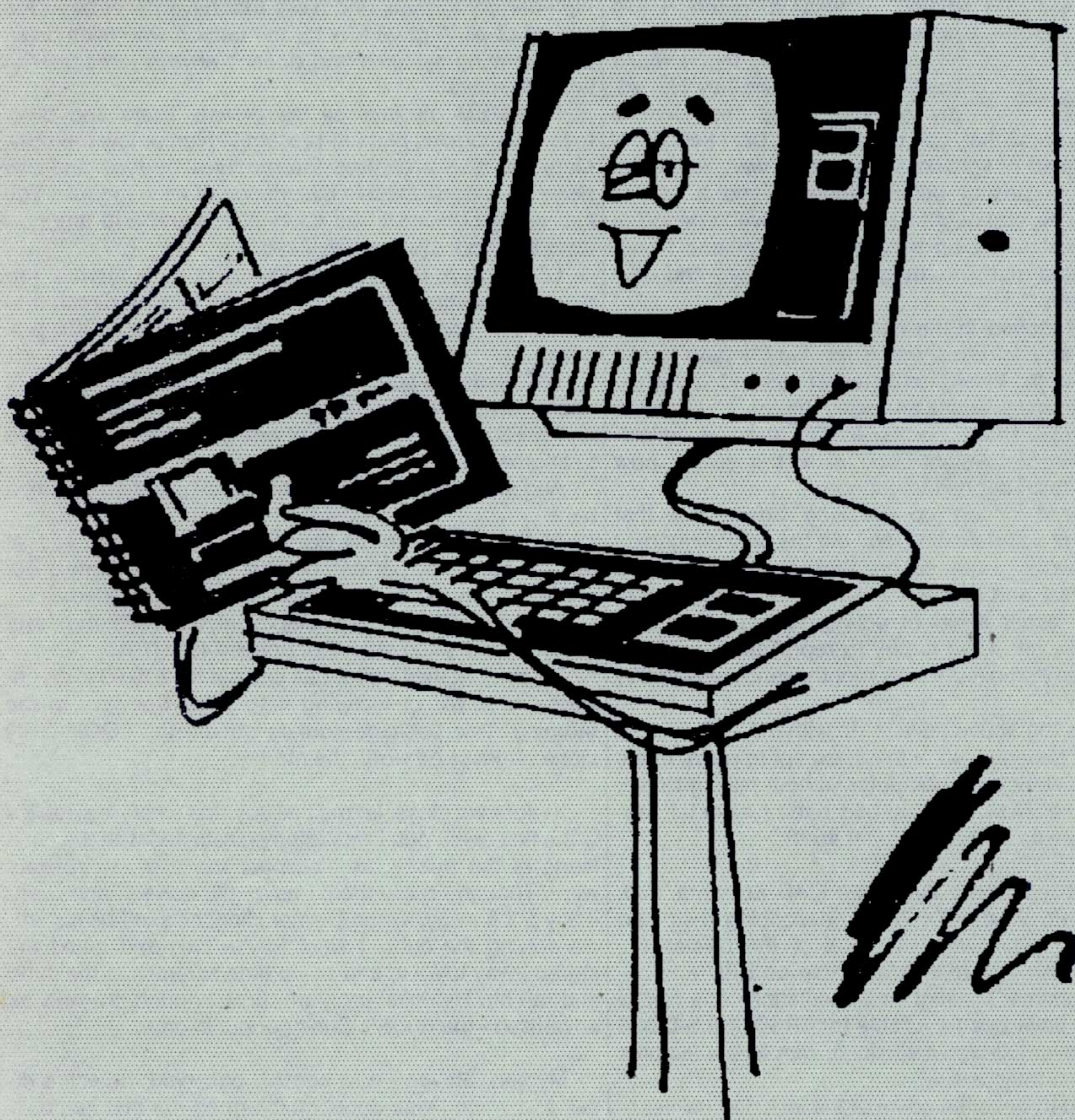


TRSTimes

Volume 6. No. 5. - Sep/Oct 1993 - \$4.00



TRSTimes is still supporting the Model I

LITTLE ORPHAN EIGHTY



As of this writing, I haven't smoked a cigarette for 18 months. I used to puff away on better than two packs per day, so it was just natural for my friends and family not too have too much faith when I told them I was quitting. But I did it. No, I didn't go to 'Shick' or wear a 'patch', I just never lit the next one.

Does that mean that I didn't have the urge to smoke? Not so, my friends. I had the nicotine

craving more times than you can imagine, but I kept fighting it off. Instead of lighting a cigarette, I went to the refrigerator for a peanut butter sandwich and a can of Classic Coke. As you might well imagine, over a period of 18 months that totals up to quite a few peanut butter sandwiches and many, many gallons of Coke - and a 30 lbs. weight gain.

One day I looked in the mirror and I saw that I was well on my way to becoming fat. Hey, I have never been fat in my life - I weighed between 185 and 190 when I played ball in my twenties and early thirties, and grew to 200 when I retired from sports. At 6'3" that is a considered a reasonable weight. But, alas, the scales now tipped at 230 lbs. That had to change.

I considered taking up smoking again, but that seemed kind of self-defeating, so I did the only thing that I knew would work - I quit eating. Well, almost! I stopped eating breakfast and lunch, and limited myself to dinner consisting of either salad or cottage cheese and fruit. Boy, did that work. I am happy to report that as of this writing I haven't smoked for 18 months, and I weigh 201 lbs.

What does the above have to do with the TRS-80, and computers in general? Not a thing, except to prove that it is possible to work on a computer without having to eat, drink or smoke. Now the only bad habit I have left, according to my wife, is the computer itself!

I have just received a flyer from MAD Software, P.O. Box 331323, Ft. Worth, TX 76163, describing their new Utility Disk #1. They write "Several of the utilities are clones of popular commands available on BSD and derived *NIX systems. Minimal implementations of a few of these utilities have been available for TRSDOS/LS-DOS6 before, but never full-blown versions, with all the features that make them truly useful."

They go on to describe some of the included utilities.

MORE displays files a screen at a time, can search through files skipping undesired output. Able to switch back and forth between numerous files.

LOOK will examine any diskette and determine what format it is recorded in. Have you ever inserted a disk and

tried to do a DIR of it, and the operating system locked-up because the disk was in an alien format? By using **LOOK**, you can avoid that. **LOOK** is able to recognize over two dozen diskette formats, and you can easily tell **LOOK** about any that we missed. If the disk is in TRSDOS/LS-DOS6 or LDOS5 format, **LOOK** also displays tables showing exactly where each file resides on the disk. **LOOK** can even identify disks that are recorded in high-density formats and disks that have never been formatted at all. A great tool for the person who uses different operating systems and doesn't always remember to label diskettes.

MAPMEM displays all the drivers, filters and other memory modules that are present in high or low memory. **Mapmem** also organizes the modules by type and displays an understandable description of each module. You can easily customize the descriptions and add any new ones that your system may need.

LS displays the files in directories in numerous formats and sorting orders. The output can be directed to the screen, printer, a file, or even through the **MORE** program.

WC reads the specified files and counts the number of characters, words, and lines that are in the files.

PIPES allows the output of one program to be passed to another as input.

And other utilities we don't have room to mention here.

All the utilities on the disk that utilize files will accept full wild-cards, output can be sent to the printer or a file, and the ability to specify default options is provided.

The Utility Disk #1 is available July 15, 1993 and costs \$25.00. - add \$4.00 shipping. Specify 4G6UT01.

Ye old (but thin) editor has just put his \$29.00 in an envelope and will be sending it to MAD Software in the morning. This package sounds good - the **LOOK** utility alone ought to be worth the money.

Before we close, let me just say that when the Rubik's Cube first came out, I was utterly fascinated with the challenge of the puzzle, as I suspect many of our readers were. Though I really tried, I never did solve it, and eventually put it away. When Delmer Hinrichs send us his article solving the cube, I went to the attic and found my stored-away toy. I spend some time scrambling the cube and ran Mr. Hinrichs' program. Sure enough, it works as advertised - the Rubik's Cube got put in order.

My son, Steven - now age 13, observed the entire ritual, and when I was done, he informed me that he could do that without a computer program. "Oh yeah, you little #@%!", I thought, "I'll show you!" So I scrambled the cube with my very best effort and handed it to him. "Take that, braggart!"

continued on page 4

TRSTimes magazine

Volume 6. No. 5 - Sep/Oct 1993

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 1993 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.

1993 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

FONTS GALORE..... 5
Gary W. Shanafelt

BORN AGAIN..... 8
Lance Wolstrup

BEAT THE GAME 11
Daniel Myers

**A BRIEF HISTORY OF HARD DRIVES
AND THEIR TROUBLES**..... 14
Roy T. Beck

RUBIK'S CUBE SOLVED..... 17
Delmer D. Hinrichs

HINTS & TIPS 24
Tipps, Spratt, Bates

HUNTING FOR BURIED TREASURE..... 26
Lance Wolstrup

ITEMS OF INTEREST..... 30
User Group Information



continued from page 2

Steven took the cube and went to his room. About 30 minutes later he came back down - with the cube solved!!!

"Great Scott!, how did he do that?" I was perplexed, but also quite pleased that my son had such puzzle prowess.

At first he wouldn't tell me, but eventually I wormed the secret out of him. He had very meticulously removed each one of the stickers, and then pasted them back on in color order to make it appear that the cube was solved.

Just goes to show you that there are more than one way to solve a puzzle!

Gary Shanafelt is once again helping the TRS-80 stay a modern computer. Now the MS-DOS machines no longer have a monopoly on TRUE-TYPE fonts. With a little bit of work, we can convert them for use with the TRS-80.

I had an absolutely marvelous time exploring the Model I emulator program running on the PC.. I plan to write some programs for it in future issues, and I hope that readers will submit some new items for it also. It is such a kick to run a Model I on my PC!

Danny Myers is at it again. This time he presents us with the solution to ZORK II. Now, I know that many of you were not aware that ZORK was such a big game - it has three parts, each installment a complete game within itself. I never got past the 'echo' room in part I, but I certainly plan to spend an evening (or two) to see where I went wrong, and then tackle part II. Boy, reading Danny's walk-through gives me renewed respect for the Infocom authors and programmers. This is one large and fun game!

If it wasn't for Roy Beck's articles, TRSTimes would be much thinner, and much less educational. Roy is the hard drive 'guru' of the Greater Los Angeles user groups and a prolific writer. He always has a three-four page article ready for somebody. 'Rambling', he calls it. If that's the case, 'Ramble on', I enjoy the information and the way you present it.

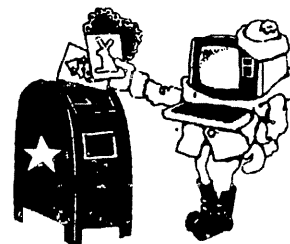
The Hints & Tips section for this issue features sorting programs from Frank Tipps, a fun program to find the date of Easter Sunday by Chris Spratt, and a tutorial on installing double-sided drives in a 4P by Kelly Bates.

Finally, we bring you another installment of 'Hunting for Buried Treasure'. This time we investigate and find that elusive, missing 256th byte from the random access FIELD statement. In the process, we create a patch program.

Thanks to all the contributors - we really do appreciate your hard work.

And now.....Welcome to TRSTimes 6.5

THE MAIL ROOM



USER GROUP KUDOS

My computing has extended to a PC which I built from parts, a 386 DX/40 with co-processor. It stands side by side with the Model 4, but has its main use in word processing using Wordperfect 5.1 which I am getting on with quite well.

The Model 4 is still used quite a bit and I have made very many friends through it. As I have said in the past, I am a much better hacker of hardware than of software which has created the friendships and I often get calls to help with TRS-80 problems.

One of those I have been helping recently was very impressed with your editorial (Little Orphan Eighty) in Volume 6, No. 4 and commented how very true it had been for him, having made several friends through his Model 4, usually because he was not sure of something and a phone call sorted it out. He lives some 175 miles from me and I have made one visit to fit double-sided drives when I stayed overnight and a second visit with my wife when we did the return trip in the same day having removed a piece of paper from one of the drives. It got attached to a disk and was inserted, never to come out without minor surgery.

Tom Ridge
Surrey, England

Congratulations on your 386. When you read the article in this issue about the Model I emulator, you will be able to to put away WordPerfect and replace it with a truly useful word processor - Model I SCRiPSIT!

Seriously, I strongly recommend attending user group meetings - I guess my editorial conveyed that. I have attended PC meetings, CoCo meetings, Commodore 64 meetings, an Apple meeting, and of couse, TRS-80 meetings. To my knowledge, most CoCo and C-64 groups are now defunct, and I won't even comment on the Apple gathering. The PC-group meetings that I have attended have, for the most part, been intolerable. They have been crowded, full of 'appliance users' who thought they were programming when they created a macro in Lotus.... Well, maybe they were! It just isn't the kind of environment that I enjoy being in, so I now confine my meetings to the ones I talked about in the aforementioned column - and they are all more or less TRS-80 oriented.

Ed.

FONTS GALORE!

by Gary W. Shanafelt

If you've been following the development of Windows on the PC clones, you've probably noticed the virtual explosion of typefaces available in TrueType format. And maybe you're envious of your friends with 486 machines who, as a result, seem to have hundreds of new lettersets available for their documents -- *many free in the public domain* -- while on your TRS-80 you're stuck with the same old Courier or boldface that you've had for the last ten years, right?

Wrong. In the last few months, it's become possible to get a lot of those MSDOS fonts up and running on TRS-80s. How? First, while there are no TRS-80 programs which support the TrueType font format, it's fairly easy to convert TrueType fonts to LaserJet bitmapped format, which is supported by a number of TRS-80 word processors. And even if you have only a dot matrix printer, a host of new public domain fonts are now available in Dotwriter format.

New LaserJet/DeskJet Fonts

Converting TrueType fonts is made possible by a clever MSDOS program written by Alexander Walter of Middletown, New Jersey, TT2BMAP.EXE. Since most TrueType fonts on the LaserJet download the selected character information in bitmapped PCL (Printer Control Language) format to the printer at print time, Walter's program simply intercepts all the character data and sends them instead to a disk file. From there, TT2BMAP creates a new bitmapped version of the font, in the size(s) you originally selected. Of course, to use the program, you need access to a MSDOS computer running Windows as well as a program like HyperCross or David Miller's new MS Utilities to transfer the newly-created fonts from MSDOS to TRS-80 disks. And you may have to use a MSDOS-based conversion program if you want to make DeskJet versions of the LaserJet fonts.

Once you have the fonts in TRS-80 format, they still have to be integrated with your favorite word processor. If you have the Computer News 80 LaserJet or DeskJet Utility Packs for Allwrite, you use the included utilities to create a width table which you then incorporate into the Allwrite printer driver. If you have SuperScripsit or ScripsitPro, David Goben at Computer News 80 can for a fee customize a driver for you. A number of converted fonts, ARChived together in both LaserJet and DeskJet format, are available in the Model 4 section of the TRS-ureTrove BBS sponsored by this publication (phone 213

664-5056, 8/N/1). All you have to do is call up the BBS and download them. A few samples from my DeskJet Plus printer:

Ondine 24 pt.

GALLIA 20 PT.

EDDA CAPS 24 PT.

KONAN 30 PT.

New Dotwriter Fonts

The new Dotwriter fonts also originate in the MSDOS world. In this case, the source is the numerous shareware fonts or lettersets that proliferated on various BBSes for the original PrintShop and PrintMaster programs. PrintShop and its imitators allowed people with MSDOS computers to do what Dotwriter on the TRS-80 had been doing for years. Dotwriter was the original TRS-80 fancy font program, allowing you to print hundreds of different lettersets on dot matrix printers, or to create your own with special font editing utilities. When Prosoft went out of business, though, its library of Dotwriter fonts disappeared off the market with it. The PrintShop fonts were not compatible with Dotwriter.

The result was that Dotwriter users were limited to what Dotwriter fonts they had purchased before Prosoft closed down its operations, with one exception. The exception was PrintShop/PrintMaster *icon* files. Several years ago, Paul F. Barnett wrote a Model III program called P2DOT which would take the icon files from either of these formats and convert them to 7- or 8-pin Dotwriter fonts. Longtime readers of this publication may remember that P2DOT was reviewed on page 28 of the July 1988 issue. Barnett also wrote a utility to convert between 7- and 8-pin Dotwriter formats. (Both these programs are available on the TRS-ureTrove BBS in the Model III section: P2DOT/ARC and DOTCON/ARC). But P2DOT wouldn't work with the -letter (as opposed to the *icon*) fonts for PrintShop or PrintMaster, for the letter fonts had a different format than the icon fonts.

That any of the letter fonts are now available in Dotwriter format is the result of the labors of Kelly Bates of Oklahoma. He has manually converted a great number of shareware PrintShop/Printmaster letter fonts to Dotwriter

format. These can also be found in the Model III section of the BBS, in five ARCHived files. PARK8/ARC and NEWS8/ARC contain various sizes from 8 to 24 points of two typefaces much like Helvetica and Bodoni respectively. The other files are DOTAG8/ARC, DOTH08/ARC, and DOTPT8/ARC. Each of these contains 6-7 individual fonts in alphabetical order, with DOTAG8 holding fonts with names ranging from A to G, DOTH08 font names from H to O, etc. You should note that these are all 8-pin versions. If you want 7-pin versions, you'll need to run them through Paul Barnett's conversion program (though results can't be guaranteed with the larger fonts because of memory limitations). Some samples:

News10 News18 News24

Park10 Park10 Italic Park24

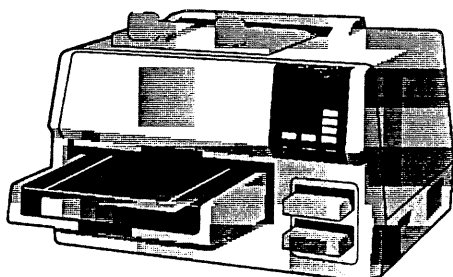
Platform TWIG

Thoreau Gatsby

MARIN Blippo

Joyce Balloon

Who can use these fonts? Anyone with Dotwriter is obviously a prime candidate. But if you have a hi-res board and are running either MDRAW (in GBASIC) or TRSDRAW (in BASICG), you can print these 8-pin Dotwriter fonts directly on your screen. The above samples were done that way; they were loaded into MDRAW, and then the whole hi-res screen was printed at 150 dpi on my DeskJet printer. Once again, TRS-80 owners continue to have access to some of the latest computer options years after Radio Shack turned their machines into orphans.



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

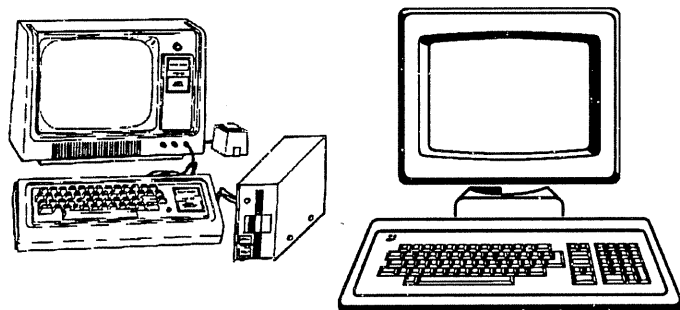
Z80 Tutor X All Z80 instructions, flags \$12.95

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

BORN AGAIN

by Lance Wolstrup



Before we get to meat of this article, let me freely admit that if you don't own a PC clone of some sort, the following will only be of limited interest to you. What we have here, my friends, is the very first MS-DOS article published in TRSTimes.

Now, why would we, faithful devotees of Models I, III and 4, publish such material? Read on, and you'll see how the 386/486 machines have finally been made useful and fit for TRS-80 consumption!

The whole thing began when Art McAninch uploaded a file to the TRSuretrove BBS. It was called 'MODEL1/ZIP, and was in MS-DOS format. Art's description of the file was something like 'emulate a TRS-80 Model I on your PC'.

Hey, that sounded interesting, so I downloaded the file. After transferring it to my 386, via SUPERCROSS, I 'unzipped' MODEL1.ZIP into a directory called MOD1 on the hard drive. I then began to examine the files that had been stored in the ZIP archive.

The manual, MODEL1.DOC, begins by describing what the author, Jeff Vavasour, has done:

"This emulator package allows your MS-DOS based PC to run as a TRS-80 Model I. It will be able to run virtually all Model I applications and software from Galaxy Invasion to Electric Pencil and the Z-80 Editor/Assembler. My aim was to be able to design a convincing 'virtual reality' in which a Model I program would not be able to tell that it was not running in the genuine machine. This virtual Model I includes the following options:

- 48k CPU with Z-80 microprocessor
- Expansion Interface (including internal clock, printer port, and floppy disk controller)
- RS 232 port
- 4 80-track single-sided disk drives
- Lower case and numeric keypad upgrades

This emulator requires CGA, DOS 2.11 or better, and at least a 7MHz clock speed, though a hard drive and at least 16MHz is strongly recommended."

Jeff includes 11 files in the ZIP to accomplish this:

MODEL1.EXE -	the emulator program itself
MODEL1.FNT -	the Model I character set
READ.ME -	copyright & registration info
SETUP.EXE -	installation & config program
CONVERT.BAS -	converts Level II ROM for use with MODEL1.EXE
MODEL1.DOC -	the manual
PUBLIC.DSK -	a Model I virtual disk containing some public domain programs
RECEIVE.EXE -	MS-DOS program to receive Model I ROM or disks from TRS-80 via parallel port (used with TRSLINK.BAS)
SPEEDUP.BAS -	patches Model I ROM to remove some built-in delays
TRANSMIT.BAS -	Model I BASIC program for sending Model I ROM and disks through the serial port
TRSLINK.BAS -	Model I BASIC program to send ROM and disks through parallel port

Reading the documentation, I realized that the emulator needed one item before it would work: the Model I ROM. Obviously, Jeff could not legally provide this software, as it is copyrighted material. However, he provides you with two methods of transferring the ROM from your Model I to the PC yourself.

The first method connects the two computers via modems (or a null modem) and then uses TRANSMIT/BAS to send the ROM image to the PC. The TRANSMIT/BAS program is also capable of sending entire disks.

The second method requires construction of a special cable so the computers can be connected via their printer ports. Complete instructions are given and, according to Roy Beck, the cable is simple to make.

At this point, the ROM has been transferred to the PC and when we execute MODEL1.EXE we can play in Level II Basic. We now need to transfer our Model I DOS or DOSes, using the same methods as above.

Once we have our favorite DOS ported to the PC, we can then begin to investigate the emulator. Jeff has done an excellent job - what you have before you is, indeed, a Model I with a very good monitor. Even the keyboard has been configured to emulate the Model I. For example, Shift 2 is now " (quote). The included program, SETUP.EXE, allows you to configure the keyboard to your liking.

So, what's the bottom line? To a PC user who has never been involved with a TRS-80, the emulator will be of

little or no interest. But, if you grew up with the Model I, this is a MUST have program. It is fun, fun, fun. I have spent the better part of a week just playing with it. Even got DANCING DEMON to work. Yes, the emulator has real TRS-80 graphics. Jeff drew the entire TRS-80 character set from scratch.

Though the entire package is shareware and can be obtained free of charge from Roy Beck's TRSuretrove BBS in Los Angeles (213 664-5056), I strongly recommend that you send Jeff Vavasour a \$25 registration fee to keep him programming this type of thing. Sending him the registration fee entitles you to an upgraded version with the following additional features:

SOUND! Many TRS-80 games used the cassette port to generate sound and speech. Now this output is routed to your PC's internal speaker. Sound may be turned on or off using F4. (You can even listen to CSAVE's if you like!)

SPEED! The registered package is a newer version of the emulator. It runs at up to 2.5 times the speed of version 2.05 (the shareware version). For software that runs too fast, a speed control is provided with the F3 option.

SNAPSHOTS! The registered version of the emulator can save and load "snapshots" of the TRS-80's memory and status. This is the "virtual cassette" option. You can save programs or games-in-progress and retrieve them instantly at any time. You can even exit the emulator or turn off the computer and come back right where you left off the next time you run MODEL1! Snapshot control is provided through the F3 menu.

You will also receive the source code, and be placed on an e-mail mailing list (available to Internet users) so that you may be notified of new developments.

To register, send \$25.00 to:
 Jeff Vavasour
 c/o Department of Physics
 University of British Columbia
 6224 Agricultural Road
 Canada V6T 1Z1

My \$25 is in the mail, and I look forward to receive the upgrade.

That was the review portion of this article. Now let me ramble on by chronicling some of my experiences with the Model I emulator.

As some of you know, I am not a hardware man. As a matter of fact, I avoid anything that even remotely involves a screwdriver. So, when the manual recommended constructing a special cable, I knew that this method was not for me. (I did manage to talk Roy Beck into making it - and present the project at the next VTUG meeting - I look forward to that.) I also knew that I was not about to move my Model I or my PC so I could hook

them up to separate phone lines - and I don't have a null modem. I was curious about the program, and I did not want to wait the two weeks until the VTUG meeting, so I began to think about another way. I am lucky enough to have a Model I, Model 4 and a PC, so transferring files from the Model I to a PC is not a problem.

The first step was to write the ROM to a file. I fired up the Model I with LDOS 5.1.4 and tried the DUMP command. Unfortunately, DUMP will not dump anything below 5500H. Now, I know that there is a patch somewhere to defeat this limitation - we may even have published it in a previous issue of TRSTimes - but I wanted to do it NOW, not run off somewhere and look in file cabinets. I realize that I could have block moved the 12K ROM code to an address higher than 5500h and then dumped it to a file, but I chose to write a short assembly language program to do the task- quick and dirty.

```

;GETROM/ASM
;copyright 1993 by Lance Wolstrup
;for TRS-80 Model I - using LDOS 5.x.x
;writes Model I ROM to disk in drive :1
;
START    ORG    7000H
          LD     HL,NAME      ;point to filename & drive
          LD     DE,FCB       ;point to 32 byte FCB buffer
          CALL   441CH        ;@FSPEC
;
INIT     LD     HL,BUFFER     ;point to 256 byte I/O buffer
          LD     B,0          ;record length is 256
          CALL   4420H        ;@INIT
;
          LD     HL,0         ;point to mem location 0
          LD     B,48         ;we will loop 48 times
                                ;0 to 2ffff = 12k
                                ;12k = 256 byte sectors x 48
LOOP0    PUSH   BC           ;save record loop counter
          LD     DE,BUFFER    ;point DE to I/O buffer
          LD     B,0          ;loop1 will transfer 256 bytes
LOOP1    LD     A,(HL)        ;get byte from ROM
          LD     (DE),A       ;and store it in I/O buffer
          INC    HL           ;point to next byte in ROM
          INC    DE           ;point to next buffer location
          DJNZ   LOOP1        ;loop until we have 256 bytes
;
WREC     LD     DE,FCB       ;point to FCB
          CALL   4439H        ;@WRITE - write buffer to file
          JR     Z,WREC1      ;jump if write succeeded
;
          LD     HL,WERR      ;point to write error message
          CALL   4467H        ;display message
          POP    HL           ;clear stack
          RET                ;and return to DOS
WERR     DEFM    'ERROR - DID NOT WRITE RECORD'
          DEFB    13
;
WREC1    POP    BC           ;restore record loop counter

```



```

        DJNZ  LOOP0      ;loop until all 48
                           ;records are written
;
CLOSE  CALL  4428H      ;@CLOSE
;
        RET              ;return to DOS
;
FCB     DEFS  32
BUFFER  DEFS  256
NAME    DEFM  'ROMIMAGE:1'
        DEFB  13
        END    START

```

Worked like a charm! In just a short time I had the ROM copied to drive :1 to a file called ROMIMAGE.

The next step was to fire up my Model 4, run SPRX/CMD, and copy the ROMIMAGE file from the Model I disk to an MS-DOS disk.

Finally, I took the MS-DOS disk, inserted it in my PC, and copied the ROMIMAGE to my MOD1 directory. Running MODEL1.EXE now brought me to Level II Basic. My program had copied the ROM just fine - everything was going well.

However, messing around with Level II Basic held my attention for only a short while. I needed to get my DOSes ported over. By looking at the code to the transfer programs, I suspected that the MODEL1.EXE used files to emulate virtual disks. Well, if that was the case, maybe I could copy my Model I disks to individual files, and then transfer them the same way I had the ROMIMAGE file. It was worth a try, so I wrote another quick and dirty assembly language program.

;GETDISK/ASM

;copyright 1993 by Lance Wolstrup

;for TRS-80 Model I - using LDOS 5.x.x

;writes entire disk in drive :0 to file in drive :1

;note that disk in drive :0 must be 35-track, single density,

;while disk in drive :1 must have at least 36 tracks

```

;
START  ORG    7000H
        LD     HL,LDOS      ;point to filename
        LD     DE,FCB       ;point to FCB
        CALL   441CH        ;@FSPEC
;
        LD     HL,BUFFER    ;point to I/O buffer
        LD     B,0          ;record length is 256
        CALL   4420H        ;@INIT
;
        LD     C,0          ;select drive :0
        CALL   4754H        ;@SELECT
;
        LD     D,0          ;track 0
        LD     B,35         ;disk has 35 tracks
LOOP0  PUSH   BC            ;save track loop
        LD     B,10         ;sector loop

```

```

LOOP1  LD     E,0           ;sector 0
        LD     A,11H        ;check for
        CP     D            ;directory track
        JR     NZ,RDSECT    ;jump if regular track
;
        CALL   4B45H        ;@RDSSEC
        JR     LOOP2        ;jump over @RDSECT
;
RDSECT CALL  4477H          ;@RDSECT
;
LOOP2  PUSH   DE            ;save track & sector
        LD     DE,FCB       ;point to FCB
        CALL   4439H        ;@WRITE
        POP    DE           ;restore track & sector
        INC    E            ;next sector
        DJNZ   LOOP1        ;repeat for 10 sectors
        POP    BC           ;restore track counter
        INC    D            ;next track
        DJNZ   LOOP0        ;repeat for 35 tracks
;
        LD     DE,FCB       ;point to FCB
        CALL   4428H        ;@CLOSE
;
        RET                ;return to DOS
;
LDOS   DEFM  'LDOS/DSK:1'
        DEFB  13
FCB     DEFS  32
BUFFER  DEFS  256
        END    START

```

I booted up the Model I with LDOS 5.1.4 in drive :0 and formatted a 36 track, single-density disk in drive :1. Then I ran the GETDISK/CMD program and, sure enough, it read each sector on each track and faithfully copied the information to a file on drive :1 called LDOS/DSK. This, I am sure, took somewhat longer than Mr. Vavasour's parallel port method. But, no matter, I had created a file containing the entire LDOS boot disk in drive :0. Now, the big question was - would it work?

I stuck the LDOS/DSK disk in my Model 4 and used SPRX/CMD to copy it to an MS-DOS disk. I then inserted that disk in my PC and copied the file, LDOS.DSK, to my MOD1 directory.

Next, I fired up MODEL1.EXE and used the menu to instruct the program to use the LDOS.DSK file as the boot disk in drive :0. I rebooted the emulator and - HOLY MOLEY - up came the LDOS 5.1.4 logo and I was prompted for the date (yes, it would not accept a date after 12/31/87), and then for the time. After taking care of this preliminary business, I was looking at LDOS Ready. My program had succeeded - DYNAMITE!!!

The next few days were spent finding my Model I disks and copying them to files so they could be transferred to the PC. I have three drives attached to my Model I, so, in order to be able to copy a non-system disk (a data disk),

I kept the LDOS 5.1.4 disk in drive :0, but began copying the disks from drive :2 to files on the disks in drive :1. This only required one change in the program. Note the bolded line in the listing - change it to read:

```
LD      C,2          ;select drive :2
```

As of this writing, I have the following DOSes running:

```
LDOS 5.3.1
LDOS 5.1.4
MULTIDOS 1.6
TRSDOS 2.3
NEWDOS/80 v2.
DOSPLUS (don't know which version)
```

The following programs seem to be running fine (note that they have not been thoroughly tested):

```
EDAS
EDTASM series 1
EDTASM (NEWDOS/80)
BASIC
TED (LDOS)
FEDII (LDOS)
SUPERZAP (NEWDOS/80)
DISASSEM (NEWDOS/80)
ZAP (MULTIDOS)
PROFILE
SCRIPSIT
VISICALC
NEWSRIPT
LESCRIPT
ENHBAS
```

I am convinced that most other programs will run as well, but since I haven't had time to transfer the rest of my collection, I'll have to leave that as speculation for the moment.

While checking if EDAS would run, I wrote a few lines of assembly code and then gave the command to get a hard copy (H#,*). Sure enough, the printer port worked and my listing was printed. The printer behaved exactly the way it was supposed to (HP laserjet series II) when hooked up to a TRS-80 Model I - it printed everything on the same line - just carriage returns and no linefeeds! Imagine that - my HP is physically hooked up to an MS-DOS clone that for the moment thinks it is a TRS-80 Model I - and the PC is sending TRS-80 commands to the printer. That's a great job of programming.

All I needed to do was to dust off and make some slight alterations to the program I wrote a couple of months ago for my Model 4 when I hooked the HP up to it. No sooner said than done - I restarted EDAS and began coding the following program. Do note that I have written the code to this and the other listings in a manner so they can be keyed in to EDAS, or any of the other edi-

tor/assemblers available, such as EDTASM series I from Radio Shack, or the NEWDOS/80 version of EDTASM.

```
;HP/ASM
;Model I utility to convert
;cr to cr + lf on HP laser jets
;& desk jets
;copyright 1993 by Lance Wolstrup
;
START  ORG    7000H
      LD      A,(37E8H)      ;read printer status
      AND     0F0H          ;mask out lower 4 bits
      CP      30H           ;check if printer available
      JR      NZ,NOAVAIL    ;jump if not
      LD      HL,CRLF       ;point to printer data
      CALL    446AH         ;@PRINT
      LD      HL,MSG1       ;point to MSG1
      CALL    4467H         ;and display it
EXIT   RET                 ;return to DOS
;
NOAVAIL LD      HL,MSG2     ;point to MSG2
      JR      EXIT         ;display msg and exit
;
CRLF   DEFB    27          ;reset
      DEFB    69          ;printer
;
      DEFB    27          ;this sequence
      DEFB    38          ;of code sets
      DEFB    107         ;lf = lf
      DEFB    49          ;cr = cr + lf
      DEFB    71          ;ff = ff
      DEFB    3
;
MSG1   DEFM    'HP PRINTER NOW INITIALIZED FOR '
      DEFM    'YOUR MODEL I'
      DEFB    13
;
MSG2   DEFM    'HP PRINTER NOT AVAILABLE'
      DEFB    13
;
      END    START
```

With HP/CMD on my boot disk, I simply turn on my HP and, when it is ready, I execute my program and the laser jet is then a TRS-80 printer - and a very good one it is - sure beats the Line Printer I with no descenders!

Though I have been positive about everything so far, I must pass on that I have not been able to get the arrow keys to work correctly. There are a couple of other annoyances as well, which I will pass on to Jeff Vavasour. Hopefully, we can get everything working correctly. That will be such fun.

Just think of it, a Model I with no hardware problems. Never again will you need to 'pink-pearl' the edgcard connectors!

The Model I born again? Could be.

BEAT THE GAME

by Daniel Myers



ZORK II

So, back for more, eh? Zork I wasn't enough...or did you get hooked? No matter, it's almost time to get started. However, first a few words of advice. The Wizard of Frobozz is a lot more troublesome than the thief ever was. This is due to his spell-casting abilities. Therefore, frequent saving of the game is recommended! Otherwise, you will be wasting a lot of time waiting for some spells to wear off.

Okay, let's go! Get the sword and lamp, and move along South, South, South, SW to the Shallow Ford. At this point, turn on your lamp, then continue South, SE to the North End of Garden. Enter the Gazebo, and get all items on the table. Then "Exit Gazebo," and go N, NE to the Shallow Ford and fill the teapot with water.

Now, head South, SW, SW, and you will be in the Carousel Room. This is a wonderful place that spins you around, so you can't be sure that you're going in the direction you specify. Fun, huh? The thing to do here is keep moving around until you get to the Riddle Room. That room is actually SE of the Carousel Room, but you aren't likely to get there by typing SE, at least not for a while. So, every time you move, and find yourself somewhere else besides the Riddle Room, you will have to return to the Carousel Room and try again. Also, before you start moving around, drop everything but the teapot and the lamp; you'll pick up the things again as you need them.

These are the directions from the various other rooms to the Carousel Room:

- Marble Hall - South
- Path Near Stream - Southwest
- Topiary - West
- Menhir Room - North
- Cobwebby Corridor - Northeast
- Cool Room - Southeast

All right, you've finally made it to the Riddle Room. Now, all you have to do is answer the riddle! Actually, it's a fairly simple riddle; just enter: Say "A Well," and you will be able to move on to the East, which is the Pearl Room. Leave the necklace for now, and continue East into the Circular Room. There is a large bucket here, just big enough for you to enter, so do that. Then pour water into the bucket, and it will rise to the top of the well.

Get out of the bucket, then go East into the Tea Room, and get all cakes except the orange one. Eat the green cake, and you will shrink down. Now, go East into the Pool Room, and throw the red cake into the pool of tears.

The water will be soaked up, and you can get the package of candies. Forget about the flask; there is no use for it.

Return West to the Posts Room, and eat the blue cake. Presto! You are returned to normal size. Stretch a little, then head NW into the Low Room. Aha, there's a robot here. Tell the robot to "Go E," then go that way yourself, and you will both be in the Machine Room. I wouldn't be surprised

if the controls to the Carousel room were here! Tell the robot to "Push Triangular." There will be a thud in the distance (something fell), and the Carousel Room is now off.

Tell the robot to go South, and do likewise. You are now in a Dingy Closet, and a red sphere sits invitingly just beyond your grasp. Try getting it, and a steel cage will come down over you (uh oh!). Not to worry, however! Once the cage is down, just tell the robot (even though

you can't see him) to "Lift Cage." Once that's done, you will be able to get the sphere.

Now, getting back to the bucket may be a problem of sorts; pushing the button has caused strange things to happen in the Low Room. So, after going North and West from the closet into the Low Room, you may have a problem when trying to go SE back to the Tea Room. Keep trying however, and you'll get there sooner or later. Then, go West from the Tea Room, enter the bucket, get the water, and the bucket will descend again. Get out, drop the Teapot, head West, pick up the necklace, then West again and NW to the Carousel Room. There is a box here (the thud you heard before). Open it, and inside is a violin. You don't need to take it now, so just leave it here for the time being.

Drop the sphere, necklace and candy, then get the sword, place mat and letter opener. Move North to the Marble Hall, picking up the brick, then continue North until you come to the Ledge in Ravine. Go up to the Tiny

Room, which has a locked door. Slide the mat under the door, then move the lid and insert the opener in the keyhole. Remove the opener, then pull the mat and get the key. Now, unlock and open the door and go North into the Dreary Room. Drop the key and opener, and get the blue sphere.

Now, it's South and Down to the ledge, then West and North into the Dragon Room (what would an adventure game be without a dragon?). He's not one of your nicer dragons, so be careful here. Hit him with the sword, then move South. He will follow you. Hit him a second time, and move South again. He's still following you, and you're almost where you want to be. Whack the scaly nuisance one more time, and head West into the Ice Room.

Okay, the dragon follows you in, gets silly over his reflection in the glacier, and proceeds to get himself drowned. You can drop the sword now, since you won't be needing it anymore. As his body washes away, go East and SE to the Carousel Room, then SW to the Cobwebby Room. Get the string, then go back NE and pick up the newspaper and the matches. Now move along NW, West, and West again into the Lava Room. Leave the ruby for now, and continue on Southward to the Volcano Bottom.

Aha, look what's here....a hot air balloon! All you need is a way to get it inflated, and you just happen to have the means to do that. Get into the basket, then open the receptacle and put the newspaper inside. Light a match, then light the newspaper with the match. Make sure you keep the receptacle open! Now, just wait until the balloon rises to the Narrow Ledge, then "Land." Tie the wire to the hook and get out of the basket.

On the ledge is a gold coin. Pick that up and head South into the Library. Get the purple book, open it, get the stamp, then drop the book. You can ignore the other books; they are all worthless. Now, back North and into the basket again. Untie the wire, and wait until the balloon rises to the Wide Ledge.

Again, "Land" and tie the wire to the hook. Get out and go South. Now, it's time for a few thrills. Put the string in the brick, then put the brick in the hole in the box. Now light a match, then light the string. Now, head North *IMMEDIATELY*. There will be an explosion, then you can safely go back and get the crown. But don't dawdle; once you have it, get back to the basket pronto! The explosion weakened the ledge, and it won't hold much longer. So, get in, untie the wire, close the receptacle, and wait until the balloon lands at the Volcano Bottom.

Well, you seem to be doing almost as much travelling around as in Zork I, and you aren't finished yet, not by a long shot! Go North, get the ruby, then East twice and SE to (yet again!) the Carousel Room. Drop off everything but the lamp, then go NW, North, North, West, and West again, and you should now be at the entrance to the bank.

The bank is tricky, so follow directions carefully here. Go NE, then East into the Safety Depository, then South into the office and pick up the portrait, and back North again. Now, "Enter Light," and you will be in a small room. At this point "Enter South Wall," and you will be in the Depository again. "Enter Light" a second time, and you will find yourself in the vault, with a pile of bills in front of you. Get the bills, then "Enter North Wall," and once more you are in the Depository.

Now, drop the bills and portrait, and go East to the East Tellers Room. Go East again to the Depository, pick up the goodies, and "Enter Light." This time, you are in the East Viewing Room. From there, go South and you will be at the bank entrance. You have robbed the bank, and kept the alarms silent.

Okay, head along East until you reach the Dragon Room, then North to the Dragon's Lair. Leave the chest for now; instead, say "Hello Princess," and wait until she leaves. Follow her, and continue to follow her until you are both in the Gazebo, then wait. The unicorn will appear, and the princess will give you the key from around its neck, and also a rose. You can drop the rose; it has no use in the game.

Well, you're getting closer to the end-game, but there are still a few things left to do. Exit the Gazebo, then move South, West, SW. Drop off some of the treasures, then go back NW, North, North, North to the Lair again. Open the chest and get the statuette, then return to the Carousel Room. From there, due South to the stairway.

This is a good place to save the game, as you are about to enter one of the nastier mazes around.

Go down the stairs, and you will be in the Oddly-Angled Room. Things are not always as they seem here, and the exact direction to go at this point is not constant. Try East first, and if you can't go that way, then try West.

Either way, you should find a club. Get the club, and from that point, go SE, NE, NW, SW. Watch the room descriptions carefully as you do this; the little diamonds on the floor should get brighter with each move you make (you are actually "running the bases" here). If they are not doing this, restore the game and try again.

When you have done it successfully, there will be a noise in the distance.

Now, you should try to go either North or South (that varies also). If one way doesn't work, try the other, and if neither works, try East or West. There will be a staircase going down, but you don't want to do that yet, so go Up instead, and head North until you are back in the Carousel Room.

Get the blue and red spheres and the candy. Drop a few things if you have to, but hang on to the club! Now, SW twice, and you are in the Guarded Room, with a lizard head in the doorway. Give the candy to the lizard, then unlock the door with the gold key. Open the door and go South, then West twice into the Aquarium. Throw the club at the aquarium, then get the clear sphere. Go East into the Wizard's Workroom.

Put each sphere on the stand of the same color, then get the black sphere that appears, and go South into the Pentagram Room. Put the sphere on the circle, and the Demon will appear. You must give him *ALL* the treasures you have collected, as well as the gold key. This will mean at least one more trip to and from the Carousel Room to get all the stuff. When you have given everything to the demon, tell the demon, "Give Me The Wand," and he will take the wand from the Wizard and give it to you.

Now, you're almost finished! Go North, East, North, North, NE, South, and you are in the Menhir Room. You need to get the Menhir out of the way for a little while, so do this: Wave the wand at the Menhir and say, "Float." The Menhir will rise up, allowing you to go SW into the kennel and get the collar. After that, go NE, then South, then Down, and Down again into the Cerberus Room.

Cerberus is no problem; just put the collar on him. Now, go East, then South, and you are in the Crypt. Turn off the lamp, and you will notice a secret door in the south wall. Open the door and go South, and you will be on the Landing. The game is over!!

Or is it? After all, there's still Zork III up ahead!

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.

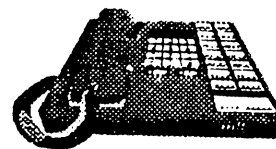
To subscribe for one year of 8 issues, send \$27 US or \$36 outside North America to REC, Att: Dr. M. Ecker, 909 Violet Terrace, Clarks Summit, PA 18411, USA or send \$10 (\$13 non-US) for 3 sample issues, creditable.

TRSuretrove BBS

8 N 1 - 24 hours

Los Angeles

213 664-5056

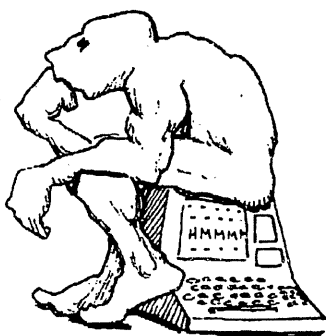


where the TRS-80 crowd meets

A BRIEF HISTORY OF HARD DRIVES AND THEIR TROUBLES

by Roy T. Beck

Introduction



Why would I bother to re-cite my personal hard drive troubles in public? Maybe it is a case of self-flagellation, maybe to complain about some of Radio Shack's design shortcomings, maybe to entertain some of you, maybe to fill this space in TRS-Times, probably some of all the above. Anyway, I will proceed.

My first contact with Radio Shack hard drives came about when I acquired one of the 5 Meg "Store Units". These units were placed by RS in many of their retail stores as a management tool to keep track of inventory, do bookkeeping, etc. I don't know why they discontinued that practice, but I have a suspicion the software was not sufficiently user friendly, and not all the managers were sufficiently computer literate to understand the hardware, keep the system working, and key in all the requisite data.

In any event, I acquired a bare box from the back room of a retail store in the area. And it was bare! No cables, no software, not even a key to turn it on! And since I didn't know beans about it, and really didn't even know the right questions to ask, I was starting from below ground zero. I finally did get the software, replaced the keyless switch, and made some cables. Then I tried to get it running, using TRSHD. What a struggle. To this day, I have never understood one of the questions the installation program asked. It wanted to know the number printed on the bottom of the box. I assumed it wanted the serial number, but refused to accept it. I then tried the catalog number, but it wouldn't buy that, either. I finally got it to accept a number made from thin air, and went on with the procedure. The installation program was written in BASIC, but I couldn't even figure out what the BASIC did with the answer I got it to accept. But I did get the HD to work, and I had a real sense of accomplishment as a result.

As time went along, I took the thing apart to study its internals (naturally), and also acquired the service manuals from RS. Now things became a little less mysterious. It was quite a while before I worked out the miserable

"three wires" RS soldered directly to its bubbles. This hookup varied with the size and manufacturer of each different bubble.

Available Sizes

Just as a review, let me list the available sizes of drives. RS produced the following sizes:

5 Meg 12 Meg 15 Meg 35 Meg 70 Meg

The 12 Meg was a little short of 12 Meg; it actually was about 11.3, but that was just the usual puffery. For a different reason, the 35 and 70 Meggers actually netted out at 32 and 64, respectively. The larger label was true on a Model II, 12, 16, etc, but due to formatting restrictions, the Models III and 4 are limited to the smaller sizes.

Since the RS boxes will accept any MFM drive bubble up to 70 Meg, there are a lot of intermediate sizes which may be fitted into an RS box. A very popular bubble is the 20 Meg ST-225, which for several years was the "way to go" with IBM clones, and therefore is available, both new and used, in good quantity.

Troubles

As time went along, I acquired other HD's for resale and also had occasion to troubleshoot them, both for myself and others. How many of you old timers remember the "pink pearl" treatment for the Model I edge connectors? That was the use of Pink Pearl (or other) brand of eraser to scrub the tarnish off of the tinned edge connectors which gradually formed as the lead in the solder oxidized. The Models III and 4 and 4P all suffer the same problem on the 50 line I/O connector, and eventually the attached hard drive grows flaky, losing data or even entirely refusing to communicate with the computer. I have had this particular trouble occur several times, and a thorough scrubbing of the edge card connector solves the problem. The cable end connectors are normally equipped with gold plated fingers, and they do not appear to cause trouble. It is just that tinned 50 line connector. If only RS had gold plated the connector as everyone does now. Oh, well.....

Other troubles I have known include ribbon cables which have been bent too often and/or severely, to the

point where a wire has been broken internally and makes intermittent or no contact. A related trouble is the fingers in the end connectors which are designed to penetrate the ribbon cable insulation and grasp the conductors inside the cable. Because there is no good way of getting the connector loose from the computer or the HD except pull on the cable, eventually trouble can occur in this connection.

Twice I have found the connector on the back of the hard drive box consisted of only 49 pins instead of the normal 50! The pins are rather delicate, and can be easily broken off if they are bent a few times. Of course, there is no good reason for bending them, but such things do happen.

A curiosity of the RS design is the red and green lamps on the front of the HD box. The green lamp serves two functions. If it is lit, the drive is Selected. If it is flickering, the drive is stepping. If it is out, then the drive either is not selected, not powered up, has a problem, or may be burned out. Or all of the above! Even more curious is the red lamp. This lamp, if lit, means the HD is Write Protected. If the lamp is dark, it normally means the HD is not write protected, or the lamp is burned out or not secure in its socket. But, surprise, if it is burned out or not making contact, the internal logic says the HD is write protected, anyway! If you ever find the HD is "write protected" when you know it isn't, test the lamp. (I have long life replacements available, which will outlast you).

There is a fuse in the internal power supply, but it is very gracious. If it ever blows, you have SERIOUS trouble, but it is extremely rare for such to happen. A more frequent trouble is the cable connector pins which are soldered into the power supply board. After long use and heat cycling, it sometimes happens that hairline cracks form in the solder which secures the pins and causes random voltage fluctuations. The HD is not happy under those conditions. Fortunately, the fix is easy. Take a soldering iron to the connector pins on the solder side of the power supply board and flow the solder around the pins. Presto, the power supply is good for many more years.

Twice I have had situations where the motor in the bubble failed to start. One time it was due to a blown transistor in the motor circuit. Another time it was apparently due to the bearing lubricant having congealed due to the drive having been stored for a long period of time. How could the latter be? The motor in a bubble has very little starting torque, apparently, in this case, less than the amount required to "break loose" the bearings. I say this because I was able to start the motor by turning on the hard drive, and then with the entire drive picked up off the table, I gave the drive an abrupt twist. The motor torque plus the inertia and my rotary jerk was enough to overcome the sticky bearings, and the motor then came up to speed. Seemed to work OK afterwards, hence my analysis! I also once advised a fellow via telephone to do this

and he also was successful. My feeling is that only the oldest RS drives with Tandon bubbles suffer this problem, but it's a trick to remember.

Other Failures

Failures of the heads and/or platters in a bubble (a crash) is the usual cause of a HD demise. Can such failures be repaired? Yes. Can data be salvaged? Maybe. What are the costs? While it is possible for a technician to open up a hard drive bubble in a clean room and repair it (new heads, new platters, new bearings, etc) the costs of doing so have to be considered. The minimum bench charge for a tech to open and repair a drive starts at about \$100 and goes up with drive size. Since there are very few new drives available of the MFM type, which is what the TRS uses, only used ones are available for replacement. The question of repair vs. replacement is really a cost tradeoff. In the small sizes, repairs are generally not economic. Can data be salvaged? Possibly, but only at great cost. Do your regular backups and don't bother with data recovery.

Restoring a Drive

Where a drive has acted up due to a bad cable connection, it is quite possible for the DOS on the HD to become corrupt, so that solving the electrical problem will not restore normal operation. Let me walk you through several scenarios.

Assuming only the DOS is corrupt and user files are OK, proceed as follows:

1. Make a new bootup disk by duplicating original DOS disk. Use this new disk to boot the machine.
2. Reset the existing partitions on the hard drive by running the
SYSTEM (DRIVE = n, DISABLE, DRIVER = "RSHARD6")
command for each partition, where n is the drive number. After each partition is restored, the machine should return to DOS ready without comment. If instead it reports "Note, Drive appears to be unformatted", then either the partition was not set correctly or the BOOT and/or DIRECTORY are corrupt. After each execution of the SYSTEM command, execute DEVICE to determine that the new partition is now available. The DEVICE command will report all the presently enabled drives, both floppy and hard. It is advisable to execute SYSGEN after each partition is restored to avoid losing your work in the event of a reboot, improper command, etc. This will progressively save the machine configuration in an invisible file on the boot floppy named CONFIG/SYS.
3. Assuming the partitions can be reset without the "unformatted" comment appearing, the next step is to

load a fresh copy of DOS onto the partition which is to become the system partition. For the sake of this example, I will assume the HD partitions were reset as :4, :5, :2, and :3, respectively. The fresh copy of DOS can be copied from the new boot disk by the command `BACKUP :0 :4(S,I)`.

4. The next step is to make the :4 partition become the system partition. The command is `SYSTEM (SYSTEM=4)` in this example. If all the hardware is working properly, the machine will now be operating from the copy of the DOS on the first HD partition.

5. Next, execute the command `SYSTEM (DRIVE=5,SWAP=1)` to make the second HD partition become drive :1. By executing the above two commands, the floppies have now become drives :4 and :5, respectively.

6. Next step is to `SYSGEN` the configuration so the machine will bootup correctly. The command is `SYSGEN (DRIVE=4)`.

Why 4? The bootup is from the first floppy, but the first floppy is presently drive :4, hence the `CONFIG/SYS` file must be created on the boot floppy in drive :4.

7. Now pull the boot disk out of the drive, turn off the computer and the hard drive and wait a bit, perhaps 30 seconds. Next, powerup the hard drive and the computer. Insert the new boot disk, press reset, and hit `ENTER`. The machine should now reboot with the hard drive as drives :0, :1, :2, and :3, and the floppies as :4 and :5. If all is well, you should next explore the recovered files on the hard drive to see if they are OK. If so, pat yourself on the back. If not, restore them from your backups. You did make backups, didn't you?

If the "Note, drive appears to be unformatted" comment appears after resetting a partition, it can mean two things; one is the `BOOT` or `DIRECTORY` of that partition is corrupt, or possibly you assigned the wrong number or location of heads and cylinders to the partition. If the former, the only option is to reformat the partition.

Use `RSFORM6 :n` to proceed. Note, this wipes out everything, good or bad, which was in the partition. It is irrevocable! If it was the latter, then reboot instead of `SYSGENing` and try again to set the partition correctly.

Design Limitations

The old, large controller boards, as used in the 5 Meg and some 12 and 15 Meg drives lacked a feature which permitted proper, shared control of two of the lines of the 50 line I/O cable. The missing feature was an open collector output in two circuits which controlled the `WAIT` line and the I/O line. As long as no other device than the hard drive was plugged into the 50 line I/O connector, every-

thing worked OK. The trouble occurred when one of these old boards shared the 50 line I/O connector with some other accessory, such as a `NEWCLOCK`, or `ORCHESTRA80`, etc. The two devices engaged in a tug-of-war over the two control lines, with everyone the loser. No damage, it just didn't work.

Apparently RS later saw the error of their ways, and the factory assembly line installed a bunch of jumpers on the later production runs of the old, large board, which apparently took care of the problem.

The later, smaller boards did not have this problem. RS designed them correctly so as to share the 50 line connector with other devices.

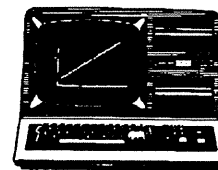
Miscellany

Autobooting is available to the Model 4, 4D or 4P user, courtesy of M.A.D. Software, Box 331323, Ft. Worth, TX 76163. Autobooting means you can bootup your hard drive without need for a floppy disk in drive :0. To have autobooting capability, you must buy a new EPROM and a piece of software. The EPROM replaces an RS ROM in your machine and the software patches some system files on the hard drive. (Note, PAL version 4P's do not need a ROM change). The installation of the ROM requires opening of the computer, but no soldering is necessary. The software is installed by a file named `HBUILD6/CMD`, and takes only a few seconds. I recommend M.A.D. Software's products.

Another nice feature is having an internal clock available so that accurate date and time are automatic at every startup. `CN-80` offers the clock and associated software. Installation of the clock also requires opening of the computer, as it is installed underneath a ROM chip; the ROM then piggybacks on top of the clock chip. No soldering, no big deal.

Summary

Altogether, the RS hard drive can be quite a satisfactory package, the biggest problem being those old tinned edge card connectors on the computer motherboard which occasionally (once a year?) need the "pink pearl" treatment. If you keep those connectors clean and shiny, the drives really will do a fine job for you. Even a 5 meg will hold a heck of a lot of software for a Model 3 or 4. Since the upper limit is 64 Meg (the boxes were labelled 70 MEG) you can store an awful lot of database and text files if you need to.



RUBIK'S CUBE SOLVED!

A BASIC Program for Model I & III by Delmer D. Hinrichs

Rubik's cube is a most interesting device from several points of view: It is a challenging puzzle, somewhat reminiscent of Sam Loyd's 15 puzzle, only in three dimensions instead of two. It provides a complicated mechanical problem; how are the parts held together, and still allowed to move freely? It has intriguing parallels to mathematical group theory, and to the theory of elemental particles in physics.

However, my intent is to provide a program that will tell you how to unscramble a Rubik's cube from any starting position. For more on its other aspects, see Douglas R. Hofstadter's article in the March 1981 issue of Scientific American.

The solution of Rubik's cube (getting all six faces of the cube to show solid colors) requires a systematic approach. There are over 43 quintillion possible states in which the cube can be; this is over 10 billion times the "More than three billion combinations" assertion on some of its labels. In addition, if the cube is disassembled and then reassembled, the possible states are increased by another factor of twelve. Clearly, random twisting of faces will never (well, hardly ever) give a solution.

How can we specify exactly what has to be done to solve the cube? First we must define our terminology: Let's call the six faces of the cube the top, bottom, left, right, front, and posterior faces (so each may be identified by its initial letter). Then to move the "cubies" (the movable parts of the cube), we can say, for example, "F+" to rotate the front face by 90 degrees clockwise, or "R-" to rotate the right face by 90 degrees counterclockwise.

There are really only three types of cubies in the cube: Center, edge, and corner. The center cubies are fixed in position, and cannot move with respect to each other. They can only rotate in place. The edge and corner cubies can move to different locations with respect to each other, and with respect to the center cubies.

To simplify things, choose a color to be "top", and orient the cube so that the center cubie of the top face is of that color. This color will remain on top throughout all

"Rubik's Cube" is a trademark of Ideal Toy Corporation.

later operations to solve the cube. For handling convenience, the posterior face is never rotated; if necessary, the whole cube is rotated on its vertical axis to make the posterior face the right or left face before rotating the (previously posterior) face. Rotating the whole cube on its vertical axis will be called "C+" or "C-", assuming that the cube is held from the bottom in the right hand. That is, "C+" rotates the whole cube in the same sense that "B+" rotates its bottom face.

Now to use the program: Load the program, Listing 1, and enter RUN. The cube will be displayed as in Figure 1. Enter the color of each cubie face as a one-letter code, and the flashing cursor will move to the next cubie face. For this and all later entries, letters may be entered as either (letter) or as Shift-(letter), and <ENTER> is not used unless it is asked for.

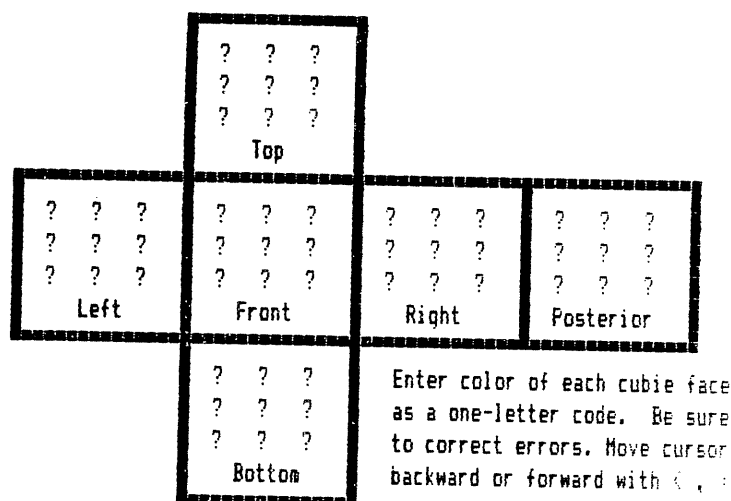


Figure 1. The initial video display, before user entry of colors for any of the cubie faces on any face of the cube. The cursor now in upper left corner of the "Top" face actually flashes. Each "?" is to be replaced by a one-letter color code.

Since different cube manufacturers use different colors, and even the same manufacturer may not always put the colors in the same orientation, you have complete freedom on where to put each color. But if you do not enter exactly nine cubie faces of each of six different colors, you will be shown an error message, and given a chance to correct the error. During entry, the cursor may be moved backward or forward in its sequence by pressing the "<" or the ">" keys (it is not necessary to shift these keys). Holding these keys down results in continuous cursor movement. To change the color of a cubie face, enter the new color over the top of the old.

If you make a more subtle error in the entry of the color of cubie faces, such as transposing the colors of two faces, the program will give an error message later

during operation. This form of error will also occur if someone has disassembled your cube and reassembled it improperly; for any random reassembly of the cube, there is only one chance in twelve that the cube can be solved.

After you have entered the colors of all cubie faces, you are given the choice of manually keying in moves for the computer to make, or of letting the computer automatically select the moves to solve the cube. If you select manual moves, you can later at any time let the computer take over.

If you let the computer select the moves, you still have two choices: You can have the computer stop after displaying each move to be made (so that the same move may be made on the physical cube), or let the computer run straight through to a solution (useful to check if a solution can be made). In addition, you can have the initial status of the cube, and all moves necessary for its solution printed on a printer, for a permanent record.

Then the solution of the cube is given. The program works in sequence; first the top edge cubies are gotten into place, then top corners, vertical edges, bottom corners, and finally bottom edges. The program displays which group it is working upon, as well as the current move to make, Figure 2. The display blinks as each new move is displayed, to attract attention. The colors shown on the cube display also change.

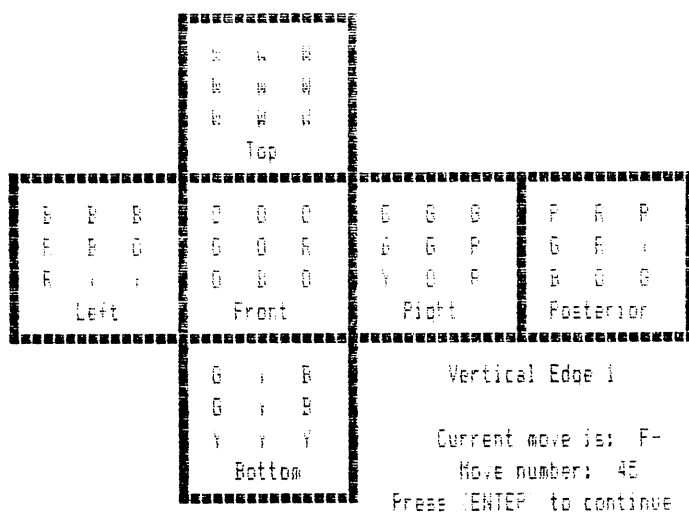


Figure 2. The video display during computer solution of the cube. The top layer has now been solved and the computer is working on the second layer.

Cubies in the first three groups are placed one at a time into the correct position, though the whole cube is rotated so that the top, front, right cubie is always the one

being worked upon. Since one bottom cubie cannot be changed without also changing other bottom cubies, they are worked upon as groups, rather than individually.

It usually requires from 130 to 200 moves to solve a completely scrambled cube. Some starting positions can be solved more quickly, Figure 3. The program makes moves at a rate of about 1.5 seconds per move (or about 2.2 seconds per move if a clock control board has not been installed). Disk systems are slightly slower. The "OUT 254,1" statement turns on the Archbold clock control board, and "OUT 254,0" turns it off. If this board is not present, these statements have no effect.

Initial Status of Cube:

```

      R B O
      R W O
      R G O

Y Y Y   B R B   Y Y Y   G O G
G G G   Y R W   B B B   Y O W
W W W   G R G   W W W   B O B

      O B R
      O Y R
      O G R
  
```

```

B-  F+  T+  L-  T-  C+  F+  F+  C+  F+  T+  L-
T-  C+  B+  F+  T+  L-  T-  R-  B+  B+  R+  F+
B+  B+  F-  C+  C+  F+  B+  B+  F-  R-  B+  B+
R+  C+  B+  B+  B+  F+  B+  B+  F-  B+  F+  B+
L+  B-  L-  B+  F+  B-  F-  B-  R-  B+  R+  C+
B+  F+  B-  F-  B-  R-  B+  R+  C+  B-  F+  B-
F-  B-  R-  B+  R+  C+  R-  B+  R+  B+  F+  B-
F-  B+  R-  B+  R+  B+  F+  B-  F-  B+  R-  B-
R+  F+  B+  B+  F-  R-  B+  R+  B+  R-  B-  R+
B-  R-  B+  B+  R+  B+  B+  R-  B-  R+  B-  R-
B+  B+  R+  B+  B+  C+  C+  L-  R+  F+  L+  R-
B+  B+  L-  R+  F+  L+  R-  C+  C+  L-  R+  F+
L+  R-  B-  L-  R+  F-  L+  R-  B-  L-  R+  F+
F+  L+  R-  C+  C+  C+  L-  R+  F+  L+  R-  B+
B+  L-  R+  F+  L+  R-
  
```

Figure 3. Optional printout of the solution of the cube. The initial status and the moves needed to solve the cube are shown.

The selection of moves is by the "brute force and awkwardness" method; can someone devise more elegant algorithms? At least, I avoided the string operations that would cause the dreaded "garbage collection" pauses. BASIC is really not at its best in pattern recognition.

Nomenclature and move sequences were mostly derived from James G. Nourse's excellent little book: "The Simple Solution to Rubik's Cube".

This is a long and difficult program to key in correctly from a listing; if readers will send me \$10, I will supply the program on tape or disk in TRS-80 Model I or Model III format. The address is:

Hinrichs Software, 2116 S.E. 377th Ave.
Washougal, WA 98671-9732

For those keying in the program, multiple lines under the same line number were formed by pressing the down-arrow. As listed, the program is a little too long to run on a 16K TRS-80; delete REMark lines until there are at least 1250 bytes of free memory before the program has been run. No REMark lines are referred to by the program.

Displays are upper/lower case for those systems that support lower case video display. Otherwise, displays will be in upper case only.

Now that you have this program, there is no excuse for leaving your Rubik's cube in a scrambled condition!

Listing 1

RUBIK/BAS

```
10 CLS :POKE 16409,0 :
PRINT@ 104,"Rubik's Cube Solved"
20 ' (c) by Delmer D. Hinrichs 1993
30 IF B GOTO 130
40 CLEAR 65 :DEFINT A-Z :
DIM C(5,8), M(303), P(53), T(25)
50 E$ = CHR$(95) + CHR$(24) :F$ = CHR$(176):
OUT 254,1 :POKE16553,255
60 POKE 16912, PEEK(16912) OR 64 :OUT 238,8 '
Model 4 Speedup
70 ' M/L Video patch by Tim Mann,
TRS-80 Computing, V1 N2 (CIE)
80 RESTORE :FOR I = 16478 TO 16507 :READ B :
POKE I,B :NEXT I
90 DATA 221,110,3,221,102,4,218,154,4,221,126,5,183,
40,1,119,121,254,32,218,6,5,254,128,210,166,4,
195,125,4
100 IF PEEK(84) < > 1 GOTO 130 ' Model III?
110 IF PEEK(15465) = 21 POKE 16414,94 :
POKE 16415,64 :GOTO 10
120 ' Set up display of cube
130 PRINT@ 15, STRING$(16,131); :
PRINT@ 320, STRING$(61,131);
140 PRINT@ 640, STRING$(61,131); :
PRINT@ 975, STRING$(16,131);
150 A$ = CHR$(191)
160 FOR I = 0 TO 896 STEP 64
170 PRINT@ I + 15, A$; :PRINT@ I + 30, A$;
180 IF I < 300 OR I > 600 GOTO 200
190 PRINT@ I, A$; :PRINT@ I + 45, A$; :
PRINT@ I + 60, A$;
200 NEXT I
210 PRINT@ 277,"Top"; :PRINT@ 582,"Left"; :
```

```
PRINT@ 596,"Front";
220 PRINT@ 611,"Right"; :PRINT@ 624,"Posterior";
230 PRINT@ 916,"Bottom"; :
PRINT@ 806,"Just a moment - -";
240 ' Put display positions of cubie faces into array P( )
250 N = 259
260 FOR I = 1 TO 6 :N = N - 177 :IF I = 2 THEN N = N + 290
270 IF I = 6 THEN N = N + 275
280 FOR J = 1 TO 9
290 P(K) = N :K = K + 1 :N = N + 4 :
IF INT(J/3) = J/3 THEN N = N + 52
300 NEXT J,I
310 FOR I = 0 TO 53 :PRINT@ P(I),"?"; :NEXT I
320 ' Entry of colors of all cubie faces by user
330 PRINT@ 738,"Enter color of each cubie face";
340 PRINT@ 802,"as a one-letter code. Be sure";
350 PRINT@ 866,"to correct errors. Move cursor";
360 PRINT@ 930,"backward or forward with < , > ";
370 I = 0 :NG = 0 :FOR K = 0 TO 25 :T(K) = 0 :NEXT K
380 GOSUB 2380
390 PRINT@ P(I),F$; :A$ = INKEY$:PRINT@ P(I),C$; :
IFA$ = ""GOTO 390
400 A = ASC(A$) :IF A = 44 OR A = 46 GOSUB 2320 :
GOTO 380
410 IF A > 95 THEN A = A - 32 :A$ = CHR$(A)
420 IF A > 64 AND A < 91 PRINT@ P(I), A$; :I = I + 1
430 IF I > 53 THEN I = 53:
PRINT@ 993," If all correct, press < ENTER > ";
440 IF A = 13 GOSUB 2910 ELSE 380
450 ' Put cubie colors into array C( ) and check for
errors
460 PRINT@ 802,"Check entry & get arrays ready";
470 FOR J = 0 TO 5
480 FOR K = 0 TO 8 :I = 9 * J + K
490 GOSUB 2380 :C(J,K) = N :N = N - 65 :
IF N > = 0 THEN T(N) = T(N) + 1
500 NEXT K,J :N = 0
510 FOR I = 0 TO 25 :IF T(I) = 9 THEN N = N + 1
520 IF T(I) < > 0 AND T(I) < > 9 THEN NG = 1
530 NEXT I :IF N < > 6 THEN NG = 1
540 IF NG PRINT@ 993,"Entry error. Please correct it";:
GOTO 330
550 ' Read Move Data into array M( )
560 FOR I = 0 TO 303 :READ M(I) :NEXT I
570 ' Select manual or automatic operation
580 GOSUB 2910 :
PRINT@ 738,"Do you want to enter the moves";
590 PRINT@ 802,"manually, or let the computer";
600 PRINT@ 866,"automatically select them?";
610 PRINT@ 933,"Choose: (M)anual/(A)uto "; :E$;
620 GOSUB 2410 :IF A = 65 GOTO 730
630 IF A < > 77 GOTO 610
640 ' Manual moves
650 GOSUB 2910 :
```



```

PRINT@ 738,"Twelve moves are possible:";
660 PRINT@ 802,"T + T- F + F- L + L- R + R- B + B-";
670 PRINT@ 866,"C+ C-, A (Top Front Left Right";
680 PRINT@ 930,"Bottom, whole Cube, or Auto)";
690 PRINT@ 997, CHR$(30); "Your Move? "; E$;
700 GOSUB 2410 :PRINT E$; :IF A=65 GOTO 730
710 B=A :GOSUB 2410 :GOSUB 2460 :GOTO 690
720 ' Set computer operation
730 GOSUB 2910 :
PRINT@ 738,"Do you want computer to stop";
740 PRINT@ 802,"& display each move (Y/N)? "; E$; :
GOSUB 2410
750 IF A=89 THEN DS=1
760 PRINT@ 866,"Do you want to print all the";
770 PRINT@ 930,"computer moves (Y/N)? "; E$; :
GOSUB 2410
780 IF A < > 89 GOTO 980
790 IF PEEK(14312) < 128 THEN PM=1 :GOTO 860
800 GOSUB 2910 :PRINT@ 738,"Printer is not ready.";
810 PRINT@ 802,"Abort (Y/N)? "; E$; :GOSUB 2410
820 IF A=89 GOTO 980
830 PRINT@ 866,"Get printer ready, hit <ENTER>";
840 GOSUB 2410 :GOTO 790
850 ' Print Cube
860 LPRINTTAB(5)"Rubik's Cube Solved"
870 LPRINTTAB(5)"=====
===== :LPRINT :LPRINT
880 LPRINTTAB(3)"Initial Status of Cube:" :LPRINT :J=0
890 FOR I=0 TO 5
900 IF I=0 OR I=5 LPRINTTAB(8);
910 FOR K=0 TO 2:LPRINT CHR$(C(I,J+K));" ";
NEXT K
920 LPRINT" ";
930 IF I=0 OR I=5 LPRINT :J=J+3 :
IF J<8 GOTO 900 ELSE LPRINT :J=0
940 IF I=4 LPRINT:J=J+3:
IF J<8 THEN I=1:GOTO 910 ELSE LPRINT :J=0
950 NEXT I
960 LPRINT :LPRINT :LPRINT"Moves used are:" :LPRINT
970 ' Computer moves
980 CM=1 :O$=" Top Edge"
990 IF C(0,1)=C(0,4) AND C(0,3)=C(0,4) AND
C(0,5)=C(0,4) AND C(0,7)=C(0,4) AND C(1,1)=C(1,4)
AND C(2,1)=C(2,4) AND C(3,1)=C(3,4) AND
C(4,1)=C(4,4) GOTO 1280
1000 FOR N=1 TO 4
1010 IF C(0,7)=C(0,4) AND C(2,1)=C(2,4) GOTO 1260
1020 IF C(0,7)=C(2,4) AND C(2,1)=C(0,4) THEN
C$="F-T+L-T-" :GOTO 1250
1030 IF C(0,5)=C(0,4) AND C(3,1)=C(2,4) THEN
C$="R-F-F-T+L-T-" :GOTO 1250
1040 IF C(3,1)=C(0,4) AND C(0,5)=C(2,4) THEN
C$="R-F-" :GOTO 1250
1050 IF C(4,1)=C(0,4) AND C(0,1)=C(2,4) THEN
C$="T+R-T-F-" :GOTO 1250

```

```

1060 IF C(0,1)=C(0,4) AND C(4,1)=C(2,4) THEN
C$="T+R-T-F-F-T+L-T-" :GOTO 1250
1070 IF C(0,3)=C(0,4) AND C(1,1)=C(2,4) THEN
C$="L+T+L-T-" :GOTO 1250
1080 IF C(1,1)=C(0,4) AND C(0,3)=C(2,4) THEN
C$="L+F+" :GOTO 1250
1090 IF C(2,5)=C(0,4) AND C(3,3)=C(2,4) THEN
C$="F-F-T+L-T-" :GOTO 1250
1100 IF C(3,3)=C(0,4) AND C(2,5)=C(2,4) THEN
C$="F-" :GOTO 1250
1110 IF C(3,5)=C(0,4) AND C(4,3)=C(2,4) THEN
C$="R+R+F-R+R+" :GOTO 1250
1120 IF C(4,3)=C(0,4) AND C(3,5)=C(2,4) THEN
C$="R+R+F-R+R+F-T+L-T-" :GOTO 1250
1130 IF C(4,5)=C(0,4) AND C(1,3)=C(2,4) THEN
C$="L+L+F+L+L+F-T+L-T-" :GOTO 1250
1140 IF C(1,3)=C(0,4) AND C(4,5)=C(2,4) THEN
C$="L+L+F+L+L+" :GOTO 1250
1150 IF C(1,5)=C(0,4) AND C(2,3)=C(2,4) THEN
C$="F+" :GOTO 1250
1160 IF C(2,3)=C(0,4) AND C(1,5)=C(2,4) THEN
C$="T+L-T-" :GOTO 1250
1170 IF C(2,7)=C(0,4) AND C(5,1)=C(2,4) THEN
C$="F+T+L-T-" :GOTO 1250
1180 IF C(5,1)=C(0,4) AND C(2,7)=C(2,4) THEN
C$="F+F+" :GOTO 1250
1190 IF C(5,5)=C(0,4) AND C(3,7)=C(2,4) THEN
C$="B-F+F+" :GOTO 1250
1200 IF C(3,7)=C(0,4) AND C(5,5)=C(2,4) THEN
C$="B-F+T+L-T-" :GOTO 1250
1210 IF C(5,7)=C(0,4) AND C(4,7)=C(2,4) THEN
C$="B+B+F+F+" :GOTO 1250
1220 IF C(4,7)=C(0,4) AND C(5,7)=C(2,4) THEN
C$="B+B+F+T+L-T-" :GOTO 1250
1230 IF C(1,7)=C(0,4) AND C(5,3)=C(2,4) THEN
C$="B+F+T+L-T-" :GOTO 1250
1240 IF C(5,3)=C(0,4) AND C(1,7)=C(2,4) THEN
C$="B+F+F+"
1250 GOSUB 2660
1260 IF N<4 THEN C$="C+" :GOSUB 2660
1270 NEXT N
1280 O$=" Top Corner"
1290 IF C(0,0)=C(0,4) AND C(0,2)=C(0,4) AND
C(0,6)=C(0,4) AND C(0,8)=C(0,4) AND C(1,0)=C(1,4)
AND C(1,2)=C(1,4) AND C(2,0)=C(2,4) AND
C(2,2)=C(2,4) AND C(3,0)=C(3,4) AND C(3,2)=C(3,4)
AND C(4,0)=C(4,4) AND C(4,2)=C(4,4) GOTO 1510
1300 FOR N=1 TO 4 :X=0
1310 IF C(0,8)=C(0,4) AND C(2,2)=C(2,4) AND
C(3,0)=C(3,4) GOTO 1490
1320 IF C(3,0)=C(0,4) AND C(2,2)=C(3,4) AND
C(0,8)=C(2,4) THEN C$="R-B+B+R+F+B+B+F-":
GOTO 1480
1330 IF C(2,2)=C(0,4) AND C(0,8)=C(3,4) AND
C(3,0)=C(2,4) THEN C$="F+B+B+F-R-B+B+R+":
GOTO 1480

```



```

1340 IF C(0,2) = C(0,4) AND C(3,2) = C(2,4) AND
C(4,0) = C(3,4) THEN C$ = "R + B-R-B-F + B + F-":
GOTO 1480
1350 IF C(3,2) = C(0,4) AND C(4,0) = C(2,4) AND
C(0,2) = C(3,4) THEN
C$ = "R + B-R-B-R-B + R + F + B + B + F-":GOTO 1480
1360 IF C(4,0) = C(0,4) AND C(0,2) = C(2,4) AND
C(3,2) = C(3,4) THEN C$ = "R + B-R-B-R-B-R + ":
GOTO 1480
1370 IF C(0,6) = C(0,4) AND C(1,2) = C(2,4) AND
C(2,0) = C(3,4) THEN C$ = "F-B + F + B + R-B-R + ":
GOTO 1480
1380 IF C(1,2) = C(0,4) AND C(2,0) = C(2,4) AND
C(0,6) = C(3,4) THEN C$ = "F-B + F + B + F + B + F-":
GOTO 1480
1390 IF C(2,0) = C(0,4) AND C(0,6) = C(2,4) AND
C(1,2) = C(3,4) THEN
C$ = "F-B + F + B + R-B + R + F + B + B + F-":GOTO 1480
1400 IF C(0,0) = C(0,4) AND C(4,2) = C(2,4) AND
C(1,0) = C(3,4) THEN C$ = "L-B + B + L + F + B + F-":
GOTO 1480
1410 IF C(1,0) = C(0,4) AND C(0,0) = C(2,4) AND
C(4,2) = C(3,4) THEN C$ = "L-B + B + L + R-B-R + ":
GOTO 1480
1420 IF C(4,2) = C(0,4) AND C(1,0) = C(2,4) AND
C(0,0) = C(3,4) THEN
C$ = "L-B + B + L + R-B + R + F + B + B + F-":GOTO 1480
1430 IF C(2,8) = C(0,4) AND C(5,2) = C(2,4) AND
C(3,6) = C(3,4) THEN C$ = "F + B + F-":GOTO 1480
1440 IF C(5,2) = C(0,4) AND C(3,6) = C(2,4) AND
C(2,8) = C(3,4) THEN
C$ = "R-B + R + F + B + B + F-":GOTO 1480
1450 IF C(3,6) = C(0,4) AND C(2,8) = C(2,4) AND
C(5,2) = C(3,4) THEN C$ = "R-B-R + ":GOTO 1480
1460 X = X + 1 :IF X < 4 THEN C$ = "B + ":GOSUB 2660:
GOTO 1430
1470 GOTO 2540
1480 GOSUB 2660
1490 IF N < 4 THEN C$ = "C + ":GOSUB 2660
1500 NEXT N
1510 O$ = "Vertical Edge"
1520 IF C(1,3) = C(1,4) AND C(1,5) = C(1,4) AND
C(2,3) = C(2,4) AND C(2,5) = C(2,4) AND C(3,3) = C(3,4)
AND C(3,5) = C(3,4) AND C(4,3) = C(4,4) AND
C(4,5) = C(4,4) GOTO 1730
1530 FOR N = 1 TO 4
1540 IF C(2,5) = C(2,4) AND C(3,3) = C(3,4) GOTO 1710
1550 IF C(2,5) = C(3,4) AND C(3,3) = C(2,4) THEN
C$ = "R-B + R + B + F + B-F-B + R-B + R + B + F + B-F-":
GOTO 1700
1560 IF C(3,5) = C(2,4) AND C(4,3) = C(3,4) THEN
C$ = "C-R-B + R + B + F + B-F-C + B-F + B-F-B-R-B + R + ":
GOTO 1700
1570 IF C(4,3) = C(2,4) AND C(3,5) = C(3,4) THEN
C$ = "C-R-B + R + B + F + B-F-C + R-B + R + B + F + B-F-":
GOTO 1700

```

```

1580 IF C(4,5) = C(2,4) AND C(1,3) = C(3,4) THEN
C$ = "C + F-B + F + B + L + B-L-B + C-B + F + B-F-B-R-
B + R + ":GOTO 1700
1590 IF C(1,3) = C(2,4) AND C(4,5) = C(3,4) THEN
C$ = "C + F-B + F + B + L + B-L-C-B-R-B + R + B + F + B-F-":
GOTO 1700
1600 IF C(1,5) = C(2,4) AND C(2,3) = C(3,4) THEN
C$ = "F-B + F + B + L + B-L-B + F + B-F-B-R-B + R + ":
GOTO 1700
1610 IF C(2,3) = C(2,4) AND C(1,5) = C(3,4) THEN
C$ = "F-B + F + B + L + B-L-B-B-R-B + R + B + F + B-F-":
GOTO 1700
1620 IF C(2,7) = C(2,4) AND C(5,1) = C(3,4) THEN
C$ = "B-R-B + R + B + F + B-F-":GOTO 1700
1630 IF C(5,1) = C(2,4) AND C(2,7) = C(3,4) THEN
C$ = "B + B + F + B-F-B-R-B + R + ":GOTO 1700
1640 IF C(3,7) = C(3,4) AND C(5,5) = C(2,4) THEN
C$ = "B + F + B-F-B-R-B + R + ":GOTO 1700
1650 IF C(5,5) = C(3,4) AND C(3,7) = C(2,4) THEN
C$ = "B-B-R-B + R + B + F + B-F-":GOTO 1700
1660 IF C(4,7) = C(3,4) AND C(5,7) = C(2,4) THEN
C$ = "F + B-F-B-R-B + R + ":GOTO 1700
1670 IF C(5,7) = C(3,4) AND C(4,7) = C(2,4) THEN
C$ = "B + R-B + R + B + F + B-F-":GOTO 1700
1680 IF C(1,7) = C(3,4) AND C(5,3) = C(2,4) THEN
C$ = "B-F + B-F-B-R-B + R + ":GOTO 1700
1690 IF C(5,3) = C(3,4) AND C(1,7) = C(2,4) THEN
C$ = "R-B + R + B + F + B-F-"
1700 GOSUB 2660
1710 IF N < 4 THEN C$ = "C + ":GOSUB 2660
1720 NEXT N
1730 X = 0 :N = 0 :O$ = "Bottom Corners"
1740 Y = 0 :Z = 0
1750 IF (C(2,8) = C(2,4) AND C(3,6) = C(3,4) AND
C(5,2) = C(5,4)) OR (C(2,8) = C(3,4) AND C(3,6) = C(5,4)
AND C(5,2) = C(2,4)) OR (C(2,8) = C(5,4) AND
C(3,6) = C(2,4) AND C(5,2) = C(3,4)) THEN Z = Z + 1:
Y = Y + 1
1760 IF (C(3,8) = C(3,4) AND C(4,6) = C(4,4) AND
C(5,8) = C(5,4)) OR (C(3,8) = C(4,4) AND C(4,6) = C(5,4)
AND C(5,8) = C(3,4)) OR (C(3,8) = C(5,4) AND
C(4,6) = C(3,4) AND C(5,8) = C(4,4)) THEN Z = Z + 1:
Y = Y + 2
1770 IF (C(4,8) = C(4,4) AND C(5,6) = C(5,4) AND
C(1,6) = C(1,4)) OR (C(4,8) = C(1,4) AND C(5,6) = C(4,4)
AND C(1,6) = C(5,4)) OR (C(4,8) = C(5,4) AND
C(5,6) = C(1,4) AND C(1,6) = C(4,4)) THEN Z = Z + 1:
Y = Y + 4
1780 IF (C(1,8) = C(1,4) AND C(2,6) = C(2,4) AND
C(5,0) = C(5,4)) OR (C(1,8) = C(2,4) AND C(2,6) = C(5,4)
AND C(5,0) = C(1,4)) OR (C(1,8) = C(5,4) AND
C(2,6) = C(1,4) AND C(5,0) = C(2,4)) THEN Z = Z + 1:
Y = Y + 8
1790 IF Z = 4 GOTO 1890
1800 X = X + 1 :IF X > 4 GOTO 2540
1810 IF Z < > 2 THEN C$ = "B + ":GOSUB 2660:

```



```

GOTO 1740
1820 IF Y=3 THEN
C$="C+R-B-R+F+B+F-R-B+R+B+B+":
GOTO 1880
1830 IF Y=5 THEN
C$="R-B-R+F+B+B+F-R-B+R+B+":GOTO 1880
1840 IF Y=6 THEN
C$="R-B-R+F+B+F-R-B+R+B+B+":GOTO 1880
1850 IF Y=9 THEN
C$="C+C+R-B-R+F+B+F-R-B+R+B-B-":
GOTO 1880
1860 IF Y=10 THEN
C$="C-R-B-R+F+B+B+F-R-B+R+B+":GOTO 1880
1870 IF Y=12 THEN
C$="C-R-B-R+F+B+F-R-B+R+B+B+":
1880 GOSUB 2660 :X=0
1890 IF C(5,0)=C(5,4) AND C(5,2)=C(5,4) AND
C(5,6)=C(5,4) AND C(5,8)=C(5,4) GOTO 2000
1900 IF C(5,0)=C(5,4) AND C(2,8)=C(5,4) AND
C(3,8)=C(5,4) AND C(4,8)=C(5,4) GOTO 1990
1910 IF C(5,0)=C(5,4) AND C(3,6)=C(5,4) AND
C(4,6)=C(5,4) AND C(1,6)=C(5,4) GOTO 1990
1920 IF C(1,8)=C(5,4) AND C(3,6)=C(5,4) AND
C(4,6)=C(5,4) AND C(4,8)=C(5,4) GOTO 1990
1930 IF C(2,6)=C(5,4) AND C(5,2)=C(5,4) AND
C(5,8)=C(5,4) AND C(4,8)=C(5,4) GOTO 1990
1940 IF C(2,6)=C(5,4) AND C(2,8)=C(5,4) AND
C(5,6)=C(5,4) AND C(5,8)=C(5,4) GOTO 1990
1950 IF C(2,6)=C(5,4) AND C(5,2)=C(5,4) AND
C(3,8)=C(5,4) AND C(5,6)=C(5,4) GOTO 1990
1960 IF C(1,6)=C(5,4) AND C(1,8)=C(5,4) AND
C(3,6)=C(5,4) AND C(3,8)=C(5,4) GOTO 1990
1970 X=X+1:IF X>4 GOTO 2540
1980 C$="C+":GOSUB 2660:GOTO 1900
1990 C$="R-B-R+B-R-B+B+R+B+B+":GOTO 1880
2000 O$=" Bottom Edges"
2010 X=0:Z=1:GOSUB 2940
2020 ON Z GOTO 2030, 2040, 2540, 2540, 2090
2030 C$="L-R+F+L+R-B+B+L-R+F+L+R-":
GOSUB 2660
2040 IF C(2,7)=C(2,4) OR C(5,1)=C(2,4) GOTO 2070
2050 X=X+1:IF X>4 GOTO 2540
2060 C$="C+":GOSUB 2660:GOTO 2040
2070 X=0:C$="L-R+F+L+R-B+B+L-R+F+L+R-":
GOSUB 2660
2080 Z=0:GOSUB 2940:IF Z<4 GOTO 2040
2090 IF C(5,1)=C(5,4) AND C(5,3)=C(5,4) AND
C(5,5)=C(5,4) AND C(5,7)=C(5,4) GOTO 2180
2100 IF C(1,7)=C(5,4) AND C(2,7)=C(5,4) AND
C(3,7)=C(5,4) AND C(4,7)=C(5,4) GOTO 2150
2110 IF C(5,3)=C(5,4) AND C(2,7)=C(5,4) AND
C(5,5)=C(5,4) AND C(4,7)=C(5,4) GOTO 2160
2120 IF C(1,7)=C(5,4) AND C(5,1)=C(5,4) AND
C(5,5)=C(5,4) AND C(4,7)=C(5,4) GOTO 2170
2130 X=X+1:IF X>4 GOTO 2540
2140 C$="C+":GOSUB 2660:GOTO 2100
2150 C$="L-R+F+F+L+R-B+B+L-R+F+L+R-B+B
+L-R+F+F+L+R-B-":GOSUB 2660:GOTO 2180
2160 C$="L-R+F+L+R-B+L-R+F+L+R-B+L-R+F
+F+L+R-B+L-R+F+L+R-B+L-R+F+L+R-B+B+":
GOSUB 2660:GOTO 2180
2170 C$="L-R+F+L+R-B-L-R+F-L+R-B-L-R+F
+F+L+R-":X=0:GOSUB 2660:GOTO 2040
2180 IF C(5,1)<>C(5,4) OR C(5,3)<>C(5,4) OR
C(5,5)<>C(5,4) OR C(5,7)<>C(5,4) GOTO 2540
2190 FOR K=0 TO 10
2200 GOSUB 2910:FOR J=0 TO 100:NEXT J
2210 PRINT@ 813,"HURRAY!!";:FOR J=0 TO 100:
NEXT J
2220 NEXT K
2230 PRINT@ 801,"The Cube is solved in";MN;"moves";
2240 IF PM LPRINT:LPRINT:
LPRINT"The Cube is solved in";MN;"moves.";
STRING$(5,10)
2250 FOR J=0 TO 2000:NEXT J
2260 PRINT@ 993,"Another Cube to solve (Y/N)? ";E$;:
GOSUB 2410
2270 IF A=89 RUN
2280 CLS:OUT 254,0
2290 END
2300 ' - - - - All Subroutines after this - - - -
2310 ' Move cursor with < or > during user entry
2320 M=A-45
2330 I=I+M:IF I<0 THEN I=0 ELSE
IF I>53 THEN I=53
2340 GOSUB 2380:PRINT@ P(I), F$;
2350 FOR J=0 TO 20:NEXT J:PRINT@ P(I), C$;
2360 P=PEEK(14368):IF P=16 OR P=64 GOTO 2330
ELSE RETURN
2370 ' Peek video
2380 N=PEEK(15360+P(I)):IF N<32 THEN N=N+64
2390 C$=CHR$(N):RETURN
2400 ' Get single character from keyboard
2410 A$=INKEY$:IF A$="" GOTO 2410
2420 A=ASC(A$):IF A>95 THEN A=A-32
2430 IF A<32 THEN A=32 ELSE IF A=59 THEN A=43
2440 A$=CHR$(A):PRINT A$;:RETURN
2450 ' Check move
2460 IF B=66 THEN Q=5:
IF A=43 OR A=45 GOTO 2580
2470 IF B=67 THEN IF A=43 OR A=45 GOTO 2610
2480 IF B=70 THEN Q=2:
IF A=43 OR A=45 GOTO 2580
2490 IF B=76 THEN Q=1:
IF A=43 OR A=45 GOTO 2580
2500 IF B=82 THEN Q=3:
IF A=43 OR A=45 GOTO 2580
2510 IF B=84 THEN Q=0:
IF A=43 OR A=45 GOTO 2580

```



```

2520 IF CM=0 RETURN
2530 ' Error routine
2540 GOSUB 2910 :OUT 254,0 :PRINT@ 814,"ERROR";
2550 IF PM LPRINT :LPRINT :LPRINT"ERROR" :PM=0
2560 GOTO 2540
2570 ' Select face to rotate
2580 IF A=45 THEN K=1 ELSE K=0
2590 L=24*(2*Q+K)+16 :K=K*8 :GOSUB 2780:
RETURN
2600 ' Rotate whole cube
2610 IF A=45 THEN K=0 ELSE K=1
2620 A=K :L=16+24*K :K=K*8 :Q=0 :R=1:
GOSUB 2780
2630 A=1-A :K=A :L=208+24*K :GOSUB 2810
2640 K=A :L=256+24*K :K=K*8 :Q=5 :R=0 :
GOSUB 2780 :RETURN
2650 ' Do move sequence specified by C$
2660 FOR M=1 TO LEN(C$) STEP 2
2670 A=ASC(MID$(C$,M+1,1)):
B=ASC(MID$(C$,M,1))
2680 MN=MN+1 :GOSUB 2710 :GOSUB 2460
2690 NEXT M :RETURN
2700 ' Display Computer moves
2710 GOSUB 2910 :Z=A :PRINT@ 743, O$;:
IF N PRINT N;
2720 PRINT@ 870,"Current move is: "; CHR$(B);
CHR$(A);
2730 PRINT@ 936,"Move number: "; MN;
2740 IF PM LPRINT CHR$(B); CHR$(A); " ";
2750 IF DS PRINT@ 994,"Press < ENTER > to
continue"; :GOSUB 2410
2760 PRINT@ 994, CHR$(30); :A=Z :RETURN
2770 ' Move Cubie faces
2780 FOR J=0 TO 1 :T=C(Q,M(K))
2790 FOR I=0 TO 2 :C(Q,M(K))=C(Q,M(K+1)):
K=K+1 :NEXT I
2800 C(Q,M(K))=T :K=K+1 :NEXT J
2810 FOR J=0 TO 2 :T=C(M(L),M(L+1))
2820 FOR I=0 TO 2:
C(M(L),M(L+1))=C(M(L+2),M(L+3)):L=L+2:NEXT I
2830 C(M(L),M(L+1))=T :L=L+2
2840 NEXT J :IF R RETURN
2850 ' Display colors of cubie faces
2860 FOR I=0 TO 5 :K=9*I
2870 FOR J=0 TO 8
2880 PRINT@ P(K+J), CHR$(C(I,J));
2890 NEXT J,I :RETURN
2900 ' Clear text from display
2910 FOR I=737 TO 993 STEP 64 :
PRINT@ I, CHR$(30);:NEXT I
2920 RETURN
2930 ' Check No. of bottom edges in position
2940 IF C(1,7)=C(1,4) OR C(5,3)=C(1,4) THEN
Z=Z+1

```

```

2950 IF C(2,7)=C(2,4) OR C(5,1)=C(2,4) THEN
Z=Z+1
2960 IF C(3,7)=C(3,4) OR C(5,5)=C(3,4) THEN
Z=Z+1
2970 IF C(4,7)=C(4,4) OR C(5,7)=C(4,4) THEN
Z=Z+1
2980 RETURN
2990 ' Data for Moves
3000 DATA 0,6,8,2,1,3,7,5, 0,2,8,6,1,5,7,3
3010 DATA 1,0,2,0,3,0,4,0, 1,1,2,1,3,1,4,1, 1,2,2,2,3,2,4,2
3020 DATA 4,2,3,2,2,2,1,2, 4,1,3,1,2,1,1,1, 4,0,3,0,2,0,1,0
3030 DATA 0,0,4,8,5,0,2,0, 0,3,4,5,5,3,2,3, 0,6,4,2,5,6,2,6
3040 DATA 0,0,2,0,5,0,4,8, 0,3,2,3,5,3,4,5, 0,6,2,6,5,6,4,2
3050 DATA 0,6,1,8,5,2,3,0, 0,7,1,5,5,1,3,3, 0,8,1,2,5,0,3,6
3060 DATA 0,6,3,0,5,2,1,8, 0,7,3,3,5,1,1,5, 0,8,3,6,5,0,1,2
3070 DATA 0,2,2,2,5,2,4,6, 0,5,2,5,5,5,4,3, 0,8,2,8,5,8,4,0
3080 DATA 0,2,4,6,5,2,2,2, 0,5,4,3,5,5,2,5, 0,8,4,0,5,8,2,8
3090 DATA 1,3,4,3,3,3,2,3, 1,4,4,4,3,4,2,4, 1,5,4,5,3,5,2,5
3100 DATA 1,3,2,3,3,3,4,3, 1,4,2,4,3,4,4,4, 1,5,2,5,3,5,4,5
3110 DATA 1,6,4,6,3,6,2,6, 1,7,4,7,3,7,2,7, 1,8,4,8,3,8,2,8
3120 DATA 1,6,2,6,3,6,4,6, 1,7,2,7,3,7,4,7, 1,8,2,8,3,8,4,8

```

Integer Variables:

A	ASCII value of user entry, temporary storage of "K".
B	ASCII value of previous user entry.
CM	Computer move flag (set if computer selects moves).
DS	Display stop flag (set if stop after each move).
I,J,K,L,M	Integer counters.
MN	Move number.
N	Integer counter.
NG	"No good" flag for testing user entries.
PM	"Print move" flag (set if moves printed on printer).
Q	Face of cube to be rotated.
R	"Return" flag (display only when cube rotation done).
T	Temporary cubie face storage when they are moved.
X	Excess move counter for error routines.
Y	Bottom corner pattern counter.
Z	Bottom corner and edge pattern counter, temporary storage of "A".

Arrays:

C(5,8)	Individual cubie faces on cube; six cube faces with nine cubie faces each.
M(303)	The list of cubie face moves when any face of the cube is rotated (in to, from notation).
P(53)	Positions for display of each cubie face on the video display screen.
T(25)	Test array, to check that exactly nine cubie faces of each of six colors were entered.

String Variables:

A\$	Graphic block for video display, user entry.
C\$	Character for flashing entry display. change string for cube face rotations.
E\$	Entry display string (underline plus backspace).
F\$	Flashing cursor for user entry (graphics block).
O\$	Group of cubies being operated on.

HINTS & TIPS

SORTING DATA

by Frank Tipps

While there are many ways to sort numeric and string data, the most common methods are probably the shell-sort and the bubble-sort. Below are examples of both methods for sorting numeric data. Both programs will work on Model I/III and 4, and could easily be converted to sort string data.

```
1 CLS
10 PRINT"SHELL SORT PROGRAM"
100 DIM A(300)
105 PRINT
106 PRINT"This program can sort up to 300 entries."
108 PRINT
110 PRINT"Enter 99999 as data to terminate data entry."
115 PRINT
120 K=0
125 K=K+1
130 PRINT"A(";K;") = ";
135 INPUT"";TP
140 IF TP=99999! THEN K=K-1:GOTO 150
145 A(K)=TP:GOTO 125
150 N=K
180 GOSUB 500
190 PRINT:PRINT
195 PRINT"          Sorted list":PRINT
219 K=0
220 FOR I=1 TO N
221 K=K+1
230 PRINT A(I)
231 IF K>=20 THEN INPUT"Touch <return> .";W$:
K=0:CLS
240 NEXT I
250 INPUT"Touch <return> to clear screen and
terminate run.";W$
255 CLS:END
500 D=1
510 D=2*D
520 IF D<=N THEN 510
530 D=INT((D-1)/2)
540 IF D=0 THEN 700
550 FOR I=1 TO N-D
560 J=I
570 L=J+D
580 IF A(J)<=A(L) THEN 640
590 X=A(J)
600 A(J)=A(L)
610 A(L)=X
620 J=J-D
```

```
630 IF J>0 THEN 570
640 NEXT I
650 GOTO 530
700 RETURN
```

```
10 'bubble sort
15 CLS
20 DIM X(300)
30 INPUT"No of data points = ? ";N
40 FOR I=1 TO N
60 PRINT"X("I") = ";:INPUT X(I)
80 NEXT I
90 NM=N-1
100 FOR I=1 TO NM
110 IP=I+1
120 FOR J=IP TO N
130 IF X(I)<X(J) THEN 200
140 H1=X(I)
150 X(I)=X(J)
160 X(J)=H1
200 NEXT J
210 NEXT I
215 PRINT
216 NP=0
220 FOR I=1 TO N
225 NP=NP+1
230 PRINT X(I)
240 NEXT I
250 END
```

WHEN IS EASTER?

by Chris Spratt

It is common knowledge that on the first Tuesday of February, the groundhog looks up out of his hole and if he sees his shadow, we then have six more weeks until Easter. Well, something like that!

Anyway, below is a Basic program that will correctly determine on which date Easter Sunday will fall in any given year. Type the code into Model 4 Basic and you'll never again have to rely on a groundhog to know when to color the eggs!

```
10 CLS
20 PRINT@(3,32),"A SPECIAL PROGRAM"
30 PRINT@(6,21),"***** THE DATE OF EASTER
SUNDAY *****"
40 PRINT@(9,32),"BY CHRIS SPRATT"
50 PRINT@(15,26),"PRESS ANY KEY TO CONTINUE ";
60 IF INKEY$ = "" THEN 60
```



```

70 CLS
80 ' -- THE DETERMINATION OF EASTER --
90 ' -- BY CHRIS SPRATT --
100 ' -- TRS-80 MODEL 4
110 PRINT@(3,28),"YEAR OF INTEREST ";:INPUT Y
120 PRINT CHR$(15)
130 A = INT(((Y/19 - INT(Y/19)) * 19) + .5)
140 B = INT(Y/100)
150 C = INT(((Y/100 - INT(Y/100)) * 100) + .5)
160 D = INT (B/4)
170 E = INT (((B/4 - INT(B/4)) * 4) + .5)
180 G = INT (((8 * B) + 13)/25)
190 J = B - D - G
200 IF Y < 1583 THEN J = 30
210 H1 = INT((((11 * J) - 4)/30)
220 H2 = INT (((7 * A) + H1 + 6)/11)
230 HH = 19 * A + J + 15 - H2
240 H = INT (((HH/29 - INT (HH/29)) * 29) + .5)
250 I = INT (C/4)
260 K = INT((((C/4 - INT (C/4)) * 4) + .5)
270 LL = 32 + 2 * E + 2 * I - K - H
280 IF Y < 1583 THEN LL = 34 + B + 2 * I - H - K
290 L = INT (((LL/7 - INT (LL/7)) * 7) + .5)
300 N = INT ((90 + H + L)/25) ]
310 PP = 19 + H + L + N
320 P = INT (((PP/32 - INT (PP/32)) * 32) + .5)
330 IF N = 4 THEN N$="APRIL"
340 IF N = 3 THEN N$="MARCH"
350 CLS
360 PRINT@(3,26),"EASTER SUNDAY IN" ;Y;"IS ...."
370 N$=N$+STR$(P)
380 PO=((80-LEN(N$))/2)
390 PRINT@(6,PO),N$
410 PRINT@(15,25),"DO YOU WANT ANOTHER YEAR
Y/N ";CHR$(14);
420 A$=INKEY$:IF A$="" THEN 420
430 IF A$ = "Y" OR A$="y" THEN CLS: GOTO 110
440 IF A$ = "N" OR A$="n" THEN CLS:END
450 GOTO 420

```

UPGRADE YOUR 4P DOUBLE-SIDED DRIVES

by Kelly R. Bates

You may want to modify your 4P for double-sided drives. My new procedure only requires the drives and the proper DOS, as far as new stuff.

In January 1986, 80 Micro published an article on how to upgrade your 4P. Several of us did it way back then. But there is a cheaper and faster way to do it, and a lot simpler, if I can explain it here. I was helping a new member of the Oklahoma City club to upgrade his drives and, just to be sure I was doing it properly, I reread the article.

On page 42, the article states "the original internal drive cable may have reusable connectors. You can use these connectors successfully if you dismantle and reassemble them correctly." The article then explains how to put the connectors back on the flat ribbon backwards.

From those few words I reexamined the procedure. Now, why not just unplug the cable and the plug the whole thing in backwards? Hey, hey, hey! It works! In other words, use the original cable as is - except for removing the little guides, 1 in each 34 pin edge card connector.

So the new procedure should be as follows:

- Open the 4P by removing 6 screws. 2 on the handle and 2 on each end.
- Remove the drive assembly (Take out about 6 more screws). Note how the flat ribbon drive cable is installed and remove it.
- Remove the drives from the drives cage.

On top of the drives cage are screw holes.

- Elongate 4 of them toward the rear of the computer about a quarter of an inch. Your new drives in most cases are a bit longer than the originals.
- Configure the new drives as Zero and One, and install them into the drives cage, but leave the screws loose so they can be tightened last.
- Install the drives cage in the computer. After the cage is installed, slide the drives forward (flush with the front bezel) and tighten the drive screws on top of the drives cage. So far so good. This procedure deviates slightly from the original article, but will work with any slim line drives. In the rare event that your new drives are shorter than the old RS drives, elongate your screw holes toward the front of the computer.

Now examine the daisy chain drive cable. The 2 edge card connectors have guides at one end. The guides were originally used to insure that the assemblers did not connect them backwards. Since we WANT to connect them backwards

- remove the connector guides. Use a lot of care so as not to damage the connector pins inside the connector. (This is the hardest part of this procedure, so take your time and do it right or you'll have to replace the connectors with that high dollar stuff!).
- Install the cable backwards. The end 34 pin edge connector goes on Drive 1, and the 2nd 34 pin edge card goes on drive 0, and the 34 pin socket connector goes where it was, except reversed (since the socket connector does not have a guide, it will plug in either way).
- Put the computer back together. The conversion is now complete.

HUNTING FOR BURIED TREASURE

PATCHER4

a Basic patching utility for Model 4

by Lance Wolstrup



I have been sitting on this idea for a couple of years. Then, and a few times since, Henry Herdegen asked me to 'fix' the Model 4 Job Control Language (JCL) so it would behave the same as JCL on Model III. The problem is that Model 4 JCL does not allow return to a calling Basic program. Oh, it will return to Basic, all-right - but it just sits there and waits at the Ready prompt.

I finally had some time, so I pulled the 'Source' off my bookshelf and began to investigate. After a few hours of intense study, I made up my mind - *no way was I going to mess with JCL.*

The original reason that Henry wanted to modify JCL was to allow the writing of a Model 4 patch program, similar to the one he has available for the Model III (TRSDOS 1.3). Hmmmm!!! Why not do it another way?

After some thinking, I decided to write a 'patch reader' program. That is, a program that will read a plain ASCII text file and then execute the patch commands contained therein. It didn't seem overly difficult, read a sequential file (the ASCII file), and use it to make modifications to a random access file (the file to be patched).

DESIGNING THE PROGRAM

OK, let's start at the beginning. First, the user needs to be prompted for which ASCII text file to use (we'll call that the 'script' file). This prompt needs to be error trapped - it is possible that the user will answer with something other than a correct filename.

The best way to test if a file exists (in Basic), is to try to open it for INPUT as a sequential file. If that produces an error, the file either does not exist, or it is bad. Either way, the ON ERROR GOTO statement will trap and recover the error. Note: Do not test the existence of a file by opening a random access file. Why? Because, if it doesn't exist - it will be created.

With the script file now open, the program must now read (and understand) the commands given. We will need a uniform way of communication with the script reader, so now is the time to invent some rules and syntax.

1. The very first line of the script list must contain the *name* of the file to patch.

2. The second line of the script list must contain a semicolon (;) followed by a brief explanation of what the patch does.

3. The third line of the script list must have the *record* and *offset* within the record where the patch begins. The syntax will be *record,offset*

4. The fourth line of the script list will contain the patch itself - that is, the new code. The bytes must be separated by spaces.

5. The fifth line of the script list will contain the code that is currently at the place where the patch will go. This is done for safety, and to be compatible with 'D and F' patch method. The bytes must be separated by spaces, and should thus line up exactly underneath the new patch code.

6. The fourth and fifth lines must be of equal length. That is, the new code in line 4 must have the exact same number of bytes as the old code in line 5.

7. To continue the patch, or make another patch in the SAME file, we will just go back to step 2

8. To install a patch on a different file, we will use the colon (:) to indicate that the work on the previous file is done, and to begin again at step 1.

Also, we will need to trap the possible errors coming from the script file, such as naming a file that doesn't exist, or lines 4 and 5 not being the same length.

This framework would work nicely - except for one thing!

THE MISSING BYTE

The information stored in a 'Random Access' file is accessed through the 'FIELD' you set up after you opened the file. You then use the GET command to retrieve a particular record into the FIELD spec. For example, you

might issue the command: GET 2,1. This tells Basic to retrieve record number 1, and use FIELD #2 to do it - and here is the rub.....

How many bytes do we have in a record? If you said 256, stand up and take a bow. That is correct. Each record has 256 bytes.

Now, what is the largest FIELD you can create? You got it. 255 bytes. *What happened to the last byte?*

The explanation might be that the FIELD is assigned to a string - and, as we all know, a string can only accomodate 255 bytes. Note that this is only a guess.

This Basic limitation set me back for a few minutes. I wanted to be able to access all 256 bytes in one segment. After all, what would I do if the patch involved the 256th character? Hmmm!

Then it came to me. The GET command obviously pulls in the entire 256 byte disk sector the record is stored in and places it somewhere in memory. The fact that FIELD only allows access to the first 255 bytes becomes irrelevant if we can find just where in memory the record is placed. Now, how do we find where a string is stored in memory? We use VARPTR. *I wonder if VARPTR will find the beginning byte of the FIELD?*

After writing a short, simple program the answer was a resounding YES. VARPTR behaves just as if it was working on a real string. The code (see line 80 in listing)

```
FI = PEEK(VARPTR(FI$) + 2) * 256 +  
PEEK(VARPTR(FI$) + 1)
```

does indeed find the very first byte of the buffer that holds the information FIELDed by FI\$ and, more importantly, all 256 bytes of the record are available starting with this memory location. In other words, FI+0 is the first byte, FI+1 is the second byte....all the way to FI+255, which is the 256th byte.

With this problem solved, I began coding PATCHER4/BAS and, as you will see, for what it does, it is an amazingly short program.

PATCHER4/BAS

```
1 'PATCHER4/BAS  
2 'copyright 1993 by Lance Wolstrup  
3 'all rights reserved  
4 'a PATCHing utility for Model 4
```

Dimension the D line and F line arrays to accomodate 80 patch bytes on a line - this should never ever be necessary - but, just in case. Also, set screen width to 80 characters and turn off the cursor. Erase screen, display program name and the copyright information. Then jump to the beginning of the program in line 100.

```
10 DIM DP(80),FP(80):  
SW = 80:A$ = CHR$(15):GOSUB 20
```

```
11 CLS:V = 0:A$ = "P A T C H E R 4":GOSUB 21:  
V = 1:A$ = "a BASIC patching utility for Model 4":  
GOSUB 21:V = 2:A$ = "Copyright (c) 1993 by Lance  
Wolstrup - all rights reserved":GOSUB 21:  
V = 3:A$ = STRING$(SW,140):GOSUB 20  
19 GOTO 100
```

The PRINT subroutines

enter routine with

V = vertical cursor position and A\$ = text

line 20 for left justified text

line 21 for centered text

line 22 for right justified text

```
20 H = 0:GOTO 23  
21 H = INT((SW-LEN(A$))/2):GOTO 23  
22 H = SW  
23 PRINT@SW*V + H,A$;:RETURN
```

The INPUT (INKEY\$) subroutine

enter with

V = vertical cursor position, H = horizontal cursor position

ML = maximum length of input

return with

I\$ = input

FL = 0 if input terminated with <ENTER>

FL = 1 if input terminated with <ESC> (shift-up arrow)

```
30 I$ = "":FL = 0:L = 0:A$ = STRING$(ML,46):GOSUB 23:  
A$ = CHR$(14):GOSUB 23  
31 A$ = INKEY$:IF A$ = "" THEN 31  
32 IF A$ = CHR$(13) THEN 39  
33 IF A$ = CHR$(27) THEN FL = 1:GOTO 39  
34 IF A$ = CHR$(8) AND L = 0 THEN 31  
35 IF A$ = CHR$(8) THEN L = L-1:H = H-1:  
I$ = LEFT$(I$,L):A$ = CHR$(46):GOSUB 23:  
A$ = "":GOSUB 23:GOTO 31  
36 IF A$ < CHR$(32) THEN 31  
37 IF L = ML THEN 31  
38 GOSUB 23:L = L + 1:H = H + 1:I$ = I$ + A$:GOTO 31  
39 A$ = CHR$(31):GOSUB 23:A$ = CHR$(15):GOTO 23
```

subroutine to find the memory location of the first byte of the 256 byte buffer filled with the GET command.. Must be FIELDed as FIELD buffernum,255 as FI\$

```
80 FI = PEEK(VARPTR(FI$) + 2) * 256 +  
PEEK(VARPTR(FI$) + 1):RETURN
```

subroutine to freeze display after error is trapped

```
90 V = V + 2:A$ = "Press <ENTER> to continue":  
GOSUB 21:H = H + LEN(A$) + 1  
91 ML = 1:GOTO 30
```


The program begins here.

Prompt for name of patch ascii file and let user respond

```
100 V=5:A$=CHR$(31):GOSUB 20:A$="Name of
patch script:":GOSUB 20:H=H+LEN(A$)+1
110 ML=23:GOSUB 30
```

If user terminated input with <ESC> erase screen and end program. If user did not supply a name by just pressing <ENTER> go back and prompt again. If user did type a name, transfer to NM\$ and place cursor on line 8.

```
120 IF FL THEN CLS:END ELSE IF I$="" THEN 110
ELSE NM$=I$:V=8
```

ok, the user supplied a name. Begin error trap and then check if file exists

```
130 ON ERROR GOTO 1000
140 OPEN"I",1,NM$
```

if we get here, the file exists, so turn off error trap and store line 1 of the patch script in NM\$

```
150 ON ERROR GOTO 0
160 INPUT#1,NM$
```

Begin error trap and check if file supplied by patch script exists.

```
170 ON ERROR GOTO 1000
180 OPEN"I",2,NM$
```

file exists, so turn off error trap and close the sequential file.

```
190 ON ERROR GOTO 0
200 CLOSE 2
```

erase everything from line 8 on down, and inform user that program is opening the requested file.

```
210 V=8:A$=CHR$(31):GOSUB 20:
A$="Opening "+NM$:GOSUB 20
```

open requested file as random access and field maximum length

```
220 OPEN"R",2,NM$
225 FIELD 2,255 AS F$
```

make sure that the R\$ array will always begin at subscript 1

```
230 CT=1
```

check for end of file

```
240 IF EOF(1) THEN 360
```

check for : (colon) - if found, close file and go back and check if new file is requested by script

```
250 INPUT#1,R$(CT):
IF R$(CT)=CHR$(58) THEN CLOSE 2:GOTO 160
```

check for ; (semi-colon) - if so, we have a patch description - display it - then go back and check for next line of the patch script

```
255 IF LEFT$(R$(CT),1)=CHR$(59) THEN V=V+2:
A$=MID$(R$(CT),2):GOSUB 21:CT=CT+1:GOTO 250
```

retrieve record number and convert it to the way Basic wants to see it (+1)

```
257 R=VAL("&H"+R$(CT))+1:CT=CT+1
```

retrieve starting position of the patch within the record and convert it to the way Basic wants to see it (+1)

```
260 INPUT#1,R$(CT):P=VAL("&H"+R$(CT))+1:
CT=CT+1
```

retriev the 'D' line. That is, the new code to be patched to the file. Then store the bytes in the DP(X) array. We can accomodate up to 80 bytes.

```
270 INPUT#1,R$(CT):DP=LEN(R$(CT)):BY=1:
FOR X=1 TO DP STEP 3:
DP(BY)=VAL("&H"+MID$(R$(CT),X,2)):BY=BY+1:
NEXT:CT=CT+1
```

retrieve the 'F' line. That is, the bytes that currently exists in the file to be patched. Then store the bytes in the FP(X) array. We can accomodate up to 80 bytes.

```
280 INPUT#1,R$(CT):FP=LEN(R$(CT)):BY=1:
FOR X=1 TO FP STEP 3:
FP(BY)=VAL("&H"+MID$(R$(CT),X,2)):BY=BY+1:
NEXT:CT=CT+1
```

display error and abort current patch if the 'D' and 'F' lines do not contain the same number of bytes.

```
285 IF DP < > FP THEN V=V+2:
A$="Error in "+NM$+" script":GOSUB 21:GOTO 350
```

no error, inform user that record will be retrieved

```
287 V=V+2:
```


A\$="Positioning to record"+STR\$(R-1):GOSUB 21

retrieve the requested record and let the subroutine in line 80 find the the beginning of the FIELD buffer

290 GET 2,R

300 GOSUB 80:BY=BY-1

check if the bytes supplied in the 'F' line match the bytes found in the buffer. If not, FL will be set to 1. An error message will be displayed and the current patch will abort. If FL=0, then the bytes matched and we prompt the user if this patch should be installed and we then get the user response

310 FL=0:FOR X=1 TO BY:

IF PEEK(FI+P-1+X-1)<>FP(X) THEN FL=1

320 NEXT V=V+1:

IF FL THEN A\$="(F)ind bytes not found":GOSUB 21:

GOTO 350 ELSE A\$=STR\$(BY)+" patch-bytes found":

GOSUB 21:V=V+2:A\$="Do you wish to install this

patch (Y/N)":GOSUB 21:H=H+LEN(A\$)

325 ML=1:GOSUB 30

check if <ESC> was pressed. If so, close all files and go back and prompt for new patch script. If the user answers 'Y', we jump to line 330 where the patch will be processed. If the answer is 'N', we will the attempt to process the next patch request.

327 IF FL THEN CLOSE: GOTO 100 ELSE

IF I\$="Y" OR I\$="y" THEN 330 ELSE

IF I\$="N" OR I\$="n" THEN V=8:

A\$=CHR\$(31):GOSUB 20:GOTO 230 ELSE 325

write the new patch bytes to their appropriate positions in the FIELD buffer, and then write the record to disk.

330 FOR X=1 TO BY:POKE FI+P-1+X-1,DP(X):NEXT

340 PUT 2,R

inform user that patch was successful. Then attempt to process the next patch request.

345 V=V+1:A\$="Patch installed":GOSUB 21

350 GOSUB 90:V=9:A\$=CHR\$(31):GOSUB 20:

GOTO 230

close all files and end with a courteous remark.

360 CLOSE

370 V=V+2:A\$="End of run":GOSUB 20:

V=V+1:A\$="Thank you for using PATCHER4":

GOSUB 20

390 END

error trap routine

1000 A\$="Unable to open "+NM\$:GOSUB 21:GOSUB 90

1010 RESUME 1020

1020 ON ERROR GOTO 0:CLOSE:GOTO 100

Now that we have the program, let me close out this article by giving a couple of examples. Note that all patches are directed to files on drive :1, and all lines are terminated with <ENTER>. Both patch scripts were created with TED. Happy patching.

PATCH1/SCR

SYS1/SYS.SYSTEM6:1

;replace REMOVE command with KILL

02,15

4B 49 4C 4C 20 20

52 45 4D 4F 56 45

:

BOOT/SYS.SYSTEM6:1

;boot with clock ON

05,53

00 00

28 04

:

SYS0/SYS.SYSTEM6:1

;boot up in upper case

00,80

20

00

PATCH2/SCR

SYS1/SYS.SYSTEM6:1

;replace REMOVE with KILL

02,15

4B 49 4C 4C 20 20

52 45 4D 4F 56 45

;replace DIR command with D

01,AA

20 20

49 52

:

;replace CAT command with C

01,76

20 20

41 54

ITEMS OF INTEREST

USERS GROUPS

SADTUG

The San Diego

TRS-80 Users Group

meets the first Sunday of each month at 2pm.

The oldest TRS user group in the US,
established in 1977,
two months after the Mod 1 exam out.

Contact Mike Baldwin

(619) 583-1578

VTUG

Valley TRS-80 Users Group

meets the third Friday of each month
at 7:30 pm at the Learning Tree University.
DeSoto & Knapp Streets, Chatsworth, CA

Contact Lance Wolstrup

(818) 716-7154

VTHG

Valley TRS-80 Hackers Users Group

Los Angeles, CA

meets the first Friday of each month at 7:00 pm

for more information

Contact Roy Beck

(213) 664-5059

OCTUG

Orange County TRS-80 Users Group

meets the third Sunday of each month at 11:00 am
in the Community Hall at the City Shopping Center
Orange, CA

Complete support for TRS-80's Model I through 4

Contact Mike Lingo

(714) 545-1059 (before 3 pm)

SAGATUG

San Gabriel Tandy Users Group

meets the second Friday of each month at 7:00 pm
at the Arcadia Park Senior Citizen's Center
405 South Santa Anita Avenue
Arcadia, CA

THE CLUB FOR TRSDOS, MS-DOS, CP/M AND LAPTOP COMPUTERS

Large public domain libraries
remote memberships available

Contact Barbara Beck

(213) 664-5059

Mid-Cities Tandy Radio Shack Users Group

**MCTRUG supports
all of the Tandy
computers plus IBM
compatibles. We
have software
available for TRS-80
Models I, III, 4,
Color Computers,
Model
2000 and
MS-DOS.**



**Write us
for more
information.
Please include your
name, mailing
address, computer
model and which
Disk Operating
System you use.**

Write to:

**MCTRUG
P.O. Box 171566
Arlington, TX 76003**

PUBLIC DOMAIN GOOD GAMES FOR MODEL I/III.

GAMEDISK#1: amazin/bas, blazer/cmd, breakout/cmd, centipede/cmd, elect/bas, madhouse/bas, othello/cmd, poker/bas, solitr/bas, towers/cmd

GAMEDISK#2: cram/cmd, fallen/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 (U.S.)

or get all 4 disks for \$16.00 (U.S.)

TRSTimes - PD GAMES

5721 Topanga Canyon Blvd. #4
Woodland Hills, CA. 91364

TRSTimes on DISK #11

Issue #11 of TRSTimes on DISK is now available, featuring the programs from the Jan/Feb, Mar/Apr, May/Jun 1993 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, & 10 are still available at the above prices

MISOSYS, Inc.

CLOSEOUT SALE ON SOFTWARE: NO RETURNS; ALL SALES FINAL

AFM: Auto File Manager data base (3)	\$10.00 D
BackRest for hard drives (3&4)	\$10.00
BSORT (3,4)	\$5.00
Cornsoft Group 5-Game Disk (3)	\$20.00
CP/M (MM) Hard Disk Drivers	\$10.00 B
CON80Z (3,4)	\$5.00
diskDISK (3,4)	\$10.00
DoubleDuty (4)	\$25.00
DSM51 / DSM4 (3,4)	\$10.00
DSMBLR / PRO-DUCE (3,4)	\$10.00
EDAS / PRO-CREATE (3,4)	\$10.00 D
Filters: Combined I & II (3)	\$5.00 B
GO:Maintenance (4)	\$15.00 B
GO:System Enhancement (4)	\$15.00 B
GO:Utility (4)	\$15.00 B
Hardware Interface Kit (4)	\$5.00
HartFORTH (3,4)	\$10.00 B
Kim Watt's Game Hits (3)	\$10.00
Lair of the Dragon (3&4)	\$10.00
Lance Miklus' Game Hits (3)	\$15.00
LDOS 5.3.1 Mod1 Upgrade kit (1)	\$20.00 B
LDOS 5.3.1 Mod3 Upgrade Kit (3)	\$20.00 B
Leo Cristopherson's Game Disk (3)	\$10.00
LS-DOS 6.3.1 Upgrade Kit (4)	\$20.00 B
LS-DOS 6.3.1 Upgrade kit (2)	\$25.00 B

LED / LS-LED (3,4)	\$5.00
LS-Host/Term (4)	\$10.00
LS-UTILITY (4)	\$10.00
MC / PRO-MC (3,4)	\$79.95 D
Mister ED (4)	\$10.00 B
MRAS / PRO-MRAS (3,4)	\$30.00 D
PowerDot (Epson or Tandy) (3)	\$5.00
PowerDraw (3)	\$5.00
PowerDriver Plus (Epson) (3)	\$5.00
PowerMail Plus (3,4)	\$15.00 D
PowerMail Plus TextMerge (3,4)	\$5.00
PowerScript (3,4)	\$10.00 B
PRO-WAM (4)	\$50.00 D
PRO-WAM Toolkit (4)	\$15.00 B
QuizMaster (3)	\$5.00
RATFOR (4)	\$10.00 D
SuperUtilityPlus (3,4)	\$15.00 D
Supreme HD Driver (3&4)	\$15.00
TBA / LS-TBA (3,4)	\$5.00 D
The Gobbling Box (3&4)	\$10.00
THE SOURCE 3-Volume Set	\$10.00 D
Toolbox/Toolbelt (3,4)	\$10.00 B
UNREL-T80 (3&4)	\$5.00
UTILITY-I (3)	\$5.00
XLR8er Software Interface Kit (4)	\$5.00 B

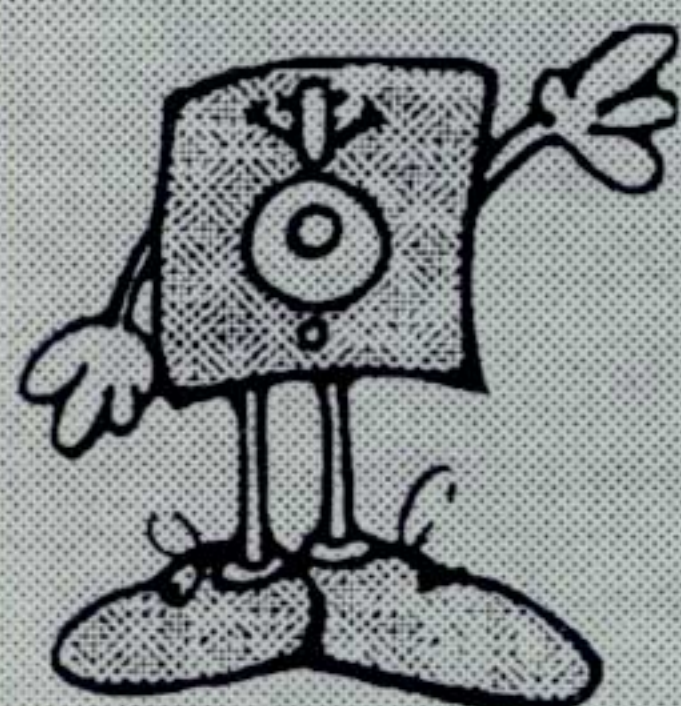
MISOSYS, Inc. Sterling, VA 20167-0239
P.O. Box 239 703-450-4181

The Fine Print: Numbers in () indicate computer support - please specify TRS-80 Model. Freight codes: A = \$3.50; B = \$4.00; C = \$4.50; D = \$5.00; All unmarked are \$3.00 each; Canada/Mexico add \$1 per order; Foreign use US rates times 3 for air shipment. Virginia residents add 4.5% sales tax. We accept MasterCard and VISA; Checks must be drawn on a US bank. COD's are cash, money order, or certified check; add \$5 for COD.

**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). You may use the 'change' feature to select (or reject) 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.

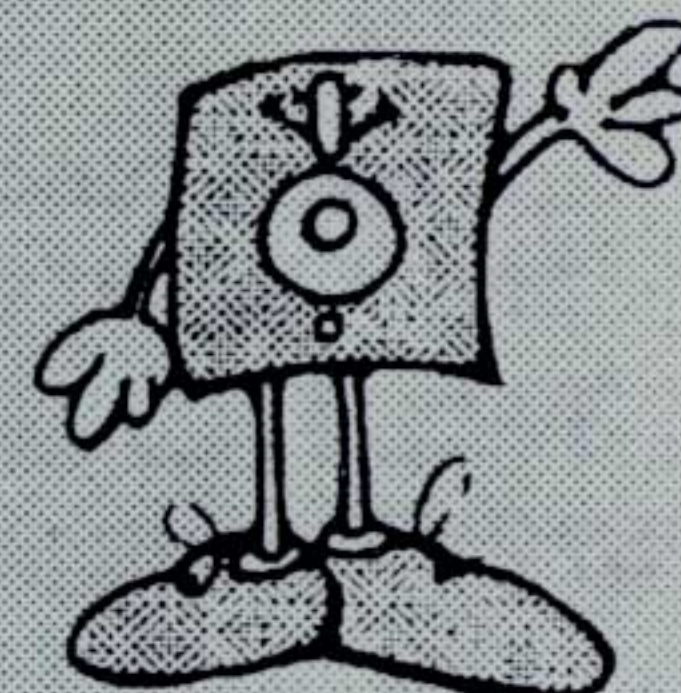
Don't be without it - order your copy today.



'DL' is available for TRS-80 Model 4/4P/4D using TRSDOS 6.2/LS-DOS 6.3.0 & 6.3.1. with 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



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.

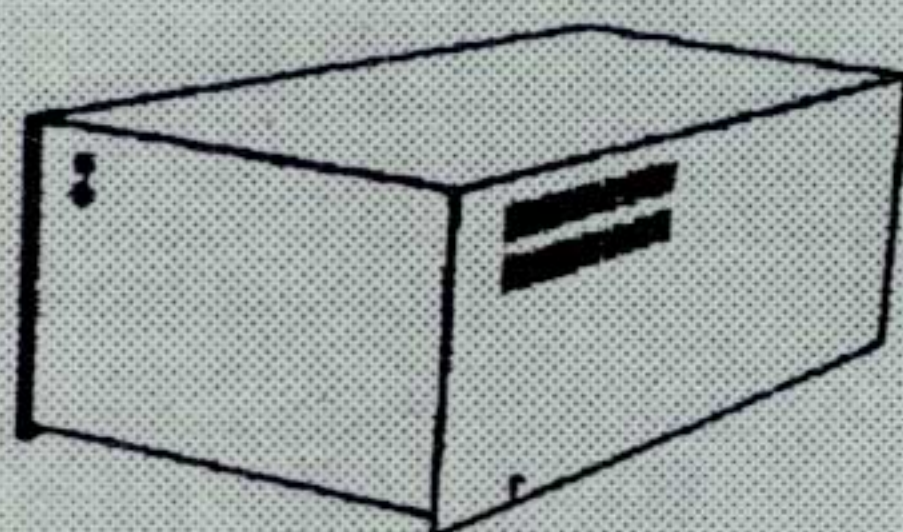
5 Meg \$175

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

