany GA, 09/23/30



Cher, El Centro CA, 05/20/46  
 June Christy, Springfield IL, 11/20/25  
 Eric Clapton, Surrey England, 03/30/45  
 Roy Clark, Meherrin TN, 04/15/33  
 Van Cliburn, Shreveport LA, 07/12/34  
 Rosemary Clooney, Maysville KY, 05/23/28  
 Natalie Cole, Los Angeles CA, 02/06/50  
 Phil Collins, London England, 01/30/51  
 Perry Como, Canonsburg PA, 05/18/12  
 Rita Coolidge, Nashville TN, 05/01/45  
 Alice Cooper, Detroit MI, 02/04/48  
 Elvis Costello, London England, 08/25/54  
 Xavier Cugat, Barcelona Spain, 01/01/00  
 Vic Damone, Brooklyn NY, 06/12/28  
 Doris Day, Cincinnati OH, 12/31/43  
 John Denver, Roswell NM, 12/31/43  
 Neil Diamond, Brooklyn NY, 01/24/41  
 Bo Diddley, McComb MS, 12/20/28  
 Fats Domino, New Orleans LA, 02/26/28  
 Donovan, Glasgow Scotland, 05/10/43  
 Bob Dylan, Duluth MN, 05/24/41  
 Billy Eckstine, Pittsburgh PA, 07/08/14  
 Don Everly, Brownie KY, 02/01/37  
 Phil Everly, Chicago IL, 01/19/38  
 Jose Feliciano, Lares Puerto Rico, 09/10/45  
 Freddy Fender, San Benito TX, 06/04/37  
 Eddie Fisher, Philadelphia PA, 08/10/28  
 Ella Fitzgerald, Newport News VA, 04/25/18  
 Roberta Flack, Black Mountain NC, 02/10/39  
 Peter Frampton, Kent England, 04/22/50  
 Connie Francis, Newark NJ, 12/12/38  
 Aretha Franklin, Memphis TN, 03/25/42  
 Bobbie Gentry, Chickasaw Co. MS, 07/27/44  
 Astrud Gilberto, Salvador Brazil, 03/30/40  
 Bobby Goldsboro, Marianna FL, 01/18/42  
 Eydie Gorme, Bronx NY, 08/16/32  
 Merle Haggard, Bakersfield CA, 04/06/37  
 Lionel Hampton, Birmingham AL, 04/12/13  
 Herbie Hancock, Chicago IL, 04/12/40  
 Skitch Henderson, Halstad MN, 01/27/18  
 Al Hirt, New Orleans LA, 11/07/22  
 Lena Horne, Brooklyn NY, 06/30/17  
 Tab Hunter, New York NY, 07/11/31  
 Burl Ives, Hunt Township IL, 06/14/09  
 Mick Jagger, Dartford England, 07/26/43  
 Waylon Jennings, Littlefield TX, 06/15/37  
 Joan Jett, Philadelphia PA, 09/22/60  
 Billy Joel, Bronx NY, 05/09/49  
 Elton John, Middlesex England, 03/25/47  
 George Jones, Saratoga TX, 09/12/31  
 Grandpa Jones, Niagara KY, 10/20/13  
 Tom Jones, Pontypridd Wales, 06/07/40  
 Chaka Khan, Great Lakes IL, 03/23/53  
 B.B. King, Itta Bena MS, 09/16/25  
 Gladys Knight, Atlanta GA, 05/28/44  
 Patti LaBelle, Philadelphia PA, 10/04/44  
 Frankie Laine, Chicago IL, 03/30/13

---

## ITEMS OF INTEREST

---

Have a hard time finding a replacement for those \$%&@\* little lamps required in the radio shack hard drives? I have just received a supply of NEW ones, rated 100,000 hours each.

Only \$2.50 each, U.S. shipping included.

Buy a spare, even if you don't need one at the moment.

**ROY T. BECK**

**2153 CEDARHURST DRIVE**

**LOS ANGELES, CA 90027**

**(213) 664-5059**

---

### FOR SALE

MODEL 4 - 64K - 2 DRIVES

**\*\* BRAND NEW \*\***

never been used - still in original carton

**ONLY \$250**

(or best offer)

**SANDY SANDUSKY**

**11277 DONA LOLA DRIVE**

**STUDIO CITY, CA 91604**

---

### FOR SALE

I have some brand new Radio Shack Cat 26-1138 Hard Drive Controllers at \$100 each plus shipping. One year guarantee. For \$25 additional, (\$125 total) I will include RSHARD5/6 driver software for use with TRSDOS (LS-DOS) and LDOS. This controller plus an MFM bubble and power supply will allow you to build your own Model III or Model 4 hard drive package.

**ROY T. BECK**

**2153 CEDARHURST DRIVE**

**LOS ANGELES, CA 90027**

**213-664-5059**

---

### PUBLIC DOMAIN GOOD GAMES

#### FOR MODEL I/III.

**GAMEDISK#1:** amazin/bas, blazer/cmd, break-out/cmd, centipede/cmd, elect/bas, madhouse/bas, othello/cmd, poker/bas, solitr/bas, towers/cmd

**GAMEDISK#2:** cram/cmd, falien/cmd, frankadv/bas, iceworld/bas, minigolf/bas, pingpong/cmd, reactor/bas, solitr2/bas, stars/cmd, trak/cmd

**GAMEDISK#3:** ashka/cmd, asteroid/cmd, crazy8/bas, french/cmd, hexapawn, hobbit/bas, memalpha, pyramid/bas, rescue/bas, swarm/cmd

**GAMEDISK#4:** andromed/bas, blockade/bas, capture/cmd, defend/bas, empire/bas, empire/ins, jerusadv/bas, nerves/bas, poker/cmd, roadrace/bas, speedway/bas

Price per disk: \$5.00, or get all 4 disks for \$16.00

**TRSTimes - PD GAMES**

**5721 TOPANGA CANYON BLVD. #4**

**WOODLAND HILLS, CA. 91364**

---



# HUNTING FOR BURIED TREASURE

## MORE PEEKING & POKING MODEL 4

### LS-DOS 6.x.x - BASIC

By Lance Wolstrup

I didn't really intend to write another article for this issue, but it just seems right to respond to the numerous requests we've had to publish a list of the TRSDOS6/LS-DOS6 Basic PEEKs & POKEs that we have uncovered over the years. Besides, I will probably find it helpful myself to have them organized and in one place. So, without further ado, here they are, plus, as a bonus, a brandnew one, which I will explain at the end of the list.

POKE &H2F, hours 'set time  
POKE &H2E, minutes 'set time  
POKE &H2D, seconds 'set time

POKE &H74, PEEK(&H74) OR 32 'force upper case  
POKE &H74, PEEK(&H74) AND 223 'force lower case

POKE &H75, PEEK(&H75) AND 223 'enable double sided prompt in format/cmd

POKE &H75, PEEK(&H75) OR 32 'disable double sided prompt in format/cmd

POKE &H76, PEEK(&H76) OR 4:  
OUT &HEC, PEEK(&H76) '40 column display  
note: POKE &H0C0C, 255 with above poke disables return to 80 column mode after CLS, NEW, and PRINT CHR\$(28)

note: POKE &H0C0C, 251 with above poke sets 40 column mode back to normal; that is, CLS, NEW & PRINT CHR\$(28) returns to 80 column mode.

POKE &H76, PEEK(&H76) AND 251:  
OUT &HEC, PEEK(&H76) '80 column display

CLEAR &HF7FF:  
POKE &H78, 134:  
OUT &H84, 134 'enable video ram permanently

CLEAR &HFFFF:  
POKE &H78, 135:  
OUT &H84, 135 'restore normal video ram

POKE &H7C, PEEK(&H7C) OR 8:  
OUT &HEC, &H72 'fast mode - 4 mhz

POKE &H7C, PEEK(&H7C) AND 247:  
OUT &HEC, 8 'slow mode - 2mhz

POKE &H7C, PEEK(&H7C) OR 16 'disable BREAK key

POKE &H7C, PEEK(&H7C) AND 239 'enable BREAK key

IF PEEK(&H7D) = 4 THEN MODEL 4  
IF PEEK(&H7D) = 5 THEN MODEL 4P

POKE &H7F, PEEK(&H7F) OR 16 'display clock  
POKE &H7F, PEEK(&H7F) AND 239 'clock display off

POKE &H7F, PEEK(&H7F) OR 64 'cursor blinks  
POKE &H7F, PEEK(&H7F) AND 191 'cursor is solid

POKE &H78B, LSB of clock screen position  
POKE &H78C, MSB of clock screen position  
note: make sure to add &HF800 to location

POKE &HB94, PEEK(&HB94) OR 5 'scroll protect 5 lines  
POKE &HB94, PEEK(&HB94) AND 248 'no scroll protect

POKE &HB94, PEEK(&HB94) OR 8 'special chr set  
POKE &HB94, PEEK(&HB94) AND 247 'space compress

POKE &HB94, PEEK(&HB94) OR 8:  
POKE &H76, PEEK(&H76) OR 8:  
OUT &HEC, PEEK(&H76) 'alternate character set

POKE &HB94, PEEK(&HB94) OR 8:  
POKE &HB76, PEEK(&H76) AND 247:  
OUT &HEC, PEEK(&H76) 'restore to special chr set

POKE &HB94, PEEK(&HB94) AND 247:  
POKE &H76, PEEK(&H76) AND 247:  
OUT &HEC, PEEK(&H76) 'restore to space compression

POKE &HB95, LSB of screen start - normally &H00  
POKE &HB96, MSB of screen start - normally &HF8  
POKE &HC06, LSB of screen start - normally &H00  
POKE &HC07, MSB of screen start - normally &HF8  
POKE &HC24, LSB of screen start - normally &H00  
POKE &HC26, MSB of screen start - normally &HF8  
POKE &HCD4, LSB of screen start - normally &H00  
POKE &HCD5, MSB of screen start - normally &HF8  
POKE &HD75, LSB of screen start - normally &H00  
POKE &HD76, MSB of screen start - normally &HF8  
POKE &HCD7, LSB of screen length - normally &H80  
POKE &HCD8, MSB of screen length - normally &H07  
POKE &HDA8, LSB of screen length - normally &H80  
POKE &HDA9, MSB of screen length - normally &H07  
note: these pokes allow any amount of scroll protection. To scroll protect R lines, add R\*80 to &HF800 and poke to screen start; then subtract R\*80 from &H780 and poke to screen length. Need I



mention that you must make sure to  
reset the normal values when done.

POKE &HB97,0 'cursor invisible  
POKE &HB97,1 'cursor visible

POKE &HB98,cursor character 'value of cursor chr

POKE &HC8D,8:CLS 'constant reverse video  
POKE &HC8D,0:CLS 'regain normal video

And now for the bonus POKE I promised at the beginning of the article....

As you can imagine, I test a lot of TRS-80 Model 4 software - program submissions as well as the public domain variety. Most are fine, well thought-out programs, but there are some that are annoying. The kind of program that I consider to be a nuisance, is one that, for example, begins to output to the printer with absolutely no warning that it is about to. To combat this particular situation, and to possibly make your programming easier, here are a couple of brandnew POKEs:

POKE &H219,&H88:  
POKE &H21A,&H0B 'send all printer output to screen

POKE &H219,&H01:  
POKE &H21A,&H0E 'reset to normal

POKE &H211,&H01:  
POKE &H212,&H0E 'send all screen output to printer

POKE &H211,&H88:  
POKE &H212,&H0B 'reset to normal

We are playing with the Device Control Blocks here. The printer DCB stores the two-byte address of the printer driver at 219H and 21AH. The video DCB stores the two-byte address of the video driver at 211H and 212H. The address of the printer driver is 0B88H, and the address of the video driver is 0E01H. Now, if we plug the address of the video driver into the printer DCB, all printer output is sent to the screen; the same holds true for the video DCB - if we plug the address of the printer driver there - all screen output is directed to the printer. Make sure that you restore the DCB values to point to the correct driver before your program terminates. Remember, the values you POKE, remain even when you leave Basic and return to DOS.

Have fun with these  
PEEKs and POKEs. How  
about submitting your  
programs that utilizes  
some of these tricks?

See you next time....



**YES, OF COURSE !**

**WE VERY MUCH DO TRS-80 !**

## **MICRODEX CORPORATION**

### **SOFTWARE**

**CLAN-4** Mod-4 Genealogy archive & charting \$69.95  
Quick and easy editing of family data. Print elegant graphic ancestor and descendant charts on dot-matrix and laser printers. *True Mod-4 mode*, fast 100% machine language. Includes 36-page manual. **NEW!**

**XCLAN3** converts Mod-3 Clan files for Clan-4 \$29.95

**DIRECT from CHRIS** Mod-4 menu system \$29.95  
Replaces DOS-Ready prompt. Design your own menus with an easy full-screen editor. Assign any command to any single keystroke. Up to 36 menus can instantly call each other. Auto-boot, screen blanking, more.

**xT.CAD** Mod-4 Computer Drafting \$95.00  
The famous general purpose precision scaled drafting program! Surprisingly simple, yet it features CAD functions expected from expensive packages. Supports Radio Shack or MicroLabs hi-res board. Output to pen plotters. *Includes a new driver for laser printers!*

**xT.CAD BILL** of Materials for xT.CAD \$45.00  
Prints alphabetized listing of parts from xT.CAD drawings. Optional quantity, cost and total calculations.

**CASH** Bookkeeping system for Mod-4 \$45.00  
Easy to use, ideal for small business, professional or personal use. Journal entries are automatically distributed to user's accounts in a self-balancing ledger.

**FREE User Support Included With All Programs !**

### **MICRODEX BOOKSHELF**

**MOD-4 by CHRIS** for TRS/LS-DOS 6.3 \$24.95

**MOD-III by CHRIS** for LDOS 5.3 \$24.95

**MOD-III by CHRIS** for TRSDOS 1.3 \$24.95

Beautifully designed owner's manuals completely replace obsolete Tandy and LDOS documentation. Better organized, with more examples, written in plain English, these books are a *must for every TRS-80 user*.

**JCL by CHRIS** Job Control Language \$7.95

Surprise, surprise! We've got rid of the jargon and JCL turns out to be simple, easy, useful and fun. Complete tutorial with examples and command reference section.

**Z80 Tutor I** Fresh look at assembly language \$9.95

**Z80 Tutor II** Programming tools, methods \$9.95

**Z80 Tutor III** File handling, BCD math, etc. \$9.95

**Z80 Tutor X** All Z80 instructions, flags \$12.95

Common-sense assembly tutorial & reference for novice and expert alike. Over 80 routines. No kidding!

**Add S & H. Call or write MICRODEX for details**  
1212 N. Sawtelle Tucson AZ 85716 602/326-3502



# HARD DRIVES FOR SALE

Genuine Radio Shack Drive Boxes with Controller, Power Supply, and Cables.  
Formatted for TRS 6.3, installation JCL included.

Hardware write protect operational.

Documentation and new copy of MISOSYS RSHARD5/6 included.  
90 day warranty.

5 Meg \$175

10 Meg \$225

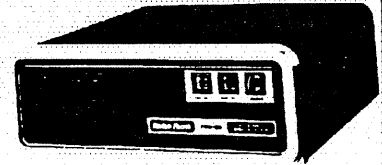
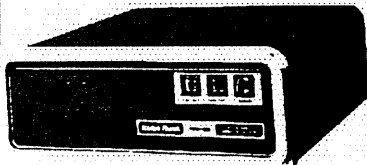
15 Meg \$275

35 Meg \$445

Shipping cost add to all prices

Roy T. Beck

2153 Cedarhurst Dr.  
Los Angeles, CA 90027  
(213) 664-5059



## MISOSYS, Inc.

### Aerocomp Hardware is now available from MISOSYS

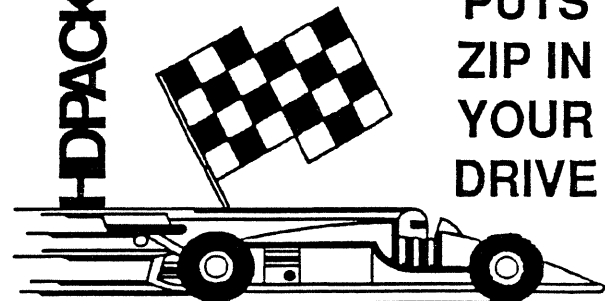
- Model I DDC Controller (DDC) \$45 + \$6S&H
- Model III/4 FDC board \$45 + \$6S&H
- Model III/4 RS232 board \$45 + \$6S&H
- Model III/4 RS232 Kit \$50 + \$6S&H
- WD1002S-SHD HDC (new) \$75 + \$5S&H
- Aerocomp 5 Meg HD \$250 + S&H
- Aerocomp 20 Meg HD \$400 + S&H
- Aerocomp 40 Meg HD \$500 + S&H
- MM CP/M 2.2 HD drivers \$29.95 + \$3S&H

### The MISOSYS Quarterly subscriptions

- Keep up to date with the latest information on MISOSYS products, programs, patches, and articles in a professional magazine format. A subscription to *TMQ* will provide you with information, news, and announcements concerning our entire product line and related machine environments. As a special for new subscribers, we'll provide you with five issues and start you off with issue V Liv, and send you three past issues at no charge. That's eight issues for the price of four! Subscriptions are: \$25 US; \$30 Canada; \$35 Europe; \$40 Australia

**MISOSYS, Inc.** Sterling, VA 20167-0239  
P. O. Box 239 703-450-4181  
orders: 800-MISOSYS

HDPACK



PUTS  
ZIP IN  
YOUR  
DRIVE

- When your hard drive files become fragmented with excessive directory extents, access speed degrades. Your program will finish in less than the optimum time. Now with our HDPACK utility, you can restore that ZIP to your computer. HDPACK will automatically, and intelligently, re-pack the fragmented files on your drive which will improve the performance of file access time.

- HDPACK provides a visual display of its de-fragging operation, which in minutes can restore a ten-megabyte directory of files to a minimum number of extents. HDPACK can even work on floppy diskettes, too.

- HDPACK, cat. no. M-33-400, is available for Model 4 DOS 6 only, and is priced at \$39.95 + \$3S&H.

# **\*\*\* DR. PATCH \*\*\***

**BRAND-NEW UTILITY FOR TRS-80 MODEL 4 AND LS-DOS 6.3.1**

**A 'MUST HAVE' FOR ALL LS-DOS 6.3.1 OWNERS.**

**DR. PATCH MODIFIES LS-DOS 6.3.1 TO DO THINGS THAT WERE NEVER BEFORE POSSIBLE.  
COMPLETELY SELF-CONTAINED - MENU-DRIVEN FOR MAXIMUM USER CONVENIENCE.  
FAST & SAFE - EACH MODIFICATION IS EASILY REVERSED TO NORMAL DOS OPERATION.**

DISABLE PASSWORD CHECK IN FORMAT/CMD  
FORMAT DOUBLE-SIDED AS DEFAULT  
FORMAT 80 TRACKS AS DEFAULT  
DISABLE VERIFY AFTER FORMAT  
CHANGE 'DIR' TO 'D'  
CHANGE 'CAT' TO 'C'  
VIEW DIR/CAT WITH (I) PARAMETER AS DEFAULT  
VIEW DIR/CAT WITH (S,I) PARAMETERS AS DEFAULT  
CHANGE 'REMOVE' TO 'DEL'  
CHANGE 'RENAME' TO 'REN'  
CHANGE 'MEMORY' TO 'MEM'  
CHANGE 'DEVICE' TO 'DEV'  
DISABLE THE BOOT 'DATE' PROMPT  
DISABLE THE BOOT 'TIME' PROMPT  
DISABLE FILE PASSWORD PROTECTION  
ENABLE EXTENDED ERROR MESSAGES

DISABLE PASSWORD CHECK IN BACKUP/CMD  
BACKUP WITH (I) PARAMETER AS DEFAULT  
BACKUP WITH VERIFY DISABLED  
DISABLE BACKUP 'LIMIT' PROTECTION  
DISABLE PASSWORD CHECK IN PURGE  
PURGE WITH (I) PARAMETER AS DEFAULT  
PURGE WITH (S,I) PARAMETERS AS DEFAULT  
PURGE WITH (Q=N) PARAMETER AS DEFAULT  
IMPLEMENT THE DOS 'KILL' COMMAND  
CHANGE DOS PROMPT TO CUSTOM PROMPT  
TURN 'AUTO BREAK DISABLE' OFF  
TURN 'SYSGEN' MESSAGE OFF  
BOOT WITH NON-BLINKING CURSOR  
BOOT WITH CUSTOM CURSOR  
BOOT WITH CLOCK ON  
BOOT WITH FAST KEY-REPEAT

**DR. PATCH IS THE ONLY PROGRAM OF ITS TYPE EVER WRITTEN FOR THE TRS-80  
MODEL 4 AND LS-DOS 6.3.1. IT IS DISTRIBUTED EXCLUSIVELY BY TRSTIMES MAGAZINE  
ON A STANDARD LS-DOS 6.3.1 DATA DISKETTE, ALONG WITH WRITTEN DOCUMENTATION.**

**\*\*\* DR. PATCH \*\*\* \$14.95**

SHIPPING & HANDLING: U.S & CANADA - NONE  
ELSEWHERE - ADD \$4.00  
(U.S CURRENCY ONLY, PLEASE)

**TRSTimes magazine - dept. DP  
5721 Topanga Canyon Blvd. #4  
Woodland Hills, CA 91367**

**DON'T LET YOUR LS-DOS 6.3.1 BE WITHOUT IT!**



# What the Heck is a 26-1138?

By Roy T. Beck

Fair Warning, this article is intended to help me sell something, so don't get huffy when you detect some advertising in it. OK?

The 26-1138 is another of Radio Shacks corporate meanderings. It is a neat little box 3 inches high, 7.5 inches wide, and 14 inches, front to rear. It contains a power supply and the late model hard disk controller (HDC) which is in most 12 and 15 Meg drives and all the 35 and 70 Meg drives sold by Tandy for use with our beloved TRS. But what good is an HDC all by itself? Read on, my friends.

Radio Shack originally offered this unit to be used with their 10 Meg drive, Cat No 25-1007 or their 20 Meg drive, Cat No 25-1041. The HDC and the two drives are all painted the same light tan as the Model 4 and 4P.

The 26-1138 HDC plus one or two of the hard drives makes a neat package alongside your Model III, 4 or 4P. They also function with the Tandy 1000 and with an accessory kit, will even work with your Model I. That's right, I said Model I! The accessory kit is the 26-1132, which contains a small cable adapter and LDOS V 5.1.3 for the Model I.

Offhand, I don't remember what brand of 10 Meg drive was in the 25-1007, but the 25-1041 contained an ST-225, which was a respectable, widely used drive until the IBM world outgrew it. After all, what self-respecting IBM type would admit, today, that he has only 20 megs?

Do you really have to have or find a 25-1007 or -1041 to operate with a 26-1138 HDC? Not necessarily. Do I have any such units for sale? No, I don't have any of the drive units. But consider: the 25-1007 and -1041 units are nothing but a neat little box with a power supply and a half-height bubble mounted inside. A power cord and two receptacles on the back complete the picture. The HDC in turn has two ribbon cables dangling outback which plug into the two receptacles on the drive box. Those of you with a little hardware expertise and a spare MFM 20 meg bubble and power supply can easily see how to connect this to a Model III or 4 if you only had the HDC unit. Well, I have a few 26-1138's available, \$100 plus shipping. And these are BRAND NEW units, already to go, with a one year warranty. For \$25 more, I will include RSHARD and an instruction manual written by yours truly. Any takers? Actually, any MFM drive from 5 to 70 megs, half height or full height will work, you just need a power supply and a box to hold the bubble and the power supply.

If any of you hardware hackers happen to have a spare ST-225 or any other good MFM drive available, consider

making up your own hard drive system from scratch, as it were.

Are there any gotchas? Yes, a few minor differences. These 26-1138 units contain the identical controllers originally incorporated in the 35 meg, etc Radio Shack units. They do not have a write protect or activity light on the front, but software write protect is available, and I never use the hardware WP switch on my other units anyway; I don't use them because they protect all partitions simultaneously, which is a pain. These units have a power on lamp on the front, but no power switch. Plug it into your master power strip. These units are set up for a maximum of two bubbles, which is the way RSHARD is written.

In conclusion, I don't know whether you will appreciate this article or not, but if you have read this far, my purpose has been served. I got you to read my advert! Come on, you hardware hackers, let's get with it and create some home brew hard drive setups.

## **TIRED OF SLOPPY DISK LABELS? TIRED OF NOT KNOWING WHAT'S ON YOUR DISK? YOU NEED 'DL'**

**'DL' will automatically read your TRSDOS6/LDOS compatible disk and then print a neat label, listing the visible files (maximum 16). Use the 'change' feature to select the filenames to print. You may even change the diskname and diskdate.**

**'DL' is written in 100% Z-80 machine code for efficiency and speed.**

**Don't be without it - order your copy today.**

**'DL' is available for TRS-80 Model 4/4P/4D using TRSDOS 6.2/LS-DOS 6.3.0 & 6.3.1. with either an Epson compatible or DMP series printer.**

**'DL' for Model 4 only \$9.95**

**TRSTimes magazine - Dept. 'DL'  
5721 Topanga Canyon Blvd. #4  
Woodland Hills, CA 91367**

# TRSTimes Special PD Collection for Model 4

---

## M4PD01

**ARCH4/CMD**, **ARCH4/DOC** - archive program for model 4, **DCTDSP/CMD** - dct display utility, **DGRAPH/BAS** - demo of direct video ram access, **DISASM/CMD** - mod 4 disassembler, **DISASM/DOC** - docs for disasm/cmd, **ERROR/ASM** - asm source code for error/cmd, **ERROR/CMD** - display complete error messages, **FKEY/ASM** - asm source code for fkey/cmd, **FKEY/CMD** - reprogram the function keys, **FKEY/DOC** - docs for fkey/cmd, **LABEL4/BAS** - disk label-maker, **MOD4BASC/BAS** - mod I/III Basic programs to mod 4, **MOD4BASC/DOC** - docs for mod4basc/bas, **SHELL/CMD** - the famous dos-shell for model 4, **SHELL/DOC** - docs for shell/cmd, **TEXEDIT4/BAS** - model 4 text editor, **VDCTL/ASM** - source code for vdctl/cmd, **VDCTL/CMD** - set, reset, point, peek & poke the video display, **VDCTL/DOC** - docs for vdctl/cmd, **VDCTL/OBJ** - obj code for vdctl/cmd

---

## M4PD02

**ANALYZER/CMD** - model 4 disk analyzer, **ANALYZER/TXT** - docs for analyzer/cmd, **BASL4/CMD** - lists a Basic program to printer no matter what format the program is saved in, **BASL4/DOC** - docs for basl4/cmd, **DISKLIST/CMD** - mod 4 disk cataloger, **DISKLIST/DOC** - docs for disklist/cmd, **DISKTOOL/CMD** - collection of mod 4 disk utilities, **DISKTOOL/TXT** - docs for disktool/cmd, **MORE/CMD** - model 4 file lister, **MORE/CCC** - c source code to more/cmd, **PMGE/CMD** - poor man's graphics editor, **PMGE/DOC** - docs for pmge/cmd

---

## M4PD03

**CODEVIEW/CMD** - great utility by Mel Patrick, **CODEVIEW/TXT** - docs for codeview/cmd, **HDEBUG/CMD** - powerful model 4 debugger, **HDEBUG/CNF** - config file for hdebug/cmd, **HDEBUG/DOC** - docs for hdebug/cmd, **NAMEIT/ASM** - source code for nameit/cmd, **NAMEIT/CMD** - disk renamer for lds & ls-dos, **NAMEIT/DOC** - docs for nameit/cmd, **PMAT/CMD** - file listing utility, **PMAT/DOC** - docs for pmat/cmd, **PMAT/PAS** - source code for pmat/cmd, **PRINTER/SET** - config file for pmat/cmd, **PURGEDUP/CMD** - purge duplicates, **PURGEDUP/TXT** - docs for purgedup/cmd, **RAMTEST4/CMD** - test model 4 ram, **RAMTEST4/DOC** - docs for ramtest4/cmd, **SCREEN/DOC** - docs for screen/fit, **SCREEN/FLT** - screen blanking filter, **SCREEN/JCL** - install for screen/fit, **SDIR4/CMD** - silly directory from Luis Garcia-Barrio, **SDIR4/DOC** - docs for sdირ4/cmd, **SRCASM/CMD** - convert edtasmm files to ascii, **SRCASM/DOC** - docs for scrasm/cmd

---

## M4PD04

**BIGPRINT/BAS** - banner program for dmp-100, **IMAGES** - datafile for bigprint/bas, **BASRUN/CMD** - needed for the profile system enhancement, **EFEC/CMD** - profile system enhancement, **EFECR/CMD** - profile system enhancement, **EFECR/TXT** - docs for profile syst enhancements, **EFER/CMD** - profile system enhancement, **EFES/CMD** - profile system enhancement, **OUTLINER/CMD** - outlining program for model 4, **OUTLINER/DOC** - docs for outliner/cmd, **PF/CMD** - file printing utility, **PF/DOC** - docs for pf/cmd, **SPLITTER/CMD** - split file into smaller files, **SPLITTER/DOC** - docs for splitter/cmd

---

## M4PD05

**DISKSYS/CMD** - optimizes system files, **DISKSYS/DOC** - docs for disksys/cmd, **DESIGNE/CMD** - low res drawing program, **DESIGNE/TXT** - docs for designe/cmd, **EXAMPLE/ASM** - example file for designe/cmd, **EXAMPLE/BAS** - example file for designe/cmd, **EXAMPLE1/ASM** - example file for designe/cmd, **SF2/CMD** - file listing utility, **SF2/DOC** - docs for sf2/cmd, **SUBMERGE/BAS** - module for submerge/cmd, **SUBMERGE/CMD** - access AL routines from Basic, **SUBMERGE/TXT** - docs for submerge/cmd, **SYSDRV/CMD** - setup new system drive, **SYSDRV/SRC** - source code for sysdrv/cmd, **TRSCAT/CMD** - model 4 disk cataloger, **TRSCAT/TXT** - docs for trscat/cmd, **VIVACE/CMD** - WittSoft's Basic compiler, **VIVACE/DOC** - short note re Lou Witt's new address, **VIVACE/FIX** - patch for vivace/cmd, **VIVACE/LIB** - module for vivace/cmd, **VIVACE/SCR** - docs for vivace/cmd, **VIVPAT/TXT** - docs for vivace/fix

---

## M4PD06

**CVT324/CMD** - convert model 3 basic to model 4, **CVT324/DOC** - docs for cvt324/cmd, **DISASM2/BAS** - disassembler written in Basic, **DMU620/CMD** - directory maintenance for model 4, **DMU620/DOC** - docs for dmu620/cmd, **QUICKOP/CMD** - hard disk optimizer, **QUICKOP/CTL** - data for quickop/cmd, **QUICKOP/TXT** - docs for quickop/cmd, **RECONFIG/CMD** - reconfiguration utility, **RECONFIG/DOC** - docs for reconfig/cmd, **ROM/CMD** - store model 4 rom in upper bank, **ROM/DOC** - docs for rom/cmd, **ROM/SRC** - source code for rom/cmd, **TIME/BAS** -



fancy time display, **TRAKZAP4/CMD** - model 4 disk zapper, **TRS31/CMD** - model 3 emulator for model 4, **TRS31/DOC** - docs for trs31/cmd, **UNFILTER/CMD** - remove filter utility, **UNFILTER/DOC** - docs for unfilter/cmd, **UNSET/CMD** - remove module in high memory, **UNSET/DOC** - docs for unset/cmd

---

#### M4PD07

**APART/BAS** - break up text file into small blocks, **APART/DOC** - docs for APART/BAS, **DIRHLP/BAS** - directory help utility, **DNAME/CMD** - change diskname, **DNAME/DOC** - documentation for dname/cmd, **GRANINFO/CMD** - display file granule info, **GRANINFO/SRC** - source code for graninfo/cmd, **HASH/CMD** - display file hash code, **HASH/DOC** - docs for hash/cmd, **HDUTIL/CMD** - hard driver park utility, **MAKECMD/CMD** - create your own dos commands, **MAKECMD/DOC** - docs for makecmd/cmd, **MAILER/BAS** - mailing list program, **MAILER/LPT** - print module for mailer/bas, **MINIOP2/CMD** - shell for model 4, **MINIOP2/SRC** - source code for miniop2/cmd, **NUMBERS/CMD** - number conversion utility, **OFFSIDE/CCC** - C source code for offside/cmd, **OFFSIDE/CMD** - epson printer utility - prints sideways, **OFFSIDE/DOC** - docs for offside/cmd, **QUIKDOS/CMD** - dos shell for model 4, **QUIKDOS/DOC** - docs for quikdos/cmd, **UPCASE/CMD** - convert file to all uppercase, **UPCASE/SRC** - source code for upcase/cmd

---

#### M4PD08

**ASSIGN/CMD** - configure your drives, **ASSIGN/DOC** - docs for assign/cmd, **DATABASE/BAS** - interesting database manage, **DIR2FILE/CMD** - read directory into a file, **DIR2FILE/TXT** - docs for dir2file/cmd, **DISKLEEN/CMD** - model 4 disk cleaner utility, **EXEDITOR/BAS** - text editor ala EX on unix, **EXEDITOR/DOC** - docs for exeditor/bas, **FILES/CMD** - add descriptions to file names, **FILES/DOC** - docs for files/cmd, **HDEMODO/CMD** - upgrade 1 edasm to model 4, **HDEMODO/DOC** - docs for hdemod/cmd, **HELLO/CMD** - model 4 file loader, **HELLO/DOC** - docs for hello/cmd, **INSTALLS/CMD** - memory module relocation utility, **INSTALLS/ASM** - source code for installs/cmd, **INSTALL/TXT** - docs for installs/cmd, **INSTALL/ASM** - module for installs/cmd, **INSTALL/REL** - module for installs/cmd, **LISTFILE/CMD** - file lister utility, **TRS2PC/BAS** - TRS-80 to PC conversion program, **VIS/CMD** - strip unprintable characters from file, **VIS/DOC** - docs for vis/doc, **WC/CCC** - c source code for wc/cmd, **WC/CMD** - word counter utility, **WC/DOC** - docs for wc/cmd

---

#### M4PD09

**DRIVETIM/CMD** - Mark Goodwin's disk drive timer, **DRIVETIM/DOC** - docs for drivetim/cmd, **EASYDOS/CMD** - dos shell for trsdos 6.2, **EASYDOS/DOC** - docs for easydos/cmd, **EASYDOS4/PAT** - patch to update easydos/cmd to ls-dos 6.3.x, **EZDOS63/CMD** - nice dos shell for ls-dos 6.3.x, **EDITDRAW/CMD** - drawing program for model 4, **EDITDRAW/DOC** - docs for editdraw, **QUICK/DOC** - docs for quikdmp/cmd & quikeps/cmd, **QUIKDMP/CMD** - setup for dmp printers, **QUIKEPS/CMD** - setup for epson printers, **SIDEWAYS/BAS** - sideways file printer for multiplan & visicalc, **SIDEWAY2/BAS** - part 2 of sideways/bas, **SORTTEXT/CMD** - list sorting program

---

#### M4PD10

**CROSSREF/CMD** - a Basic x-ref utility, **CROSSREF/TXT** - docs for crossref/cmd, **CTRAINER/BAS** - c interpreter written in Basic, **DS4/CMD** - double-sided drive initializer, **GREP/CMD** - unix-like text searcher, **GREP/TXT** - docs for grep/cmd, **HELPMATE/BAS** - source code for helpmate/cmd, **HELPMATE/CMD** - editor for /hlp files, **MAILABEL/BAS** - mailing label program, **MAILABLE/DOC** - docs for mailabel/bas, **MOD4NOTE/CMD** - leave notes on the screen, **MULTI/BAS** - model 4 database, **MULTI/DOC** - docs for multi/bas, **PAGE/CMD** - lists ascii files one page at a time, **PROGMCMD/BAS** - program commander file utility, **PROGMCMD/DOC** - docs for progmcmd/bas, **SCRUBBER/CMD** - drive cleaning program, **XTEND/CMD** - add new capabilities to Rom Basic, **XTEND/SRC** - source code for xtend/cmd, **XTEND/TXT** - docs for xtend/cmd

## TRSTimes Special PD Collection for Model 4

1 to 10 volumes    \$3.00 per volume + \$2.00 postage

TRSTimes magazine - dept. PD4  
5721 Topanga Canyon Blvd., Suite 4  
Woodland Hills, CA 91367



# **GRAPHICS-90**

## **NOW for Model I or III**

The definitive graphics program that Radio Shack **SHOULD** have released is now, through the courtesy of author Larry Payne and the hard work of Gary Shanafelt, available from TRSTimes. Our agreement is to sell this wonderful package at **COST**. Neither Larry Payne, Gary Shanafelt nor TRSTimes will make a dime on this venture. The price charged will cover the cost of reproducing the manual, disks, and mailing only. The only one profiting is **YOU!!**

### **GRAPHICS-90**

**Model III \$ 9.75**

**Model I \$ 11.50**

**Shipping: Europe: add \$4.50 air mail or \$2.75 surface mail**

**Asia, Austrailia & New Zealand: add \$5.25 air mail or \$3.50 surface mail**

**Please specify Model I or III and DOS preference**

### **TRSTimes magazine - dept. GR90**

**5721 Topanga Canyon Blvd. #4**

**Woodland Hills, CA 91367**

#### **RECREATIONAL & EDUCATIONAL COMPUTING**



REC is the only publication devoted to the playful interaction of computers and 'mathemagic' - from digital delights to strange attractors, from special number classes to computer graphics and fractals. Edited and published by computer columnist and math professor Dr. Michael W. Ecker, REC features programs, challenges, puzzles, program teasers, art, editorial, humor, and much more, all laser printed. REC supports many computer brands as it has done since inception Jan. 1986. Back issues are available.

To subscribe for one year of 8 issues, send \$27 US or \$36 outside North America to REC, Att: Dr. M. Ecker, 909 Violet Terrace, Clarks Summit, PA 18411, USA or send \$10 (\$13 non-US) for 3 sample issues, creditable.

#### **TRSTimes on DISK #10**

Issue #10 of TRSTimes on DISK is now available, featuring the programs from the Jul/Aug, Sep/Oct & Nov/Dec 1992 issues:

**TRSTimes on DISK is reasonably priced:**

**U.S. & Canada: \$5.00 (U.S.)**

**Other countries: \$7.00 (U.S.)**

**Send check or money order to:**

**TRSTimes on DISK  
5721 Topanga Canyon Blvd. #4  
Woodland Hills, CA 91367**

**TRSTimes on DISK #1, 2, 3, 4, 5, 6, 7, 8 & 9  
are still available at the above prices**