Programming Issue

A CWC/I PUBLICATION
OCTOBER 1984
USA $2.95 CAN $3.50

HOT CoCo

THE MAGAZINE FOR TRS-80 COLOR COMPUTER® AND MC-10® USERS.

Zounds. Sounds!

Real Bells And Whistles for Your CoCo

Dynacalc:

Eat Your Heart Out, Lotus 1-2-3

Spell 'N Fix II:

What's It Worth? You Decide

Plus: How To Make Programs Auto-Execute

0 A 74470 12067

The affordable step up in home computing.

Radio Shack's Best Is Now

Expand Your Programming Power! Our new 64K Extended BASIC Color Computer 2 gives superb features to serious programmers—at a very low price. It attaches easily to any television, and has an electric typewriter-quality keyboard in a compact, white case. Using the built-in Extended BASIC language, you can write sophisticated programs and easily create high-resolution graphics. Produce drawings, charts, diagrams and animation with simple, one-line commands. "Draw" with eight different colors, create musical tones, analyze data and much more. Or select from our large library of instant-loading Program Pak™ cartridges for games, education and budgeting.

Deluxe Joystick

39.95
Each
26-3012

TV not included

64K Extended BASIC Color Computer 2

AS LOW AS $26 PER MONTH

259.95
26-3127

Prices apply at participating Radio Shack stores and dealers. OS-9 is a trademark of Microware and Motorola, Inc. CitiLine is a service mark of Citicorp.

TRS-80® Color Computer Even Better.

The Heart of a Professional Disk-Based System. Add a Color Computer disk drive for an extra 156,000 characters of storage and quicker access to your data. You can add up to four disk drives for a vast 626,000 characters of total storage. For advanced programmers, our new OS-9 operating system (26-3030, $69.95) allows you to utilize the full 64,000-character memory for assembly language programs. OS-9 includes a text editor, assembler and debugger. You can develop and edit your own assembly language programs, convert them into machine language for execution, then test them for errors.

Easily Expandable. Best of all, your 64K Color Computer 2 will grow with your future needs. Add our Deluxe Joysticks for more accurate cursor control and faster response with your favorite games. Or choose our Multi-Pak Interface, which lets you connect up to four Program Paks to your Color Computer at once. When you're ready to change programs, just move the selector switch. Go on-line with the world with our deluxe RS-232 Program Pak (26-2226, $79.95). You can communicate with national information services and local bulletin boards by telephone—just add a modem. You can also add a printer, a Color Mouse cursor controller and more!

See It Today. The 64K Extended BASIC Color Computer 2 is available today at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers nationwide.

Multi-Pak Interface
179.95
26-3024

Disk Drive #0 Kit
349.95
26-3029
AS LOW AS $28 PER MONTH
Radio Shack CITILINE

Send me a free TRS-80 Computer Catalog.
Mail To: Radio Shack, Dept. 85-A-105
300 One Tandy Center, Fort Worth, Texas 76102

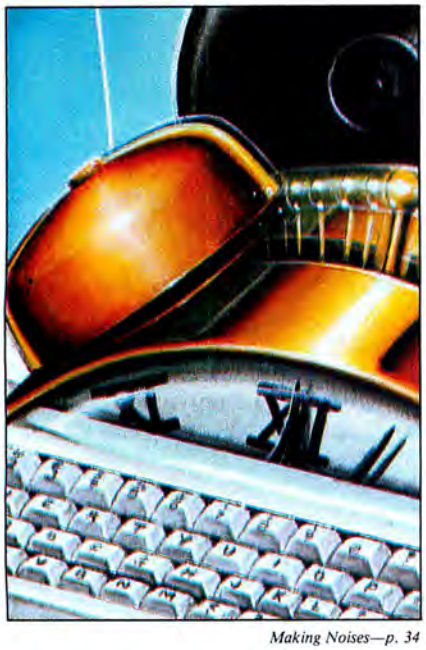
NAME _____
COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
TELEPHONE _____

Radio Shack®
The biggest name in little computers®
A DIVISION OF TANDY CORPORATION

4

HOT CoCo

Vol. 2 No. 5 October 1984



Making Noises—p. 34

DEPARTMENTS

ARTICLES

Cover art by William A. Giese

This symbol indicates the program's placement on the Instant CoCo loader, available on cassette. See our Instant CoCo ad for details.

HOT CoCo (ISSN 0740-3186) is published monthly by C.W. Communications, 80 Pine St., Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $24.97 for one year, $38 for two years, and $53 for three years. In Canada and Mexico, $27.97—one year only, U.S. funds. Application to mail at second class postage rates is pending at Peterborough, NH and additional mailing offices. Nationally distributed by International Circulation Distributors. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail), please inquire. In South Africa contact *HOT CoCo*, P.O. Box 782815, Sandton, South Africa 2146. **All subscription correspondence should be addressed to *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737. Please include your address label with any correspondence.** Postmaster: Send address changes to *HOT CoCo*, Subscription Services, P.O. Box 975, Farmingdale, NY 11737.

Telewriter-64™

the Color Computer Word Processor

- **3 display formats: 51/64/85 columns × 24 lines**
- **True lower case characters**
- **User-friendly full-screen editor**
- **Right justification**
- **Easy hyphenation**
- **Drives any printer**
- **Embedded format and control codes**
- **Runs in 16K, 32K, or 64K**
- **Menu-driven disk and cassette I/O**
- **No hardware modifications required**

THE ORIGINAL

Simply stated, Telewriter is the most powerful word processor you can buy for the TRS-80 Color Computer. The original Telewriter has received rave reviews in every major Color Computer and TRS-80 magazine, as well as enthusiastic praise from thousands of satisfied owners. And rightly so.

The standard Color Computer display of 32 characters by 16 lines without lower case is simply inadequate for serious word processing. The checkerboard letters and tiny lines give you no feel for how your writing looks or reads. Telewriter gives the Color Computer a 51 column by 24 line screen display with *true lower case characters*. So a Telewriter screen looks like a printed page, with a good chunk of text on screen at one time. In fact, more on screen text than you'd get with Apple II, Atari, TI, Vic or TRS-80 Model III.

On top of that, the sophisticated Telewriter full-screen editor is so simple to use, it makes writing fun. With single-letter mnemonic commands, and menu-driven I/O and formatting, Telewriter surpasses all others for user friendliness and pure power.

Telewriter's chain printing feature means that the size of your text is never limited by the amount of memory you have, and Telewriter's advanced cassette handler gives you a powerful word processor without the major additional cost of a disk.

...one of the best programs for the Color Computer I have seen...
— Color Computer News, Jan. 1982

TELEWRITER-64

But now we've added more power to Telewriter. Not just bells and whistles, but major features that give you total control over your writing. We call this new supercharged version Telewriter-64. For two reasons.

64K COMPATIBLE

Telewriter-64 runs fully in any Color Computer — 16K, 32K, or 64K, with or without Extended Basic, with disk or cassette or both. It automatically configures itself to take optimum advantage of all available memory. That means that when you upgrade your memory, the Telewriter-64 text buffer grows accordingly. In a 64K cassette based system, for example, you get about 40K of memory to store text. So you don't need disk or FLEX to put all your 64K to work immediately.

64 COLUMNS (AND 85!)

Besides the original 51 column screen, Telewriter-64 now gives you 2 additional high-density displays: 64 × 24 and 85 × 24!! Both high density modes provide all the standard Telewriter editing capabilities, and you can switch instantly to any of the 3 formats with a single control key command.

The 51 × 24 display is clear and crisp on the screen. The two high density modes are more crowded and less easily readable, but they are perfect for showing you the exact layout of your printed page, *all on the screen at one time*. Compare this with cumbersome "windows" that show you only fragments at a time and don't even allow editing.

RIGHT JUSTIFICATION & HYPHENATION

One outstanding advantage of the full-width screen display is that you can now set the screen width to match the width of your printed page, so that "what you see is what you get." This makes exact alignment of columns possible and it makes hyphenation simple.

Since short lines are the reason for the large spaces often found in standard right justified text, and since hyphenation is the most effective way to eliminate short lines, Telewriter-64 can now promise you some of the best looking right justification you can get on the Color Computer.

FEATURES & SPECIFICATIONS:

Printing and formatting: Drives any printer (LPVII/VIII, DMP-100/200, Epson, Okidata, Centronics, NEC, C. Itoh, Smith-Corona, Terminet, etc).

Embedded control codes give full dynamic access to intelligent printer features like: underlining, subscript, superscript, variable font and type size, dot-graphics, etc.

Dynamic (embedded) format controls for: top, bottom, and left margins; line length, lines per page, line spacing, new page, change page numbering, conditional new page, enable/disable justification.

Menu-driven control of these parameters, as well as: pause at page bottom, page numbering, baud rate (so you can run your printer at top speed), and Epson font. "Typewriter" feature sends typed lines directly to your printer, and Direct mode sends control codes right from the keyboard. Special Epson driver simplifies use with MX-80.

Supports single and multi-line headers and automatic centering. Print or save all or any section of the text buffer. Chain print any number of files from cassette or disk.

File and I/O Features: ASCII format files — create and edit BASIC, Assembly, Pascal, and C programs, Smart Terminal files (for uploading or downloading), even text files from other word processors. Compatible with spelling checkers (like Spell 'n Fix).

Cassette verify command for sure saves. Cassette auto-retry means you type a load command only once no matter where you are in the tape.

Read in, save, partial save, and append files with disk and/or cassette. For disk: print directory with free space to screen or printer, kill and rename files, set default drive. Easily customized to the number of drives in the system.

Editing features: Fast, full-screen editor with wordwrap, block copy, block move, block delete, line delete, global search and replace (or delete), wild card search, fast auto-repeat cursor, fast scrolling, cursor up, down, right, left, begin line, end line, top of text, bottom of text; page forward, page backward, align text, tabs, choice of buff or green background, complete error protection, line counter, word counter, space left, current file name, default drive in effect, set line length on screen.

Insert or delete text anywhere on the screen without changing "modes." This fast "free-form" editor provides maximum ease of use. Everything you do appears immediately on the screen in front of you. Commands require only a single key or a single key plus CLEAR.

...truly a state of the art word processor... outstanding in every respect.
— The RAINBOW, Jan. 1982

PROFESSIONAL WORD PROCESSING

You can no longer afford to be without the power and efficiency word processing brings to everything you write. The TRS-80 Color Computer is the lowest priced micro with the capability for serious word processing. And only Telewriter-64 fully unleashes that capability.

Telewriter-64 costs $49.95 on cassette, $59.95 on disk, and comes complete with over 70 pages of well-written documentation. (The step-by-step tutorial will have your writing with Telewriter-64 in a matter of minutes.)

To order, send check or money order to:

Cognitec
704 N. Nob St.
Del Mar, CA 92014 ↙ 121

Or check your local software store. If you have questions, or would like to order by Visa or Mastercard, call us at (619) 755-1258 (weekdays, 8AM-4PM PST). Dealer inquiries invited.

(Add $2 for shipping. Californians add 6% state tax. Allow 2 weeks for personal checks. Send self-addressed stamped envelope for Telewriter reviews from CCN, RAINBOW, 80-Micro, 80-U.S. Telewriter owners: send SASE or call for information on upgrading to Telewriter-64. Telewriter-compatible spelling checker (Spell 'n Fix) and Smart Terminal program (Colorcom/E) also available. Call or write for more information.)

HOT CoCo

PUBLISHER
Jeff DeTray

EDITOR-IN-CHIEF
Michael E. Nadeau

MANAGING EDITOR
Janet Fiderio

REVIEW EDITOR
Mark E. Reynolds

NEW PRODUCTS EDITOR
Celeste Wrenn

TECHNICAL EDITORS
Peter Paplaskas,
Guier Wright,
Keith Johnson (Instant CoCo)

EDITORIAL OPERATIONS MANAGER
Jack Burnett

ADVERTISING: 1-800-441-4403
Sales Manager: Raino E. Wirein
West Coast Office: 1-415-328-3470 or 3471
160 Marsh Road
Menlo Park, CA 94025
Sales Manager: Giorgio Saluti
Sales Representatives: Allison Walsh,
Karen Letendre

PRODUCTION DIRECTOR: Nancy Salmon
Assistant Production Manager/Manufacturing:
Susan Gross
Typesetting Manager: Dennis Christensen
Film Preparation Manager: Robert M. Villeneuve
Photography Manager: Nathaniel Haynes

CREATIVE DIRECTOR
Christine Destrempes

DESIGN MANAGER
Joyce Pillarella

DESIGNERS
Susan Hays, Dion Owens, Maurelle Godoy

VICE PRESIDENT/GENERAL MANAGER
Debra Wetherbee

VICE PRESIDENT/FINANCE
Roger Murphy

ASSISTANT TO VICE PRESIDENT/GM
Matt Smith

ASSISTANT TO VICE
PRESIDENT/FINANCE
Dominique Smith

MARKETING MANAGER
Pamela Esty

DIRECTOR OF CIRCULATION
William P. Howard

ASSISTANT CIRCULATION MANAGER
Frank S. Smith

DIRECT & NEWSSTAND
SALES MANAGER
Ginnie Boudrieau; 1-800-343-0728

DIRECTOR OF CREDIT, SALES,
AND COLLECTION
William M. Boyer

DIRECTOR OF PUBLIC RELATIONS
James Leonard

FOUNDER
Wayne Green

DIGRESSIONS

WHAT YOU DON'T KNOW MIGHT SOMEDAY HELP YOU

I have a confession to make: I am not a great programmer. In fact, I probably never will be. I don't *want* to write my own programs; I can buy anything that I need, so why bother?

However, I can follow the flow of a Basic listing. I am familiar with the syntax of most Basic commands, and I can debug a program or change it to suit my needs.

To many people, programming is an enjoyable pastime. This is great; we need more hobbies that are as creative as programming. But I'll stick to stamps and Studebakers.

You don't have to be a programmer, but a little knowledge of Basic, and sometimes Assembly language, can mean the difference between frustration with an unexpected problem and the confidence that comes with being self-reliant for your computing problems.

Some examples? Okay: Say you just typed in what looks like an interesting program from a book or magazine. You've checked and double-checked the listing for accuracy, but you continue to get an SN error message. Did you type in something wrong, or did the publication make a mistake? A little knowledge of command syntax might help you track down the problem.

Once you've solved the syntax problem, what if you discover the program isn't as good as you thought because the screen format was poorly designed? If you know how characters are allocated on the video screen, you probably need only to adjust some PRINT statements.

The list of examples is endless. The point is that the more you know about your Color Computer, the more you will benefit from owning one.

If you are a programmer, pay a little attention to what you type into your computer when you enter programs from this magazine. You'll be surprised at how much you learn. (I taught myself most of what I know about programming this way.) Also, read the entire accompanying articles, not just the instructions on how to use the program. You might pick up some piece of information that will help you later.

I'm not trying to twist your arm if you really have no interest in programming. Not everyone who drives a car is a mechanic. But most people know how to change a flat tire or check their oil. In the same vein, computer owners should know some simple elements of Basic programming.

Who knows: Maybe once you know a little about it, you'll like programming.—*Michael E. Nadeau* ■

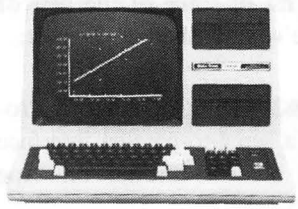
HOT CoCo is a member of the CW Communications/Inc. group, the world's largest publisher of computer-related information. The group publishes 52 computer publications in 19 major countries. Members of the group include: Argentina's *Computerworld/Argentina*; Australia's *Australia Computerworld, Australian Micro Computer Magazine, Australian PC World and Directories*; Brazil's *DataNews* and *MicroMundo*; China's *China Computerworld*; Denmark's *Computerworld/Danmark* and *MicroVerden*; Finland's *Mikro*; France's *Le Monde Informatique, Golden* (Apple) and *OPC* (IBM); Germany's *Computerwoche, Microcomputerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business* and *Commodore Magazine*; Italy's *Computerworld Italia*; Japan's *Computerworld Japan* and *Perso ComWorld*; Mexico's *Computerworld/Mexico* and *CompuMundo*; Netherland's *CW Benelux* and *Micro/Info*; Norway's *Computerworld Norge* and *MikroData*; Saudi Arabia's *Saudi Computerworld*; Singapore's *The Asian Computerworld*; Spain's *Computerworld/Espana* and *MicroSistemas*; Sweden's *ComputerSweden, MikroDatorn* and *Min Hemdator*; the UK's *Computer Management* and *Computer Business Europe*; United States: *Computerworld, HOT CoCo, inCider, InfoWorld, jr, MacWorld, Micro MarketWorld, Microcomputing, PC World, PC Jr. World, RUN, 73 Magazine* and *80 Micro*.

From Computer Plus to YOU . . .

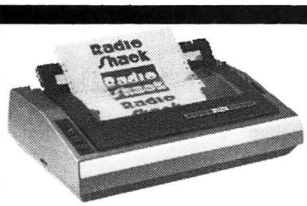
PLUS *after* PLUS *after* PLUS

Model 100 8K $495
Model 100 24K $625

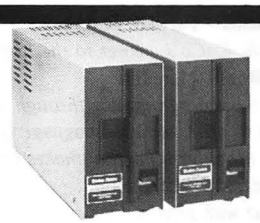
Color Computer II
w/64K Ext. Basic $205



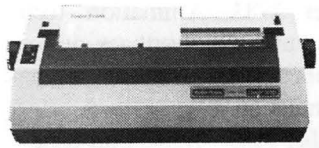
Model 4 16K $679
Model 4 64K
2 Disk & RS232 $1020



DMP120 $385
DMP200 $520



Color Computer Disk Drive
Drive 0 $289 Drive 1 $220



DWP210 $489

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS

Model 4 Portable 64K w/2 Drives	1020
Model 2000 2Dr	2299
Model 12 1 Drive	2360
Model 16B 1Dr 256K	3965

MODEMS

Hayes Smartmodem II	215
AC-3	125
DC Modem I	89
DC Modem II	160

PRINTERS

Silver Reed EXP400 D.W. Par.	309
Silver Reed EXP550 D.W. Ser.	430
CGP115	159
CGP220 Ink Jet	545
DMP110	305
Toshiba 1340 (24 wire head)	779
Gemini 10X	289
Gemini Powertype	345
Panasonic P1091	315
Smith Corona Fastext	190
Prowriter 8510	345
Okidata and Epson	CALL

ETC.

Disk Drive Controller	139
Extended Basic Kit	39.95
PBH Ser/Par Conv.	69
64K Ram Chips	62.95
Deluxe Keyboard	35.95
HJL Keyboard	79.95
CCR-81 Recorder	52
Deluxe Joystick (each)	35.95
Joysticks (pair)	22
Video Plus (monitor adapter)	24.95
Video Plus IIC	39.95
Amdek Color 1+ Monitor	299
Amdek Video 300 Green	145
Amdek Video 300 Amber	159
Taxan Color 210 Monitor	265
Taxan Green	130
Taxan Amber	139

SOFTWARE (Tape Version)

The King	26.95
Screen Print (specify printer)	19.95
Buzzard Bait	27.95
World of Flight	29.95
Colorpede	29.95
Juniors Revenge	28.95
Pac Attack	24.95
Block Head	26.95
Lunar Rover Patrol	24.95
Lancer	24.95
Typing Tutor	23.95
Galagon	24.95
Scott Adams Adventures	19.95
Sea Dragon	34.95
Colorcome	49.95
Telewriter 64	49.95
O-Pak (disk)	34.95
Key-264K	39.95
Deft Pascal	79.95
Elite-Calc	59.95
VIP Writer	69.95
VIP Calc	69.95
VIP Terminal	49.95
VIP Database (disk)	59.95
Graphicom	29.95

Order any 2 software pieces listed and take 10% off their listed price. All Radio Shack software 10% off list. Send for complete list.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

18

TRS-80 is a registered trademark of Tandy Corp.

Back Issues

Yes, back issues of *HOT CoCo* are available for all months. Here's a short list of some of the best of what we've published in the past:

June 1983—The CoCo Word Processor, a serial-to-parallel interface project, and a tutorial on tape reliability
July 1983—How to upgrade your CoCo to 64K
August 1983—Speech synthesis without hardware
September 1983—Disk utilities, character generator
October 1983—Animation techniques, build a biofeedback device
November 1983—Nuclear submarine simulation
December 1983—Education issue
January 1984—Programs for the investor and businessman
February 1984—Simulate Extended Color Basic on Color Basic CoCos
March 1984—How a disk stores information, create your own word-search puzzles
April 1984—Peripherals Buyer's Guide, how to shop for a disk drive
May 1984—OS-9 review, Financial Transactions Tracker program
June 1984—Simulations issue, how to build an Atari joystick interface
July 1984—Build your own lowercase modification
August 1984—Your disk drive as a graphics tool
September 1984—Buyer's Guide to Educational Software

In each back issue, you'll also find our regular features, reviews of popular software and hardware, and dozens of useful programs that are yours for the typing in.

Each back issue costs $3.50 plus $1 shipping and handling. On orders of 10 or more back issues, there is a flat $7.50 shipping and handling fee. Send your orders to *HOT CoCo*, Attn.: Back-Issue Orders, 80 Pine St., Peterborough, NH 03458.

Instant CoCo

This directory lists all programs available on HOT CoCo*'s October Instant CoCo cassette. See our ad on page 81 for more details.*

SIDE A

SIDE B

Tips on Entering Our Programs

Having trouble entering our listings from the magazine? Here are a few tips that might help.

First, we print all our Basic listings in the CoCo's 32-column format. This means that each line should appear the same on the screen as it does in the magazine. If a line on your screen does not match the same line in the magazine, reread what you typed; you might have made an error.

Second, make sure the program is for your computer. Read the System Requirements box. The information in this box represents the minimum system configuration needed to run that particular program. Also, read the article thoroughly before typing in the program. Sometimes the article contains instructions vital to making the typed-in listing work. For instance, some CoCos will not accept the high-speed POKE (POKE 65495,0). The article for a program using this POKE will tell you to change those POKEs to 65494,0 if your computer will not work at the faster speed.

Anyone who owns the new CoCos with the 1.2 ROMs, have noticed poor keyboard response in some published programs. To solve this, you can insert this line: FOR Z=1TO4:POKE340+Z,255:NEXT after any line that makes reference to PEEK 338-345. This loop will slow down a Basic program. Another way is to directly insert a POKE xxx,255, where xxx is any keyboard location between 338 and 345. Example: IF PEEK(341)=251 THEN Y=Y-1. Change to: IF PEEK(341) =251 THEN POKE341,255:Y=Y-1.

Assembly listings usually require an editor/assembler to enter them into your CoCo. The two most common editor/assemblers are Radio Shack's EDTASM+ and The Micro Works' SDS80C. An Assembly listing assembled using the SDS80C will probably not run under EDTASM+.

If all the above fails, send us a printout or a detailed description of the problem you experience along with any error messages. We'll try to work it out for you. We cannot help you if you have modified the original program.■

The up-arrow indicates exponentiation on your Color Computer. However, our printer does not have an up-arrow and prints a caret instead. When entering programs from *HOT CoCo*, please change all carets to up-arrows.

Article submissions from our readers are welcomed and encouraged. Inquiries should be addressed to: *HOT CoCo* Submissions Editor, 80 Pine Street, Peterborough, NH 03458. Include an SASE for a copy of our writer's guidelines. Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased. Authors of reviews should contact the *HOT CoCo* Review Editor, 80 Pine Street, Peterborough, NH 03458.

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737.

Change of Address: Send old label or copy of old address and new address to: *HOT CoCo*, P.O. Box 975, Farmingdale, NY 11737. Please give eight weeks advance notice.

Dealers: Contact Ginnie Boudrieau, Bulk Sales Manager, *HOT CoCo*, Pine St., Peterborough, NH 03458. (800) 343-0728.

Problems with Advertisers: Send a description of the problem and your current address to: Magazine, Rt. 101 & Elm Street, Peterborough, NH 03458. ATTN.: Rita B. Rivard, Customer Service Manager. If urgent, call 1-800-441-4403.

There's more to OS-9 than meets the eye.

File Handlers Toolbox **$85.00**

The File Handlers Toolbox: a new utility command toolbox specially designed for OS-9 users who do a lot of file manipulation. The package is a collection of twelve OS-9 command programs, including equivalents of some of the most popular UNIX* utilities that are not included in the basic OS-9 command set. Most of the programs are useful as "filters" using the OS-9 pipeline facilities.

Entertainment Pack **$85.00**

Entertainment Pack I is a collection of programs written in Basic09 for the OS-9 Operating System. The package consists of games and other interesting programs that are not only entertaining but serve as excellent instructional examples of Basic09 programming techniques. All programs include complete source files and can be easily edited to run on standard alphanumeric or graphics terminals.

CIS Cobol/Forms 2 **$400.00**

CIS COBOL, which meets the ANSI standard for Level One Cobol plus selected features from Level Two, is ideal for microcomputers. Combined with the FORMS 2 package it creates a timesaving, fast and convenient screen oriented business application system.

Relocatable Macro Assembler **$125.00**

At last — a full feature relocatable macro assembler and linkage editor for OS-9. RMA permits sections of assembly language programs to be independently assembled to "relocatable object files". The linkage editor takes any number of program sections and/or library sections and combines them into a single executable OS-9 memory module. Global data and program references are automatically resolved in the process. RMA also supports conditional assembly and library source files.

The Official OS-9 Manual Set **$40.00**

The complete, unabridged OS-9 manual set direct from Microware. This three manual set contains complete information on writing device discriptors, disk drivers and full explanations of how OS-9 works. A great addition to the serious OS-9 programmers library.

The BASIC09 Tour Guide **$18.95**

Map out your route through the Mercedes of Basics . . . Basic09 with **the official Basic09 Tour Guide.** Skillfully written in a friendly and easy to read style this book will put you in the drivers seat in no time. Fasten your seatbelt, sit back and enjoy the ride to perfecting your programming skills.

That's just the beginning . . .

GET EYE TO EYE WITH MICROWARE AND OS-9

microware®

OS-9™

137

MICROWARE SYSTEMS CORPORATION • 1866 N.W. 114th STREET • DES MOINES, IA 50322
TELEPHONE 515/224-1929 • TELEX 910-520-2535

OS-9 and BASIC09 are trademarks of Microware and Motorola. UNIX is a trademark of Bell Laboratories, Inc.

Feedback

Avast, You Pirates!

I haven't seen any firm statistics on piracy; some say pirated copies of software outnumber legitimate copies 10:1, others say 20:1. I always thought authors and publishers faked most of these numbers as an excuse for poor sales, but I was wrong.

Last year, I wrote a graphics puzzle, Strip Tease, that I advertised for only one month. Because of the game's adult theme, no store or software company wanted to move it.

Over the next three months I sold exactly 71 copies, and received responses from almost 400 other people interested in the game. The price was $14, and I never advertised the fact that anyone who bought Strip Tease also received a second game free. I made about $219 for a month's worth of design and Assembly-language programming. The game just wasn't a hit.

But a curious thing happened. Because of the difficulty of the puzzle, people started writing and calling for hints to the solution, like they do for adventure games. Now remember, I only sold 71 copies, and I produced and distributed all of these, so I know exactly who the purchasers are. As a matter of fact, I imbedded the legitimate buyer's name and a serial number in every copy I sold.

I received 22 requests (most by phone) for hints to the solution. Only one of these requests (a nasty letter—I remember it well) was from someone who actually paid for the game. If my math is correct, a 22:1 ratio would mean that there are at least 1,562 copies of my puzzle in existence, although I was only paid for 71.

I always thought that copyright pirates were clever, sneaky, code-breaking people. I know 21 such individuals who couldn't solve my puzzle, and who also gave me their names and addresses.

If people steal my stuff like this, how much do they steal across the whole CoCo market? How many illegal copies have they made of great games? Does price mean anything? Are they more likely to steal when the price is $14 for two pieces of software, or when it's $34.95 for a single program?

If anyone has access to Strip Tease, you might want to load the program and type in the following:

```
10 FOR I=&H3357 TO &H336E:PRINT
  CHR$(PEEK(I));:NEXT
RUN
```

This will display the original purchaser's name. In many cases, you won't even know the guy.

Britt Monk, CDP
Morristown, TN

Scripsit-to-Okidata Interface

I have had a difficult time interfacing my Color Scripsit disk to my Okidata printer. Perhaps what I've learned will be of help to others.

Scripsit uses its own printer driver rather than the one in the Color Basic ROM. The Scripsit driver doesn't signal a full-stop bit between characters. This is no problem with Radio Shack printers, but it doesn't work with many other brands.

I disassembled part of Scripsit and added the following line to DOS/BAS:

```
15 POKE &HEBC,&H8D :POKE &HEBD,6
:POKE &HEBE,&H12
```

Be sure to save the modified version on disk. This patch is for Scripsit version 1.03.

For those who are interested, the printer driver starts at $0E95 and is called with the character to print in the A register.

Eric Hazen
Madison, WI

Shift/O

In the July 1984 Doctor ASCII section, a reader said that he could not use the letter O key when he used the speedup POKE 65495,0. Using a shifted O corrects the problem. It's a little annoying to use often, but it beats taking the computer apart.

Bradley Roe
El Paso, TX

Possum Runs On Disk

I found it surprisingly easy to convert "Possum Run" (*HOT CoCo*, March 1984, p. 50) to run on disk.

I relocated the machine-language routines by adding 10,000 (decimal) to all machine-language references (i.e., line 0, clear 200, 26000; line 5025, TT=26300; etc.).

Then I added $800 to each absolute graphics-screen reference in the machine-language strings (5000–5126) to offset the amount of memory the disk system uses.

These references are easy to spot. They are the four-digit numbers following each 8E (LDX) and 8C (CMP). For example, in line 5020, the first six hex numbers are 8E0C41, which translate to LDX#$0C41. Adding $800 results in 8E1441.

David Sonnenshine
Oroville, CA

Joystick Interface Tip

In his article, "Atari Joystick Interface" (*HOT CoCo*, June 1984, p. 82), James Barbarello uses a 4016 CMOS quad analog switch IC. I've had trouble finding these chips locally, and I don't like to come up with the standard $10 mail order just to get a 35¢ chip.

Fortunately, Radio Shack sells a CMOS quad bilateral switch IC (part #276-2466, chip #4066) for about $1, and you can use it without making any changes in Mr. Barbarello's fine article.

Ron Snell
Cortland, OH

Statistical Analysis Help

Does anyone out there know of a complete statistical analysis program for a 64K cassette-based CoCo with printer? I could really use the help in my work, and would appreciate any information you can give.

As a relative newcomer to computers and programming, I've found *HOT CoCo* most informative. In fact, if it weren't for your magazine, my computer and I would still be staring at each other, each wondering what

the other could do. Many thanks to you and your contributors.

Stephen M. Hicks
5969 Nelda St. #1
Simi Valley, CA 93063

There are several programs out there that should more than do the job. If you read the review of Dynacalc in this issue, you might run out and buy a disk drive.

You might also take a look at any of the following:

- *SuperStat, from Skyline Marketing, 4510 W. Irving Park Road, Chicago, IL 60641, 16K, Extended Color Basic, $29.95 cassette (reviewed on p. 22 of our May 1984 issue);*
- *Statgraf, from Sugar Software, 2153 Leah Lane, Reynoldsburg, OH 43068, 32K, Extended Color Basic, $24.95 cassette, $29.95 disk (reviewed on p. 29 of our March 1984 issue); or*
- *Elite Calc, from Elite Software, Box 11224, Pittsburgh, PA 15238, 16K-64K, Extended Color Basic, $59.95, disk or cassette (reviewed on p. 19 of our December 1983 issue).*

There are certainly other fine programs that will analyze data for you. Individuals and manufacturers who have more information on the subject are invited to pass it along to Mr. Hicks.—eds.

Color Down Under

On p. 13 of the March 1984 Feedback section, you published a letter entitled "Them Ol' PMODE4 Blues," concerning the problems caused by using the CoCo with the P.A.L. television system here in Australia. High-resolution graphics in PMODE4 come out black and white or blue and white.

However, I know of a program that corrects this problem in most cases. It gives the four colors in the PMODE3 graphic screen and the screen option of 1.1 or 1.0.

The program is called Convert and is available from

Robin Brown
4/98 Vernon St.
Nundah,
Queensland, 4012
Australia

I've used it on many games and found it most satisfactory.

Henry Cope
Zilzie, Queensland
Australia

Smashout Help

I just finished reassembling "Smashout" (*HOT CoCo*, November 1983, p. 80) according to the information you gave in the February 1984 Feedback section (p. 8).

Now, does anyone out there know how I can get the game to score off the top of the blocks, like Atari's Breakout does?

A. Hartsough
4620 Butler Road
Fort Wayne, IN 46808

More Investment Help

I have written many of my own programs to analyze stocks, but I thought your "Stock Transactions Tracker" (*HOT CoCo*, January 1984, p. 58) and "Stock Market Simulator" (*HOT CoCo*, June 1984, p. 58) were excellent. Both could be valuable and entertaining for investment clubs.

Please continue publishing programs and articles on financial systems and stocks. *HOT CoCo* is a great magazine—one I wouldn't be without.

Ron Komas
Richfield, WI

Autodialing

For the past few months, I've been trying to convert a Model III program that automatically dials a series of phone numbers for a selected prefix. So far, I've been unsuccessful.

If anyone out there has such a program, or knows where I can buy one, I would appreciate hearing about it.

Mike MacNeil
3596 Bloomfield
Windsor, Ontario N9C 1R7

Puget Sound *HOT CoCo*

I would like to hear from anyone in the mideastern Puget Sound area who has a fairly complete collection of *HOT CoCo*. I want to check the tables of contents.

Please phone 523-9222 (home) or 773-4196 (work), or write.

Dane Brooke
74 7547 35 N.E.
Seattle, WA 98115

Programming Pin 11

I find your magazine very informative, and I look forward to getting my new copy each month.

Barry Becker stated in his article, "Sound Advice" (*HOT CoCo*, April 1984, p. 74), that the *TRS-80 Technical Reference Manual* (CoCo) does not explain how to program pin 11 of PIA U4 (bit 1 = single bit sound output) as an input or output.

However, the second paragraph on p. 4 of the manual tells you just how to do so.

Kirk Thompson
White Rock, NM

Income Tax Program

I have developed a federal income tax program and would like to send free evaluation copies to some of your readers. I need to know how it runs on all Color Computer system configurations.

John W. Gregg
AlphaByte
1008 Alton Circle
Florence, SC 29501
803-662-9500

Pet CoCo

I have a Color Computer at home and use a Pet in my programming courses at school, but I'm finding it difficult to keep the differences between the two Basics straight.

I can't find a program that will let me translate my Pet programs so they will run on my CoCo. Is there anyone out there who knows where I can get such help?

William Lowe
7 Crowland Drive
Rexdale, Ontario M9W 2S5

Light Cycles Search

I'm looking for a program like the Tron game, Light Cycles. Can anyone out there help?

Cory Laursen
HC-34 Box 67
Alliance, NE 69301

Send your letters to Feedback, HOT CoCo, *80 Pine St., Peterborough, NH 03458.*

NOW the One Person Who Cares Most About Your Computer Can Protect Its Data —

YOU!

Introducing... Memory Minder from J&M Systems
The most advanced disk drive testing program on the market.

Until recently, only trained technicians using special equipment could check your disk drives for potential or actual problems. But now, thanks to major breakthroughs by J&M Systems, Ltd. and Dysan Corporation, YOU can be your own computer's technician.

Memory Minder is the most comprehensive disk diagnostic program available for microcomputers. With Memory Minder, you can periodically test disk drives to monitor long-term drift of head alignment, index hole timing, spindle speed, directional seek, and many other parameters. Then, if your head is out of alignment, Memory Minder can accurately align it once again. Early detection and correction of these problems will help protect you against costly data loss and down time.

And with Memory Minder, you don't need a technical background to perform these tests. Simply follow the easy instructions in the manual or on the screen's menu. Now you can use state-of-the-art software to care for your hardware.

Memory Minder is currently available for IBM, TRS-80, Kaypro and Sanyo systems. Only from J&M Systems. Prices start at $79.

_____ I'd like more information on Memory Minder. Please send me your complete brochures.

_____ Send me information on other fine products from J&M Systems.

I have the following computer: _____

Name _____

Address _____

City _____ State _____ Zip _____

101

J & M SYSTEMS, LTD.

137 UTAH NE
ALBUQUERQUE, N.M. 87108
505/265-1501

See List of Advertisers on page 97

The Basic Beat

THE FOR... NEXT LOOP AND SET GRAPHICS

by James W. Wood

Last month you saw some lessons in addition, subtraction, multiplication, and division. Program Listing 1 is a brief review. Each example contains only one math operation.

What happens if you use multiplication and addition in one problem? Look at Program Listing 2. You might have expected line 10 to print 14, the result of adding three and four and then multiplying the sum times two, but it doesn't work that way. The computer follows the mathematical rules of order of operations.

The computer performs all multiplications and divisions before additions and subtractions. For example, it calculates line 10 as four times two equals eight, eight plus three equals 11. In line 30 it performs the two multiplications first and then adds the products.

An Extended Color Basic machine also has an exponent key: the up-arrow. When you use the up-arrow as a math operation, you must have a number on both sides of the arrow. PRINT 2 ↑ 3 represents two to the third power, or 2*2*2, which equals eight. The up-arrow on most printouts appears as an inverted caret (Λ) in a listing. The CoCo will calculate a power function before multiplication and division. Program Listing 3 is for Extended Color Basic computers.

Is there something that can change these rules of order of operations? The computer performs any operation within parentheses first. Therefore, in working math problems, use this order: parentheses, powers, multiply and divide, and then add and subtract from left to right.

Program Listing 4 is your math operation quiz program. Use a pencil and paper to calculate the answers before you run the program.

Program Listing 5 is a brief review of IF...THEN. It prints the numbers from 1–1,000. Remember the purpose of the semicolon in line 20? It prints the numbers close together; not in two columns as a comma would, or with each number on a different line, as results when there is no punctuation.

```
10 A=3
20 B=4
30 PRINTA+B
40 PRINTB-A
50 PRINTA*B
60 PRINTA/B
```

Program Listing 1

```
10 PRINT3+4*2
20 PRINT3*4+2
30 PRINT2*3+4*5
```

Program Listing 2

```
10 PRINT5+4*2^3
```

Program Listing 3

```
10 PRINT5+6/3+7
20 PRINT2*3-2
30 PRINT2*3*4-5*4
40 PRINT17-12/4
```

Program Listing 4

As Program Listing 6 runs it prints the same numbers as Listing 5. This is a FOR...NEXT loop. The FOR statement in line 10 must contain a variable, a beginning number, TO, and an ending number. A NEXT statement must follow the FOR statement. The FOR and NEXT mark the start and end of a FOR loop.

When you run the program, A first equals one. The next A in line 30 sends the program back to line 10, which sets A equal to two. This loop continues, adding one to the value of A each time, until it reaches 1,000. Listing 5 takes 25 seconds to run, while Listing 6 runs in 15 seconds. FOR...NEXT loops are faster than IF...THEN statements.

Program Listing 7 uses a FOR... NEXT loop as a delay. Since there are no other statements between the FOR and NEXT, the program simply takes time to count to 500 before changing the screen color.

Notice the STEP in line 10 of Program Listing 8. This increases the value of A by 10 each time. The printout will be 0, 10, 20, 30, and so on.

How do you decrease a FOR... NEXT loop? Program Listing 9 will count from 20 to 1. To do so, line 10 must have the first number larger than the second, and a negative number must follow the STEP command.

Program Listings 10 and 11 produce exactly the same results. Each program prints a six-column-by-six-row group of numbers, arranged in a multiplication table.

In Listing 10 you'll see the dreaded nested FOR...NEXT loop. The inner loop (B) must go from one to six first. After B equals six, the program goes through line 40 to line 50, which sends print to the following screen line. Then line 60 sends the A loop back to line 10 where A is equated to two. B will vary from one to six again before A increases to three, and this continues until both A and B equal six.

Listing 11 gets the same results as Listing 10, but it uses IF...THEN commands. Which program is easier to understand? I'll use FOR loops for-

SEND FOR FREE CATALOG

TCE

TM

Dealer inquiries invited

ABC'S IN COLOR

In the ABC program, all 26 letters spring up in color to the familiar ABC tune. Then, colorful detailed pictures depicting each individual letter of the alphabet appear one by one. Your child's fascination will mount as he or she correctly presses the letter on the keyboard and is rewarded with a musical tune before the next detailed picture is drawn line by line onto the screen: AIRPLANE for A, BUS for B, CLOWN for C and so on to ZEBRA for Z. Truly a must program for the preschool to first grade age group!

CoCo 16K ECB . . . . . Tape: $19.95 Disk: $25.95

CRISS–CROSS MATH

As the program begins, your child is presented with a nine square playing board. It is your choice as to which square you choose. After a choice is made, a MATH PROBLEM appears in the square. You score your first X by answering the problem correctly. If your answer is incorrect, the square clears and your opponent is allowed his choice of squares. The game is over when three squares vertically, horizontally, or diagonally are won by the same player. When playing against the computer, every answer you get wrong is won by the computer. Multi-level ADDITION AND SUBTRACTION program.

CoCo 16K . . . . . Tape: $12.95

FRACTIONS

SIDE ONE: Fraction Lessons, explains fractions with the aid of graphics. Child studies the different ways fractions can be represented. Lessons include:

IMPROPER FRACTIONS
MIXED FRACTIONS
PROPER FRACTIONS

Many educators have praised the use of motion and color to display the fractional equivalents.

SIDE TWO: Fraction practice, offers a random computer generated quiz.

Atari 16k . . . . . Tape: $19.95
CoCo 16k . . . . . Tape: $19.95

JOYSTICK DRAW

Joystick Draw is the simple way to explore your artistic talents! Program operation is easy enough for a child to use, but effective enough that TCE uses it to design many sophisticated high-resolution graphic screens. Joystick Draw's design allows you or your child to save those masterpieces for future revisions or for use in other programs (instructions included). Your child will spend many hours enjoying this program and at the same time improving his or her eye hand coordination! You will find Joystick Draw to be an easy way to design those more sophisticated graphics for your own programs!

CoCo 16 ECB . . . . . Tape: $16.95

SPELL BOMBER

As captain of your ship, you must destroy the enemy bomber by spelling the mystery word. In this exciting and educational game the bomber gets closer with each inaccurate letter. You have only EIGHT tries to guess the mystery word or your ship will be bombed! If you guess the word correctly, GENERAL QUARTERS will sound and your ship will fire a missile to destroy the bomber. Three levels are available: EASY, MEDIUM, and HARD. Challenging for all ages!

Atari 16K . . . . . Tape: $18.95
CoCo 16k ECB . . . . . Tape: $18.95 Disk: $22.95
Vic 20 13k . . . . . Tape: $18.95

SPELLING BEE

The word is pronounced vocally and it is up to you to type in the correct spelling. If wrong, the computer will be your friend and flash the word on the screen for just an instant. OK! Try typing the word in again. STILL WRONG! The computer wants success and allows you to see the word again this time a little longer. If you just can't spell the word, the computer realizes you need to learn to spell the word and leaves the word on the screen for you to copy. Try your best and the computer has a surprise for your reward!

SPELLING BEE I . . . GRADE 1 & 2 SPELLING BEE III . . . GRADE 5 & 6
SPELLING BEE II . . GRADE 3 & 4 SPELLING BEE IV . . . GRADE 7 & 8

CoCo 16k ECB . . . . . TAPE: $16.95 Each

TC–INVENTORY

Many insurance companies offer a discount for policy holders which have complete inventories on file. TC – Inventory is designed to help you organize, maintain, and compile the personal belongings of your home. Program is user friendly and menu driven. TC – Inventory allows input for location of item, price of item, serial number of item, date of purchase, and a text written description of the item. Don't put off recording your personal belongings until its too late. Requires printer for hard copy.

CoCo 32k ECB . . . . . Tape: $16.95

TEACHING CLOCK

Torn between teaching time on a digital or a conventional (face and hands) clock? Well, this program combines the two using high resolution graphics and prompts! Your child will learn to tell time with the aid of a specially designed CLOCK! Child enters the time, if wrong, the center of the clock displays a graphic aid. If the child is correct a musical reward is heard. Program offers three levels: hours, quarter hours, and five minute intervals.

Apple 48k . . . . . Disk: $19.95
Atari 32k . . . . . Tape: $16.95
CoCo 16k ECB . . . . Disk: $19.95 Tape: $16.95

MasterCard

Additional Educational Software available for Color Computer, TDP 100, Atari ®, Apple ®, Commodore 64 ®, and VIC 20 ®.

VISA®

390

P.O. Box 2477 Gaithersburg, Maryland 20879 (301) 963-3848

The Right Stuff

HOT CoCo magazine is the stocking stuffer that gives color computerists information for improving their computing skills. From programming to game playing, you won't find a more exciting and practical gift for the Color Computer users on your shopping list.

Every month **HOT CoCo** publishes business applications, games, programming utilities and educational programs for children and adults. Plus **HOT CoCo** offers software and hardware evaluations and comparisons that help Color Computer owners make the right buying decisions.

As a source of useful software and a tutor for the novice and intermediate user, **HOT CoCo** will be used and appreciated throughout the year.

Take advantage of this special gift giving offer. Order now and give 12 months of **HOT CoCo** for only **$24.97.**

YES! I'd like to give an exciting and practical gift!

Send a one year (12 month) subscription to **HOT CoCo** to:

Name_____

Address_____

City_____ State_____ Zip_____

☐ Check Enclosed ☐ MC ☐ AE ☐ VISA ☐ Bill me $24.97 for 12 issues

Please make checks payable to HOT CoCo

Card #_____ Exp. Date_____

Signature_____

My Name_____

Address_____

City_____ State_____ Zip_____

Canada & Mexico $27.97, 1 year only, US funds drawn on US bank. Foreign surface $44.97, 1 year only, US funds drawn on US bank. Foreign airmail please inquire. All gift subscriptions begin with the January 1985 issue.

*TRS80 Color Computer is a registered trademark of the Radio Shack Division of Tandy Corp.

64AR4

CW Communications/Peterborough • HOT CoCo • PO Box 975 • Farmingdale, NY 11737

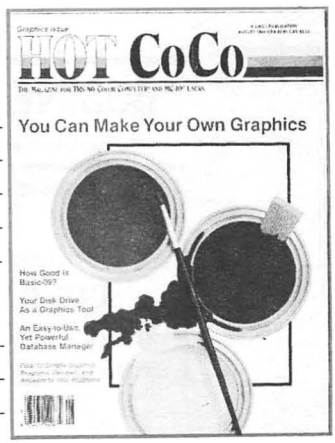
YES! I'd like to give an exciting and practical gift! Send a one year (12 issue) subscription to HOT CoCo for $24.97.

☐Check Enclosed ☐MC ☐VISA ☐AE
☐Bill me $24.97 for 12 issues

Card #_____ Exp. Date_____
Signature_____
My Name_____
Address_____
City_____ State_____ Zip_____

Please enter a one year gift subscription to:

Name_____
Address_____
City_____ State_____ Zip_____



Canada & Mexico $27.97, 1 year only, US funds drawn on US bank. Foreign surface $44.97, 1 year only, US funds drawn on US bank. Foreign airmail please inquire. All gift subscriptions begin with the January 1985 issue. Please make checks payable to HOT CoCo.

64AB4C

CW Communications/Peterborough • HOT CoCo • PO Box 975 • Farmingdale, NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

First Class Permit No. 73 Peterborough, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

CW Communications/Peterborough
HOTCoCo
PO Box 975
Farmingdale, NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

First Class Permit No. 73 Peterborough, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

CW Communications/Peterborough
HOTCoCo
PO Box 975
Farmingdale, NY 11737

ever whenever possible.

Program Listing 12 uses a nested FOR...NEXT loop to flash the screen through its eight colors at a rate that you determine in line 10. Lines 20 and 30 limit this rate. Line 40 is the loop for the screen color. Line 60 is the loop for the time delay. Line 70 shows how you can combine two NEXT statements. Since FOR A was before FOR T, NEXT T must precede NEXT A. One loop must be completely inside the other (nested).

Your First Graphics Command

SET is the slowest graphics command, but it is simple and sometimes more suited to the program than any other method. To create graphics you position small blocks of color to light up patterns or figures.

The SET command divides the screen into 64 columns numbered 0 to 63 and 32 rows numbered 0 to 31. This is a total of 32*64 or 2,048 small rectangles of light that you can turn on.

Three numbers must follow the command: i.e., SET(X,Y,Z). The first value determines the column. Column 0 is at the left side of the screen, column 63 at the right. The second number represents the row. Row 0 is at the top and row 31 at the bottom. The third number indicates your color choice. Figure 1 is a color chart.

```
10 A=A+1
20 PRINTA;
30 IF A=1000 THEN STOP ELSE 10
```

Program Listing 5

```
10 FOR A=1 TO 1000
20 PRINTA;
30 NEXTA
```

Program Listing 6

```
10 CLS RND(8)
20 FOR A=1 TO 500
30 NEXT A
40 GOTO10
```

Program Listing 7

```
10 FOR A=0 TO 500 STEP 10
20 PRINTA;
30 NEXTA
```

Program Listing 8

```
10 FOR A=20 TO 1 STEP-1
20 PRINTA;
30 NEXTA
```

Program Listing 9

```
10 FOR A=1 TO 6
20 FOR B=1 TO 6
30 PRINT A*B;
40 NEXT B
50 PRINT
60 NEXT A
```

Program Listing 10

```
10 A=A+1
20 B=B+1
30 PRINTA*B;
40 IF B=6 THEN 50 ELSE 20
50 B=0
60 PRINT
70 IF A=6 THEN END ELSE 10
```

Program Listing 11

```
10 INPUT"RELATIVE DELAY (1 TO 10
00)";D
20 IF D<1 THEN 10
30 IF D>1000 THEN 10
40 FOR A=1 TO 8
50 CLS A
60 FOR T=1 TO D
70 NEXT T,A
80 GOTO10
```

Program Listing 12

```
10 CLS0
20 SET(0,0,1)
30 SET(63,0,2)
40 SET(0,31,3)
50 SET(63,31,4)
60 GOTO60
```

Program Listing 13

```
10 CLS0
20 FOR A=0 TO 63
30 FOR B=0 TO 9
40 SET(A,B,4)
50 NEXT B,A
60 FOR A=0 TO 63
70 FOR B=10 TO 21
80 SET(A,B,5)
90 NEXT B,A
100 FOR A=0 TO 63
110 FOR B=22 TO 31
120 SET(A,B,3)
130 NEXT B,A
140 GOTO140
```

Program Listing 14

```
10 CLS0
20 C=4
30 FOR B=0 TO 31
40 IF B=10 THEN C=C+1
50 IF B=22 THEN C=C-2
60 FOR A=0 TO 63
70 SET(A,B,C)
80 NEXT A,B
90 GOTO90
```

Program Listing 15

```
10 CLS0
20 FOR A=0 TO 63
30 SET(A,6,2)
40 SET(A,7,3)
50 NEXTA
60 GOTO60
```

Program Listing 16

```
10 CLS0
20 FOR A=0 TO 63
30 SET(A,7,2)
40 SET(A,8,3)
50 NEXTA
60 GOTO60
```

Program Listing 17

```
10 CLS0
20 FOR A=1 TO 31
30 SET(A,A,3)
40 SET(A*2,A,4)
50 SET(A/2,A,5)
60 NEXTA
70 GOTO70
```

Program Listing 18

```
10 CLS8
20 RESET(5,5)
30 FOR A=5 TO 10
40 RESET(A,10)
50 NEXT A
60 GOTO60
```

Program Listing 19

```
10 CLS0
20 FOR A=22 TO 30
30 SET(A,7,8)
40 SET(A,13,8)
50 NEXT A
60 FOR A=8 TO 12
70 SET(22,A,8)
80 SET(30,A,8)
90 NEXT A
100 SET(21,9,8):SET(31,9,8)
110 SET(25,9,8):SET(27,9,8)
120 FOR A=24 TO 28
130 SET(A,11,8)
140 NEXT A
150 FOR A=14 TO 23
160 SET(26,A,1)
170 NEXT A
180 FOR A=16 TO 25
190 SET(A,16,1)
200 NEXT A
210 FORA=27 TO 33
220 SET(A,15,1)
230 NEXT A
240 SET(18,15,1):SET(33,16,1)
250 FORA=22 TO 25
260 SET(A,23,1)
270 NEXT A
280 FOR A=27 TO 30
290 SET(A,22,1)
300 NEXT A
310 FOR A=1 TO RND(50)
320 IF RND(2)=1 THEN RESET(25,9)
 ELSE RESET(27,9)
330 FOR A=1 TO RND(50)
340 IF RND(2)=1 THEN SET(25,9,8)
 ELSE SET(27,9,8)
350 GOTO310
```

Program Listing 20

The Basic Beat

"Lighting each of the 2,048 positions individually would be a rather lengthy program, but fortunately there is an easier way."

In Program Listing 13, line 20 lights the upper left corner green, line 30 lights the upper right yellow, line 40 lights the lower left blue, and line 50 lights the lower right red.

Lighting each of the 2,048 positions individually would be a rather lengthy program, but fortunately there is an easier way. Program Listing 14 uses FOR...NEXT loops to create a red, white, and blue screen. Lines 20–50 place red on the top, lines 60–90 create the white middle and lines 100–130 do the blue bottom. Line 140 keeps the computer busy so that the OK message and a green stripe do not mess up my great display.

Listing 14 is straightforward but a little tedious. Lines 20, 60, and 100 are all the same program. Listing 15 creates the same display, but a little cleverness with IF...THEN statements changes the colors at the correct time.

One fact complicates SET graphics. You cannot SET two different colors in one PRINT@ position. There are 512 PRINT@ positions (numbered 0 to 511) on the screen. There are 32 PRINT@ positions across and 16 down. This is half the number of SET positions in each direction. Therefore, each PRINT@ position contains four SET positions in a pattern of two across by two down.

Program Listings 16 and 17 show the problems that can result from this one limitation. Both listings create a yellow stripe and a blue stripe adjacent to each other across the screen. Line 40 in Listing 16 makes a wide blue stripe as it erases the yellow created in line 30.

Listing 17 does draw adjacent yellow and blue lines. What's the difference? Listing 16 places yellow at six SET positions down and blue at seven SET positions down, but both occupy the same PRINT@ position and cannot be different colors.

1 green
2 yellow
3 blue
4 red
5 buff
6 cyan
7 magenta
8 orange

Fig. 1. The Eight Available Colors and Their Corresponding Numbers.

To make a guide sheet, find the SET position page in your Color Basic manual. Darken the horizontal and vertical lines between SET positions 1 and 2. Next darken both lines between SET positions 3 and 4.

Continue darkening lines in this pattern (every other line) over the entire sheet. You'll have 512 boxes, each containing four SET positions in a 2-by-2 pattern. I photocopied such a sheet to use as a graphic design page on which I can draw with colored pencils. I can color any small square any of the eight colors, but no two boxes within an enclosed darkened four-box area can be different colors. They can only be some combination of black and one color.

Program Listing 18 will help you draw diagonals. It's handy for houses.

RESET turns SET graphics off. Two numbers, the horizontal and vertical coordinates, follow the RESET command. These are the same as for SET, except that you don't need a number for the color, because RESET turns any color to black.

Program Listing 19 demonstrates how to reset one point in line 20 and how to darken a strip in lines 30–50. Can you program the CoCo to RESET an area by using nested FOR...NEXT loops? I wrote Program Listing 20 for some reason, I forgot why. It must display possibilities of SET and RESET.

Next month I'll show you how to develop a small graphic game. A game requires a player to interact with the computer, but INPUT is the only command since the reintroduction of the Basic Beat that lets you do so, and it has its limitations.

I'll also give you some ways to control simple arcade action with and without joysticks. ■

✓ *See List of Advertisers on page 97*

BY SCOTT L. NORMAN

THE DO-MOST-EVERYTHING SPREADSHEET—DYNACALC

Add enhancements to a great Flex program, convert it to run on RS DOS, and what do you have?

Dynacalc v. 5.0:3
Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
64K, one disk drive (minimum)
$99.95

Dynacalc has been one of the best pieces of Flex software around. Now this powerful spreadsheet has been converted to run under the standard CoCo DOS, and something has been gained in the translation. The new Dynacalc features a versatile, easy-to-use graphics package, making it a simple matter to graph the data in any portion of a sheet. This is probably the first meaningful example of integrated software for the Color Computer.

I especially like Dynacalc's graphics routine, but there are lots of other nice things about the program, too. Computer Systems Center has refined many features of the Flex-based version and added several useful mathematical and logical functions.

It is even possible to use a joystick or mouse to move the cursor about over a worksheet.

Getting on the Air

Dynacalc still reminds me of Flex software in many ways. The program's master disk itself is copy-protected, but you can use it to make as many bootable working copies as you please.

You build working copies of Dynacalc by running a disk routine called Create. This routine also handles the installation chores, allowing you to customize the program to your own printer and disk drives. You can temporarily change any of the selections from within the program once it is running.

Code numbers identify four generic types of printers: Radio Shack, Epson/Gemini, Okidata, and C. Itoh/NEC. Owners of most other dot-addressable machines can probably use one or another of the codes, and tinker a bit with DIP switches to get satisfactory results. There are a few subtleties that you must deal with before you can get the best-looking graphs, even on the standard printers.

The selection of drive-stepping speeds ranges from an unidentified zero (the fastest) to an equally unidentified three (the slowest, and supposedly the default, but see below). Were I a betting man, I'd say that these correspond to 6, 12, 20, and 30 milliseconds per step; in any event, exercise caution when making a selection.

It is very worthwhile to make up a practice spreadsheet or two and then save and reload the data to make sure that your drives can handle a new speed. My drives (one MPI, one Radio Shack/TEC) work perfectly well at setting 2.

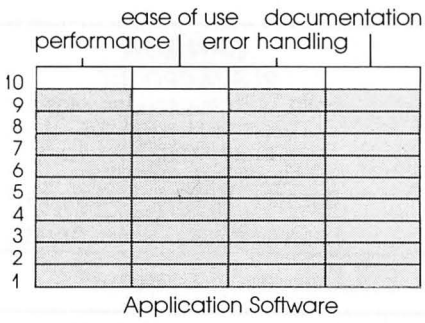
Dynacalc does not actually run under the Color Computer's original DOS (the one in ROM). Instead, it automatically loads its own operating system (operationally identical to the Radio Shack version) into high RAM from the disk.

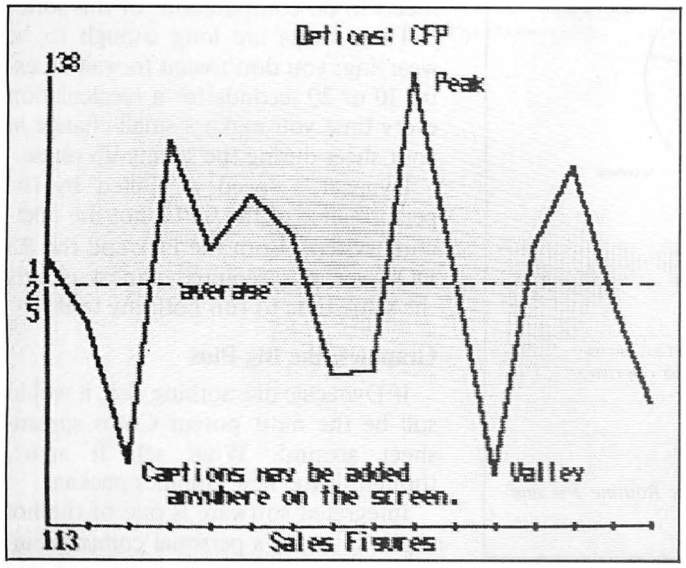
A small bug in the current version of the program requires that you always respond to the question, "Change disk step rate?" with a Y—even if you intend to use the slowest value, in which case you would then specify setting 3. Apparently, the system can assign the wrong default value if left to itself.

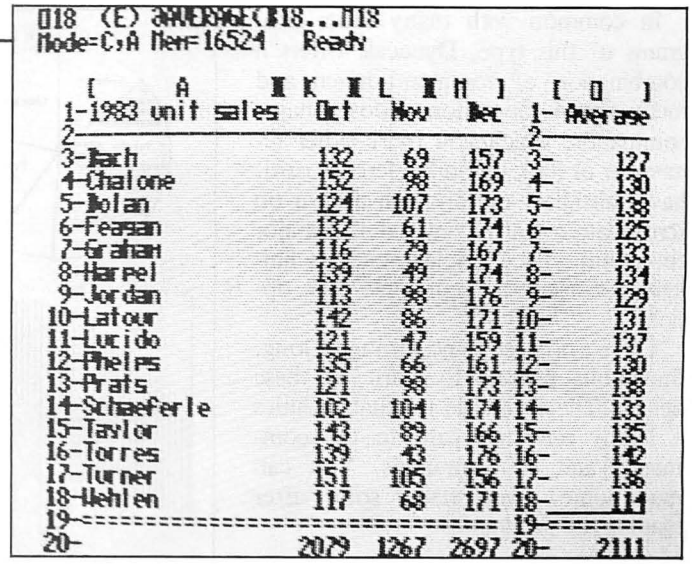
Performance and Ease of Use

To set up any spreadsheet, you must fill various cells with values (numbers), labels (pieces of text), or formulas that express relationships between other cells. These programs can perform speedy recalculations of all the relationships whenever you change a piece of input information.

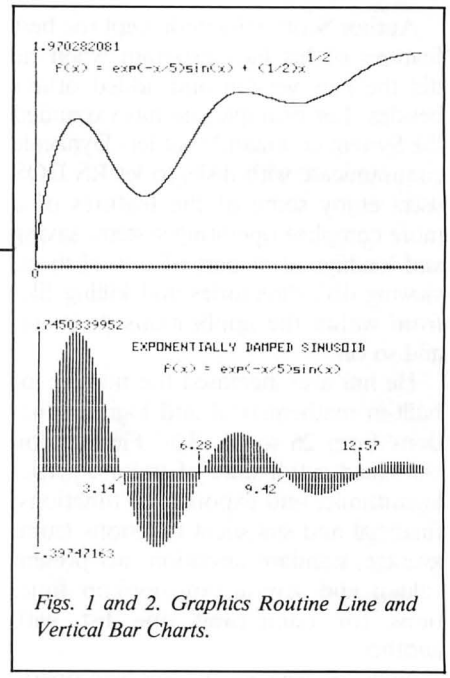
Spreadsheets differ chiefly in the number and power of their mathematical and logical functions, display for-







Photos. Dynacalc's High-Resolution-Screen Displays



Figs. 1 and 2. Graphics Routine Line and Vertical Bar Charts.

matting options, and modes of interaction with the operating system and other programs. Dynacalc rates very high scores on all counts.

It loads itself in two stages: first the actual spreadsheet and graphics program, and then some help files. These contain several screens of on-line reminders about command syntax, function notation, and so on. You can add almost 8,500 bytes to the RAM available to a spreadsheet if you delete the help screens, but I wouldn't advise doing so unless you are strapped for space. The new Dynacalc has a *lot* of command options and functions, and even though the manual is organized in a simple alphabetical fashion, it's nice to have assistance available at the touch of a key.

The program uses a 51-by-24 display format, which shows 21 rows and five columns of data (default column width is nine spaces). The remaining space is occupied by typical spreadsheet tools: row- and column-coordinate markers, status messages, indications of the available command options, an editing area, and so on. A blank sheet has 256 possible rows and columns, although you can only use 3,000 or so of the resulting cells at any one time—a limitation common to 64K computers of all varieties.

There are five methods of moving the cursor (a highly visible oscillating bar) around a Dynacalc worksheet. As in the Flex version, you can press an arrow key once to advance by a single cell or use a keyboard GOTO command to jump to any specified address. The new Dynacalc also offers an auto-repeat function, so simply holding one of the arrow keys down for a half-second or so causes the cursor to scoot off.

One of the status lines on the display keeps you informed of the current location, but the movement is so fast that it can be very difficult to control the cursor's stopping point precisely. This technique is probably best for browsing through a large sheet.

In the fourth method, you plug a joystick or mouse into either of the CoCo's joystick ports. Pressing the fire button transfers cursor control to the peripheral device, but you cannot move beyond the portion of the spreadsheet visible on the video display. To go further afield, you must first return to keyboard control to shift your vantage point.

The last mode of cursor movement is limited in a different way, but still serves an important purpose. A locate-label command jumps the cursor to a location that you specify not by its coordinates, but rather by its contents—as long as it contains a text string, that is.

The command is /L (major Dynacalc commands begin with a slash character), and it produces a screen prompt for the search-target string. This displays the address of the first "hit," along with its entire contents, and you then can choose to jump the cursor to that location or look for the next occurrence of the target.

/L is a substring search, and will locate the target no matter where it occurs in a label. You can also specify wild-card characters and search to distinguish between upper- and lowercase versions of the same letter. This command is useful for locating one item in a large, complex sheet—very much like the "find" feature of most word processors.

Dynacalc has a complete set of commands for formatting and editing a worksheet, replicating formulas so that you can use a few keystrokes to perform similar calculations on many cells, loading and saving disk files, and generally behaving as a mature spreadsheet calculator should.

In common with many other programs of this type, Dynacalc offers a combination of command-driven and menu-driven operation. Most major commands, which you must either remember or look up on a reference card, have subsidiary options that appear on screen after you've entered the major command. Since display space is limited, a single letter represents each option.

There's no need to go through a long, drawn-out process to learn all these details. The Dynacalc manual includes a handy reference card to the commands and subcommands. You can manipulate spreadsheets soon after loading the program.

Enhancements

Author Scott Schaeferle kept the best features of his Flex program when he did the new version and added others besides. For example, he has expanded the System command that lets Dynacalc communicate with disks to let RS DOS users enjoy some of the features of a more complete operating system: saving and loading all or part of a worksheet, viewing disk directories and killing files from within the applications program, and so on.

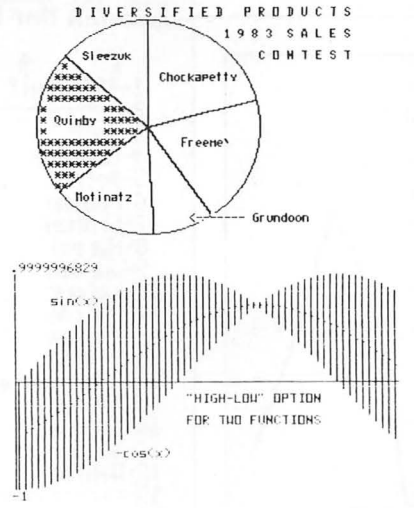
He has also increased the number of built-in mathematical and logical functions from 26 to 36. The Flex edition contained a full suite of trigonometric, logarithmic, and exponential functions; financial and statistical functions (sum, average, standard deviation, net present value); and several table-lookup functions for comparing one list with another.

Now you also get the Boolean operations AND, OR, NOT, and E(xclusive) OR, and the very useful IF decision-making function, among others. You can construct spreadsheets that contain much of the power of Basic's IF... THEN...ELSE statement.

There have also been a number of more subtle changes, mostly for the better. I have the impression that the procedures for editing formulas and labels have been improved, for one thing. Unfortunately, it is also true that some of the key-redefinition sequences have been changed in a manner that guarantees some confusion for old Flex hands. hands.

Some of Dynacalc's new key selections are more logical than others, but in any event they should pose few problems to people who don't constantly switch back and forth between the two versions.

Another modest but welcome improvement is the new keysaver command, /K, with which you can define a sequence of keystrokes that you want repeated several times—a macro, if you will. You can use it, for example, to insert or delete several rows or columns in the middle of an existing spreadsheet.



Figs. 2 and 3. Graphics Routine Pie and High/Low Bar Charts.

Keysaver lets you have the command sequence repeated automatically. Suppose you want 10 new columns. The procedure is to enter the command, /IC, which generates the first one, then type /K9 before you do any further work on your sheet. This repeats the previous command nine times, and there are your new columns.

You can use keysaver with any Dynacalc commands, although it seems best suited to things like format modification, cell blanking, and generating borders and underlines in title areas.

Although I have not run any standardized benchmarks to check Dynacalc's speed, I did time the program as it went through some sample calculations involving a lot of floating-point math.

First, I computed 100 angles from 0 to 6.28 radians and their sines. Then I tried a 50-row table in which each row contained an integer n, two raised to the nth power, and the nth root of the latter (which was identically two, of course). In other words, the three columns of the table were as follows:

n 2↑n (2↑n)↑(1/n)

I deliberately avoided simplifying things by using cell references; each 2↑n had to be calculated once for the middle column and again for the last one.

Dynacalc required approximately 9.8 seconds to do the angle-and-sine calculation, and about 18.5 seconds for the powers-of-two table. Therefore, you might want to disable the automatic-recalculation feature when building spreadsheets to do computations of this sort.

These times are long enough to be wearying; you *don't* want to wait an extra 10 or 20 seconds for a recalculation every time you make a small change in your sheet during the setting-up phase.

Dynacalc's speed is limited by the performance of the CPU, not the operating system. Both the Flex and the RS DOS versions required almost exactly the same time to run both my tests.

Graphics, the Big Plus

If Dynacalc did nothing else, it would still be the most potent CoCo spreadsheet around. What sets it apart, though, is the new graphics package.

Integrated software is one of the hot topics in today's personal computer industry. Most 16-bit machines can run one or more packages that can pass data back and forth between word processors, spreadsheets, data-base managers, and graphics routines—or some combination thereof.

That sort of interaction can offer you a tremendous amount of freedom to organize information and is responsible for much of the excitement of products like Lotus 1-2-3 and the Apple Macintosh.

But integrated software also takes a lot of memory. For the most part, those with 8-bit computers have been left out in the cold. The best you could do was to try to standardize data-file formats, hoping that it wouldn't be too difficult to write translation routines that would let the output of one program serve as input to another.

Dynacalc's graphics routines take care of at least part of the problem. They let you produce several types of graphs from spreadsheet data. You can specify which part of a sheet is to be graphed, of course, and it doesn't matter whether the numbers were entered as raw data or found by calculation.

You can then print the graphs or save them on disk, and full-screen editing lets you add captions, legends, or other annotations.

Several spreadsheet programs, including the Flex version of Dynacalc, let you turn a column of data into a horizontal bar graph. Such graphs appear right on the sheet itself, and usually take the form of rows of asterisks or other characters.

In contrast, the new graphics routines generate four different types of high-resolution graphs (see Figs.): line charts, vertical bar charts, pie charts, and the special high/low bar graphs commonly used to depict stock prices.

1983 unit sales	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total	Average	Best	Worst
Bach	136	139	119	161	130	104	84	121	95	115	75	161	1440		161	75
Chalone	120	170	152	170	182	102	89	157	162	129	64	158			182	64
Dolan	188	157	103	112	161	122	99	145	145	103					188	97
Feagan	105	94	127	115	157	97	61	132	113						174	61
Graham	135	135	183	116	151	104	86	149							183	63
Harpel	134	102	190	161	180	85										
Jordan	105	109	188	171	120											
Latour	112	128	124	129												
Lucido	158	110														
Phelps	167															
Prats																75
Schaeferle															193	78
Taylor														145	190	88
Torres					131								1620	135	177	105
Turner				127	131						75	178	1635	136	190	75
Wehlen			145	142	154			137	125	106	60	151	1495	125	154	60
	2312	2166	2387	2321	2401	1699	1439	2276	2242	2011	1318	2631	25203	2100		

DYNACALC®

FOR CoCo DOS

NO OTHER OPERATING SYSTEM NEEDED!

THE BEST OF BOTH WORLDS!

available from

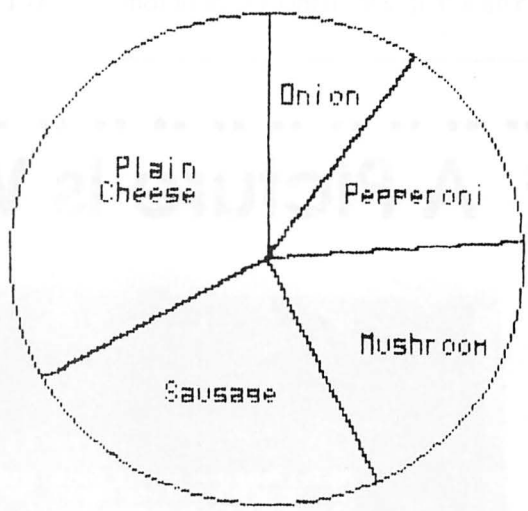
CSC

COMPUTER SYSTEMS CENTER
13461 Olive Blvd.
Chesterfield, MO 63017 USA
(314) 576-5020

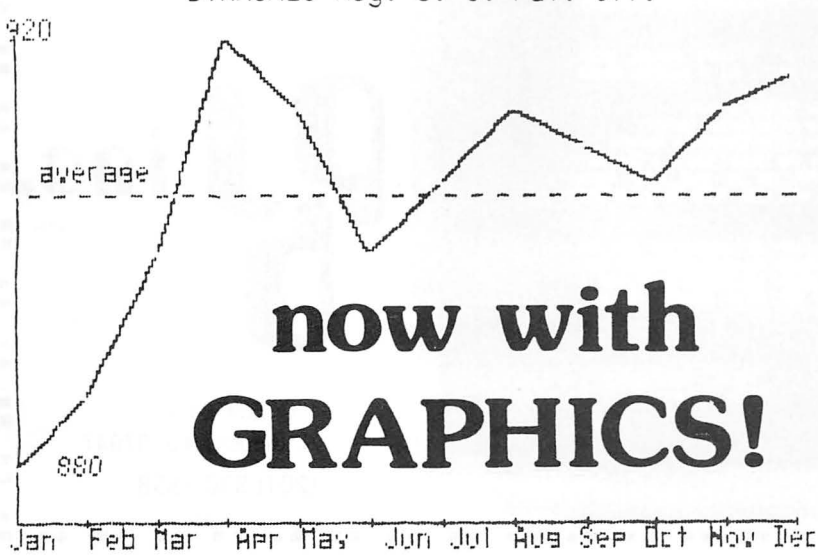
✓506

or your local DYNACALC dealer

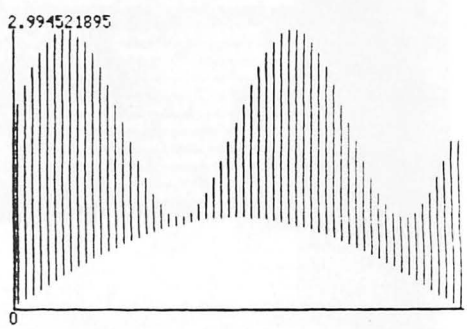
$99.95
Price ~~$150~~ postpaid in US & Canada.
Outside North America add $10 postage.



DYNACALC Reg. U. S. Pat. Off.



now with GRAPHICS!



CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular Tel. (514) 287-1563
Canadian Toll Free 800-361-5155

✓ See List of Advertisers on page 97

(Actually, these should be called high/low/close charts; you can graph a third range of spreadsheet cells to identify yet another set of data points, such as a stock's daily closing price.)

The graphs are drawn on a separate video page and are stored and printed separately from the spreadsheet itself. The routines are not perfect—I would prefer a different sort of automatic scaling, for instance—but they are quite good and convenient to use.

In creating charts and graphs, I originally configured Dynacalc to send information to my C. Itoh Prowriter at 9,600 baud and encountered no difficulties at all with spreadsheets. At first, there was a problem with graphics: I was unable to get anything sensible out of the printer unless I slowed things down to 2,400 baud.

The problem arose because I was using the Prowriter's optional 1K buffer to store data from the computer. When I switched to the single-print-line buffer mode by moving one of the printer's DIP switches, everything worked beautifully at 9,600 baud. Owners of other printers should experiment.

You will have to change the print font from time to time, however. Because video pixels and printed characters have different aspect ratios (height-to-width relationships), there will be some distortion between the video display and the printout. It's acceptable in most cases, but, with normal fonts, a pie chart that looks circular on the screen will print as a squashed, elliptical object.

The best solution is to shift the printer into condensed mode, usually 17 characters per inch. This minimizes the distortion, although it reduces the size of the entire graph.

The Bottom Line

Dynacalc has many other features, and I should mention the program's ability to generate disk files that a word processor can read.

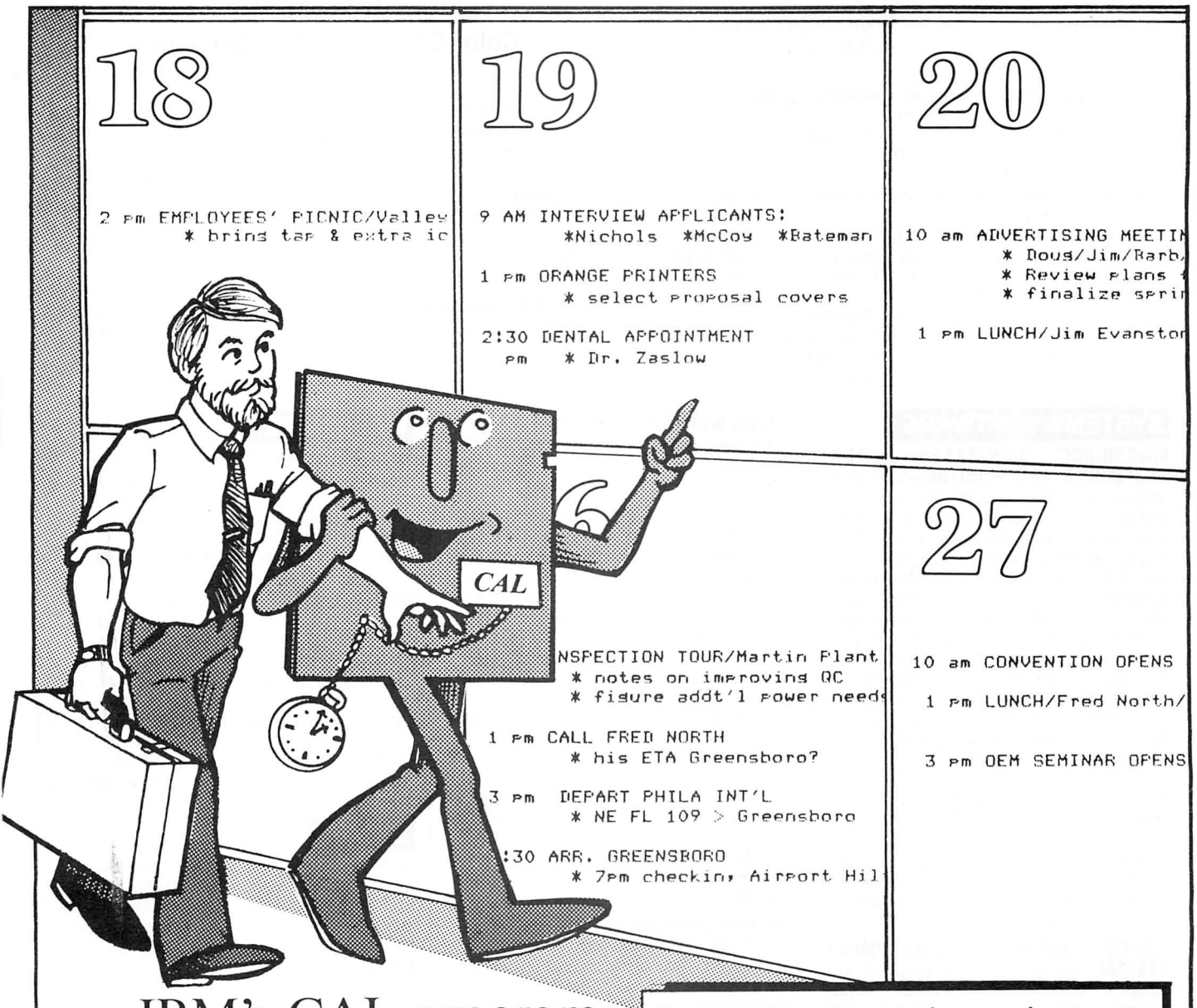
The output-to-textfile command, /O, produces an ASCII version of a spreadsheet with a single carriage return at the end of each line. You can change several print parameters, such as page length and width, and incorporate the resulting file into other documents. I tried this with the ASCII I/O version of Telewriter-64, and it worked quite well.

As I have indicated, this is a full-featured program, quite capable of performing in a professional environment. The error handling is excellent—you have to work at it to lose any of your data—and the documentation is very good, if crowded.

It would be nice to have better graph-scaling routines and to be able to use the PBJ Word-Pak for an 80-column spreadsheet display, but those are just my own fetishes. This is still a great product and my leading contender for the nonexistent title of Color Computer Program of the Year. ■

> *"This is still a great product and my leading contender for the nonexistent title of Color Computer program of the year."*

WHO? WHAT? WHEN? WHERE?
Let CAL Help You Prepare. . .



JBM's CAL program ($69) is the fastest, simplest, most comprehensive schedule manager available for OS9* (all levels).

MasterCard

VISA

For more information or to place an order, contact:

Dept. HC 13

the JBM group

The JBM Group, Inc.
Continental Business Center
Front & Ford Streets
Bridgeport, PA USA 19405
TEL: 215-337-3138
TWX: 510-660-3999

190

VISA/MASTERCHARGE accepted. PA res. add 6% sales tax.
US orders, add $5.00 postage and handling.

* OS9 is a registered trademark of Microware Corp.

See List of Advertisers on page 97

NEW GOOD STUFF FOR EVERY COLOR COMPUTER

Turn your Color Computer into a graphic design center with the ease of a keystroke! **MagiGraph** makes it simple to create highly detailed figures up to and including an entire high-resolution screen. Designed for those with some experience in Basic and Assembly Language programming, **MagiGraph** includes lots of special features:

- **A full set of logical and pixel manipulation functions** simplifies the development of complex figures.
- **An editor** lets you zoom in and work on every detail of your design. Toggle between the "macro" and "micro" screens for perspective on your creations.
- **Nine animation buffers** allow you to preview each sequence to ensure continuity and smooth flow.
- **Versatile I/O routines** store a graphic screen on cassette or floppy disk; recall it later for use by another program or revise it with **MagiGraph**.

If you're looking for the finest graphic development utility available for your Color Computer, THIS IS IT. Maximize your machine's potential, while you push your imagination to the limit — with **MagiGraph**!

By Kevin Dooley. Cassette **$34.95** (16K required); Disk **$39.95** (32K Extended Color BASIC required); Amdisk cartridge **$44.95.**

CSPOOL
Color Computer Print Spooler

Stop Waiting Around for the Printer! **CSPOOL** allows you to use your printer and computer concurrently, takes only 26 bytes of Color Basic's memory, and gives you 32K of print buffer. It's like having two computers in one! By intercepting characters sent to the printer and storing them in the upper 32K of RAM, **CSPOOL** allows you to run other programs while your printer is doing its job. **CSPOOL** is FREE with the purchase of a 64K RAM UPGRADE KIT from The Micro Works, or it may be purchased separately on cassette or diskette for **$19.95**. Requires 64K; not for FLEX or OS9.

64K MEMORY UPGRADE KIT: For Rev. levels E, ET, NC, TDP-100s, and Color Computer II. Eight prime 64K RAM chips, instructions, and CSPOOL: **$64.95.**

SYSTEMS SOFTWARE

MACRO-80C: DISK-BASED EDITOR, ASSEMBLER AND MONITOR—With all the features the serious programmer wants, this package includes a powerful 2-pass macro assembler with conditional assembly, local labels, include files and cross referenced symbol tables. MACRO-80C supports the complete Motorola 6809 instruction set in standard source format. Incorporating all the features of our Rompack-based assembler (SDS-80C), MACRO-80C contains many more useful instructions and pseudo-ops which aid the programmer and add power and flexibility. The screen-oriented editor is designed for efficient and easy editing of assembly language programs. MACRO-80C allows global changes and moving/copying blocks of text. You can edit lines of assembly source which exceed 32 characters. DCBUG is a machine language monitor which allows examining and altering of memory, setting break points, etc.

Editor, assembler and monitor—along with sample programs—come on one Radio Shack compatible disk. Extensive documentation included. By Andy Phelps. **$99.95**

SDS-80C: SOFTWARE DEVELOPMENT SYSTEM—Our famous editor, assembler and monitor in Rompack. Like MACRO-80C, it allows the user to write, assemble and debug assembly language programs with no reloading, object patching or other hassles. Supports full 6809 instruction set. Complete manual included. **$89.95**

MICROTEXT: COMMUNICATIONS VIA YOUR MODEM! Now you **can** use your printer with your modem! Your computer can be an intelligent printing terminal. Talk to timeshare services or to other personal computers; print simultaneously through a second printer port; and re-display text stored in memory. Download text to Basic programs; dump to a cassette tape, or printer, or both. Microtext can be used with any printer or no printer at all. It features user-configurable duplex/parity for special applications, and can send any ASCII character. You'll find many uses for this general purpose module! ROMPACK includes additional serial port for printer. **$59.95**

MICRO WORKS COLOR FORTH

- Faster to program in than Basic
- Easier to learn than Assembly Language
- Executes in less time than Basic

The MICRO WORKS COLOR FORTH is a Rompack containing everything you need to run Forth on your Color Computer. COLOR FORTH consists of the standard Forth Interest Group (FIG) implementation of the language plus most of FORTH-79. It has a super screen editor with split screen display. Mass storage is on cassette. COLOR FORTH also contains a decompiler and other aids for learning the inner workings of this fascinating language. It will run on 4K, 16K, and 32K computers. And COLOR FORTH contains 10K of ROM, leaving **your** RAM for **your** programs! There are simple words to effectively use the Hi-Res Color Computer graphics, joysticks, and sound.

Includes a 112-page manual with a glossary of the system-specific words, a full standard FIG glossary and complete source listing.

MICRO WORKS COLOR FORTH . . . THE BEST! From the leader in FORTH, Talbot Microsystems. **$109.95**

MACHINE LANGUAGE

MONITOR TAPE: A cassette tape which allows you to directly access memory, I/O and registers with a formatted hex display. Great for machine language programming, debugging and learning. It can also send/receive RS232 at up to 9600 baud, including host system download/upload. 19 commands in all. Relocatable and reentrant. **CBUG TAPE: $29.95**

MONITOR ROM: The same program as above, supplied in 2716 EPROM. This allows you to use the entire RAM space. And you don't need to reload the monitor each time you use it. The EPROM plugs into the Extended Basic ROM Socket or the Romless Pack I. **CBUG ROM: $39.95**

SOURCE GENERATOR: This package is a disassembler which runs on the Color Computer and generates your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. A 16K system is required for the use of this cassette. **80C Disassembler: $49.95**

HARDWARE

PARALLEL PRINTER INTERFACE—Serial to parallel converter allows use of all standard parallel printers. PI80C plugs into the serial output port, leaving your Rompack slot free. You supply the printer cable. **PI80C: $59.95**

SUPER-PRO KEYBOARD—$69.95 (For computers manufactured after Oct. 1982, add $4.95)

ROMLESS PACKS for your custom EPROMS — call or write for information.

BOOKS

6809 ASSEMBLY LANGUAGE PROGRAMMING, by Lance Leventhal, $18.95

TRS-80 COLOR COMPUTER GRAPHICS, by Don Inman, **$14.95**

ASSEMBLY LANGUAGE GRAPHICS FOR THE TRS-80 COLOR COMPUTER, by Don Inman, **$14.95**

STARTING FORTH, by L. Brodie, **$17.95**

GAMES

ZAXXON—The real thing. Excellent. What more can we say? Cassette requires 32K. **$39.95**

STAR BLASTER—Blast your way through an asteroid field in this action-packed Hi-Res graphics game. Available in ROMPACK; requires 16K. **$39.95**

PAC ATTACK—Try your hand at this challenging game by Computerware, with fantastic graphics, sound and action! Cassette requires 16K. **$24.95**

HAYWIRE—Have fun zapping robots with this Hi-Res game by Mark Data Products. Cassette requires 16K. **$24.95**

ADVENTURE—*Black Sanctum* and *Calixto Island* by Mark Data Products. Each cassette requires 16K. **$19.95** each.

CAVE HUNTER—Experience vivid colors, bizarre sounds and eerie creatures as you wind your way through a cave maze in search of gold treasures. This exciting Hi-Res game by Mark Data Products requires 16K for cassette version. **$24.95**

THE MICRO WORKS

P.O. Box 1110-D
Del Mar, CA 92014
(619) 942-2400 196

California Residents add 6% Tax

Master Charge/Visa and COD Accepted

BY GARY A. LUDWICK

HOW DO YOU SPELL "VALUE"?

Star-Kits' Spell 'N Fix gives you lots of bang for as many bucks as you wish to spend.

Spell 'N Fix II
Star-Kits
P.O. Box 209-R
Mt. Kisco, NY 10549
914-241-0287
32K disk (includes original 16K version)
price determined by customer

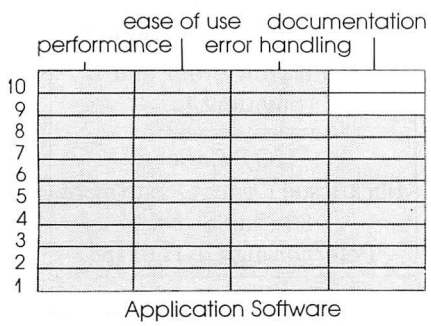
How do you review a piece of free software? Even though the *HOT CoCo* review rating graph doesn't include a scale for "value," the ratio between price and performance is a significant factor in determining the worth of any piece of software.

Peter Stark of Star-Kits Software is using a whole new marketing approach. Send him a formatted CoCo disk and he'll send you his latest version of Spell 'N Fix, absolutely free!

Is there a catch? Sort of, but we'll talk about that a little later.

Performance

The disk you'll receive actually contains two programs: the original 16K Spell 'N Fix and the updated (and considerably faster) 32K Spell 'N Fix II. Each works in a different way, and each has some outstanding attributes of its own, but I'll concentrate on Spell 'N Fix II.



It's a menu-oriented spelling checker. When you first run it, it offers you the options of correcting a document's spelling or looking up words in Spell 'N Fix's 20,000-word dictionary. It then asks if you want to create a corrected file.

You can also tell the program to check anything with a space or carriage return on either side of it, or to check only those groups of letters that reasonably seem to be words.

Once you've set the parameters, Spell 'N Fix II begins printing your text on the upper half of the screen. As it writes each word, the program checks it against the dictionary file. When it finds a word that it doesn't recognize, the text stops scrolling and the unknown word is displayed in the lower half of the screen, followed by a series of choices: A adds the word to the dictionary, F fixes the word, I ignores the word and continues the program, or Q quits the program.

If you choose to fix the word, you can either type in the correction or go to the dictionary to look it up. The program shows you where the word should appear in the dictionary, and from there you can scroll forward and backward until you find the correct spelling.

Each word displayed on the screen is numbered, and once you find the correct spelling, just touching that number puts the correction in place of the misspelling in your text.

Spell 'N Fix II is fast and accurate. It's also extremely flexible. Although designed to work with most standard word processors that generate ASCII text files, Spell 'N Fix (both versions) allows for various deviations. For instance, some word processors imbed special characters in the text for printer control or to signify the end of a paragraph. You can ask Spell 'N Fix II to ignore such characters in the proofing process.

Ease of Use

Simply put, Spell 'N Fix II couldn't be much easier to use. On-screen menus guide you almost every step of the way, and most operations are automatic. Star-Kits has even provided several utilities on the program disk to handle some special needs.

For example, Spell 'N Fix II (unlike its predecessor) cannot automatically add new words to its master dictionary. It does, however, let you create and add a file of words. Then, with a utility program called Append, you can combine your own dictionaries into one large text file. A machine-language program called Addwords automatically alphabetizes this file into your master dictionary.

You also get these utilities:

- Build, which acts as a mini word processor, allowing you to create text files on the disk;

● List, which prints disk files on the screen, without invoking the spelling correction; and
● Reset, which does with software what your reset button does mechanically. These utility programs give Spell 'N Fix II a power and flexibility seldom found in any dictionary program, regardless of price.

Error Handling

I couldn't crash Spell 'N Fix II. The only way I could get into trouble was to use the program on a single-disk setup, and try to guess where disk swaps should occur.

But this didn't work. When the program went back to the disk, expecting my text file, it found its own program text and dutifully began feeding it in for proofing.

The answer for single drive owners? The cure is now a part of every current version; although it is undocumented in my copy of Spell 'N Fix II. Simply copy the dictionary file onto your text-holding disk. It will occupy roughly half your disk storage space, but is well worth the loss if you're a single-drive owner in need of a spelling-correction program.

In every case, intentional or accidental, Spell 'N Fix II fielded each error and displayed a clear description of the problem.

Documentation

The program comes with no documentation—at least none you can hold in your hand. There is, however, a file called manual/txt on the disk, and you can print it out yourself. That's not a bad idea for a "free" program, but the information contained is pretty skimpy. You'll get nine single-spaced pages, which are enough to get you up and running, and to answer some of your more fundamental questions. If you decide to send at least $25 to Star-Kits, you'll receive the expanded manual.

Is There Really Free Software?

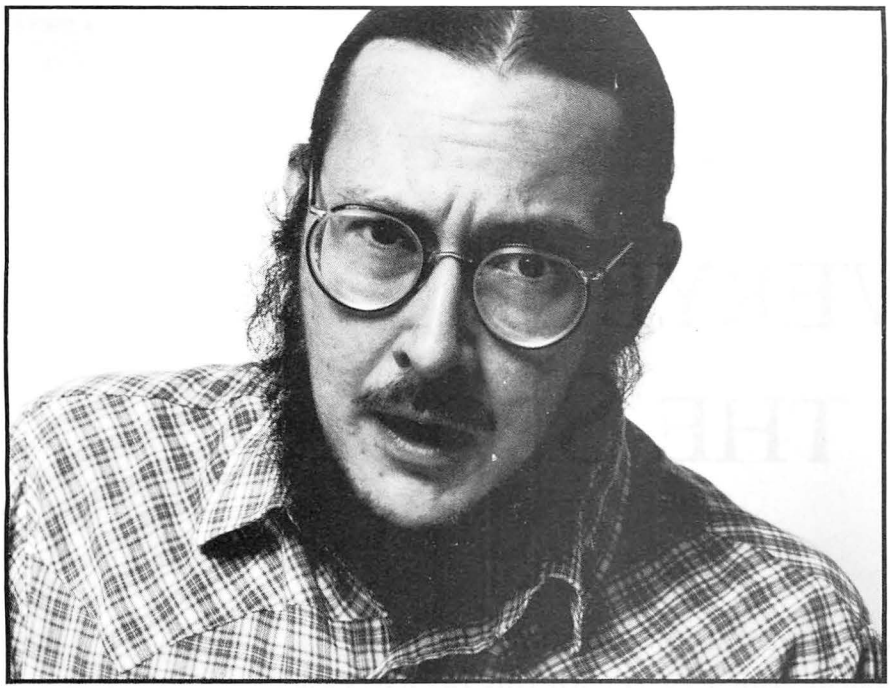
Peter Stark is hoping not. He is attempting to end-run a system that requires software writers today to have huge advertising budgets and large dealer-distribution networks. He is asking each person who gets a copy—either directly from him or from a friend—to send his company a contribution based on what they think the program is worth.

It's worth far more than $25. In my opinion, it's worth a good deal more than $75. Side-by-side comparisons with Electric Webster on a Model III show few significant differences. The Electric Webster is somewhat faster, and a little fancier, but it also costs $150 in its basic form. And Table 1 will show you how Spell 'N Fix II stacks up against Radio Shack's Color Dictionary—there really is no comparison.

Spell 'N Fix II is a superb piece of programming, and, more important, it does everything a good dictionary/spelling program should do. I suggest you send Star Kits your disk and a $50 check. You won't be disappointed. ■

	Radio Shack Color Dictionary	Spell 'N Fix II
Checks Scripsit files	yes	yes
Checks other text-processor files	no	yes
Checks Basic data files	no	yes
Checks files larger than memory	no	yes
Upper & lowercase display	no	yes
Add words to dictionary	no	yes
Delete words from dictionary	no	yes
Create custom dictionary	no	yes
Error-free dictionary	no	yes
Usable with foreign languages	no	yes
Words checked & fixed in one pass	no	yes
Shows suspect words in context	yes	yes
Program contained on one disk	no	yes
Lets you look up words in dictionary	yes	yes
Look up while correcting	no	yes
Use DIR/CMD while in program	no	yes
Uses standard Basic file format	no	yes

Table 1. A Comparison of Spell 'N Fix II and Radio Shack's Color Dictionary



DON'T GET BURNED

THE COLOR BURNER FROM GREEN MOUNTAIN MICRO

Lately I've been hearing that you want to program erasable read-only memories (EPROMs). It seems you want to create your own program cartridges, or make changes to your Basic ROMs, or turn your CoCo into some different animal.

The problem is, most EPROM programmers cost over $100, and $100 is big dues to pay. You want to burn EPROMs, not get burned in price — or quality.

So, I've put together the Color Burner, an EPROM programmer that will burn all the "27" family — 2716, 2732, 2764, 27128. Yes, it will also burn 68764 replacements for your Basic ROMs and, no, it won't break your budget.

Although my Color Burner doesn't cost a whole lot, you won't get burned over quality. I don't cut corners in hardware. I use the best fiberglass boards, with gold edges, protective solder masking and silk-screened legends. Before I send you a Color Burner, I test it by actually programming an EPROM.

So how can it be good if it's so inexpensive? First of all, you can only get a Color Burner from Green Mountain Micro. No dealers are adding to its price. Second, it isn't fancy. No high-tech power supplies are in sight. You've got to add three homely, low-tech 9-volt batteries to get it going. Finally, it won't set new standards of complexity. It's simple, hardworking and reliable.

You can get your Color Burner complete or *a la carte*: try an assembled and tested unit, a kit, or just a bare board. Order it with or without programming software. Both kits and assembled units come with over 40 pages of documentation, complete program listings, and schematics. Nothing is hidden.

You'll burn those EPROMs, you won't get burned, and my technical support staff will keep you from getting burned up if you have a question or need help.

You Won't Get Burned with The Color Burner

- Assembled/tested with software, $69.95.
- Assembled/tested only, $64.95.
- Complete kit with software, $56.95.
- Complete kit only, $49.95.
- Board/documentation with software, $30
- Board/documentation, $23
- Bare board only, $20
- ColorPack 8/16K ROM/RAM cartridge kit, $19.95.
- 2716 and 2732 EPROMs available.

Specifications:

Programs 24/28-pin EPROMs, providing 21/25-volt programming pulses under software control. Includes unwired personality module. Requires three 9-volt batteries (not included). Tape software supports 2716 through 27128 and 68764/66 EPROM families, and requires 32/64K Extended Color Basic.

ALSO AVAILABLE FROM GREEN MOUNTAIN MICRO

Lowerkit II*, $79.95 /$49.95 kit

64K Color memory upgrade kit, $49.95

MC-10 added 4K upgrade kit, $19.95

CoCoPort interface, $49.95 /$39.95 kit

RAM/ROM pack, $29.95 /$19.95 kit

Scroll-A-Roll software video text display, $24.95

TV Buff II*, improved to handle virtually all monitors, $14.95

Color Quaver, Software Music Synthesizer, $19.95

(Add $2.50 shipping and handling)
**Specify Color Computer or CoCo II*

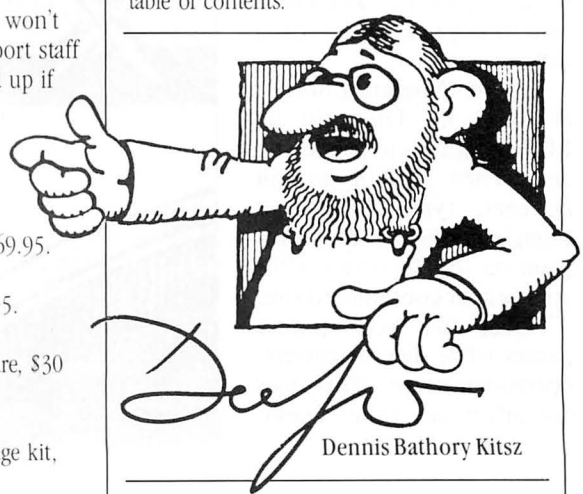
The Micro Language Lab Presents

"Learning the 6809"

- Authored by Color Computer Magazine columnist Dennis Bathory Kitsz
- 24 half-hour lessons on twelve cassettes
- 236 pages of text and complete dox for all the smart chips, including 6809E, 6821, 6847, and 6883
- 35 sample programs ready to load, list, and run
- 1,500 programmed question and answer system
- Learn not only how to put a dot on a screen, but also where and why.

$99.00 (plus $3.50 shipping and handling). *Dealer and club inquiries invited.*

Not sure? Request our sample page and table of contents.



Dennis Bathory Kitsz

Green Mountain Micro

Bathory Road, Box H ↙98
Roxbury, Vermont 05669
802-485-6112

Hours: 9am-5pm, Monday-Friday
COD/VISA/MASTERCARD
TRS-80 is a trademark of Tandy Corporation

↙ See List of Advertisers on page 97

BY FRANK TIPPS

EVERYBODY INTO THE SPOOL!

Now you can print and do something else simultaneously on your CoCo with this print-spooler.

Katherine Mahoney

Printing one program while editing another is easy to do if you have a print spooler. This position-independent routine lets you do it without spending money on a hardware print buffer or a software spooler.

Type in Program Listing 1 and run it. Then do an LLIST with your printer on line. When the OK prompt appears, type NEW and begin entering another program on the keyboard while your printer continues to run.

Notice that your printer pauses while disk or cassette operations occur. This does not affect any other operation going on. You can even run another program while printing takes place.

64K RAM
16K RAM (with modification)
Disk Color Basic
(Extended Color Basic with modification)

Initialization

Program Listing 2 shows the routine in Assembly language. With interrupts disabled, the buffer pointers are set. Lower RAM locations are set with linkage for the IRQ vector and the PRINT vector. Also, the IRQ interrupt is enabled. This causes an interrupt every 16.67 milliseconds. The baud rate is set for 9,600 baud. After each interrupt execution returns to Basic.

Print Function

When Basic wants to issue a character to a device, it jumps to a subroutine through the low RAM location &H167. I've changed this to point to the PRCHR routine. This routine checks the device number in location &H6F to see if the printer is to be used. If the de-

vice is not the printer, device −2, the routine jumps to the ROM at location &HCB4A, the ROM location for a disk system. For a tape system, the ROM location to continue is &H8273.

If the character is destined for the printer, it is put in a circular buffer in the upper 32K of RAM, which is shared with the ROMs. There are two pointers for this buffer: PRHEAD and PRTAIL.

PRHEAD points to the next open slot in the buffer; PRTAIL points at the next character to be issued. If the two pointers are equal after a character has been added to the buffer, the buffer is full. If the two pointers are equal before a character is issued from the buffer, the buffer is empty.

The buffer is a large array. However, because the pointers are reset to the beginning, &H8000, the buffer is circular if they ever reach the end at location &HFF00.

If the character just added was a carriage return, &H0D, the line-printer column position, &H9C, is reset. Otherwise, the column position is incremented and checked for the maximum value in &H9B. These steps are necessary for compatibility with other ROM functions such as TAB.

After a character has been added to the buffer, the routine returns to the original caller of the PRINT function in the ROM.

```
10 CLEAR 200,&H7F4F
20 DB$="1A508E8000AF8C1BAF8C1A30
8C58BF01D3008C13BF0168C637F7FF03
C6011D7961CAF39800008000 3414"
30 DC$="D66FC1FE2632AE8CF11A50B7
FFDFA780B7FFDE1CAF8CFF00260038E80
00AC8CDC27FBAF8CD581 0D2708"
40 DD$="0C9CD69CD19B2502 0F9C3514
326239351 47ECB4AAE8CBCAC8CB72733
F6FF2254252DB7FFDFA680B7FF"
50 DE$="DE8CFF00260038E8000AF8C9E
BDA2FB5FBDA2FDC6034045F445958BD
A2FD35045A26F2BDA2FB7ED7BC"
60 AD=&H7F50
70 DA$=DB$:GOSUB130
80 DA$=DC$:GOSUB130
90 DA$=DD$:GOSUB130
100 DA$=DE$:GOSUB130
110 EXEC &H7F50
120 END
130 I=1
140 FOR A=AD TO AD+40
150 D=VAL("&H"+MID$(DA$,I,2))
160 POKE A,D
170 I=I+2
180 NEXT A
190 AD=AD+41
200 RETURN
```

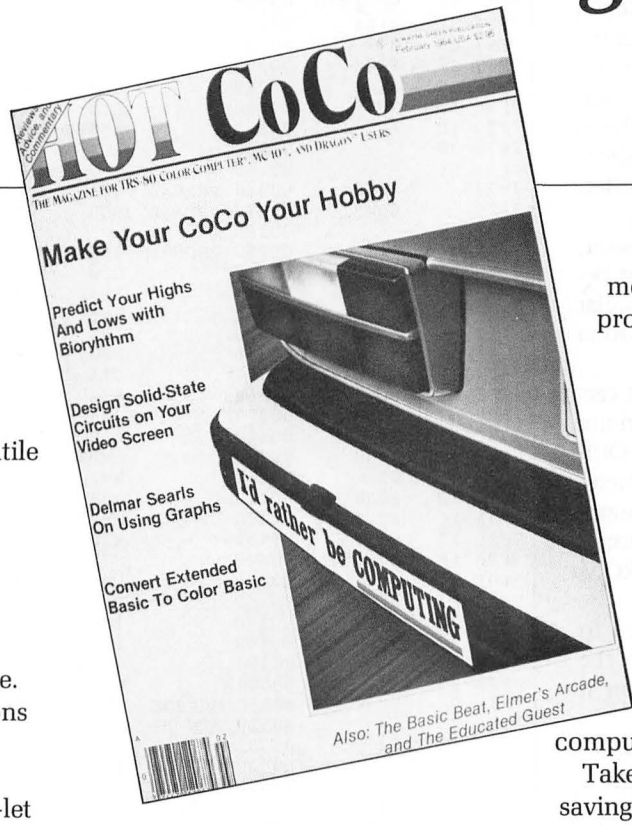
Program Listing 1. Print Spooler, Basic Version

When an IRQ occurs, the ROM vectors through RAM location &H10C to the IRQPR routine. This routine checks for an empty buffer. If the buffer is not empty, it checks for a busy printer. If either condition is true, the routine continues the ROM IRQ service routine at &HD7BC. Again, this is for a disk sys-

```
7F50                    00100          ORG     $7F50
7F50 1A   50            00110 INIT     ORCC    #$50     DISABLE INT
7F52 8E   8000          00120          LDX     #$8000   SET POINTERS
7F55 AF   8C 1B         00130          STX     <PRHEAD,PCR
7F58 AF   8C 1A         00140          STX     <PRTAIL,PCR
7F5B 30   8C 58         00150          LEAX    <IRQPR,PCR GET IRQ VECTOR
7F5E BF   010D          00160          STX     $10D     SET IRQ VECTOR
7F61 30   8C 13         00170          LEAX    <PRCHR,PCR GET PRINT VECTOR
7F64 BF   0168          00180          STX     $168     SET PRINT VECTOR
7F67 C6   37            00190          LDB     #$37     ENABLE IRQ INT
7F69 F7   FF03          00200          STB     $FF03
7F6C C6   01            00210          LDB     #1       SET BAUD RATE
7F6E D7   96            00220          STB     $96
7F70 1C   AF            00230          ANDCC   #$AF     UNMASK INT
7F72 39                 00240          RTS              RETURN
                        00250 *
7F73      8000          00260 PRHEAD   FDB     $8000
7F75      8000          00270 PRTAIL   FDB     $8000
                        00280 *
7F77 34   14            00290 PRCHR    PSHS    X,B      SAVE REG
7F79 D6   6F            00300          LDB     $6F      GET DEV NUM
7F7B C1   FE            00310          CMPB    #-2      CHECK FOR PRINTER
7F7D 26   32            00320          BNE     PRRTN    SKIP IF NOT PRT
7F7F AE   8C F1         00330          LDX     <PRHEAD,PCR GET HEAD ADR
7F82 1A   50            00340          ORCC    #$50     MASK INT
7F84 B7   FFDF          00350          STA     $FFDF    SWAP MEM BANKS
7F87 A7   80            00360          STA     ,X+      SAVE CHAR
7F89 B7   FFDE          00370          STA     $FFDE    RESTORE MEM BANK
7F8C 1C   AF            00380          ANDCC   #$AF     UNMASK INT
7F8E 8C   FF00          00390          CMPX    #$FF00   CHECK HEAD ADR
7F91 26   03            00400          BNE     CKFULL   SKIP IF NOT OVERFLOW
7F93 8E   8000          00410          LDX     #$8000
7F96 AC   8C DC         00420 CKFULL   CMPX    <PRTAIL,PCR SEE IF FULL
7F99 27   FB            00430          BEQ     CKFULL   LOOP IF FULL
7F9B AF   8C D5         00440          STX     <PRHEAD,PCR SAVE HEAD ADR
7F9E 81   0D            00450          CMPA    #$0D     CR?
7FA0 27   08            00460          BEQ     CLRPOS   SKIP IF CR
7FA2 0C   9C            00470          INC     $9C      LPT POSITION
7FA4 D6   9C            00480          LDB     $9C
7FA6 D1   9B            00490          CMPB    $9B      MAX COUNT
7FA8 25   02            00500          BLO     RTN      SKIP IF LESS
7FAA 0F   9C            00510 CLRPOS   CLR     $9C      CLR LPT POSITION
7FAC 35   14            00520 RTN      PULS    B,X      RESTORE REG
7FAE 32   62            00530          LEAS    2,S      BUMP RETURN
7FB0 39                 00540          RTS
                        00550 *
7FB1 35   14            00560 PRRTN    PULS    B,X      RESTORE REG
7FB3 7E   CB4A          00570          JMP     $CB4A
                        00580 *
7FB6 AE   8C BC         00590 IRQPR    LDX     <PRTAIL,PCR GET TAIL ADR
7FB9 AC   8C B7         00600          CMPX    <PRHEAD,PCR CHECK FOR EMPTY BUFF
7FBC 27   33            00610          BEQ     GOIRQ    CONT IF EMPTY
7FBE F6   FF22          00620          LDB     $FF22    CHECK PRT BUSY
7FC1 54                 00630          LSRB
7FC2 25   2D            00640          BCS     GOIRQ    CONT IF BUSY
7FC4 B7   FFDF          00650          STA     $FFDF    SWAP MEM BANKS
7FC7 A6   80            00660          LDA     ,X+      GET CHAR
7FC9 B7   FFDE          00670          STA     $FFDE    RESTORE MEM BANKS
7FCC 8C   FF00          00680          CMPX    #$FF00   CHECK TAIL ADR
7FCF 26   03            00690          BNE     SAVEX    SKIP IF NOT OVERFLOW
7FD1 8E   8000          00700          LDX     #$8000   RESET TAIL ADR
7FD4 AF   8C 9E         00710 SAVEX    STX     <PRTAIL,PCR SAVE TAIL ADR
7FD7 BD   A2FB          00720          JSR     $A2FB    ISSUE MARK
7FDA 5F                 00730          CLRB
7FDB BD   A2FD          00740          JSR     $A2FD
7FDE C6   08            00750          LDB     #8       GET BIT COUNT
7FE0 34   04            00760 SAVB     PSHS    B        SAVE COUNT
7FE2 5F                 00770          CLRB             CLEAR BIT OUTPUT
7FE3 44                 00780          LSRA             SHIFT CHAR
7FE4 59                 00790          ROLB             SHIFT BIT
7FE5 58                 00800          ASLB
7FE6 BD   A2FD          00810          JSR     $A2FD
7FE9 35   04            00820          PULS    B        GET BIT COUNT
7FEB 5A                 00830          DECB             DEC BIT COUNT
7FEC 26   F2            00840          BNE     SAVB     LOOP IF NOT 0
7FEE BD   A2FB          00850          JSR     $A2FB    ISSUE STOP BIT
7FF1 7E   D7BC          00860 GOIRQ    JMP     $D7BC
          0000          00870          END
```

Program Listing 2. Print Spooler, Assembly Version

Now you can learn how to use your color computer for more than just games... with HOT CoCo magazine.



With the right information on programming utilities, debugging, and graphics there's no limit to what you can do with your color computer. **HOT CoCo** gives you that information. It can make your computer a versatile tool that you'll find indispensible. **HOT CoCo** is packed with:

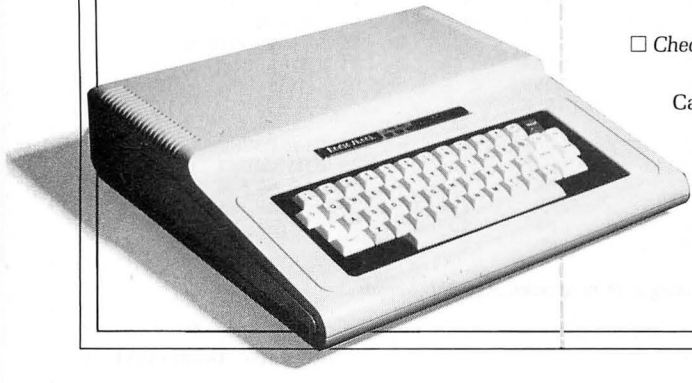
- **Business application programs**—to help you understand what the color computer can do at the office. You can use these applications immediately because they're written in plain English.
- **Home management help**—let **HOT CoCo** show you how everyday chores can be done on your machine. You'll be surprised at just what you can do and just how much time can be saved with your Color Computer.
- **Programming tips & tutorials**—**HOT CoCo** will show you how to program. It's loaded with programming techniques and hints to help the novice and expert programmer write and improve their programs.
- **New product reviews & announcements**—if you're looking for equipment to expand the use of your computer, **HOT CoCo** reviews numerous hardware and software products each month. Plus, **HOT CoCo**'s new product announcements let you comparison shop at home—spend more time at your computer and less time in computer stores.

And **HOT CoCo** is loaded with challenging games to provide hours of fun and excitement for your whole family.

Let **HOT CoCo** show you how much time you can save with your color computer. Order **HOT CoCo** today!

Take advantage of this money-saving offer. Get 12 issues of **HOT CoCo** for only $24.97. A 13th issue is yours FREE with pre-payment (check or credit card). Use the attached order card, the coupon, or call

TOLL FREE 1-800-258-5473.
IN NH CALL 1-924-9471.



☐ ***YES! I want more use from my CoCo!***

Send me 12 issues of **HOT CoCo** for $24.97 now! I understand that with payment enclosed or credit card order I will receive a FREE issue making a total of 13 issues for $24.97.

☐ *Check Enclosed* ☐ *MC* ☐ *VISA* ☐ *AE* ☐ *Bill Me $24.97 for 12 issues*

Card #_____ Exp. Date_____

Signature_____

Name_____

Address_____

City_____ State_____ Zip_____

Canada & Mexico $27.97, 1 year only, US funds drawn on US bank.
Foreign Surface $44.97, 1 year only, US funds drawn on US bank.
Foreign airmail please inquire. Please allow 6–8 weeks for delivery.

HOT CoCo • PO Box 975 • Farmingdale, NY 11737 **34AF4**

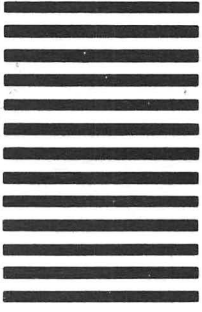
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

First Class Permit No. 73 Peterborough, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

CW Communications/Peterborough
Hot CoCo
PO Box 975
Farmingdale, NY 11737



tem. The location for a cassette system is &H894C.

The routine fetches a character from the buffer and updates the PRTAIL pointer. The routine issues the 8-bit character to the printer using sequences in the Basic ROM. This works with the 1.0 and 1.1 ROMs. The called sequences are the same, but you need to set your printer for 8 bits.

If you want to stop the spooler, just reinitialize it by doing an EXEC &H7F50. However, there might still be a partial line in your printer that hasn't been issued. Do a PRINT#−2," " to print the line held in your printer. All the buffer data will be lost.

Calling This Routine

To use this routine with your Assembly-language routines, you must observe the calling sequence. The routine has been set up to work with the ROM. When the ROM wants to issue a character, it calls a subroutine. The subroutine then calls a subroutine in low RAM. This places two return addresses on the stack.

This routine removes the top return address and returns to the original caller of the print function. Your Assembly-language routine must place two return addresses on the stack. It should use the following sequence:

```
        LDA   #$20      GET CHAR
        JSR   PRINT     PRINT CHAR/
                        RETURN
              :
PRINT   JSR   $167      CALL SPOOLER
        RTS             DUMMY...
```

Modifications

This routine was written to communicate with the printer at 9,600 baud. If your printer cannot accept characters at this speed, change the value of the baud rate according to Table 1. Make sure your printer switches are set for the proper baud rate as well.

The routine can be made to work with 16K or 32K systems, but you must decide how much RAM to use for the buffer. If the buffer is too small, you might not get much from this routine when printing a lot of information. With the changes that were indicated, you can make the routine work with a cassette system.

This routine lets you do two tasks at once: printing and performing another function. Because your computer shares the time between the two, the speed at which each operates is affected. However, by giving up a little speed, you gain a lot of versatility. ■

Baud Rate	POKE 150
1,200	40
2,400	18
4,800	7
9,600	1

Table 1. Data to Change Baud Rates

Address correspondence to Frank Tipps, 1837 Cartlen Drive, Placentia, CA 92670.

"The routine can be made to work with 16K or 32K systems, but you must decide how much RAM to use for the buffer."

✓ See List of Advertisers on page 97

BY PHILIP MCLAUGHLIN

MAKING NOISES

Sirens, rasps, explosions, screeches, swooshes and thumps —add some audio zip to your CoCo programs.

Computer graphics get so much attention that it is easy to overlook the effectiveness of sound in programming. Sound effects are a powerful means of getting attention and evoking emotional response.

While Color Basic has two commands, SOUND and PLAY, that produce sounds, this article will concentrate on PLAY. It is the more powerful of the two, and is more difficult to program and understand. The chapter devoted to PLAY in the Extended Basic user's manual explains how to use the command for reproducing simple, single-voice musical passages. By contrast, the focus here is on sound as noises; sirens, whistles, buzzers, swooshes, explosions, and so on. These are the sort of sound effects useful in computer games. Although the emphasis is non-musical, you can adapt some of the methods to simulate polyphonic (many-voiced) music.

The accompanying program demonstrates these sound effects in the order they are presented. (Type PCLEAR1 before loading.) It also acts as a library of fundamental effects that you can customize for your own programs.

Noise making with PLAY depends on the manipulation of strings and string variables, so I will review these techniques as part of the first examples. (Experienced programmers might want to skip these.) On the other hand, if you have any doubts about variable types and string operations, you should reread the appropriate chapters in the first Color Basic book.

The PLAY Command

The syntax of PLAY is like that of a PRINT statement. For instance, the command PRINT "A" causes the computer to print the character "A" on the screen while PLAY "A" results in a tone with a pitch corresponding to the musical note A. Of course, PRINT will print out almost any sequence of characters, but PLAY is more particular. Feeding it characters it doesn't understand such as a musical note that doesn't exist (PLAY "Z") causes an error message.

You can also specify the pitch of sound to be played using a number from 1 to 12. For instance: PLAY "3".

The command correctly processes characters, not values or ordinary numeric variables. The quotation marks indicate a literal string, telling the computer to take what is between them as it is without applying numerical interpretation. Literal strings can be referred to and manipulated as string variables, which the computer recognizes by a dollar sign as the last character of the variable name. For example:

```
10 SP$ = "3"
20 PLAY SP$
```

is perfectly legal.

Building complicated sounds is a matter of building complicated character strings for PLAY to process.

```
150 PLAY"1;2;3;4;5;6;7;8;9;10;11;12;"
```

In this example, line 150 plays an ascending series of pitches. The semicolon acts as a delineator between the characters representing pitches. Though mandatory here, you can often delete the semicolon if the meaning of the string is still clear. In fact it is better to do so for the sake of memory and processing time. For readability, however, the examples in this article contain all the delineators, whether needed or not.

Octaves

The computer has a greater range than just 12 pitches. You get other ranges by selecting an octave from one to five (lowest to highest). When PLAY is processing a string of characters and encounters the letter O, it expects a character representing a numeral from one to five in the next space. For instance, (PLAY "O1;1") plays the lowest pitch available and (PLAY"O5;12") the highest. So there are 5 times 12, or 60 pitches.

Suppose you want to play all the tones available from the lowest to the highest. Since there are various methods for feeding sound parameters to PLAY, I have programmed this example three different ways.

```
10 PLAY "O1;1;2;3;4;5;6;7;8;9;10;11;12;"
20 PLAY "O2;1;2;3;4;5;6;7;8;9;10;11;12;"
30 PLAY "O3;1;2;3;4;5;6;7;8;9;10;11;12;"
40 PLAY "O4;1;2;3;4;5;6;7;8;9;10;11;12;"
50 PLAY "O5;1;2;3;4;5;6;7;8;9;10;11;12;"
```

This gets the job done, but wastes space. Remembering that you can substitute string variables for strings of characters you can make a more compact version.

```
5 SP$ = "1;2;3;4;5;6;7;8;9;10;11;12"
10 PLAY "O1"
15 PLAY SP$
20 PLAY "O2;" + SP$
30 PLAY "O3;" + SP$
40 PLAY "O4;" + SP$
50 PLAY "O5;" + SP$
```

The first octave (lines 10 and 15) is done differently from the rest. In lines 20 and following, an elementary string operation of concatenation (the symbol is the plus sign) is performed to join the string of characters denoting the octave to those denoting the pitch. Notice that you must include the semicolon delineator to properly separate the parameters. Chapter 12 of *Getting Started with Color Basic* explains concatenation and other string operations more fully.

This example is a little better but it still contains some redundant code. Experienced programmers might prefer to put the PLAY function in a loop, rather than typing it out five times. The immediate difficulty is that ordinary program loops such as the FOR...NEXT type use numeric rather than string variables. Fortunately there is a function that converts values to the characters that represent them: STR$ (Y$ = STR$(X)).

You can translate this statement to read, "The string variable Y$ now stands for the printable characters needed to show the value represented by the ordinary numeric variable X." Notice the use of the string sign (dollar sign) on both sides of the equation. Only similar variable types can be equated. Mismatching them earns a type mismatch or TM ERROR.

A Controlled Loop

Lines 230–260 in the Program Listing use an STR$ conversion to step the character following the letter O from one to five. You say that the octave parameter is controlled by the loop and you can control numerous other sound factors by similar methods. Be sure to observe the following points:

- There are two types of variables, string and numeric. They are not compatible.
- String variables represent printable characters much like ordinary variables represent numerical values.
- PLAY cannot process numerical values directly, but variables can be converted from numerical type to string type with the STR$ function.
- Addition of strings or string variables causes them to be joined together or concatenated.
- Errors in construction of strings usually become obvious by substituting PRINT for PLAY.

Faster and Faster

The more interesting the sound, the more sonic information it packs into short spans of time. Accomplish this by using the T (for tempo) parameter to speed things up.

Tempos are available from one, the slowest, to 255, the fastest. Once you set a tempo (such as by PLAY "T100"), it remains at that speed until another tempo change. The default tempo is two, so what you have been hearing so far is pretty slow.

The next example sets the tempo ever

System Requirements

16K RAM
Extended Color Basic

faster, playing lines 230–260 as a subroutine:

```
320 PLAY "T20":GOSUB 230
330 PLAY "T40":GOSUB 230
335 PLAY"T100":GOSUB 230
340 PLAY"T150":GOSUB 230
350 PLAY"T255":GOSUB 230
```

Once up to speed, this sound becomes a full-fledged siren (hard to improve upon as an attention getter). Variations on sirens illustrate some further capabilities of the PLAY command.

Louder

Volume control is available with commands such as PLAY "V31". Thirty-one is the loudest setting available and one is the faintest. As was the case with tempo, the sound remains at a given volume until you reset it. You can create sounds of interesting texture by rapidly changing volume. Sounds played at only one volume level sound flat. Line 460 from the Program Listing varies the volume along with the octave of the siren. Note the arithmetic operation inside the conversion function STR$(O*6). This is legal if the result is an integer.

As manipulations become more complex, the technique of replacing PLAY with PRINT becomes increasingly helpful. For instance, if you multiplied the variable O (for octave) by seven instead of six to derive V (for volume), you'd get values higher than the permitted maximum of 31, resulting in an FC ERROR.

Once a playable string variable is defined in a program, you can use it over and over again as a building block. For example:

```
P$(1) = 1;2;3;4;5;6;7;8;9;10;11;12;"
```

This is the familiar ascending tone sequence, hereafter called P$(1). This is an array or subscripted variable. You can also define a descending sequence:

```
P$(2) = "12;11;10;9;8;7;6;5;4;3;2;1;"
```

Lines 510–540 from the Program Listing use these two elements for a full up and down siren.

Warbling Siren

Lines 610–640 of the Program Listing are a rearrangement of the basic siren. Instead of playing only ascending scales up all the octaves before starting down, they play both upscale and downscale between octave changes.

Intersecting Siren

To explore noises made from other sequences of tones consider the following series:

```
1,12,2,11,3,10,4,9,5,8,6,7,7,6,8,5,9,4,10,3,11,2,
12,1
```

Every other number starting from the first goes up in sequence, while every other number starting with the second goes down. The sequence is symmetrical toward the middle. It is called "INTERSECTING SIREN" in the Program Listing (lines 700–720).

When played fast, this sounds like two different tone sources, one going up while the other is going down. Many of the better tricks of noise making are based on the fact that the ear has trouble telling if rapidly successive tones are generated sequentially or simultaneously.

Incidentally, most of the examples in the program have an INKEY$ function such as the one in line 720. This is just so the sound will quit playing when you press a key. Since polling the keyboard takes processing time, the sounds are altered slightly.

Polyphony

People often want to know if a single-voice sound output like that of the Color Computer can generate harmony. You can approximate this effect by exploiting the ear's tendency to blend successive sounds. Here are some chords taken from guitar fingering charts.

G Chord:

```
810 PLAY "O1;G;O3;O4;C;E"
```

G# Minor Chord:

```
840 PLAY "O1;G#;O3;G#;B;O4;E"
```

And here is a dissonance:

```
870 PLAY "O1;G;O2;G#;A;B"
```

These strings must be played very rapidly for the blending effect to work. Notes whose frequencies are in simpler ratios blend with the least clashing or dissonance. You can vary the qualities of sound effects by using these effects. Consonances tend to sound open or bell-like, while dissonances sound clanky, constricted, or even ominous.

Rasps

Rasps are regular sequences of notes and pauses. They form the basis for many useful sound effects. Motor or helicopter noises, or the zapping sounds when certain weapons are fired in arcade games, are modified rasps. Other possible applications are buzzers, zippers, ripping cloth, chainsaws, and creaking doors. So far you have been concerned with the pitches of sounds played. But the major characteristics of a rasp are determined by the length of the tones and the length of the pauses between tones.

The PLAY function includes a pause of selectable length. For example: PLAY "1;P100;1". This plays two tones of the pitch 1 with a pause of 100 between them. Beware that the highest numbers represent the shortest pauses (like conventional music notation.) The range allowed is from 1 to 255, just like tempo. Here is an example of a single-note rasp:

```
900 SP$ = "1;P2;1;P2;1;P2;1;P2;1"
910 PLAY"T250"
920 PLAY SP$
930 GOTO 920
```

Add interest to rasps by using multiple tones between the pauses. This is from the multinote rasp example in the Program Listing:

```
980 SP$ = "C;P2;E;P2;G;P2;C;P2;E;P2;G;P2;"
```

It happens that the tones used are consonant, corresponding to a musical chord, so the sound is relatively open. You can change the mood of this effect by using dissonant tones:

```
990 SP$ = "C;P2;C#;P2;D;P2;C;P2;C#;
P2;D;P2;"
```

Changing Tone Length

You control the length of tones played by the L parameter. Once L is set by a sequence such as PLAY "L50" all tones following will be of that length until it is reset. The range is from 1 to 255, with one being the longest. Both L (note length) and P (pause length) use a fractional system so that four, (representing a quarter note or quarter rest) is twice as long as eight (eighth note or rest).

Note how L, P, and T (length, pause, and tempo) interact. Tempo changes affect both tone lengths and pauses, so they may be regarded as relative parameters. The length of pauses is specified for each pause, while the note length parameter remains the same until it is reset. By controlling these three factors independently, you can produce many types of rasp noises ranging from slow ticks to rapid buzzes.

Since so many of the more interesting effects are achieved by speeding up the tempos of tone sequences, there is a fundamental difference between speeding up the tempo of PLAY, and speeding up the playback of a sound recorded on tape or record. If you play back a tape or record at high speed, you increase the

tempo and raise the pitch of the sound simultaneously. But PLAY's tempo parameter does not affect the pitches.

Envelope Effects

When PLAY processes strings of sounds at high speed, the original pitches make only a minor contribution to the overall result.

For example, the musical note A is a sound with 440 cycles per second. Now suppose that another note or different frequency (say, the note G) is played in a rasp so that G is repeated 440 times per second. In this case the overall shape of the waveform, or envelope, could simulate an A tone that might completely overshadow the original G.

In fast machine-language programs, the control of such effects is a method for producing polyphonic output. Basic is a little slow to be versatile in this regard, but you can hear envelope effects at the fastest tempos. Sometimes they are unanticipated side effects, but if understood, you can control them. The rasp sound is an example because its overall frequency is the repetition rate of its note-pause cycle.

Noisy Sounds

The examples so far have been very regular or repetitious in both tone sequence and time structure. Such sounds are rarely heard in nature, although they are characteristic of man-made machinery. Most natural sounds have more noise content. Noise in information theory is very much like static in electronics. A noisy signal is one that is contaminated with a certain amount of random or unpredictable background noise. White noise is a signal of completely random impulses, or pure static. Hisses, swooshes, applause, crackling fires, running waters, and rustling winds are sounds with high randomness content.

A primitive way to add randomness to noises is to throw a pair of dice and construct PLAY statements with pitches corresponding to the resulting numbers. This isn't necessary because Color Basic has a built-in random-number generator. The statement X = RND(12) sets the numeric variable X equal to a number from 1 to 12. You can use the STR$ function to convert this value to a playable string character.

```
2520 SP$=STR$(RND(12))+";"
2525 RETURN
```

Long strings with random elements can be built by concatenation.

```
2500 R$=R$+SP$:GOSUB 2520
```

This can be repeated as long as the string remains fewer than 256 characters long.

The random element needn't be restricted to just pitches. You can control pauses, tempos, volume, and all the other parameters. This substitution in the example above causes the octave to be varied randomly.

```
2530 R$="O"+STR$(RND(5))+";"+STR$
(RND(12))+";"
```

Long strings of random pitches at low octave and high tempo sound like the rumbling aftermath of an explosion. Short strings with random pauses and pitches played at high octaves sound like shattering glass. Moderate tempos and octaves with pauses sound like crickets or squeaky springs. The program contains several variations of such sounds. They will sound different every time it is run because the random variables are reinitialized.

Other Sources

You can get programmable sounds from many sources, sometimes with interesting results. Consider the following data from a scientific experiment:

```
DATA C,A,C,T,G,T,A,A,A,G,C,T,A,A,
C,T,T,A...
```

Readers who have had a course in college or high school biology might recognize these letters as initials for the sequence of subunits in a DNA molecule. Sequences similar to this make up the genetic code that determines the characteristics of all living things. The initials stand for adenine, guanine, cytosine, and thymine.

Arbitrarily, line 3120 of the listing substitutes E for T. One way of feeding playable character sequences to PLAY that hasn't been demonstrated yet is to

Program Listing. Noise Library

```
1Ø CLEAR1ØØØ:CLS6:PRINT@98,"'MAK
ING NOISES MADE SIMPLE'";:PRINT@
192,"AN ACCOMPANIMENT TO THE MAG
AZINEARTICLE OF THE SAME NAME.";
:PRINT@288,"A TONE OF MODERATE L
OUDNESS IS  NOW BEING PLAYED.";:
PRINT@384,"";
12 PRINT"":PRINTCHR$(239);"ADJUS
T VOLUME":PRINTCHR$(255);"PRESS
ANY KEY TO CONTINUE
14 PLAY"L8;V15;O2;T15Ø"
15 PLAY"V25;O1;A;P2;V5;O5;G":IFI
NKEY$=""THEN15
17 Z=RND(-TIMER)
2Ø GOTO1ØØØØ
99 '
1ØØ CLS6:PRINT"STEP UP SCALE":SP
$="STRING TO BE PLAYED='1; 2; 3;
 4; 5; 6; 7; 8; 9; 1Ø; 11; 12;'"
:PRINTSP$
11Ø PLAY"V2Ø;T4;L2"
15Ø PLAY"1;2;3;4;5;6;7;8;9;1Ø;11
;12;"
198 RETURN
199 '
2ØØ CLS6:PRINT"STEP UP SCALE BY
OCTAVES":PRINTSP$:PRINT"IT WILL
BE PLAYED IN EACH OCTAVE FROM 1
TO 5
225 PLAY"T5"
23Ø FOR LOOP=1TO5
24Ø O$="O"+STR$(LOOP)
25Ø PLAYO$
255 PLAY"1;2;3;4;5;6;7;8;9;1Ø;11
;12;"
26Ø NEXT
298 RETURN
299 '
3ØØ CLS6:PRINT"SAME THING BUT FA
STER.":PRINT"STEP UP SCALE BY OC
TAVES.
31Ø PRINT"TEMPO WILL BE 1Ø, THEN
 2Ø,THEN  4Ø, THEN 8Ø, THEN 1ØØ,
 THEN 15Ø,THEN 25Ø.
32Ø PLAY"T1Ø;L4":GOSUB33Ø:PLAY"T
2Ø":GOSUB33Ø:PLAY"T4Ø":GOSUB33Ø:
PLAY"T8Ø":GOSUB33Ø
322 PLAY"T1ØØ":GOSUB33Ø
324 PLAY"T15Ø":GOSUB33Ø:PLAY"T25
Ø":GOSUB33Ø
325 RETURN
33Ø FOR LOOP=1TO5
34Ø O$="O"+STR$(LOOP)
35Ø PLAYO$
36Ø PLAY"1;2;3;4;5;6;7;8;9;1Ø;11
;12;"
37Ø NEXT
38Ø RETURN
399 '
4ØØ CLS6:PRINT"ASCENDING SCALE U
P ALL OCTAVES.":PRINT"REPEATED A
T HIGH SPEED TO MAKE  CONTINUOUS
 SOUND.":PRINT"PRESS ANY KEY TO
STOP
42Ø PLAY"T25Ø;L25Ø"
43Ø GOSUB33Ø:IFINKEY$=""THEN43Ø
45Ø Z$=INKEY$:PRINTCHR$(255);"WI
TH VOLUME CONTROLLED
46Ø FORO=1TO5:PLAY"O"+STR$(O)+";
V"+STR$(O*6)+";"+P$(1):NEXT
47Ø FORO=1TO5:PLAY"O"+STR$(O)+";
V"+STR$(36-O*6)+";"+P$(1):NEXT
48Ø IFINKEY$=""THEN46Ø
498 PLAY"V2Ø":RETURN
499 '
5ØØ CLS6:PRINT"ASCENDING DECENDI
NG SIREN.":PRINT"STRINGS PLAYED
ARE "P$(1):PRINT"AND  "P$(2)
51Ø PLAY"T25Ø"
52Ø FORO=1TO5:PLAY"O"+STR$(O)+";
"+P$(1):NEXT
53Ø FORO=5TO1 STEP-1:PLAY"O"+STR
$(O)+";"+P$(2):NEXT
54Ø IFINKEY$=""THEN52Ø
55Ø Z$=INKEY$
56Ø PRINTCHR$(255);"SAME THING B
UT WITH INVERTED  ORDER
57Ø FORO=1TO5:PLAY"O"+STR$(O)+";
"+P$(2):NEXT
58Ø FORO=5TO1 STEP-1:PLAY"O"+STR
$(O)+";"+P$(1):NEXT
59Ø IFINKEY$=""THEN57Ø ELSE RETU
RN
599 '
6ØØ CLS6:PRINT"ASCENDING DECENDI
NG SIREN WITH  WARBLE":PRINT"STR
INGS PLAYED="P$(1):PRINT"AND  "P
$(2)
61Ø PLAY"T25Ø
```

Listing continued

Listing continued

```
62Ø FORO=1TO5:PLAY"O"+STR$(O)+";
"+P$(1)+P$(2):NEXT
625 IFINKEY$<>""THENRETURN
63Ø FORO=5TO1 STEP-1:PLAY"O"+STR
$(O)+";"+P$(2)+P$(1):NEXT
64Ø IFINKEY$=""THEN62Ø
65Ø RETURN
699 '
7ØØ CLS6:PRINT"INTERSECTING SIRE
N":PRINT"STRING PLAYED=";P$(4)
71Ø PLAY"O3;T25Ø"
72Ø PLAYP$(4):IF INKEY$=""THEN 7
2Ø
73Ø PRINTCHR$(255);"BY OCTAVES":
Z$=INKEY$
74Ø FORO=1TO5:PLAY"O"+STR$(O)+";
"+P$(4):NEXT
745 IFINKEY$<>""THENRETURN
75Ø FORO=5TO1 STEP-1:PLAY"O"+STR
$(O)+";"+P$(4):NEXT
76Ø IFINKEY$=""THEN74Ø
798 RETURN
799 '
8ØØ CLS6:PRINT"POLYPHONIC EFFECT
S
8Ø5 PRINT"G MAJOR CHORD"
81Ø C$="O1GO3GO4CEO1GO3GO4CE"
815 PLAY"T2ØØ"
82Ø PLAYC$:IFINKEY$=""THEN82Ø
83Ø PRINT"G SHARP MINOR CHORD"
84Ø C$="O1G#O3G#BO4EO1G#O3G#B"
85Ø PLAYC$:IFINKEY$=""THEN85Ø
86Ø PRINT"DISSONANCE"
87Ø C$="O1GO2G#ABO1GO2G#AB"
88Ø PLAYC$:IFINKEY$=""THEN88Ø
898 RETURN
899 '
9ØØ CLS6:PRINT"RASPS":PRINT@64,"
1 SINGLE NOTE":PRINT"2 MULTI TON
E":INPUTMZ:ONMZ GOTO91Ø,98Ø
9Ø5 RETURN
91Ø CLS6:PRINT"SINGLE NOTE RASPS
":PRINT"STRING PLAYED="P$(5):SP$
=P$(5)
915 PRINT"TEMPO=5Ø,LOW OCTAVE":P
LAY"T5Ø;O1;V31;"
92Ø PLAYSP$:IFINKEY$=""THEN92Ø
925 PRINTCHR$(255);"TEMPO 15Ø":P
LAY"T15Ø":Z$=INKEY$
93Ø PLAYSP$:IFINKEY$=""THEN93Ø
935 PRINTCHR$(255);"TEMPO INCREA
SED FROM 2Ø TO 255
94Ø FORT=2ØTO255 STEP2Ø:PLAY"T"+
STR$(T)+";"+SP$:NEXT
945 PRINTCHR$(255);"CHANGING OCT
AVES
95Ø FORO=1TO4:PLAY"O"+STR$(O)+";
"+SP$:NEXT:IFINKEY$=""THEN95Ø
955 PRINTCHR$(255);"UP AND DOWN"
:Z$=INKEY$
96Ø FORO=1TO4:PLAY"O"+STR$(O)+";
"+SP$:NEXT:FORO=4TO2 STEP-1:PLAY
"O"+STR$(O)+";"+SP$:NEXT:IFINKEY
$=""THEN96Ø
965 PRINTCHR$(255);"HIGH OCTAVE
WITH VOLUME CHANGE":PLAY"T255":Z
$=INKEY$
97Ø FORV=1TO31STEP6:PLAY"V"+STR$
(V)+";"+SP$:NEXT:IFINKEY$=""THEN
97Ø
975 PLAY"T2Ø;V2Ø":RETURN
98Ø CLS6:PRINT"MULTI NOTE RASP":
PRINTCHR$(239);"CONSONANT":SP$=P
$(6):PRINT"STRING PLAYED ="SP$
985 PLAY"L64":GOSUB915
99Ø PRINT"MULTI NOTE RASP":PRINT
CHR$(239);"DISSONANT":SP$=P$(7):
PRINT"STRING PLAYED ="SP$
995 GOSUB915
998 PLAY"L2":RETURN
999 '
1ØØØ CLS6:PRINT"RANDOM TONES":PR
INT"SOUND WILL BE DIFFERENT EVER
Y   TIME":GOSUB25ØØ:SP$=R$(1):PR
INT"STRING PLAYED="SP$
1Ø1Ø PRINT"TEMPO=25Ø","HIGH OCTA
VE":PLAY"T25Ø;O5"
1Ø2Ø PLAYSP$:IF INKEY$=""THEN 1Ø
2Ø
1Ø3Ø PRINTCHR$(255);"LOW OCTAVE"
:PLAY"O1"
1Ø4Ø PLAYSP$:IF INKEY$=""THEN 1Ø
4Ø
1Ø5Ø PRINTCHR$(255);"WITH VOLUME
 CONTROL
1Ø6Ø FORV=6TO31 STEP5:PLAY"V"+ST
R$(V)+";"+SP$:NEXT:IFINKEY$=""TH
EN1Ø6Ø
1Ø7Ø PRINTCHR$(255);"OCTAVE ALSO
 CONTROLLED
1Ø8Ø FORV=31TO4 STEP-5:PLAY"V"+S
TR$(V)+";O"+STR$(INT(6.5-V/6))+"
;"+SP$:NEXT:IFINKEY$=""THEN1Ø8Ø
1Ø9Ø RETURN
1999 '
2ØØØ CLS6:PRINT"RANDOM NOTES AND
 PAUSES":PRINT"THIS SOUND WILL
BE  DIFFERENT  EVERY TIME.":GOSU
B251Ø:PRINT"THE STRING PLAYED TH
IS TIME="R$(3)
2Ø1Ø PLAY"T25Ø;V2Ø"
2Ø2Ø PLAYR$(3):IFINKEY$=""THEN2Ø
2Ø
2Ø3Ø Z$=INKEY$:PRINTCHR$(255)"WI
TH CHANGING VOLUME
2Ø4Ø FORV=1TO31 STEP5:PLAY"V"+ST
R$(V)+";"+R$(3):NEXT
2Ø5Ø IFINKEY$<>""THEN2Ø8Ø
2Ø6Ø FORV=31TO1 STEP-5:PLAY"V"+S
TR$(V)+";"+R$(3):NEXT
2Ø7Ø IFINKEY$=""THEN2Ø4Ø
2Ø8Ø PLAY"V2Ø":RETURN
24ØØ 'filling strings with rando
m variables
241Ø GOSUB25ØØ,251Ø
242Ø RETURN
25ØØ R$(1)="":FORX=1TO2Ø:GOSUB25
2Ø:R$(1)=R$(1)+R$:NEXT:RETURN
251Ø R$(3)="":FORX=1TO15:GOSUB25
3Ø:R$(3)=R$(3)+R$:NEXT:RETURN
252Ø R$ =STR$(RND(12))+";":RETUR
N
253Ø R$="O"+STR$(RND(5))+";"+STR
$(RND(12))+";P"+STR$(RND(255))+"
;":RETURN
2999 '
3ØØØ 'other sources
3Ø1Ø CLS6:PRINT"OTHER SOURCES OF
 PLAYABLE       STRINGS":PRINTST
RING$(32,223);:PRINT"1 TRANSFER
RNA":PRINT"2 BEETHOVEN":PRINT"Ø
BACK TO LAST MENU"
3Ø2Ø INPUTMU:ON MU GOSUB31ØØ,33Ø
Ø
3Ø3Ø IFMU=ØTHENRETURNELSE3ØØØRET
URN
31ØØ CLS6:PRINT"SOUND GENERATED
FROM SCIENTIFIC DATA":PRINT"TRAN
SFER RNA":PRINT"TEMPO 8
311Ø PLAY"T8;O3;L2;"
312Ø DATA C,A,C,E,G,E,A,A,A,G,C,
E,A,A,C,E,E,A,G,C,A,E,E,A,A,C,C
313Ø DATA E,E,E,E,A,A,G,E,E,A,A,
A,G,A,E,E,A,A,G,A,G,A,A,C,C,A,A,
C,A,C,C,E,C,E,E,E,A,C,A,G,E,G,A
314Ø RESTORE:FOR LOOP=1 TO 69
315Ø READ SP$:PLAY SP$
316Ø NEXT
317Ø PRINTCHR$(239);"TEMPO 25Ø":
PLAY"T25Ø"
318Ø RESTORE:FOR LOOP=1TO69:READ
SP$:PLAYSP$:NEXT:IFINKEY$=""THEN
318Ø
319Ø Z$=INKEY$:PRINTCHR$(239)"LO
W OCTAVE":PLAY"O1
32ØØ RESTORE:FORLOOP=1TO69:READS
P$:PLAYSP$:NEXT:IFINKEY$=""THEN3
2ØØ
321Ø RETURN
33ØØ CLS6:PRINT@35,"PASSAGES FRO
M SHEET MUSIC";:PRINT@128,"Ø ESC
APE TO LAST MENU":PRINT"1  BEETH
OVEN AT NORMAL TEMPO":PRINT"2  F
OUR TIMES NORMAL (TEMPO 8)":PRIN
T"3   16 TIMES NORMAL (TEMPO 32)
":PRINT"4   64 TIMES NORMAL (TEM
PO 128)"
331Ø INPUTMU:IFMU=ØTHEN3ØØØ
```

Listing continued

READ them from DATA statements. Sixty-nine data items in the program represent this sort of information, so that number limits the loop. (See lines 3140–3160.)

This particular noise is midway in quality between the last highly random or noisy examples, and the earlier, more regular sounds.

The option that lets you specify pitches by musical letter names is convenient for reading published sheet music. A passage from the score of Beethoven's *Fifth Symphony* is included in the program. Instructions for translations of this sort are found in the Extended Basic user's manual, so I won't repeat them here. You can get some odd effects by playing musical passages at high tempos.

Building Special-Purpose Noise

Often you want a complicated noise. Particular effects are desirable for games and educational programs and might require a combination of sounds, perhaps including some of the general types demonstrated so far.

The first step is to decide what sort of effect you want. Think about what types of sounds might go into it and in what order, and try to relate them to general types. Are the desired sounds regular or noisy? Smooth or raspy? Harmonious or dissonant? What about volume or pitch changes?

Programming the sound involves some trial and error, but patience and close listening can obtain something like what is desired starting from the general noise types.

For example, suppose you want some sort of ominously descending noise followed by an explosion. After considering the possible components of this sort of effect, you might decide on the following sequence of basic types: a descending rasp, a short pause, a loud percussion, and a rumbling echo.

A candidate for the first sound might be a variation of the standard rasp with the pitches descending. Line 4210 defines such a string as the string variable C$. Line 4230 plays the string five times. To do so, it uses the Execute or X option. In a PLAY statement, "XC$;" causes the string named C$ to be executed or played. Each time, the octave is lower and the tempo faster. The fact that the sequence of tones is dissonant makes it sound more ominous as the tempo increases. (See line 4230.)

Line 4240 inserts a pause to add suspense before the impact and changes the tempo. The impact is a loud short dissonance.

Now you need a long rumble or echoing peal. It calls for extended sound with high noise content and low pitch. Strings of random pitches sometimes work well for this purpose. But just for fun, use the converted data for the DNA molecule because it has similar characteristics. This loop in line 4215 constructs a playable string called DNA$ from the data already in the program (line 3120). The RESTORE command in line 4215 sets the data reader back to the first item. Once defined it can be executed. Add interest by causing the volume to swell a time or two before fading, as in line 4245.

After the explosion dies down, crickets start chirping (line 4250) indicating a return to quiet. The particular string played was once produced by the RANDOM function. Because it sounded so much like crickets, I printed it out and saved it for occasions such as this.

The accompanying program is structured for easy use as a library of basic effects for building custom sounds. I've sorted the examples into types whose routines you can find via the program lines specified by Table 1. Frequently used strings are defined in lines 10000 to 10100. Other than this and the other instances noted above, the routines are self-contained, and you can enter and execute individual routines if the entire program is too large for your computer. With a little imagination, they can be starting points for all sorts of interesting, arresting, melodious, cacophonous, exciting, annoying, strange, realistic, otherworldly, or just plain odd, noises. ■

Address correspondence to Philip McLaughlin, 712 Roberts St., Denton, TX 76201.

100	Simple scale
200	Simple scale by octaves
300	Faster and faster
400	Continuous siren
500	Up and down siren
600	Warbling siren
700	Intersecting siren
800	Polyphony
900	Single and multinote rasps
1000	Random pitches
2000	Random pitches and pauses
3000	Other sources of sounds
	3100 DNA molecule
	3300 Beethoven
4000	Building special-purpose sounds

Table 1. Subroutine Menu

Listing continued

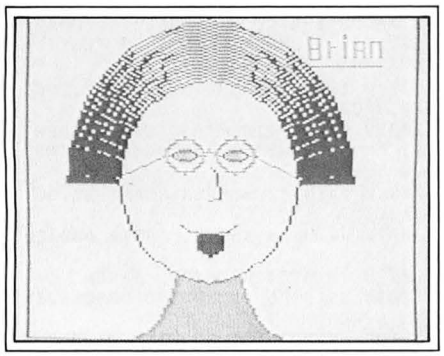
```
332Ø PLAY"T2;O2":IFMU=2THENPLAY"
T8"
333Ø IFMU=3THENPLAY"T32"
334Ø IFMU=4THENPLAY"T128"
335Ø RESTORE:FORZ=1TO69:READZ$:N
EXT
336Ø FORPO=1TO9Ø:READP$:PLAYP$:N
EXT:GOTO33ØØ
337Ø PLAY MP$
338Ø DATAV3Ø,L8,B-,B-,B-,L2,V25,
E-,V2Ø,F,O1,B-,L4,O2,V15,P1,B-,O
3,E-,D,E-,F,C,P64,C,O2,B-,P64,B-
,O3,E-,D,E-,F,C,C,O2,B-
339Ø DATAP1,O2,L2.,G,L4,F#,E,D,P
99,D,P99,D,P99,L2.,G,L4,F#,E,D,P
99,D,P99,D,P99,L2.,O3,C,O2,L4,B,
A,G#,P99,G#,P99,G#,P99,L2,A,P99,
O3,L4..,C,O2,L16,A,L2,G,L4..,A,L
16,'F#
3999 '
4ØØØ CLS6:PRINT"SPECIAL PURPOSE
NOISES":PRINTSTRING$(32,223);:PR
INT"1 TRAILING SIREN AND IMPACT"
:PRINT"2 TRAILING RASP AND IMPAC
T":PRINT"Ø BACK TO LAST MENU"
4Ø1Ø INPUTMU:ON MU GOSUB41ØØ,42Ø
Ø
4Ø2Ø IFMU=ØTHENRETURNELSE4ØØØ
41ØØ CLS6:PRINT"IMPACT
411Ø GOSUB24ØØ
412Ø 'PLAY"V3Ø;O5;T25Ø;"+P$(1)+"
O1;"+R$(1)
413Ø PLAY"V3;O5;T4;L4;12;T5Ø;V2Ø
;"+P$(2)+"V1Ø;O4;T1ØØ;"+P$(2)+"V
1Ø;O3;"+P$(2):PLAY"O2;"+P$(2):PL
AY"T2;P2;T25Ø;V31;O1;"+R$(1)+"V2
Ø;"+R$(1):PLAY"T15Ø;V15;"+R$(1)+
R$(1)
414Ø PLAY"V31;"+R$(1):PLAY"V1Ø;O
2;"+R$(1)
415Ø PLAY"V31;O1;"+R$(1)+"V31;O2
;"+R$(1)
416Ø PLAY"V2Ø;O1;XR$(1);XR$(1);X
R$(1);V15;XR$(1);XR$(1);XR$(1);V
1Ø;XR$(1);XR$(1);XR$(1);V5;XR$(1
);XR$(1);XR$(1);
417Ø GOTO4ØØØ
42ØØ CLS6:PRINT"TRAILING RASP WI
TH IMPACT
421Ø C$="12;P1;11;P1;1Ø;P1;9;P1;
8;P1;7;P1;6;P1;5;P1;4;;P1;3;P1;2
;P1;"'INTITAL STRINGS
4214 DNA$=""
4215 RESTORE:FOR LOOP=1 TO 69:RE
AD Z$:DNA$=DNA$+Z$:NEXT
422Ø PRINT"DOWNSCALE RASP WITH I
NCREASING  TEMPO"
423Ø PLAY"V2Ø;L2Ø;;T2Ø;O5;XC$;T4
Ø;O4;XC$;T8Ø;O3;XC$;T16Ø;O2;XC$;
T25Ø;O1;XC$;
424Ø PLAY"T4;P1;T25Ø;L25Ø":CLS8:
PLAY"V31;O1;CC#DCBO5A#O1DD#ECC#O
3BO1CDEF#FE#
4244 CLS2:PLAY"T4;P1;L25Ø;T25Ø":
CLS5:PRINT"RUMBLE DERIVED FROM D
NA DATA"
4245 PLAY"V31O1;XDNA$;V15;XDNA$;
V31T5CT25ØV1Ø;XDNA$;V25;XDNA$;V1
5;XDNA$;V1Ø;XDNA$;V5;XDNA$;V1;XD
NA$;"
425Ø PLAY"O5;V1":PRINT"'CRICKETS
' FROM RANDOM FUNCTION":FORX=1TO
15 STEP.5:PLAY"V"+STR$(INT(X))+"
;2;6;1Ø;6;5;6;11;7;3;11;2;1Ø;11;
2;8;1;1Ø;9;2;2":NEXT
426Ø GOTO4ØØØ
4999 '
1ØØØØ DIMP$(12)'define often use
d strings
1ØØ1Ø P$(1)="1;2;3;4;5;6;7;8;9;1
Ø;11;12;"'ASCENDING SCALE
1ØØ2Ø P$(2)="12;11;1Ø;9;8;7;6;5;
4;3;2;1"'DESCENDNG SCALE
1ØØ3Ø P$(3)="B;A;G;F;E;D;C"'DESC
ENDING NOTES
1ØØ4Ø P$(4)="1;12;2;11;3;1Ø;4;9;
5;8;6;7;7;6;8;5;9;4;1Ø;3;11;2;12
;1;"'INTERSECTING SEQUENCE
1ØØ5Ø P$(5)="1;P2;1;P2;1;P2;1;P2
;1;P2;1;P2;1;P2;1;P2;1;P2;1;"'SI
MPLE RASP
1ØØ6Ø C$="GO+CEO-;":P$(6)=C$+"P1
;"+C$+"P1;"+C$+"P1;"'CONSONANT R
ASP
1ØØ7Ø P$(7)="CC#D;P1;CC#D;P1;CC#
D;P1;"
1ØØ8Ø P$(9)="1;P8;1;P8;":P$(8)="
XP$(9);XP$(9);XP$(9);XP$(9);XP$(
9);XP$(9);"'CONSTRUCTED SIMPLE R
ASP
1ØØ9Ø P$(11)="1;P255;1;P255;1;P2
55;1;P255;":P$(1Ø)="XP$(11);XP$(
11);XP$(11);XP$(11);"'SHORT CONS
TRUCTED RASP
1Ø1ØØ P$(12)="C;D;E;F;G;A;B;"'SI
MPLE SCALE
19999 '
2ØØØØ 'main menu
2ØØ1Ø PLAY"V2Ø;O2"
2ØØ2Ø CLS6:PRINT@9,"*MAIN MENU*"
;:PRINT@64," DEMONSTRATIONS OF S
OUNDS FROM   MAGAZINE ARTICLE.":
PRINT@16Ø," 'LAZY BUTTON' PLAYS
ALL BUT THE FIRST TWO SOUNDS IN
SEQUENCE.":PRINT@257,CHR$(255);"
TO STOP SOUNDS PRESS ANY KEY";CH
R$(255);
2ØØ3Ø PRINT@329,"**CHOICES**";:P
RINT@352,"  1 SCALES, SIRENS, AN
D RASPS":PRINT"  2 BUILDING COMP
LICATED SOUNDS":PRINT"  3 'LAZY
BUTTON'
2ØØ4Ø INPUTMU:ONMU GOTO2ØØ5Ø,22Ø
ØØ,3ØØØØ
2ØØ5Ø CLS6:PRINT"**SCALES, SIREN
S, AND RASPS**":PRINT@64,"1 SIMP
LE SCALE
2ØØ6Ø PRINT"2 SIMPLE SCALE BY OC
TAVES
2ØØ7Ø PRINT"3 ABOVE FASTER AND F
ASTER
2ØØ8Ø PRINT"4 EXTENDED SIREN
2ØØ9Ø PRINT"5 ASCENDING DESCENDI
NG SIREN
2Ø1ØØ PRINT"6 SIREN WITH WARBLE
2Ø11Ø PRINT "7 INTERSECTING SIRE
N
2Ø12Ø PRINT"8 POLYPHONY
2Ø13Ø PRINT"9 SINGLE AND MULTI T
ONE RASPS
2Ø14Ø PRINT"Ø BACK TO MAIN MENU
2Ø145 PRINT"":PRINT"  PRESS ANY
KEY TO STOP SOUNDS":PRINT""
2Ø15Ø INPUTMU:ONMU GOSUB1ØØ,2ØØ,
3ØØ,4ØØ,5ØØ,6ØØ,7ØØ,8ØØ,9ØØ
2Ø16Ø IFMU=ØTHEN2ØØØØELSE2ØØ5Ø
21999 '
22ØØØ 'submenu 2
22Ø1Ø CLS6:PRINT@Ø,"**BUILDING C
OMPLICATED SOUNDS**";
22Ø2Ø PRINT@64," 1 RANDOM NOTE S
EQUENCES
22Ø3Ø PRINT" 2 RANDOM NOTES AND
PAUSES
22Ø4Ø PRINT" 3 OTHER SOURCES
22Ø5Ø PRINT" 4 SPECIAL PURPOSE N
OISES
22Ø6Ø PRINT" Ø BACK TO MAIN MENU
"
22Ø65 PRINT"":PRINT"  PRESS ANY
KEY TO STOP SOUNDS":PRINT""
22Ø7Ø INPUTMU:IFMU=ØTHEN2ØØØØ
22Ø8Ø ONMU GOSUB1ØØØ,2ØØØ,3ØØØ,4
ØØØ
22Ø9Ø GOTO22ØØØ
29999 '
3ØØØØ GOSUB3ØØ:GOSUB4ØØ:GOSUB31Ø
ØØ:GOSUB5ØØ:GOSUB31ØØØ:GOSUB6ØØ:
GOSUB31ØØØ:GOSUB7ØØ:GOSUB31ØØØ:G
OSUB8ØØ:GOSUB31ØØØ:GOSUB9ØØ:GOSU
B31ØØØ:GOSUB1ØØØ:GOSUB31ØØØ:GOSU
B2ØØØ:GOSUB3ØØØ:GOSUB4ØØØ
3ØØ1Ø GOTO2ØØØØ
31ØØØ PRINT"PRESS KEY TO CONTINU
E";
311Ø5 FORT=1TO1ØØ:NEXTT:Z$=INKEY
$
3111Ø IFINKEY$=""THEN3111ØELSERE
TURN
```

END

BY BRIAN RUPERT

CREATE-A-FACE

Here's something to keep your children entertained on a rainy day—an easy-to-use face maker.



Create-A-Face is a simple program that lets children make up funny faces on the TV screen and gives an outstanding display of PMODE3 graphics. It begins with an outline of the screen, a blank face, and a neck. A name appears in the upper-right corner and you can use the DRAW command to change it.

There are 29 options in all. The first 26 are labeled with letters A to Z, and the last three with numbers 1 through 3. You cannot use some of the options singly. For example, you must use option V with option M. Table 1 gives a complete list of options.

I used the PEEK statement instead of INKEY$, so you can hold the desired key instead of having to press it repeatedly. This lets you use options D, L, X, Y, H, G, W, and I more fully. ■

Address correspondence to Brian Rupert, R.D. 1, Box 425, Kittanning, PA 16201.

System Requirements

16K RAM
Extended Color Basic

Program Listing. Create-A-Face

```
5Ø PMODE 3,1
6Ø PCLS
7Ø SCREEN 1,1
8Ø POKE65495,Ø
9Ø H=6Ø
1ØØ M=6Ø
11Ø 'FRAME
12Ø LINE(Ø,191)-(255,191),PSET
13Ø LINE(Ø,191)-(Ø,Ø),PSET
14Ø LINE(Ø,Ø)-(255,Ø),PSET
15Ø LINE(255,Ø)-(255,191),PSET
16Ø CIRCLE(121,96),6Ø  'HEAD
17Ø PAINT (121,96),5
18Ø 'SHOULDERS
19Ø CIRCLE(4Ø,14Ø),6Ø,6,1,.Ø4,.2
5
2ØØ CIRCLE(2ØØ,14Ø),6Ø,6,1,.25,.
48
21Ø CIRCLE(121,96),6Ø,6
22Ø PAINT(12Ø,17Ø),6,6
23Ø CIRCLE(121,96),6Ø,8
24Ø GOSUB 245Ø
25Ø 'MAIN ROUTINE
255 FOR Z= 1 TO 7:POKE338+Z,255:
NEXT
26Ø IF PEEK(339)=254 GOSUB 64Ø
27Ø IF PEEK(34Ø)=254 GOSUB 59Ø
28Ø IF PEEK(341)=254 GOSUB 56Ø
29Ø IF PEEK(342)=254 GOSUB 7ØØ
3ØØ IF PEEK(343)=254 GOSUB 74Ø
31Ø IF PEEK(344)=254 GOSUB 77Ø
32Ø IF PEEK(345)=254 GOSUB 83Ø
33Ø IF PEEK(338)=253 THEN POKE 3
38,255: GOSUB 94Ø
34Ø IF PEEK(339)=253 GOSUB 1ØØØ
35Ø IF PEEK(34Ø)=253 GOSUB 112Ø
36Ø IF PEEK(341)=253 GOSUB 116Ø
37Ø IF PEEK(342)=253 GOSUB 12ØØ
38Ø IF PEEK(343)=253 GOSUB 129Ø
39Ø IF PEEK(344)=253 GOSUB 137Ø
4ØØ IF PEEK(345)=253 GOSUB 142Ø
41Ø IF PEEK(338)=251 GOSUB 15ØØ
42Ø IF PEEK(339)=251 GOSUB 158Ø
43Ø IF PEEK(34Ø)=251 GOSUB 162Ø
44Ø IF PEEK(341)=251 GOSUB 165Ø
45Ø IF PEEK(342)=251 GOSUB 169Ø
46Ø IF PEEK(343)=251 GOSUB 172Ø
47Ø IF PEEK(344)=251 GOSUB 175Ø
48Ø IF PEEK(345)=251 GOSUB 189Ø
49Ø IF PEEK(338)=247 THEN POKE33
8,255: GOSUB 22ØØ
5ØØ IF PEEK(339)=247 GOSUB 225Ø
51Ø IF PEEK(339)=239 GOSUB 23ØØ
52Ø IF PEEK(34Ø)=239 GOSUB 235Ø
53Ø IF PEEK(341)=239 GOSUB 238Ø
54Ø IF PEEK(34Ø)=247 GOTO 184Ø
55Ø GOTO 25Ø
56Ø 'MOUTH
57Ø CIRCLE(121,121),2Ø,8,.3
58Ø RETURN
59Ø 'NOSE
6ØØ DRAW"BM123,87;S4C8F4"
61Ø LINE (129,1Ø5)-(12Ø,1Ø6),PSE
T
62Ø LINE(126,9Ø)-(129,1Ø5),PSET
63Ø RETURN
64Ø CIRCLE(1Ø5,8Ø),1Ø,8,.4'EYES
65Ø PAINT(1Ø5,8Ø),6,8
66Ø CIRCLE(139,8Ø),1Ø,8,.4
67Ø PAINT(139,8Ø),6,8
68Ø RETURN
69Ø 'HAIR ROUTINE
7ØØ IF H>=96 THEN RETURN
71Ø CIRCLE(121,96),H,8,1,.5
72Ø H=H+1
73Ø RETURN
74Ø 'CHIN
75Ø CIRCLE(121,14Ø),15,8,.4,.15,
.4
76Ø RETURN
77Ø 'EYE BROWS
78Ø FOR A=68 TO 72
79Ø CIRCLE(1Ø5,A),15,8,.4,.5,1
8ØØ CIRCLE(138,A),15,8,.4,.5,1
81Ø NEXT A
82Ø RETURN
83Ø 'CHEEKS
84Ø COLOR 7,8
85Ø FOR C=1 TO 4Ø
86Ø A=RND(2Ø):B=RND(2Ø)
87Ø PSET(87+A,9Ø+B)
88Ø NEXT C
89Ø FOR C=1 TO 4Ø
9ØØ A=RND(2Ø):B=RND(2Ø)
91Ø PSET(14Ø+A,9Ø+B)
92Ø NEXT C
93Ø RETURN
94Ø 'TALKING ROUTINE
95Ø CIRCLE(121,121),2Ø,8,.3
96Ø PAINT(121,121),7,8
97Ø FOR X=1 TO 55:NEXT X
98Ø PAINT(121,121),5,8
99Ø RETURN
1ØØØ 'BLINKING ROUTINE
1Ø1Ø CIRCLE(1Ø5,8Ø),1Ø,8,.4
1Ø2Ø CIRCLE(139,8Ø),1Ø,8,.4
1Ø3Ø PAINT(1Ø5,8Ø),8,8
1Ø4Ø PAINT(139,8Ø),8,8
1Ø5Ø FOR X=1 TO 8Ø:NEXT X
1Ø6Ø PAINT(1Ø5,8Ø),6,5
1Ø7Ø IF PPOINT(144,7Ø)=8 THEN1Ø9
Ø
1Ø8Ø PAINT(139,8Ø),6,5
1Ø9Ø CIRCLE(1Ø5,8Ø),1Ø,8,.4
11ØØ CIRCLE(139,8Ø),1Ø,8,.4
111Ø RETURN
112Ø 'FROWN ROUTINE
113Ø CIRCLE(121,121),2Ø,5,.3
114Ø CIRCLE(121,121),2Ø,8,.3,.5,
1
```

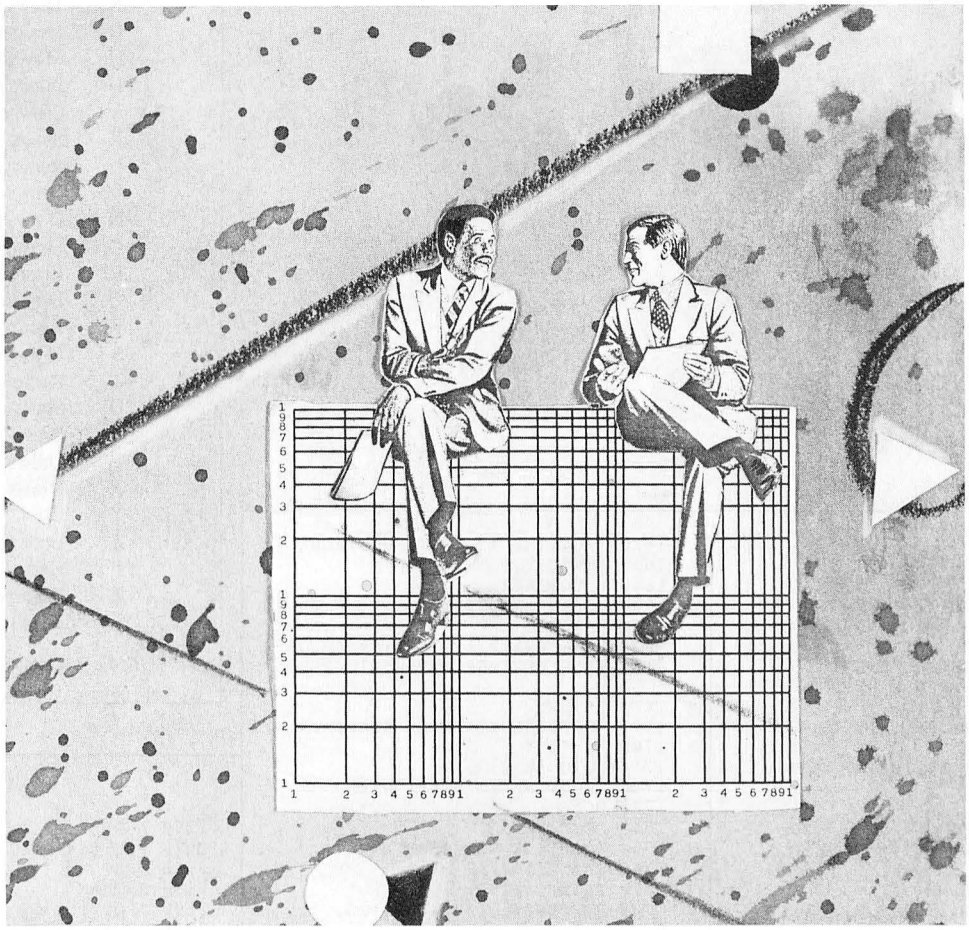
A	Draws eyes
B	Draws nose
C	Draws mouth
D	Draws hair
E	Draws chin
F	Draws eyebrows
G	Draws cheeks
H	Moves lips
I	Blinks eyes
J	Frowns
K	Smiles
L	Removes hair
M	Draws ears
N	Draws patch
O	Draws glasses
P	Erases glasses
Q	Draws mustache
R	Removes mustache
S	Draws beard
T	Removes beard
U	Starts over
V	Erases ears
W	Draws whiskers
X	Draws mohawk
Y	Erases mohawk
Z	Ends program
1	Draws earrings
2	Erases earrings
3	Draws random face

Table 1. Create-A-Face Options

```
115Ø RETURN
116Ø 'SMILE ROUTINE
117Ø CIRCLE(121,121),2Ø,5,.3
118Ø CIRCLE(121,121),2Ø,8,.3,1,.
5
119Ø RETURN
12ØØ 'HAIR LOSS ROUTINE
121Ø IF H<=61 THEN H=61:GOTO 127
Ø
122Ø CIRCLE(121,96),H,5,1,.5
123Ø H=H-1
124Ø IF PPOINT(55,1ØØ)=7 THEN  1
3ØØ
125Ø RETURN
126Ø GOSUB 13ØØ
127Ø CIRCLE(121,96),6Ø,8,1
128Ø RETURN
129Ø 'EAR ROUTINE
13ØØ CIRCLE(55,1ØØ),8,8,1.3,.1,.
9
131Ø CIRCLE(185,1ØØ),8,8,1.3,.62
,.4
132Ø PAINT(187,97),7,8
133Ø PAINT(5Ø,97),7,8
134Ø IF H=62 THEN 136Ø
135Ø RETURN
136Ø GOTO 112Ø
137Ø 'ROUTINE TO DRAW PATCH
138Ø CIRCLE(147,42),9Ø,7,.3,.218
5,.4
139Ø CIRCLE(14Ø,63),22,7,1.3,.Ø4
,.48
14ØØ PAINT(14Ø,78),8,7
141Ø RETURN
142Ø 'GLASSES ROUTINE
143Ø COLOR 6,5
144Ø CIRCLE(1Ø5,8Ø),15,6,.8
145Ø CIRCLE(139,8Ø),15,6,.8
146Ø LINE(92,8Ø)-(62,92),PSET
147Ø LINE(151,8Ø)-(177,92),PSET
148Ø CIRCLE(12Ø,82),7,6,1,.6,.9
149Ø RETURN
15ØØ 'ROUTINE TO REMOVE GLASSES
151Ø CIRCLE(1Ø5,8Ø),15,5,.8
152Ø CIRCLE(139,8Ø),15,5,.8
153Ø LINE(92,8Ø)-(62,92),PRESET
154Ø LINE(151,8Ø)-(177,92),PRESE
T
155Ø CIRCLE(12Ø,82),7,5,1,.6,.9
156Ø IF PPOINT(14Ø,78)=8THEN 137
Ø
157Ø RETURN
158Ø 'MUSTACHE ROUTINE
159Ø D$="C8"
16ØØ DRAW"BM1ØØ,118;XD$;S6U4R1D4
R1U4R1D4R1U5R1D4R1U5R1D4R1U6R1D5
R1U5R1D5R1U5R1D5R1U5R1D5R1U5R1D5
R1U5R1D5R1U5R1D5R1U5R1D5R1U5R1D5
R1U5R1D5R1U5R1D5R1U5R1D5R1U5R1D6
R1U5R1D6R1U5R1D5R1U4R1D5R1U4"
161Ø RETURN
162Ø 'ROUTINE TO ERASE MUSTACHE
163Ø D$="C5"
164Ø GOTO 16ØØ
165Ø 'BEARD ROUTINE
166Ø B$="C8"
167Ø DRAW"BM112,127;XB$;S9D4R1U4
R1D5R1U5R1D6R1U6R1D5R1U5R1D4R1U4
"
168Ø RETURN
169Ø 'ROUTINE TO ERASE BEARD
17ØØ B$="C5"
171Ø GOTO 167Ø
172Ø 'ROUTINE TO START OVER
173Ø PCLS
174Ø GOTO 5Ø
175Ø 'ROUTINE TO ERASE EARS
176Ø PAINT(187,97),5,8
177Ø PAINT(5Ø,97),5,8
178Ø CIRCLE(55,1ØØ),8,5,1.3,.1,.
9
179Ø CIRCLE(185,1ØØ),8,5,1.3,.62
,.4
18ØØ IF PPOINT(55,95)<>8 THEN 18
3Ø
181Ø CIRCLE(55,1ØØ),8,8,1.3,.6,.
9
182Ø CIRCLE(185,1ØØ),8,8,1.3,.6,
.9
183Ø RETURN
184Ø 'END PROGRAM
185Ø POKE 65494,Ø
186Ø CLS
187Ø PRINT @ 236,"BYE-BYE";
188Ø END
189Ø 'WHISKERS ROUTINE
19ØØ COLOR 8,5
191Ø FOR X=1 TO 15
192Ø A=RND(2Ø):B=RND(2Ø)
193Ø PSET(75+A,11Ø+B)
194Ø NEXT X
195Ø FOR X=1 TO 15
196Ø A=RND(2Ø):B=RND(2Ø)
197Ø PSET(82+A,117+B)
198Ø NEXT X
199Ø FOR X=1 TO 15
2ØØØ A=RND(2Ø):B=RND(2Ø)
2Ø1Ø PSET(88+A,127+B)
2Ø2Ø NEXT X
2Ø3Ø FOR X=1 TO 15
2Ø4Ø A=RND(23):B=RND(23)
2Ø5Ø PSET(1Ø8+A,128+B)
2Ø6Ø NEXT X
2Ø7Ø FOR X=1 TO 15
2Ø8Ø A=RND(2Ø):B=RND(2Ø)
2Ø9Ø PSET(13Ø+A,125+B)
21ØØ NEXT X
211Ø FOR X=1 TO 15
212Ø A=RND(2Ø):B=RND(2Ø)
213Ø PSET(141+A,117+B)
214Ø NEXT X
215Ø FOR X=1 TO 15
216Ø A=RND(2Ø):B=RND(2Ø)
217Ø PSET(147+A,11Ø+B)
218Ø NEXT X
219Ø RETURN
22ØØ 'ROUTINE TO DRAW MOHAWK
221Ø IF M>=95 THEN RETURN
222Ø CIRCLE(121,96),M,8,1,.73,.7
9
223Ø M=M+1
224Ø RETURN
225Ø 'ROUTINE TO ERASE MOHAWK
226Ø IF M<=61 THEN 127Ø
227Ø CIRCLE(121,96),M,5,1,.73,.7
9
228Ø M=M-1
229Ø RETURN
23ØØ 'ROUTINE TO DRAW EARRINGS
231Ø R$="C6"
232Ø DRAW"BM55,1Ø8;XR$;S13D6U2L3
R5"
233Ø DRAW"BM185,1Ø8;XR$;D6U2L3R5
"
234Ø RETURN
235Ø 'ROUTINE TO ERASE EARRINGS
236Ø R$="C5"
237Ø GOTO 232Ø
238Ø 'ROUTINE TO DRAW RANDOM
     FACE
239Ø FOR X=1 TO 15
24ØØ ON INT(RND(17)) GOSUB 64Ø,5
9Ø,56Ø,7ØØ,74Ø,77Ø,137Ø,112Ø,116
Ø,129Ø,142Ø,158Ø,165Ø,189Ø,22ØØ,
23ØØ,83Ø
241Ø Q=RND(-TIMER)
242Ø NEXT X
243Ø RETURN
244Ø 'ROUTINE TO DRAW 'BRIAN'
245Ø DRAW"BM187,12;S9C6D6R3U3L3R
3U3L3"
246Ø DRAW"BM199,15;S9D4U3R2"
247Ø DRAW "BM2Ø8,15;D4BM2Ø8,11D1
"
248Ø DRAW "BM214,15;D4U4R3D4U2L3
"
249Ø DRAW"BM224,15;D4U4R3D4"
25ØØ DRAW"BM186,29;R2Ø"
251Ø RETURN
```

END

BY PHILIP N. WILCOX



LEARNING CURVE

Be sure of your production costs and time, and calculate your production needs for the future.

If your business is producing a product and you are not using learning-curve theory, you should be. You could lead the field in price competition and at the same time be sure that you aren't reducing your prices too quickly.

T.P. Wright developed learning-curve theory in the 1930s, and put it into practice during World War II. He was studying the cost of building aircraft when he discovered that with double production quantities, production costs came down a certain fixed percent.

The formula he developed was $y=a*x^b$, where y is the hours to produce any given unit, x is the successive number of that unit, a is the hours to produce the first unit and b is the power to which you raise x. You need no more math because the Program Listing, Learning Curve, does it for you.

It calculates the slope on which you are currently producing, if you input the hours it takes to produce two completed units. It calculates the hours needed to produce any unit in the future and the cost of producing a quantity of units. If you want to know the unit number for which you know only the hours, it also tells you that.

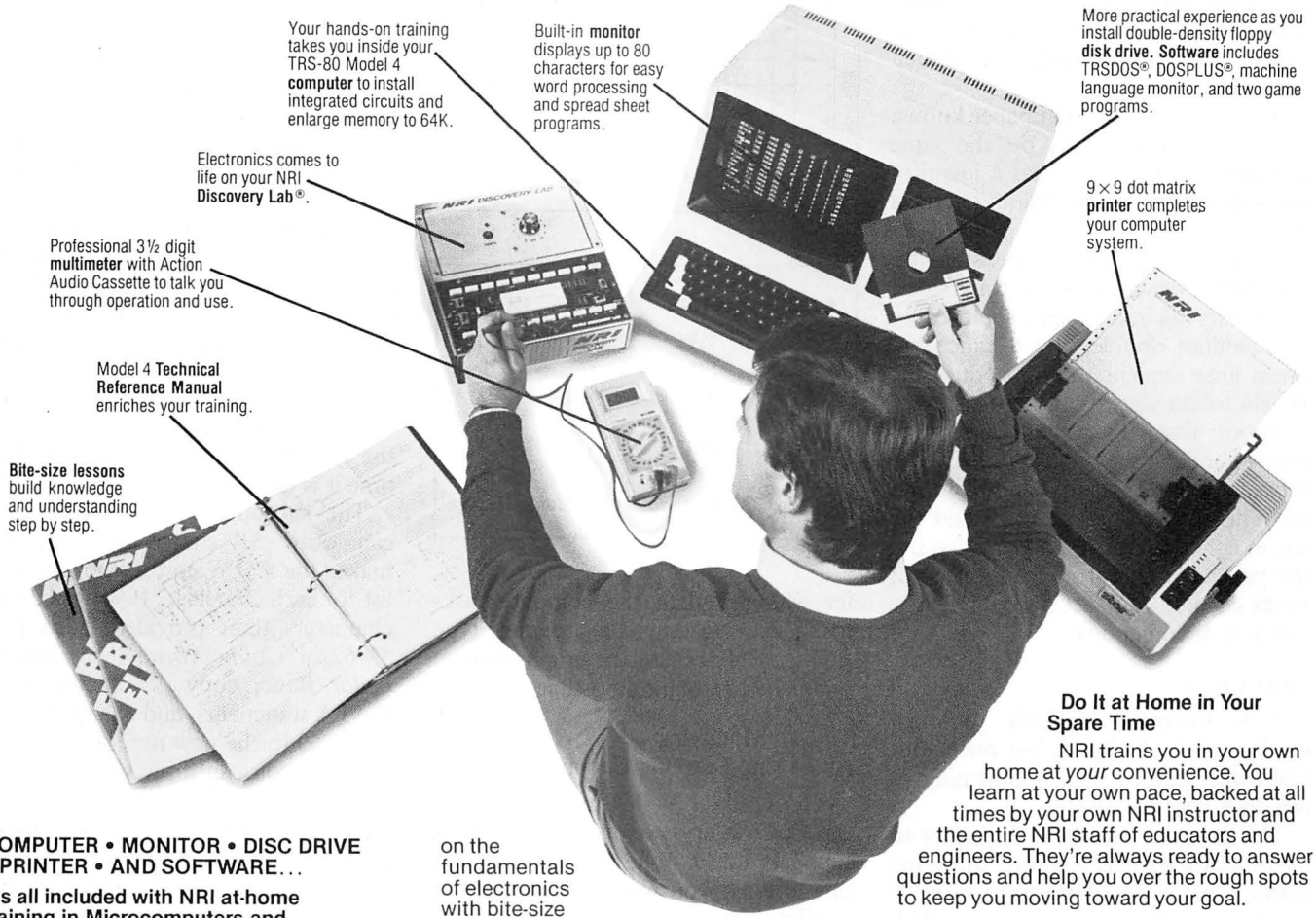
Once you decide on a curve for your product, there are other ways to approximate unit hours without running the program every time. The most common is to graph the values.

The only trouble is that learning usually happens in a logarithmic fashion. If you plot units and hours on regular arithmetic scales, they produce a curve with a continually changing radius, making it impossible to continue the line past the known values.

Enter the logarithm. The slide rule that you used in the days before calculators had scales set far apart on low num-

System Requirements
32K RAM
Extended Color Basic
Printer (optional)
Screen-Print Program

ONLY NRI GIVES YOU TOTAL COMPUTER SYSTEM TRAINING



COMPUTER • MONITOR • DISC DRIVE • PRINTER • AND SOFTWARE...

It's all included with NRI at-home training in Microcomputers and Microprocessors. And *only* with NRI training.

If you really want to learn how to work on computers, you have to learn how to work on computer systems. Only NRI includes a complete system with your training...total training that gives you hands-on experience with computers, peripherals, and software.

Get Inside the Famous TRS-80 Model 4 Computer

You'll work with one of the most popular computers ever designed. Learn programming techniques for trouble-shooting . . . go inside to expand its RAM from 16K to a powerful 64K . . . even install a double density floppy disc drive.

You'll become familiar with computer languages...run software that demonstrates computer operation...even become proficient at modifying programs to fit your business or family computing needs.

Add High Performance Peripheral Printer

Only NRI gives you total system training with the flexible Gemini 10X printer as part of your hands-on learning. Working with it, you'll get practical experience in adding peripherals, learn maintenance and servicing operations. And you'll use it for hard copies of important programs, business records, and word processing.

Understanding Built On Experience

You start with the basics, rapidly build on the fundamentals of electronics with bite-size lessons to master advanced concepts like digital logic, microprocessors, and computer memories.

You'll use the exclusive NRI Discovery Lab® to see what makes transistors tick, build and test working electronic circuits that duplicate key computer circuitry, construct digital logic circuits that demonstrate computer performance. And, you'll probe into the very heart of your computer using the Discovery Lab and professional digital multimeter included in your training program.

Do It at Home in Your Spare Time

NRI trains you in your own home at *your* convenience. You learn at your own pace, backed at all times by your own NRI instructor and the entire NRI staff of educators and engineers. They're always ready to answer questions and help you over the rough spots to keep you moving toward your goal.

Free Catalog Tells Much More... Send Today

See all the interactive training equipment you get in this state-of-the-art course in today's technology. Look over the lesson outlines and descriptions of the exciting experiments you'll perform. And look at other high-tech career fields like Robotics, TV/Video/ Audio Servicing and Data Communications.

Mail the coupon today to see how NRI can give you the skills and confidence you need to start a good-paying career as a microcomputer technician.

NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, N.W., Washington, DC 20016
We'll give you tomorrow.

All Career courses approved under GI bill.
☐ Check for details

CHECK ONE FREE CATALOG ONLY

- ☐ Computer Electronics with Microcomputers
- ☐ Data Communications
- ☐ Robotics & Industrial Controls
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronics Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics • Mobile • Aircraft • Marine
- ☐ CB & Cellular Radio
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction

Name (Please Print) Age

Street

City/State/Zip Accredited by the National Home Study Council 194-104

TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

See List of Advertisers on page 97

bers and close together on high ones. Using the slide rule, you were adding or subtracting the logarithms of the numbers shown, but the results were as though you had multiplied or divided the numbers.

A certain type of graph paper known as log/log paper works on the same principle (see Fig. 1). It has a logarithmic scale on the horizontal and vertical axes.

An interesting thing happens as you plot hours versus units on this paper. The points fall in a straight line because as quantities double, the distance between lines remains the same, just like Wright's discovery about unit cost.

Notice that the distance between points 1 and 2 is the same as the distance between points 2 and 4, 4 and 8, and so on. This makes it easier to extend the line in either direction, and you can use the paper to make rough estimates of hours or plot your actual hours to be sure you are on course.

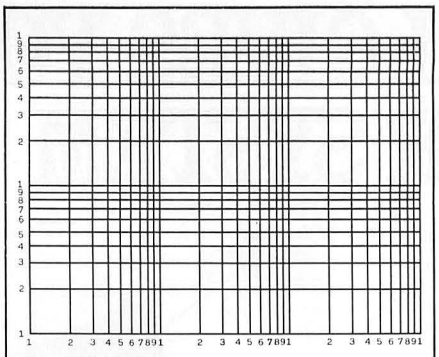


Fig. 1. Log/log Graph

UNIT	HOURS	UNIT	HOURS
1	755	6	300
2	529	7	277
3	429	8	259
4	370	9	244
5	330	10	231

Table 1. The Bear Motel Unit/Hours History

Enter CoCo

It is extremely difficult to draw log/log graphs by hand, but once your CoCo has learned to do something, it can do it very efficiently.

This program uses your computer to make all of the calculations, draw the graph, label it, and even print it out if you have a screen-print program. This should significantly improve your cost estimates.

Figure 1 is a log/log graph drawn by a CoCo. I typed the numbers along the side and top, to allow adequate space for the graph. Most numbers will be more than one digit, so label the graph by hand and keep it as large as possible on your screen.

Read hours on the vertical scale and units on the horizontal. The graph starts with 1 where the axes cross, and the lines get progressively closer together as the values increase, just like on the slide rule. The 1 could indicate 1, 10, 100, 1000, etc., and the next 1 along the scale will be 10 times greater than its predecessor.

Your CoCo tells you the values of the starting and ending points of the line it draws in the bottom title. It is up to you to label the graph or to learn to read it without labels.

The entire program is menu driven and prompts you for all necessary information. You need only run it, select the function you want to perform, and enter whatever data the computer asks for. Before loading the program, type PCLEAR 4 because the graphs require the highest resolution possible.

Tips on Modifying And Using Learning Curve

The USING ####.# in line 210 tells the CoCo that you want to show four numbers ahead of the decimal point and one after. If you want to use larger or smaller numbers than allowed, enter more cross hatches before or after the decimal point to avoid an error (see page 129 of *Going Ahead with Extended Color Basic*). As is, you can go from .1 to 9999.9 units, which should be enough for most applications.

In line 200, if you want a clear screen that is a different color than green and do not want the remainder of the line to turn green after the data is printed, you must place a semicolon at the end of the line (see lines 20–40).

Lines 3000–3220 set up and calculate the plot points. The graphics window only covers from 1–100 on the vertical axis and from 1–1000 on the horizontal. Each of the variables is in turn swapped into the variable P, starting with the vertical axis.

Dividing the first number to go through the plot-point loop by 10 and testing it repeatedly scales it down until it is between 1 and 100. The counter, C, keeps track of the number of divisions required to place it in the window. The second number to go through the plot-point loop is simply divided by 10 raised to the power C. This keeps it proportional to the first number.

This subroutine also takes care of the special case in which P = 1, by making P = 0 so it falls on the axis. The same formula calculates the plot points that locates the grid lines. The plot point is then swapped back into the original variable. The same logic works for the horizontal axis.

There is some danger in using this method of scaling points to a fixed grid, because some of the points could fall outside of the window. This rarely happens though, because real learning-curve slopes are seldom steep enough to cause this problem. If this is a problem to you, the only way out is to recalculate the grid, scaling it to your points each time it is drawn.

The subroutine from 5500 to 5700 can draw 51 characters. Line 5520 makes the CoCo start at the top of the list for each character. If you need more characters than provided, read pages 53–62 of *Going Ahead with Extended Color Basic*, copy the format of the DATA statements, and change the 51 in line 5560 to the new number of characters that you can draw with your additions.

Lines 6000–6270 use a screen-print program to produce a copy of the graph on your printer. This works with Custom Software Engineering's program for the C.Itoh or NEC printers.

If you have a different screen-print program or printer, you must modify this section to operate with your equipment. It is also menu driven and returns to the main menu on completion of the printout.

Line 6054 clears space for the program so that Basic does not overwrite it later.

Lines 7000–7086 provide a name and listing of the lines in which you can find each of the variables in the program. You need the variable table if you decide to adapt this general program to some specific application. There is no need to type in these lines. You can also conserve memory by calculating the grid instead of using lines 5042–5132.

Lines 7500–7540 state the formula used. Again, they are purely for reference and need not be typed in.

Examples

The following example gives you some test data to verify that the program is free from errors.

Suppose that you have invented a better bear trap, the Bear Motel, and that the world is beating a path to your door to buy it. You have built 10 units and kept accurate records, as shown in Table 1.

Run the program and select item 1 from the first menu that appears, so that you can calculate the slope from the cost history.

The computer prints the function selected at the top of the screen and prompts for one of the known unit numbers: Press 4 and enter. The next prompt is for the hours for unit 4: Press 370 (from Table 1) and enter. Now you see the prompt for the other known unit number: Press 8 and enter. Finally, enter 259 hours for unit 8.

The answer appears instantly; you are on a 70 percent learning curve. It is always a good idea to check more values; therefore, after a few seconds the program prompts you to choose what you would like to do next. If you press C, you continue doing slope calculations. M allows you to select another function from the menu, and P draws a graph of the latest results calculated.

For now, press C to continue the same calculation. Go through the table trying various combinations of numbers until you are satisfied that you are on a 70-percent learning curve. Then press M to return to the menu to select another function.

Next, assume your accountant asks you when you will build a unit for less than 100 hours. Select function 2 from the menu. When asked for the slope, enter 70. For the known unit, enter 1, for unit 1 hour, enter 755, and for hours for the unknown unit, enter 99.8. The answer is unit 51.

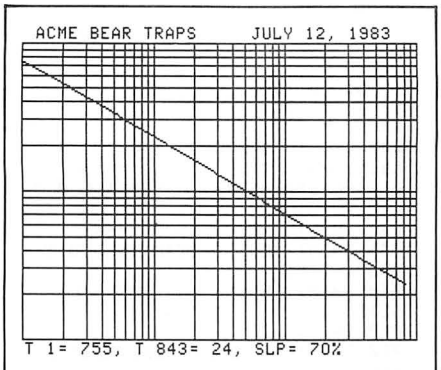


Fig. 2. Sample Learning Curve Graph

A Canadian fur trapper wants to buy your next 20 traps (units 11–30). He wants a price for all 20 traps. Select function 3 from the menu and when prompted, enter 70 for the slope, 1 for the known unit, 755 for the hours for unit 1, and 11 for the unit number for which you would like to know the hours. The answer is 219.8 hours.

Press M to return to the menu. Select function 4 to calculate the cumulative total. The program prompts you for the slope, for which you should enter 70. Enter 11 for the starting unit number, 219.8 for the hours for unit 11, and 30 for the ending unit number.

This time the computer takes a little longer since it must calculate hours for each individual unit. The calculation time increases in direct proportion to the number of units entered. The answer is 3,301 Cumulative Hours.

Your market research shows that you will sell 843 Bear Motels during the upcoming year. In order to determine the unit hours required to build unit 843, select item number 3 from the menu. When prompted, enter 70 for the slope, 1 for the known unit, 755 for the hours for unit 1, and 843 as the unit for which you would like to know the hours. The answer is 23.6 hours.

This time, instead of returning to the menu, press P to plot the results. The CoCo responds by asking you to enter the top title for the chart.

Basic only allows you to enter 32 letters per line. However, the characters this program draws are proportional and take up less space. You should be able to enter at least 34 per line. If you need longer titles, try it. At most, the last characters will just overwrite each other and you will have to try again.

After you have typed the title, press enter. If you have done everything right, your screen should look just like Fig. 2.

This program provides the essential calculations and plotting routines for a useful business technique that you can tailor to your individual needs.

You might want to plot more than two points, in which case you need to convert the scalar variables to arrays. Or perhaps you need lot totals, mid points, or averages. Then you need to add a few calculations or a machine-language subroutine for cumulative totals. You might need to add more characters or perhaps lowercase letters to the labeling routine.

If you do not have a C.Itoh or NEC printer, you have to change the screen-print section. I am interested in hearing about any changes and improvements in the program. ■

Write to Philip Wilcox at 16665 Olive Circle, Fountain Valley, CA 92708.

Program Listing. Learning Curve

```
10 CLS0
20 PRINT @73, "LEARNING CURVE";
30 PRINT @138,"(C) 1983 BY:";
40 PRINT @200,"PHILIP N. WILCOX"
;
50 '            16665 OLIVE CIRCL
E
60 '            FOUNTAIN VALLEY,
CA 92708
70 FOR J=1 TO 2300: NEXT
80 U1$="0":U2$="0":H1$="0":H2$="
0":E$="=":C$=",":SP$=" ":T$="T":
SLP$="SLP":P$="%"
90 GOTO 2000
95 '
100 REM **** SUBROUTINE TO CALCU
LATE AN UNKOWN UNIT NUMBER ****
110 CLS:PRINT@69,"CALCULATE UNIT
 NUMBER":PRINT
120 INPUT "ENTER SLOPE ";S
130 B=LOG(S/100)/LOG(2)
140 INPUT "ENTER A KNOWN UNIT NU
MBER ";U1
150 PRINT "ENTER THE HOURS FOR U
NIT";U1
160 INPUT H1
170 INPUT "ENTER THE HOURS FOR T
HE UNKNOWN UNIT ";H2
180 A=H1/U1^B
190 U2=EXP(LOG(H2/A)/B)
200 CLS
210 PRINT @102, "UNIT":PRINT @10
7,USING"####.#";U2:PRINT @114, "
@";H2 "HOURS"
220 PRINT @171, "BASED ON"
230 PRINT @266, "SLOPE = ";S "%"
240 PRINT @326, "UNIT";U1 "@";H1
 "HOURS"
250 GOSUB 2500
260 GOTO 110
270 '
500 REM **** SUBROUTINE TO CALCU
LATE THE VALUE OF AN UNKNOWN SLO
PE ****
510 CLS:PRINT@72,"CALCULATE SLOP
E":PRINT
520 INPUT "ENTER ONE OF THE KNOW
N UNIT     NUMBERS ";U1
530 PRINT "ENTER THE HOURS FOR U
NIT";U1
540 INPUT H1
550 INPUT "ENTER THE OTHER KNOWN
 UNIT      NUMBER ";U2
560 PRINT "ENTER THE HOURS FOR U
NIT";U2
570 INPUT H2
580 CLS
590 B=LOG(H1/H2)/(LOG(U1)-LOG(U2
)): S=EXP(B*LOG(2))*100
600 PRINT @106, "SLOPE = ": PRIN
T @114, USING"###.#";S :PRINT @1
19,"%"
610 PRINT @171, "BASED ON"
620 PRINT @ 262, "UNIT";U1 "@";H
1 "HOURS"
630 PRINT @326, "UNIT";U2 "@";H2
```

Listing continued

Listing continued

```
 "HOURS"
640 GOSUB 2500
650 GOTO 510
660 '
1000 REM **** SUBROUTINE TO CALC
ULATE CUMULATIVE TOTALS ****
1010 CLS:PRINT@70,"CALCULATE CUM
ULATIVE HOURS":PRINT
1020 INPUT "ENTER SLOPE ";S
1030 INPUT "ENTER STARTING UNIT
NUMBER ";U1
1040 PRINT "ENTER THE HOURS FOR
UNIT";U1
1050 INPUT H1
1060 INPUT "ENTER ENDING UNIT NU
MBER ";U2
1070 B=LOG(S/100)/LOG(2):A=H1/U1
^B:HC=0
1080 CLS:PRINT @ 103, "COMPUTING
..."
1090 FOR I=U1 TO U2 : Q=I^B: HC=
HC+Q: NEXT
1100 HC=A*HC
1110 CLS
1120 PRINT @39, USING"########.#
";HC:PRINT @47," CUM HOURS"
1130 PRINT @107, "BASED ON"
1140 PRINT @170, "SLOPE =";S"%"
1150 PRINT @229, "UNIT";U1 "@";H
1 "HOURS
1155 PRINT @297, "THROUGH UNIT";
U2
1160 B=LOG(S/100)/LOG(2):A=H1/U1
^B:H2=A*U2^B
1170 GOSUB 2500
1180 GOTO 1010
1190 '
1500 REM**** SUBROUTINE TO CALCU
LATE HOURS FOR A  GIVEN UNIT ***
*
1510 CLS:PRINT@70,"CALCULATE UNI
T HOURS":PRINT
1520 INPUT "ENTER SLOPE ";S
1530 INPUT "ENTER A KNOWN UNIT N
UMBER ";U1
1540 PRINT "ENTER THE HOURS FOR
UNIT";U1
1550 INPUT H1
1560 INPUT "ENTER THE UNIT NUMBE
R THAT YOU  WOULD LIKE TO KNOW T
HE HOURS FOR ";U2
1570 B=LOG(S/100)/LOG(2):A=H1/U1
^B:H2=A*U2^B
1580 CLS
1590 PRINT @ 99, "UNIT";U2:PRINT
 @109, "@":PRINT @111, USING"###
###.#";H2:PRINT @119, " HOURS"
1600 PRINT @ 171, "BASED ON"
1610 PRINT @ 266, "SLOPE = ";S "
%"
1620 PRINT @ 326, "UNIT";U1 "@";
H1 "HOURS"
1630 GOSUB 2500
1640 GOTO 1510
1650 '
2000 CLS: REM**** MAIN MENU ****
2010 PRINT @100,"PRESS NUMBER BY
 FUNCTION"
2020 PRINT @132,"TO CALCULATE:"
2030 PRINT @232,"1. SLOPE"
2040 PRINT @264,"2. UNIT NUMBER"
2050 PRINT @296,"3. UNIT HOURS"
2060 PRINT @328,"4. CUMULATIVE T
OTALS"
2065 PRINT @360,"5. PRINT GRAPH"
2070 R$=INKEY$: IF R$="" GOTO 20
70
2080 R=VAL(R$): IF R<1 OR R>5 TH
EN GOTO 2070
2090 ON R GOTO 510,110,1510,1010
,6000
2100 '
2500 FOR J=1 TO 2300: NEXT
2510 PRINT @385, "PRESS C TO CON
TINUE, M TO GOTO  MENU, P TO PLO
T RESULTS
2520 R$=INKEY$: IF R$="" THEN GO
TO 2520
2530 IF R$="C" THEN RETURN
2540 IF R$="M" THEN GOTO 2000
2550 IF R$="P" THEN GOTO 3010
2560 GOTO 2520
2570 '
2990 X=U2:U2=U1:U1=X:X=H2:H2=H1:
H1=X:RETURN: REM**** ADJUSTS FOR
 HIGHER UNIT NUMBER BEING ENTERE
D FIRST ****
3000 REM**** SET-UP SUBROUTINE F
OR CALCULATION OF PLOT POINTS **
**
3010 X=INT(S+.5):S$=STR$(X):CLS
3020 IF U1>U2 THEN GOSUB 2990
3030 P=INT(U1+.5):U1$=STR$(P):C=
0:GOSUB 3210
3040 U1=P
3050 X=INT(U2+.5):U2$=STR$(X)
3060 P=U2/10^C:GOSUB 3170
3070 U2=P
3080 P=INT(H1+.5):C=0:H1$=STR$(P
):GOSUB 3150
3090 P=181-P:H1=P
3100 Y=INT(H2+.5):H2$=STR$(Y)
3110 P=H2/10^C:GOSUB 3170
3120 P=181-P:H2=P
3130 GOTO 3500
3140 REM**** SUBROUTINE TO CALCU
LATE Y-AXIS PLOT POINTS ****
3150 IF P <= 100 GOTO 3170
3160 IF P > 100 THEN P=P/10:C=C+
1:GOTO 3150
3170 IF P=1 THEN LET P=0:IF P=0
THEN GOTO 3190
3180 P=85*(LOG(P)/LOG(10))
3190 RETURN
3200 REM**** SUBROUTINE TO CALCU
LATE X-AXIS PLOT POINTS ****
3210 IF P <=1000 GOTO 3170
3220 IF P >1000 THEN P=P/10:C=C+
1:GOTO 3210
3230 '
3500 CLS:PRINT:PRINT:PRINT"IN 34
 CHARACTERS OR LESS ENTER  TOP T
ITLE":PRINT@169,"***EXAMPLE***"
3510 PRINT @ 225, "ACME BEAR TRA
PS  JULY 12, 1983":LINEINPUT TT$
: REM **** TOP TITLE ****
3520 BT$=T$+U1$+E$+H1$+C$+SP$+T$
+U2$+E$+H2$+C$+SP$+SLP$+E$+S$+P$
:REM**** BOTTOM TITLE ****
3530 '
5000 REM**** GRAPHICS PLOTTING S
UBROUTINE ****
5010 PMODE4,1
5020 PCLS
5030 SCREEN 1,0
5040 LINE (0,11)-(255,181),PSET,
B
5042 LINE (26,11)-(26,181),PSET
5044 LINE (41,11)-(41,181),PSET
5046 LINE (51,11)-(51,181),PSET
5048 LINE (59,11)-(59,181),PSET
5050 LINE (66,11)-(66,181),PSET
5052 LINE (72,11)-(72,181),PSET
5054 LINE (77,11)-(77,181),PSET
5056 LINE (81,11)-(81,181),PSET
5058 LINE (85,11)-(85,181),PSET
5060 LINE (111,11)-(111,181),PSE
T
5062 LINE (126,11)-(126,181),PSE
T
5064 LINE (136,11)-(136,181),PSE
T
5066 LINE (144,11)-(144,181),PSE
T
5068 LINE (151,11)-(151,181),PSE
T
5070 LINE (157,11)-(157,181),PSE
T
5072 LINE (162,11)-(162,181),PSE
T
5074 LINE (166,11)-(166,181),PSE
T
5076 LINE (170,11)-(170,181),PSE
T
5078 LINE (196,11)-(196,181),PSE
T
5080 LINE (211,11)-(211,181),PSE
T
5082 LINE (221,11)-(221,181),PSE
T
5084 LINE (229,11)-(229,181),PSE
T
5086 LINE (236,11)-(236,181),PSE
T
```

Listing continued

Listing continued

```
5088 LINE (242,11)-(242,181),PSE
T
5090 LINE (247,11)-(247,181),PSE
T
5092 LINE (251,11)-(251,181),PSE
T
5100 LINE (0,155)-(255,155),PSET
5102 LINE (0,140)-(255,140),PSET
5104 LINE (0,130)-(255,130),PSET
5106 LINE (0,122)-(255,122),PSET
5108 LINE (0,115)-(255,115),PSET
5110 LINE (0,109)-(255,109),PSET
5112 LINE (0,104)-(255,104),PSET
5114 LINE (0,100)-(255,100),PSET
5116 LINE (0,96)-(255,96),PSET
5118 LINE (0,70)-(255,70),PSET
5120 LINE (0,55)-(255,55),PSET
5122 LINE (0,45)-(255,45),PSET
5124 LINE (0,37)-(255,37),PSET
5126 LINE (0,30)-(255,30),PSET
5128 LINE (0,24)-(255,24),PSET
5130 LINE (0,19)-(255,19),PSET
5132 LINE (0,15)-(255,15),PSET
5200 LINE (U1,H1)-(U2,H2),PSET
5210 DRAW "C1;S4;BM2,8": REM ***
* POSITION PEN ****
5220 TITLE$=TT$
5230 GOSUB 5500
5240 DRAW "C1;S4;BM2,189"
5250 TITLE$=BT$
5260 GOSUB 5500
5270 IF INKEY$="" GOTO 5270 ELSE
 GOTO 2000
5280 '
5500 REM**** SUBROUTINE FOR DRAW
ING CHARACTERS ON GRAPHICS SCREE
N ****
5510 FOR LNGTH=1 TO LEN(TITLE$)
5520 SCHAR=0
5530 RESTORE
5540 READ SCHAR$,FCHAR$
5550 IF SCHAR$=MID$(TITLE$,LNGTH
,1) THEN DRAW FCHAR$: GOTO 5570
5560 SCHAR=SCHAR+1: IF SCHAR<51
THEN 5540
5570 NEXT: RETURN
5600 DATA " ","BM+7,0"
5602 DATA"0","BM+1,0H1U4E1R2F1D4
G1L2BM+6,0"
5604 DATA"1","BM+1,0R1NR1U6G1BM+
6,5"
5606 DATA"2","NR4U1E1R1E2U1H1L2G
1BM+7,5"
5608 DATA"3","BM+0,-1F1R2E1U1H1N
L1E1U1H1L2G1BM+7,5"
5610 DATA"4","BM+3,0U2NR1L3U1E3D
3BM+4,3"
5612 DATA"5","BM+0,-1F1R2E1U2H1L
3U2R4BM+3,6"
5614 DATA"6","BM+4,-5H1L2G1D4F1R
2E1U1H1L3BM+7,3"
5616 DATA"7","U1E4U1L4BM+7,6"
5618 DATA"8","BM+1,0H1U1E1H1U1E1
R2F1D1G1NL2F1D1G1L2BM+6,0"
5620 DATA"9","BM+0,-1F1R2E1U4H1L
2G1D1F1R2BM+4,3"
5622 DATA",","BM+2,0NU1G1BM+6,-1
"
5624 DATA"=","BM+1,-2R3BM-3,-2R3
BM+4,4"
5626 DATA"%","BM+0,-5U1R1D1L1BM+
0,+5U1E4U1BM+0,6L1U1R1D1BM+3,0"
5628 DATA"/","U1E4U1BM+3,6"
5630 DATA"-","BM+0,-3R4BM+3,3"
5632 DATA"A","U4E2F2D2NL4D2BM+3,
0"
5634 DATA"B","U6R3F1D1G1NL3F1D1G
1L3BM+7,0"
5636 DATA"C","BM+1,-0H1U4E1R2F1B
M+0,+4G1L2BM+6,0"
5638 DATA"D","NL1U6NL1R3F1D4G1L3
BM+7,0"
5640 DATA"E","NR4U3NR2U3R4BM+3,6
"
5642 DATA"F","U3NR2U3R4BM+3,6"
5644 DATA"G","BM+1,-0H1U4E1R2F1B
M+0,2NL1D2G1L2BM+6,0"
5646 DATA"H","U3NU3R4NU3D3BM+3,0
"
5648 DATA"I","BM+1,0R1NR1U6NL1R1
BM+4,6"
5650 DATA"J","BM+0,-1F1R1E1U5NL1
R1BM+3,6"
5652 DATA"K","U3NU3R1NE3F3BM+3,0
"
5654 DATA"L","NU6R4U1BM+3,1"
5656 DATA"M","U6F2ND1E2D6BM+3,0"
5658 DATA"N","U6F1D1F2D1F1NU6BM+
3,0"
5660 DATA"O","BM+1,0H1U4E1R3F1D4
G1L3BM+7,0"
5662 DATA"P","U6R3F1D1G1L3BM+7,3
"
5664 DATA"Q","BM+1,0H1U4E1R2F1D3
G1NH1NF1G1L1BM+6,0"
5666 DATA"R","U6R3F1D1G1L2NL1F3B
M+3,0"
5668 DATA"S","BM+0,-1F1R2E1U1H1L
2H1U1E1R2F1BM+3,+5"
5670 DATA"T","BM+2,0U6NL2R2BM+3,
6"
5672 DATA"U","BM+0,-1NU5F1R2E1U5
BM+3,6"
5674 DATA"V","BM+0,-6D2F1D1F1ND1
E1U1E1U2BM+3,6"
5676 DATA"W","NU6E2NU1F2U6BM+3,6
"
5678 DATA"X","U1E4U1BM-4,0D1F4D1
BM+3,0"
5680 DATA"Y","BM+0,-6D2F2ND2E2U2
BM+3,6"
5682 DATA"Z","NR4U1E4U1L4BM+7,6"
5684 DATA"?","BM+0,-5E1R2F1D1G2B
M+0,2D1BM+5,-1"
5686 DATA"!","BM+2,1U1BM+0,-2U4B
M+5,6"
5688 DATA".","BM+2,0U1BM+5,1"
5690 DATA":","BM+2,-1U1BM+0,-2U1
BM+5,5"
5692 DATA";","BM+1,1E1U1BM+0,-2U
1BM+5,4"
5694 DATA"'","BM+1,-5E2BM+4,7"
5696 DATA"+","BM+2,-1U2NU2NL2R2B
M+3,3"
5698 DATA"(","BM+2,0H2U2E2BM+3,6
"
5700 DATA")","E2U2H2BM+6,6"
5710 '
6000 REM **** SUBROUTINE TO ACCE
SS SCREEN PRINT PROGRAM ****
6005 IF SKIP= 1 THEN GOTO 6080
6010 CLS:PRINT @71,"PREP TAPE TO
 LOAD"
6015 PRINT @134,"SCREEN PRINT PR
OGRAM"
6020 PRINT @260,"PRESS ANY KEY W
HEN READY"
6030 R$=INKEY$: IF R$="" THEN 60
30
6050 CLS: PRINT @104,"LOADING PR
OGRAM":PRINT @170,"PLEASE WAIT"
6054 CLEAR 200,&H7D8F
6056 CLOADM"GSPRP ",&H7D90-&H600
6058 SKIP=1
6060 DEF USR0=&H7D90
6070 DEF USR1=&H7D92
6080 CLS: PRINT @100,"PRESS NUMB
ER BY FUNCTION"
6090 PRINT @132,"TO PRINT:"
6100 PRINT @232,"1. NORMAL IMAGE
"
6110 PRINT @264,"2. INVERSE IMAG
E"
6120 PRINT @296,"3. DOUBLE SIZE"
6130 R$=INKEY$: IF R$="" THEN 61
30
6140 R=VAL(R$): IF R<1 OR R>3 TH
EN 6130
6150 CLS: SCREEN 1,0
6160 ON R GOTO 6200,6220,6240
6200 X=USR0(0)
6210 GOTO 80
6220 X=USR1(0)
6230 GOTO 80
6240 POKE32181,2:POKE32189,95:PO
KE32206,4:POKE32211,47:POKE32341
,48:POKE32421,18:POKE32422,18
6250 X=USR0(-128)
6260 POKE32181,1:POKE32189,223:P
OKE32206,2:POKE32211,111:POKE323
41,24:POKE32421,38:POKE32422,13
6270 GOTO 80
7000 REM **** VARIABLE TABLE ***
*
7010 ' VAR          NAME
LINE NO.
7020 ' A        1ST UNIT HOURS
180,1070,1100,1160,1570
7022 ' B        EXPONENT
130,180,590,1070,1090,1160,1570
7024 ' BT$      BOTTOM TITLE
3520,5250
7026 ' C        COUNTER/EXPONENT
3030,3060,3080,3110,3160,3220
7028 ' C$       "COMMA"
80,3520
7030 ' FCHAR$   FOUND CHARACTER
5540,5550
7032 ' E$       EQUAL SIGN
80,3520
7034 ' H1       LOW UNIT HOURS
160,180,240,540,590,620,1050,107
0,1150,1160,1550,1570,1620,2990,
3080,3090
7036 'H2        HIGH UNIT HOURS
170,210,570,590,630,1160,1590,29
90,3100,3110,3120
7038 'H1$       HOURS AS STRING
80,1050,1550,3080,3090,3510
7040 'H2$       HOURS AS STRING
80,1160,1570,3100,3520
7042 'HC        HOURS CUMULATIVE
1070,1090,1100,1120
7044 'I         INCREMENT
1090
7046 'J         PAUSE LOOP
70,2500
7048 'LNGTH     TITLE LENGTH
5510,5550
7050 'P         PLOT POINT
3030,3040,3060,3070,3080,3090,31
10,3120,3150,3160,3170,3180,3210
,3220
7052 'P$        PERCENT SIGN
80,3520
7054 'Q         CALC RESULT
1090
7056 'R         VAL(R$)
2080,2090,6140,6160
7058 'R$        INKEY$ RESPONCE
2070,2080,2520,2530,2540,2550,61
30,6140
7060 'S         SLOPE
120,130,230,590,600,1020,1070,11
40,1160,1520,1570,1610,3010
7062 'S$        SLOPE STRING
3010,3520
7064 'SCHAR     COUNTER
5520,5560
7066 'SCHAR$    SEARCH CHARACTER
5540,5550
7068 'SLP$      "SLP" FOR BT$
80,3520
7070 'SP$       SPACE
80,3520
7072 'T$        CHARACTER "T"
80,3520
7074 'TITLE$    TITLE FOR DRAW
5220,5250,5510,5550
7076 'TT$       TOP TITLE
7270 'U1        LOW UNIT NUMBER
140,150,180,240,520,530,590,620,
1030,1040,1070,1090,1150,1160,15
30,1540,1570,1620,2990,3020,3030
,3040
7078 'U2        HIGH UNIT NUMBER
190,210,550,560,590,630,1060,109
0,1160,1560,1570,1590,2990,3020,
3050,3060,3070
7080 'U1$       LOW UNIT STRING
80,3010,3520
7082 'U2$       HIGH UNIT STRING
80,3080,3520
7084 'X         EXCHANGE XPOINTS
2990,3010,3050
7086 'Y         EXCHANGE YPOINTS
3100
7100 '
7500 REM **** FORMULA ****
7510 ' Y=AX^B, WHERE Y= UNKNOWN
UNIT HOURS
7520 '                A= HOURS FO
R FIRST UNIT BUILT
7530 '                X= ANY UNIT
 NUMBER
7540 '                B= EXPONENT
```

END

BY ELLEN AFTAMONOW

DON'T BUG ME

Debugging is one of the least enjoyable and most frustrating phases of programming. No matter how careful you are, you always manage to hit a wrong key. But debugging is not all that bad if you do it in a rational manner.

The easiest bug to find is the syntax (SN) error. The SN error occurs most often when you mistype something on a line, so check for characters that resemble each other. The letter B looks like the numeral 8, the letter I looks like the numeral 1, M looks like N, and the letter O looks like the numeral 0.

Sometimes you've typed in everything correctly, but the unforgiving machine still insists that there is an SN error because some commands need spaces between them. For example, line 110 of the Program Listing will not run as:

```
FORT=JTOK
```

unless you add spaces. Changing line 110 to:

```
FORT=J TO K
```

works just fine. If in doubt, use spaces. You can always delete them later.

The type-mismatch (TM) error, while not as common as the SN error, is easy to find. Just remember that anything written in quotation marks that is not preceded by a PRINT statement must have a $ notation, so look for a missing or mistyped $. For instance, INKEYS="I" produces a TM error, while INKEY$="I" does not.

The function-call (FC) error is the hardest to find. The FC error is often not in the line that the computer lists, but actually in a previous line. First check the listed line. If it is accurate, see if it contains a string. In the Program Listing, an error in line 20 or 30 (A$ and B$) would show up as an FC error in line 80, because this is the first place that the strings are used. Lines 20 and 30 just define the strings.

Therefore, when you tell the computer to:

```
DRAW"BM92,68;"+A$+B$
```

in line 80, the computer says that it can't do so because A$ or B$ was written incorrectly in line 20 or 30. If after checking these lines you still do not find the error, change line 80 to read:

```
DRAW"BM92,68;"+A$
```

If the line executes, do the same for B$:

```
DRAW"BM92,68;"+B$
```

If you get an FC error, then line 30 has the error.

The same is true in line 140 with:

```
PLAY C$+D$+E$
```

An FC error will be listed for line 140 because it is here that the program executes the command. The actual error might have occurred in line 40, 50, or 60, where C$, D$, and E$ are defined. Always go back and check the line in which you first defined a string variable. You can quickly determine the guilty string variable by executing each one at a time. For instance, if:

```
PLAY C$
```

executes, you know that either D$ or E$ are incorrectly defined.

A GET statement that is not dimensioned large enough will also generate an FC error. Change line 10 in the program to read:

```
DIM R(1,1)
```

and see what happens.

If your program has GET and PUT

```
10 DIMR(1,2)
20 A$="C7E8F8L16"
30 B$="C8BR3ØBD16H4R8G4"
40 C$="T4O2L8EL4GP4O3L2E"
50 D$="L4DCCL8CO2L4.A"
60 E$="P4O3L4DL8CO2L4BO3L2C"
70 PMODE 3,1;PCLS:SCREEN1,1
80 DRAW"BM92,68;"+A$+B$
90 CIRCLE(122,84),16,8,1,.1,.42
100 J=1:K=1000
110 FORT=JTOK:NEXTT
120 GET(92,58)-(112,68),R,G
130 PUT(132,58)-(152,68),R,PSET
140 PLAY C$+D$+E$
150 GOTO 150
```

Program Listing. Sample Errors

Defeat programming dirty work by taking the pain out of debugging. Here's a thorough guide for you.

statements and runs without any error messages, but you get garbage on the screen, chances are that the coordinates of the GET and PUT statements are not of the same length. If line 130 read:

```
PUT(132,58)-(153,69),R,PSET
```

you would get garbage on the screen. The GET rectangle is 20 by 10 units long, while your PUT rectangle is 21 by 11 units long.

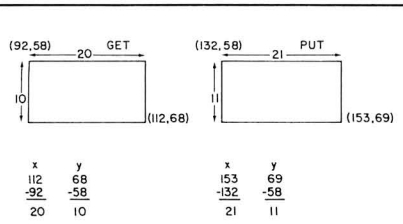


Fig. 1. GET/PUT Rectangles

If you get an FC error in line 80 but want to see if the rest of the program runs, you cannot bypass the error by typing RUN 100. There might be strings, dimensions, or other information vital to the program in previous lines. So, if you ask the computer to RUN 100, you get an FC error of your own making.

You can also get an FC error if the coordinates are out of range:

```
DRAW"BM260,260;"
```

will not work because the graphics screen goes only to 256,192.

Whenever you enter a long line number, make sure it is entered completely before typing in the next line number. Otherwise, line 1120 could become line 120 or line 20. Your original line 120 will then be replaced by what should be line 1120.

The TRON (trace on) function is an excellent tool. Use it to make sure the program runs in its proper order. You won't be able to see a graphics screen while it is in use, but you can still follow the program's progress. Don't forget to type TROFF (trace off) when you are finished.

Here's a hint to help find an error in a line containing a lot of characters. Put the line in the edit mode. Then press the space bar as you go along. That way, you won't lose your place on the screen. ■

Address correspondence to Ellen Aftamonow, 46 Howe St., Milford, CT 06460.

BY JAMES J. BARBARELLO

ROM HACKER

Part III

In this third part of the "ROM Hacker" series, I'll show you how to construct the CoCo Mouse. The Mouse is a two-wheeled, motorized vehicle. The Mouse gets its instructions from the CoCo through the Master Interface, which translates those instructions into movement.

You're familiar with Logo, aren't you? Well, for those of you who aren't, Logo is a computer application in which a triangle-shaped object (turtle) moves about the screen. It is told where and how far to move by the operator (usually a child). When it moves, it leaves a line, and thus is capable of drawing patterns (sometimes very intricate).

Logo has its own syntax, sort of an abbreviated language. A similar program (Mouse Logo) controls the CoCo Mouse. With it, you'll have your mechanized rodent moving about your computer room floor in the most amusing manner. Mouse Logo also has its own syntax, which a youngster can easily use. It also lets you create procedures (a series of commands). You can imbed procedures within other procedures for quite intricate movement. Finally, so you don't have to key in all those commands every time you turn on your CoCo, Mouse Logo lets you save and retrieve up to 20 procedures from tape.

One other nice feature about this project is that it shows you how to control high-current devices like motors, relays, solenoids, and the like. Once you understand the operation of the Mouse Interface, you can modify it to do some other things around the house.

System Requirements

16K RAM
Extended Color Basic

Motor Control

A dc motor is a simple device to understand. There are two terminals to which you apply a dc voltage. The motor specifications include input voltage and current. For instance, you might see a motor rated at 6 volts dc, 100 milliamps. To have the motor run at its rated speed, your power supply must deliver 6 volts dc at 100 milliamps. If it doesn't, the motor either runs slower than rated or not at all.

A dc motor's leads are polarity sensitive. Apply the voltage with one polarity and the motor shaft rotates clockwise, reverse the polarity and the shaft rotates counterclockwise.

The Mouse

The project is simply two motors, each with its own wheel. To allow each wheel to go forward and backward, you must have a method of applying voltage to each motor and reversing the polarity of the applied voltage. The Master Interface is not a likely candidate since it can provide only about 10 milliamps per peripheral line. You need a buffer between the PIA lines in the Master Interface and the motors.

The schematic diagram of Fig. 1 shows our Mouse Interface. It consists of two identical circuits, one for each motor. Furthermore, each motor circuit contains two identical PIA peripheral line buffer circuits. Let's see how they work by looking at relays K1 and K2. Note that their normally open (NO) contacts are connected to a positive voltage supply, and their normally closed (NC) contacts are connected to ground. "Normally" means that the relay is not energized. In this state, the NC contact is connected to the common contact. When you energize the relay, this contact is opened, and the NO contact now connects to the common contact.

If you start out with both relays unenergized, 0 volts (ground) are applied to both leads of the motor. With no voltage potential across it, the motor stays at rest. Similarly, if you energize both relays, 5 volts are across both leads

Construct this mechanized rodent and use "Mouse Logo" to delight even the oldest child.

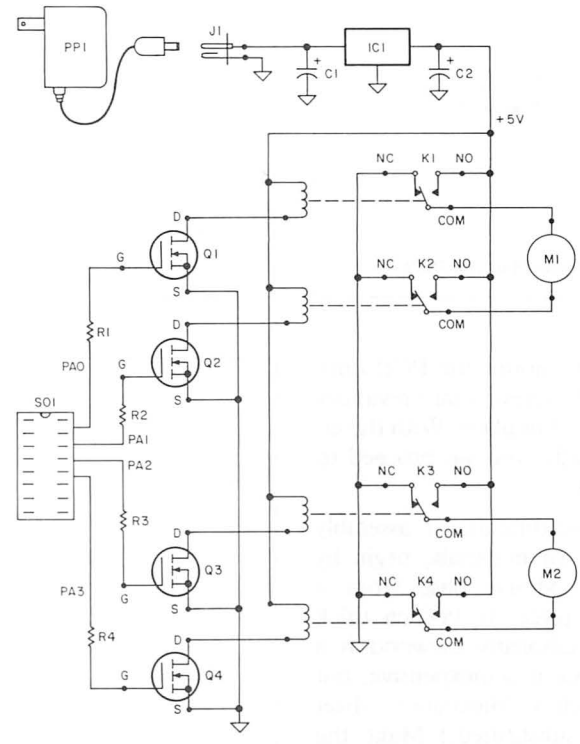


Fig. 1. Mouse Interface

of the motor. But there is still no *difference* in potential across the motor, so it stays at rest. Now, let's leave K1 energized and de-energize K2. One side of the motor has + 5 volts applied, and the other has 0 volts. This 5-volt potential difference causes the motor shaft to rotate. Then, if you de-energize K1 and energize K2, you reverse the potential across the motor, and the shaft rotates in the opposite direction.

How do you energize the relays? You could use a very sensitive relay that the Master Interface PIA could power directly, but such relays are expensive. Instead, use a standard, low-cost relay and a buffer device called a power MOSFET. MOSFET stands for metal oxide semiconductor field effect transistor.

The power MOSFET is like a transistor, but with some very different characteristics. If you apply at least 2 volts to the gate (G) of the MOSFET, you close the almost ideal semiconductor switch between its drain (D) and source (S). An ideal switch would have 0 ohms resistance. The MOSFET switch has 3.2 ohms resistance maximum (very close to ideal). Aso, it can pass 400 milliamps continuously, more than enough to energize a relay. You would have to use cascaded transistors to approach the same performance. At about $1.50 each, the power MOSFET is a bargain and makes interfacing simple.

As you see in Fig. 1, the coil of K1 is in series with Q1. So when you turn on Q1, it allows current to flow through K1's coil, energizing it. When you turn off Q1, the current path is broken and K1 is de-energized. The control signals for Q1 through Q4 are obtained from PIA peripheral lines PA0 through PA3. If you make PA0 high and PA1 low, the motor is energized. If you then make PA0 low and PA1 high, the motor reverses direction. If you make PA0 and PA1 both high or low, the motor stops.

The motor I used is a 6-volt type. To handle the relays and motors, you need a lot more power than the CoCo can provide without straining its power supply. That power is provided by PP1, a 9-volt dc power-pak adaptor (one of those "cube" power supplies that plugs directly into the wall). You take the raw 9 volts dc and regulate it down to 5 volts. This is more in line with the motor requirements, and the 1-volt difference is of no consequence in our application.

While the power MOSFET can pass up to 400 milliamps, the relay contacts can handle 1 amp at 125 volts! So this buffering scheme lets you control devices much more demanding than our motors. By using a relay with a similar coil current but larger relay contacts, you could control many household machines and appliances.

Building the Mouse

The project consists of two separate items: the Mouse Interface and the Mouse itself. Let's begin with the Mouse Interface. (See Table 1 for the list of materials.) The printed circuit board (PCB) replaces the cover of the case identified in the list of materials. So to start, size the PCB blank and locate the four mounting holes using the case cover. Now fabricate the PCB using the pattern of Fig. 2. When done, mount the components as shown in Fig. 3. Note that the four power MOSFETs are inserted into DIP socket SO2. When inserted correctly, you will be able to read the marking on the MOSFETs if you hold the PCB so the relays are to your left. Be sure to observe the proper polarity for capacitors C1 and C2, and IC1. (Pay special attention to IC1's tab.)

The final step is connecting a four-conductor cable to the PCB. The cable can be any four-conductor type, including a coiled telephone cord. To allow freedom of movement, however, the cable should be at least 15 feet in length; 25 feet would be even better. The coiled cord might be your best option, since it will be less prone to interfering with the Mouse as it cavorts about. With the

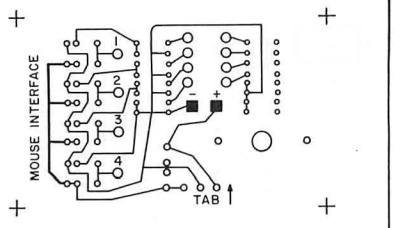


Fig. 2. Mouse Interface PCB

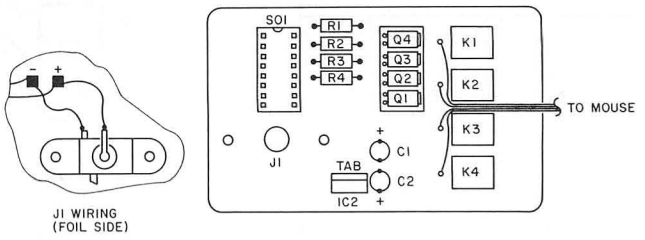


Fig. 3. Mouse Interface Component Placement

cable connected, mount the PCB onto the case using the screws that previously held the case cover in place. With the interface completed, you can proceed to the Mouse itself.

If you use the dual-motor assembly cited in the list of materials, begin by fabricating the Mouse base from a 12-inch square piece of ¼-inch thick stock. (Luan mahogany plywood is a good choice since it is inexpensive, but any material such as Masonite or sheet plastic can be substituted.) Make the cutouts in the material as shown in Fig. 4. The final shape of the base is up to you. If you want to add a "body" to the mouse, plan your base so it will accommodate that structure.

Obtain four 1½-inch long machine screws and 12 machine nuts. Attach the wheels to the motor assembly and make a trial fit onto the base. Make sure the unit mounts securely to the base and the wheels move freely. If not, remove the assembly and remachine. Since the Mouse has only two wheels, it can tip forward and back. So the base will need two more points of support. Referring to Fig. 4, detail A, mount these supports. They can be casters or ordinary stiff wire formed as shown. Finally, mount the five-lug terminal strip on the base in back of the assembly closest to where the motor leads exit. Connect the interface's cable to the Mouse as shown in Fig. 5.

These instructions are meant as a guide. You can make your base smaller or larger, octagonal or triangular. You don't even have to use the dual-motor assembly; any two 6-volt dc motors and appropriate wheels will do. Be a bit creative in giving life to this mechanical rodent.

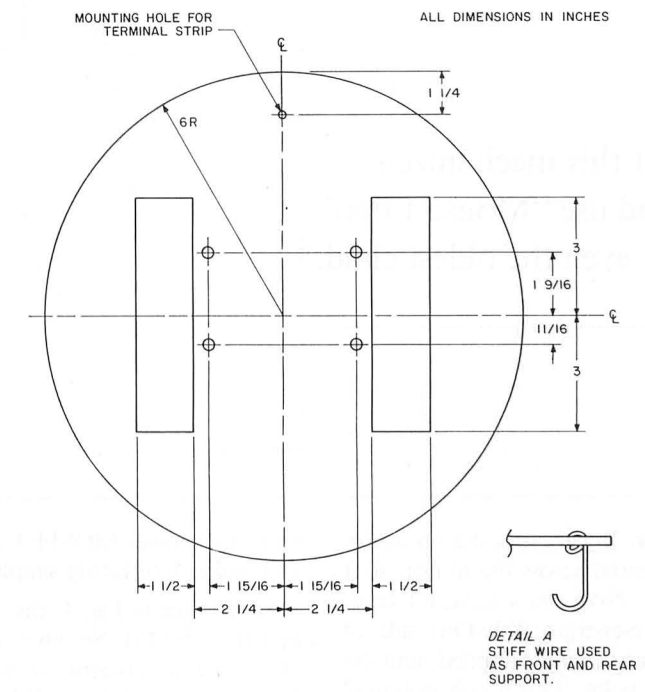


Fig. 4. Mouse Base

The Program

Type in the program shown in Listing 1 and save it using the name MLOGO.

As you've done before, POKE C+1, 0 to select the data-direction register of the Master Interface PIA, side A (you do not use PIA side B for this project). Then, POKE C,255 sets all lines to outputs, POKE C+1,4 reselects the peripheral register and POKE C,0 initializes all output lines to logic level 0 (ground).

Line 20 also defines array elements C$(0) through C$(10), the available movement commands (Table 2). To understand these commands, refer back to Fig. 1. If you use the array element number as the number POKEd to the PIA, let's see what would happen if you selected C$(10).

Ten decimal is 1010 binary. POKEing 10 decimal to the PIA would make PA3=1, PA2=0, PA1=1, and PA0=0. PA0 and PA1 control one motor; PA2 and PA3 control the other. PA0 de-energizes relay K1, PA1 energizes relay K2. (This energizes Motor M1.) Similarly, PA2 and PA3 energize motor M2. Our wiring scheme is such that this produces forward rotation in each motor. Therefore, C$(0) causes both motors to be de-energized (0 decimal = 0000 binary). Thus the definition of C$(0) as S stands for stop.

Array elements C$(3) and C$(7) are defined as X, since they produce results already defined by other elements. For instance, C$(3) produces a binary code 0011. The 00 de-energizes motor M2, and the 11 de-energizes motor M1. So C$(3) is a redundant stop command. C$(7) (0111 binary) duplicates C$(4) (0100 binary), since motor M1 is de-energized and motor M2 is energized.

Using Mouse

Begin by connecting your Master Interface to your unpowered CoCo. Then attach the Master Interface's DIP connector to the Mouse Interface DIP socket. (The DIP cable will be facing *away* from the relays.) Now plug the power pak dc connector into the Mouse Interface and then plug the power pak into the wall. Place the mouse in the

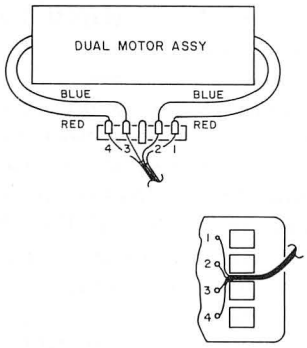


Fig. 5. Mouse Interface Interconnection

Reference	Description	Part #	Source
R1–R4	2.2Kohm, ¼-watt resistor	271-1325	RS
C1,C2	1.0 μF tantalum electrolytic capacitor	272-1434	RS (JA)
IC1	7805 + 5-volt regulator (TO220 case)	276-1770	RS (JA)
IC2–IC5	IRFD-1Z3 power MOSFET (four-pin DIP)	276-2073	RS
K1–K4	SPDT microminiature PC relay	275-240	RS
SO1, SO1	16-pin DIP socket	276-1998	RS (JA)
PP1	9-volt dc power-pak adaptor	273-1651	RS (JA)
J1	Chassis mount jack (5mm OD, 2.1mm ID)	274-1549	RS
M1	Dual motor assembly	TM22K638	H/R
WH1, WH2	Wheel	TM22K677	H/R
TS1	Five-lug terminal strip	274-688	RS

Miscellaneous: 1 square foot of ¼-inch plywood, PC board, 15–25 feet of 4 conductor wire (see text), case (Radio Shack P/N 270-230), solder, etc.

SOURCES: RS: Radio Shack

JA: Jameco Electronics, 1355 Shoreway Road, Belmont, CA 94002. 415-592-8097. Call for ordering information (alternate source for parts where cited; check catalog for part numbers). H/R: Herbach & Rademan Inc., 401 East Erie Ave., Philadelphia, PA 19134. 215-426-1700. Call for ordering information.

Table 1. Mouse Project List of Materials

center of the floor. If possible, suspend the cable from above the mouse, leaving enough slack for it to travel around freely. Turn on your CoCo; load and run the MLogo program.

You are first asked "Load procedure file (Y/N)?." Press N. The screen clears with the title "Mouse Logo command mode." Below the title are the instructions "(HELP = Instructions, End to end)." Below the title is a question mark. Enter F 20. (Make sure there is a space between F and 20.) Your mouse should proceed forward for a short distance and then stop. If it rotates left or right, one of the motor connections is reversed. If it goes backward, both motor connections are reversed. Change connections until an F command produces forward motion of both wheels.

Now enter F20 (no space). The message "Command not recognized" appears above the question mark. Enter HELP. The help screen immediately appears. After you've read the help screen, press enter to return to the command mode. Enter MAKE TEST. The make screen now appears. Enter the following commands (press enter after each command and include spaces where shown, but do not enter the commas):

F 20, S 10, R 20, END

You will now see "Procedure complete. Press enter to continue." Press enter to return to the command mode. Enter DO TEST. The screen clears and the message "Executing test" appears. Your mouse goes forward, waits a bit, and then turns right. When it stops, the screen returns to the command mode.

Now enter MAKE SQUARE. When the make screen appears, enter the commands:

TEST, S 30, TEST, S 30, TEST, S 30, TEST, END

Press enter to return to the command mode and enter DO SQUARE. Your mouse should perform the Square routine.

Enter LIST ALL. The message "Procedure not available" appears above the question mark. Below the question mark you'll see "Procedures: Test square." Enter LIST TEST. The list screen and the listing of TEST (F 20/S 10/R 20/END) appear. Press enter to return to the command mode and enter LIST SQUARE. The listing of SQUARE (TEST/S 30/TEST/S 30/TEST/S 30/TEST/END) appears. Press enter to return to the command mode.

Since the mouse is a nonprecision device, the R 20 command in test will probably not have produced an exact right-angle turn. Let's revise Test by entering MAKE TEST. The make screen appears and states "Procedure exists. Redo (Y/N)?." Press N and the command screen returns with the message "Redo test aborted" above the question mark. Enter LIST TEST and note it hasn't changed. Again enter MAKE TEST, this time pressing Y in response to the redo question. Enter the commands:

F20, S 30, R 14, END

Press enter to return to the command mode and enter DO SQUARE. Note that there is a longer delay between movements (you changed S 10 to S 30), and the right turn is less severe. Now, at the command mode, enter END. The screen asks "Save procedure file (Y/N)?." Press N. The screen now says "No save. Program ended (enter GOTO 30 to reenter)." Enter GOTO 30 and note you are returned to the command mode.

Enter LIST ALL and note that both procedures (Test and SQUARE) are still there. Enter END and respond Y to the save question. The screen will now advise "Prepare cassette recorder. Press enter when ready." Prepare your cassette recorder for recording (make sure all three plugs are inserted, giving the CoCo control over your cassette deck's motor) and press enter. The message "SAVING..." appears, and after a short time it is followed by "DONE." The program then ends.

Rerun the program. As it begins it asks "Load Procedure File (Y/N)?." Press Y. The "Prepare cassette recorder" message again appears. Position the tape before the procedure file and press enter. The screen advises "Searching...," and then Loading...." When the load is complete, you are presented with the command mode screen. Enter LIST ALL and note that the procedures Test and Square are available.

Summing It Up

The Mouse project can be very useful in educational applications. It requires the youngster to understand, plan, and revise. Also, the physically moving mouse will delight even the oldest child. You can modify the mouse by drilling a hole in the base and inserting a pen or pencil. If the mouse is then set on a large piece of paper you can change the

mechanized rodent into an artist, drawing all kinds of interesting designs. When using the mouse, it is a good idea to suspend the cable a few feet over the mouse. In this manner, the cable will not get caught under the mouse as it moves and rotates.

The more practically minded among you can modify the interface to control household appliances and the like. While I've only included four relays, you could expand this to 16 (one for each of the PIA lines) by simply duplicating the buffer/relay circuits 16 times instead of four. If you do, be sure your power supply can accommodate the maximum number of relays that will be energized at any given time.

Eds. note: Due to unforeseeable circumstances (a broken wrist on the author and construction cost factors), the fourth "ROM Hacker" project, a robot arm, will be delayed. We will continue the series as soon as possible. We apologize for any inconvenience this postponement causes.

Also, anyone who has had difficulties ordering parts from Spectrum Projects for the projects in this series should contact Spectrum again. The complete cable and connectors are now available. ■

Address correspondence to J.J. Barbarello, RD#1, Box 241H, Tenent Road, Englishtown, NJ 07726.

MLogo Command	Action Produced
S	Stop. Both motors off.
F	Forward. Both wheels move forward.
B	Back. Both wheels move backward.
R	Right. Left wheel moves forward, right moves backward.
L	Left. Right wheel moves forward, left moves backward.
FR	Forward Right. Right wheel moves forward, left motor off.
FL	Forward Left. Left wheel moves forward, right motor off.
BR	Back Right. Right wheel moves backward, left motor off.
BL	Back Left. Left wheel moves backward, right motor off.

Table 2. Available Movement Commands

```
1Ø CLS:PMODE Ø:PCLEAR 1:CLEAR 8Ø
ØØ:BL$=STRING$(32,128):DIMP$(2Ø)
,PN$(2Ø),P(2Ø),S(2Ø)
2Ø C=&HCØØØ:POKE C+1,Ø:POKE C,25
5:POKE C+1,4:POKE C,Ø:C$(Ø)="S":
C$(1)="BL":C$(2)="FR":C$(3)="X":
C$(4)="BR":C$(5)="B":C$(6)="R":C
$(7)="X":C$(8)="FL":C$(9)="L":C$
(1Ø)="F":GOSUB6ØØ
3Ø CLS:PRINT@4,"MOUSE LOGO COMMA
ND MODE":PRINT@32,"(help=INSTRUC
TIONS, end TO END)":PRINTBL$;
4Ø PRINT@96,"  "C$:PRINT" ":IFF=
1THENF=Ø:IFPN<>ØTHENPRINT@16Ø,"P
ROCEDURES:":FORI=1TOPN:PRINTLEFT
$(PN$(I)+STRING$(7,32),8);:NEXT
5Ø PRINT@128,;:INPUT C$:IF C$="H
ELP"THEN5ØØELSEIFINSTR(C$,"MAKE
")<>ØTHEN15ØELSE IF INSTR(C$,"DO
 ")THEN3ØØELSEIFINSTR(C$,"LIST "
)<>ØTHEN4ØØELSEIFC$="END"THEN 7Ø
Ø
6Ø S=INSTR(C$," "):IFS=Ø THEN 8Ø
ELSES=S-1
7Ø FORI=ØTO1Ø:IFC$(I)=LEFT$(C$,S
)THEN9ØELSENEXT
8Ø C$="COMMAND NOT RECOGNIZED":G
OTO 4Ø
9Ø T=VAL(RIGHT$(C$,LEN(C$)-S)):I
FT<1ORT>99THEN8Ø
1ØØ POKE C,I:FORI=1TO1Ø*T:NEXT:P
OKE C,Ø:GOTO3Ø
14Ø '**MAKE**
15Ø S=INSTR(C$," "):A$=LEFT$(RIG
HT$(C$,LEN(C$)-S),8):C$=""
16Ø CLS:PRINT"make procedure: ";
A$:PRINTBL$;:TMP=PN
17Ø IFPN=ØTHENPN=1:PN$(1)=A$:GOT
O2ØØELSE FOR I=1TOPN:IFPN$(I)<>A
$THENNEXT:PN=PN+1:PN$(PN)=A$:GOT
O2ØØ
18Ø PRINT@128,"PROCEDURE EXISTS.
 REDO (Y/N)?";
19Ø GOSUB8ØØ:IF Q$="N"THENC$="RE
DO "+PN$(I)+" ABORTED":GOTO 3Ø E
LSE PN=I:P$(PN)=""
2ØØ PRINT@128,"  "C$:PRINT" ":PR
INT@16Ø,;
21Ø INPUT C$:IF INSTR(C$,"END")<
>ØTHEN26ØELSES=INSTR(C$," "):IFS
=Ø THEN 23Ø ELSE S=S-1
22Ø FORI=ØTO1Ø:IFC$(I)=LEFT$(C$,
S)THEN24ØELSENEXT
23Ø IFPN=ØTHEN29ØELSEFORI=1TOTMP
:IFC$=PN$(I)THENP$(PN)=P$(PN)+CH
R$(I+127):GOTO2ØØELSENEXT:GOTO29
Ø
24Ø T=VAL(RIGHT$(C$,LEN(C$)-S)):
IFT<1ORT>99THEN29Ø
25Ø P$(PN)=P$(PN)+CHR$(I)+CHR$(T
):GOTO 2ØØ
26Ø P$(PN)=P$(PN)+CHR$(255):PRIN
T@327,"PROCEDURE COMPLETE":IFTMP
>PN THEN PN=TMP
27Ø PRINT@483,"PRESS enter TO CO
NTINUE...";
28Ø C$=INKEY$:IFC$=""THEN28ØELSE
IFASC(C$)=13THEN3ØELSE28Ø
29Ø C$="COMMAND NOT RECOGNIZED":
GOTO2ØØ
3ØØ IFPN=ØTHEN8ØELSEFORI=1TOPN:I
FINSTR(C$,PN$(I))=ØTHENNEXT:GOTO
8Ø
31Ø CLS:PRINT@324,"EXECUTING ";P
N$(I)
32Ø S=1:A$=P$(I):S(1)=I:P=1:FORI
=2TO2Ø:P(I)=Ø:S(I)=Ø:NEXT
33Ø N=ASC(MID$(A$,P,1)):IF N=255
 THEN 36Ø
34Ø IFN<128 THEN T=ASC(MID$(A$,P
+1,1)):POKE C,N:FORX=1TO1Ø*T:NEX
T:POKEC,Ø:P=P+2:GOTO33Ø
35Ø IF N>147 THEN POKE C,Ø:C$="E
RROR":GOTO 3Ø ELSE N=N-127:P(S)=
P+1:S=S+1:P=1:A$=P$(N):S(S)=N:GO
TO 33Ø
36Ø S=S-1:IFS=ØTHEN3Ø ELSE A$=P$
(S(S)):P=P(S):GOTO33Ø
39Ø '**LIST**
4ØØ S=INSTR(C$," "):A$=RIGHT$(C$
,LEN(C$)-S):C$=""
41Ø FORI=1TOPN:IFA$<>PN$(I)THENN
EXT:C$="PROCEDURE NOT AVAILABLE"
:F=1:GOTO3Ø
42Ø CLS:PRINT"procedure: "A$:A$=
P$(I):IFA$=CHR$(255)THENPRINT"/E
ND":GOTO27Ø
43Ø FORI=1TOLEN(A$)-1:IFASC(MID$
(A$,I,1))<127THENPRINTC$(ASC(MID
$(A$,I,1)));:I=I+1:PRINTASC(MID$
(A$,I,1));CHR$(8)"/";:GOTO45Ø
44Ø PRINTPN$(ASC(MID$(A$,I,1))-1
27)"/";
45Ø NEXT:PRINT"END":GOTO27Ø
5ØØ CLS:PRINT"********** H E L
P **********MOVEMENT COMMAND S
YNTAX= XX YY  XX=COMMAND, YY=INC
REMENT (1-99) COMMANDS: F=FORWAR
D, B=BACK                R=RIGHT,
L=LEFT,S=STOP NOTE: YOU CAN COMB

INE B,L,R OR  F,L,R (EX: BL=BACK
 LEFT)"
51Ø PRINT:PRINT"DIRECT COMMANDS:
 DO, MAKE, LIST SYNTAX: DO NAME
DOES PROCEDURE  'NAME' (NAME <=
8 CHARACTERS).  MAKE NAME MAKES
A PROCEDURE     'NAME' (END MAKE
 WITH 'END')    LIST NAME LISTS
THE PROCEDURE.":GOTO27Ø
6ØØ CLS:PRINT@11,"MOUSE LOGO":PR
INTBL$:PRINT@13Ø,"LOAD PROCEDURE
 FILE (Y/N)?..."
61Ø GOSUB 8ØØ:IF Q$="N"THENRETUR
NELSEGOSUB9ØØ
62Ø PRINT@264,"SEARCHING...":OPE
N"I",#-1,"LOGODATA":PRINT@264,"L
OADING...":INPUT#-1,PN
63Ø FORI=1TOPN:INPUT#-1,PN$(I),L
:FORJ=1TOL:INPUT#-1,D:P$(I)=P$(I
)+CHR$(D):NEXTJ,I:CLOSE:GOTO3Ø
7ØØ CLS:PRINT@11,"MOUSE LOGO":PR
INTBL$:PRINT@13Ø,"SAVE PROCEDURE
 FILE (Y/N)...":IFPN=ØTHENQ$="N"
ELSEGOSUB8ØØ
71Ø IFQ$="N"THENQ$="NO SAVE. ":G
OTO73Ø
72Ø GOSUB9ØØ:PRINT@264,"SAVING..
.":OPEN"O",#-1,"LOGODATA":PRINT#
-1,PN:FORI=1TOPN:X=LEN(P$(I)):PR
INT#-1,PN$(I),X:FORJ=1TOX:D=ASC(
MID$(P$(I),J,1)):PRINT#-1,D,:NEX
TJ,I:CLOSE:Q$="DONE. "
73Ø PRINT@13Ø,Q$;"PROGRAM ENDED.
":PRINT" ":PRINT@258,"(ENTER GOT
O 3Ø TO REENTER)":PRINT@36Ø,;:EN
D
8ØØ Q$=INKEY$:IFQ$=""THEN8ØØ
81Ø IFQ$<>"Y"ANDQ$<>"N"THEN8ØØEL
SERETURN
9ØØ PRINT@13Ø,"PREPARE CASSETTE
RECORDER":PRINT@162,"PRESS enter
 WHEN READY..."
91Ø Q$=INKEY$:IFQ$=""THEN91Ø
92Ø IF ASC(Q$)<>13THEN92ØELSERET
URN
```

Program Listing. MLogo

✓ See List of Advertisers on page 97

BY LARRY LANDWEHR

THE AUTO-EXECUTE STORY

How would you like to be able to insert a tape into your cassette player, type CLOADM, and have your Assembly-language program begin executing independently?

There are two obvious ways of doing this. The first way gets into the nitty-gritty of Assembly language. When the Basic interpreter executes a CLOADM instruction it has to leave a return address on the computer's hardware stack so that after the computer has executed a CLOADM instruction it knows where to go to find its next instruction.

If you know where this return address is, you can modify it to point to the program just loaded. After the CLOADM instruction, control does not return to the interpreter. Instead, the program begins executing. This technique can be tricky, and reports are that it is not very reliable.

Another method, the one I prefer, uses the RAM hooks of the Basic operating system. Basic uses quite a bit of random memory for tables, buffers, scratch areas, and so on. It also uses a few locations for storing jump instructions. At certain points within the Basic interpreter control transfers to a RAM address. Usually this address contains a jump instruction to return control back to ROM. But RAM is changeable. If

System Requirements

64/32K RAM
Extended Color Basic
EDTASM+
Editor/Assembler

Program Listing. Graph: Auto Execute Version

```
                        00000 ************************
                        00010 *    PROGRAM:  GRAPH   *
                        00020 *AUTO EXECUTING VERSION *
                        00030 ************************
                        00040
0182                    00050          ORG     386      RAM HOOK
0182 7E   2332          00060          JMP     START    AUTO EXECUTE
                        00070
2328                    00080          ORG     9000
          2800          00090 VIDRAM   EQU     10240    VIDEO RAM
          4000          00100 VND      EQU     16384    VIDEO END
2328                    00110 XVAL     RMB     1        CATESIAN COORDINATES
2329                    00120 YVAL     RMB     1        OF BIT TO SET
232A      01            00130 POWERS   FCB     001      POWERS OF 2
232B      02            00140          FCB     002
232C      04            00150          FCB     004
232D      08            00160          FCB     008
232E      10            00170          FCB     016
232F      20            00180          FCB     032
2330      40            00190          FCB     064
2331      80            00200          FCB     128
                        00210 *CLEAR VIDEO
2332 86   00            00220 START    LDA     #0
2334 8E   2800          00230          LDX     #VIDRAM
2337 A7   80            00240 CLEAR    STA     ,X+
2339 8C   4000          00250          CMPX    #VND
233C 26   F9            00260          BNE     CLEAR
                        00270 *SWITCH IN VIDEO
233E B7   FFC6          00280          STA     65478
2341 B7   FFC8          00290          STA     65480
2344 B7   FFCB          00300          STA     65483
2347 B7   FFCC          00310          STA     65484
234A B7   FFCF          00320          STA     65487
234D B7   FFD0          00330          STA     65488
2350 B7   FFD2          00340          STA     65490
                        00350 *SELECT GRAPHICS MODE G6R
                        00360 *SET VDG REGISTER
2353 B7   FFC0          00370          STA     65472
2356 B7   FFC3          00380          STA     65475
2359 B7   FFC5          00390          STA     65477
                        00400 *SET CONTROL REGISTER
235C B6   FF22          00410          LDA     65314
235F 84   07            00420          ANDA    #7
2361 8A   F0            00430          ORA     #240
2363 B7   FF22          00440          STA     65314
                        00450 *START IN CENTER
2366 86   80            00460          LDA     #128
2368 B7   3410          00470          STA     13328
236B 86   80            00480          LDA     #128
236D B7   2328          00490          STA     XVAL
2370 86   60            00500          LDA     #96
2372 B7   2329          00510          STA     YVAL
                        00520 *MAIN LOOP
                        00530 *SLOW DOWN
2375 86   FF            00540 MAIN     LDA     #255
2377 C6   08            00550          LDB     #8
2379 4A                 00560 SLOW     DECA
237A 81   00            00570          CMPA    #0
237C 26   FB            00580          BNE     SLOW
237E 5A                 00590          DECB
237F C1   00            00600          CMPB    #0
2381 26   F6            00610          BNE     SLOW
                        00620 *READ KEYBOARD
                        00630 *UP ARROW?
2383 C6   F7            00640          LDB     #$F7
2385 F7   FF02          00650          STB     $FF02
2388 B6   FF00          00660          LDA     $FF00
238B 81   F7            00670          CMPA    #$F7
```

You've probably wondered how some games execute automatically. Here's how one technique works.

```
238D 27   26        00680          BEQ     UP
                    00690 *DOWN ARROW?
238F C6   EF        00700          LDB     #$EF
2391 F7   FF02      00710          STB     $FF02
2394 B6   FF00      00720          LDA     $FF00
2397 81   F7        00730          CMPA    #$F7
2399 27   27        00740          BEQ     DOWN
                    00750 *LEFT ARROW?
239B C6   DF        00760          LDB     #$DF
239D F7   FF02      00770          STB     $FF02
23A0 B6   FF00      00780          LDA     $FF00
23A3 81   F7        00790          CMPA    #$F7
23A5 27   28        00800          BEQ     LEFT
                    00810 *RIGHT ARROW?
23A7 C6   BF        00820          LDB     #$BF
23A9 F7   FF02      00830          STB     $FF02
23AC B6   FF00      00840          LDA     $FF00
23AF 81   F7        00850          CMPA    #$F7
23B1 27   29        00860          BEQ     RIGHT
23B3 20   C0        00870          BRA     MAIN
                    00880
                    00890 * RANGE CHECKS
23B5 F6   2329      00900 UP       LDB     YVAL
23B8 C1   00        00910          CMPB    #0
23BA 27   B9        00920          BEQ     MAIN
23BC 5A             00930          DECB
23BD F7   2329      00940          STB     YVAL
23C0 20   27        00950          BRA     PROC
23C2 F6   2329      00960 DOWN     LDB     YVAL
23C5 C1   BF        00970          CMPB    #191
23C7 27   AC        00980          BEQ     MAIN
23C9 5C             00990          INCB
23CA F7   2329      01000          STB     YVAL
23CD 20   1A        01010          BRA     PROC
23CF F6   2328      01020 LEFT     LDB     XVAL
23D2 C1   00        01030          CMPB    #0
23D4 27   9F        01040          BEQ     MAIN
23D6 5A             01050          DECB
23D7 F7   2328      01060          STB     XVAL
23DA 20   0D        01070          BRA     PROC
23DC F6   2328      01080 RIGHT    LDB     XVAL
23DF C1   FF        01090          CMPB    #255
23E1 27   92        01100          BEQ     MAIN
23E3 5C             01110          INCB
23E4 F7   2328      01120          STB     XVAL
23E7 20   00        01130          BRA     PROC
                    01140
                    01150 * BIT MAPPING FUNCTION
23E9 F6   2329      01160 PROC     LDB     YVAL
23EC 86   20        01170          LDA     #32      32 BYTES/ROW
23EE 3D             01180          MUL
23EF 1F   01        01190          TFR     D,X
23F1 4F             01200          CLRA
23F2 F6   2328      01210          LDB     XVAL
23F5 57             01220          ASRB             INT(X/8)
23F6 57             01230          ASRB
23F7 57             01240          ASRB
23F8 C4   1F        01250          ANDB    #$1F     CLEAR TOP 3 BITS OF B
23FA 30   8B        01260          LEAX    D,X
23FC 30   89 2800   01270          LEAX    VIDRAM,X WHICH BYTE
2400 58             01280          ASLB
2401 58             01290          ASLB
2402 58             01300          ASLB
2403 F0   2328      01310          SUBB    XVAL
2406 CB   07        01320          ADDB    #7       WHICH BIT IN BYTE
2408 108E 232A      01330          LDY     #POWERS
240C E6   A5        01340          LDB     B,Y      MASK TO SET BIT IN BYTE
240E EA   84        01350          ORB     ,X
2410 E7   84        01360          STB     ,X       SET PIXEL
2412 16   FF60      01370          LBRA    MAIN
          0000      01380          END
                                                    END
```

you modify this instruction to point to the program you want to run, control goes to that program and it is executed.

Two remaining questions need to be answered: Where are these locations, and how do you modify them? The first answer is easy. Look at any decent memory map of your computer for three byte locations where a jump instruction resides. The location used in this article is 386 to 388. There are others that will work just as well, so you might want to experiment a bit.

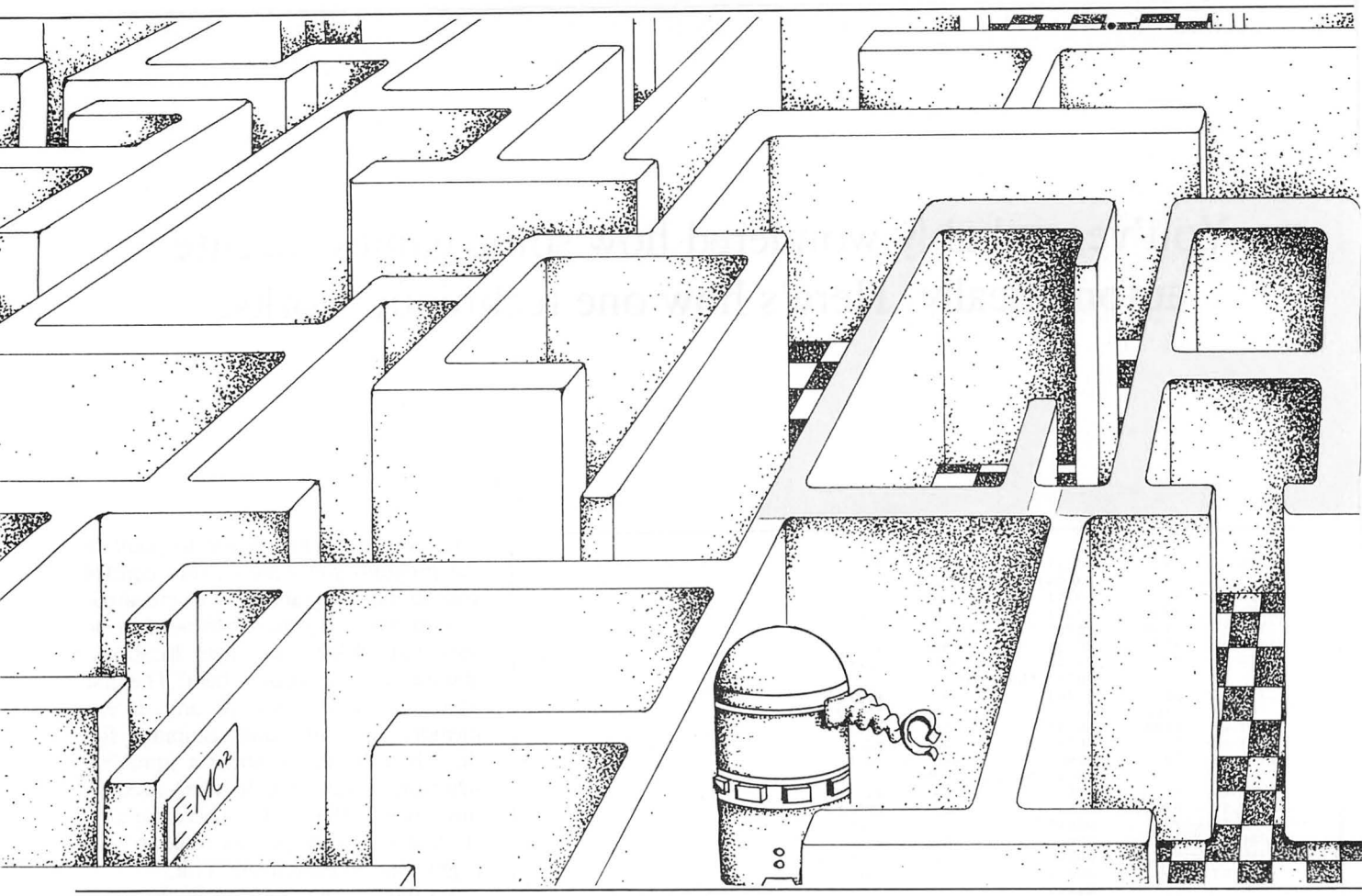
To modify this location is also very simple. You need to use an origin statement and a jump instruction in the Assembly-language program.

Program Listing 1 is a simple program that turns your computer into a toy resembling an etch-a-sketch. The first two lines modify the RAM hook. The rest is the program itself. You know when this program works. The screen changes from white to black and a dot appears in the middle of the screen. To draw lines use the arrow keys.

Try pressing the reset button while the program is running. Notice what happens? Notice the way the keyboard is read. I explain this technique and other interesting things in my book called *In Assembly Language* (available from The Dataman, 420 Ferguson Ave. N., Hamilton, Ontario, Canada L8L 4Y9).

One final caveat. Do not locate your program in RAM areas used by Basic since that will cause the Basic interpreter to crash. This area is located in the lowest 1.5K of RAM. Again, any good memory map will show it. ■

Address correspondence to Larry Landwehr, P.O. Box 27545, Minneapolis, MN 55427.



Last month I introduced you to Carl the Robot and the world he lives in. I also explained Robot Talk, the language Carl understands. This month I'll explain how to use the program ShowCarl, which lets you enter and run programs for Carl. Using ShowCarl you can see Carl move around the screen as he avoids walls and picks up beepers. The solutions to the challenge problems posed last month appear in the sidebar.

If you missed Part I, you can get a good idea of the syntax of Robot Talk by examining the solutions to the challenge problems. To help you I have repeated Tables 1–3, which list all the Robot Talk commands. The numbering of figures and tables is continued from the first part.

Using the Program ShowCarl

To get up and running, run the Basic portion of ShowCarl (Program Listing 1). This loads the machine-language portion of ShowCarl (Program Listing 2). Then you are faced with a picture of Carl in his world.

The system commands available in ShowCarl are listed in Table 4. They are explained below:

● *Entering and running Carl programs:* To write a Carl program just type it in. ShowCarl checks the syntax of each line as you enter it. You can use line numbers between 1 and 999. If you decide to change a line, just enter a new line with the same line number. To delete a line, enter the line number alone. To delete an entire program, enter the command NEW.

While writing a Carl program, you will overwrite the picture of Carl's world. Don't worry. The picture is reconstructed when you run your Carl program.

To run a Carl program, type RUN. To list part or all of your program use the command LIST, just like Basic. You can list all or part of a program. You can even list your program to a printer with the command LLIST. Save or load Carl programs from tape with the commands CSAVE and CLOAD. File names are optional. You can even skip a file on tape with the command SKIPF.

You can stop a running Carl program by pushing the @ key. The @ key substitutes for the break key within ShowCarl. (The break key still works and stops ShowCarl from operating.)

● *Building a world for Carl:* Before

System Requirements

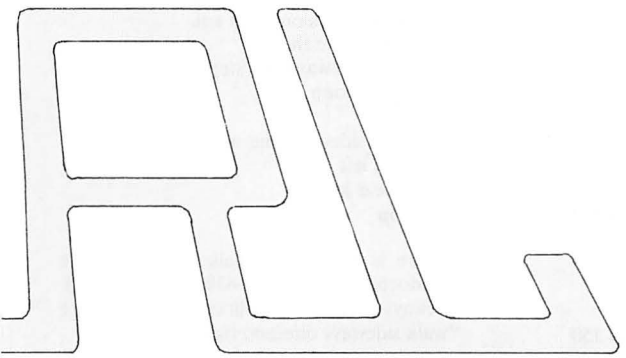
Extended Color Basic
EDTASM+

STEP
TURN LEFT
TURN RIGHT
PICK BEEPER
PUT BEEPER
STOP

Table 1. Carl's Action Commands

GOTO
GOSUB
RETURN
DO n TIMES
NEXT

Table 2. Carl's Unconditional Control Statements



the Robot

PART II

BY DAVID MEREDITH

You know a little about how the Robot Talk language works, now here's the program to use.

Dion LeGros

running a Carl program you will want to create an interesting world for Carl. You can add walls or beepers to Carl's world or fill Carl's bag with beepers by using the world builder. To enter the world-builder mode, type WORLD. You will see a picture of Carl's world and above it a menu of world-building commands.

To turn Carl push T. To move Carl forward push M, then enter the number of steps to move. You can go right through walls, but you won't be permitted to move outside of Carl's world.

To place beepers on the ground, make Carl point at the place you want the beepers, push B, then enter the number of beepers you want at this location. To erase all beepers from a location, place zero beepers there.

To change the number of beepers in Carl's bag, push G then enter the number of beepers you want in the bag.

To add a wall running north-south to Carl's world, make Carl point east or west toward the desired south end of the wall, push W, then enter the length of the wall. Adding an east-west wall is similar. Start by pointing Carl north or south toward the desired east end of the wall, then push W and enter the length of the wall.

To erase an existing wall, make Carl point to it and push E.

To create a new world for Carl with no walls or beepers, push N.

You can save a world on tape by pushing S and following the directions that appear on the screen. You can name your world. You can reload a saved world by pushing L and following the directions on the screen. Should you enter S or L accidentally and want to return to the world builder without any tape operations, push @.

Finally, to return from the world builder to the mode permitting entry and running of programs, just push R.

● *Single stepping and immediate mode:*

FACING NORTH	NOT FACING NORTH
FACING EAST	NOT FACING EAST
FACING SOUTH	NOT FACING SOUTH
FACING WEST	NOT FACING WEST
FRONT IS CLEAR	FRONT IS BLOCKED
RIGHT IS CLEAR	RIGHT IS BLOCKED
LEFT IS CLEAR	LEFT IS BLOCKED
BEEPER HERE	BEEPER NOT HERE
BAG EMPTY	BAG NOT EMPTY

Table 3. Conditions That Can Be Used in the Conditional Command: IF Condition THEN Line Number

NEW
RUN
LIST
LLIST
CSAVE
CLOAD
WORLD
STEP
IMMEDIATE

Table 4. System Commands for ShowCarl

Solutions to Challenge Problems

1. Teach Carl to jump hurdles until he comes to a beeper on the ground (Fig. 8).

The pseudo-code that directs Carl to jump hurdles is very short:

```
While beeper not here do
  If front is blocked then
    Jump
  Else
    Walk
Endwhile
Stop
```

Here is how to jump over a hurdle:

```
Turn left
While right is blocked do walk
Turn right
Walk
Turn right
While front is clear do walk
Turn left
```

Here's the translation to robot talk:

```
10 IF BEEPER HERE THEN 70
20 IF FRONT IS CLEAR THEN 50
30 GOSUB 100
40 GOTO 10
50 WALK
60 GOTO 10
70 STOP
100 TURN LEFT
110 IF RIGHT IS CLEAR THEN 140
120 WALK
130 GOTO 110
140 TURN RIGHT
150 WALK
160 TURN RIGHT
170 IF FRONT IS BLOCKED THEN 200
180 WALK
190 GOTO 170
200 TURN LEFT
210 RETURN
```

The subroutine Jump begins at line 100.

2. Show Carl how to escape from a maze. Figure 9 shows a typical maze. Assume a beeper marks the end of the maze.

There is a classic strategy for escaping a maze: Put your right hand on a wall, then go right if you can, go left if you must:

```
10 IF BEEPER HERE THEN 500
20 IF FRONT IS BLOCKED THEN 50
30 WALK
40 GOTO 10
50 TURN LEFT
100 IF BEEPER HERE THEN 500
110 IF RIGHT IS BLOCKED THEN 150
120 TURN RIGHT
130 WALK
140 GOTO 100
150 IF FRONT IS BLOCKED THEN 180
160 WALK
170 GOTO 100
180 TURN LEFT
190 GOTO 150
500 STOP
```

3. A beeper was left somewhere in Carl's world. Find it without using the north or east boundary walls.

First go to the southwest corner, then follow a diagonal zig-zag pattern looking for the beepers. Figure 10 shows how the zig-zag goes.

```
10 IF FACING WEST THEN 40
20 TURN LEFT
30 GOTO 10
40 GOSUB 900
50 TURN LEFT
60 GOSUB 900
70 TURN LEFT
80 GOSUB 700
90 GOSUB 400
100 GOSUB 500
110 IF FRONT IS CLEAR THEN 100
120 TURN RIGHT
130 GOSUB 700
140 GOSUB 400
150 GOSUB 600
160 IF FRONT IS CLEAR THEN 150
170 GOTO 70
400 WALK
410 TURN LEFT
420 TURN LEFT
430 RETURN
500 GOSUB 700
510 TURN RIGHT
520 GOSUB 700
530 TURN LEFT
540 RETURN
600 GOSUB 700
610 TURN LEFT
620 GOSUB 700
630 TURN RIGHT
640 RETURN
700 IF BEEPER HERE THEN 730
710 WALK
720 RETURN
730 PICK BEEPER
740 STOP
900 IF FRONT IS BLOCKED THEN 930
910 WALK
920 GOTO 900
930 RETURN
```

The subroutine at 900 takes Carl to a wall. Lines 10–60 place Carl in the southwest corner facing south. The subroutine at 700 checks for a beeper and stops Carl if one is found, otherwise causes Carl to take a step. The subroutine at 400 causes Carl to walk one step and turn around. The subroutine at 500, invoked only when Carl is facing west, makes Carl zig-zag one step west then one step north and end up facing west. The complementary subroutine at 600, called only with Carl facing south, has Carl zig-zag one step south then one step east and end up facing south. Lines 70–170 cause Carl to follow diagonal paths, first southwest then northeast, until a beeper is found.

4. Tell Carl how to go in a counter-clockwise spiral, dropping beepers until he confronts a wall or his bag is empty. The letters in Fig. 11 show how he should go.

Let's begin with a pseudocode solution to the problem. First define a procedure step (lines 500–550 below):

```
If Bag is empty then halt
else put beeper
  if front is blocked then halt
  else walk
    return
```

Now use step to walk a spiral:

```
Step
Turn left
Loop
  Step
  Walk sideways one step left
  If beeper here then
    Walk sideways one step right
    Repeat loop
  Else
    Walk sideways one step right
    Turn left
    Repeat loop
End loop
```

Here is the Robot Talk version of the pseudocode. Lines 400–430 execute "walk sideways one step left;" lines 300–330 execute "walk sideways one step right."

```
10 GOSUB 500
20 TURN LEFT
30 GOSUB 500
40 GOSUB 400
50 IF BEEPER HERE THEN 100
60 GOSUB 300
70 TURN LEFT
80 GOTO 30
100 GOSUB 300
110 GOTO 30
300 TURN RIGHT
310 WALK
320 TURN LEFT
330 RETURN
400 TURN LEFT
410 WALK
420 TURN RIGHT
430 RETURN
500 IF BAG EMPTY THEN 550
510 PUT BEEPER
520 IF FRONT IS BLOCKED THEN 550
530 WALK
540 RETURN
550 STOP
```

5. Teach Carl to multiply. Write a Robot Talk program that makes Carl move a distance equal to the product of the distances to the next two beepers.

I'll use a recursive definition of multiplication: If $x=0$ or $y=0$ then $xy=0$ else if $x=1$ then $xy=y$ else if $y=1$ then $xy=x$ else $xy=x(y-1)+x$.

Start with a pseudocode version of multiplication, using a recursive procedure Multiply that moves Carl from his current location a distance equal to the product of the distances to the next two beepers. Carl will pick up the first of the beepers whose distances he is multiplying, and he will drop a beeper where he stops.

Multiply assumes that the distance to the next beeper is at least one. If Carl is standing on a beeper, then he simply stops without calling Multiply, since the distance to the first beeper is zero.

I'll also use a procedure Add as explained in Part I that moves Carl a distance equal to the sum of the distances to the next two beepers, picking up the second beeper and dropping a beeper where he stops:

```
If beeper here then stop
Multiply
Stop
```

```
Procedure Multiply

Walk
If beeper here then
   Pick beeper
   Walk to next beeper
   Put beeper
   Return
Else
   Walk to first beeper
   Carry it back one step
   Walk back to starting point
   Multiply
   Walk back to starting point
   Add
   Return
```

Now for Robot Talk:

```
10 IF BEEPER HERE THEN 30
20 GOSUB 100
30 STOP
100 WALK
110 IF BEEPER NOT HERE THEN 200
120 PICK BEEPER
130 IF BEEPER HERE THEN 160
140 WALK
150 GOTO 130
160 PUT BEEPER
170 RETURN
200 IF BEEPER HERE THEN 230
210 WALK
220 GOTO 200
230 PICK BEEPER
240 GOSUB 500
250 WALK
260 PUT BEEPER
270 IF FRONT IS BLOCKED THEN 300
280 WALK
290 GOTO 270
300 GOSUB 500
310 GOSUB 100
320 GOSUB 500
330 IF FRONT IS BLOCKED THEN 360
340 WALK
350 GOTO 330
360 GOSUB 500
370 GOSUB 700
380 RETURN
500 TURN LEFT
510 TURN LEFT
520 RETURN
700 GOSUB 800
710 PUT BEEPER
720 RETURN
800 IF BEEPER HERE THEN 900
810 WALK
820 GOSUB 800
830 WALK
840 RETURN
900 PICK BEEPER
910 IF BEEPER HERE THEN 980
920 PUT BEEPER
930 WALK
940 IF BEEPER HERE THEN 970
950 WALK
960 GOTO 940
970 PICK BEEPER
980 RETURN
```

Lines 100–380 contain the routine Multiply; lines 700–980 contain Add. The subroutine at 500 turns Carl 180 degrees.

ShowCarl has some special features not available in Basic. You can single step through a Carl program by entering the command STEP. Each program line is displayed on the screen above Carl's world before the line is executed. To execute the displayed line, push the space bar. To stop single stepping a program, push the @ key. Single stepping aids in program debugging.

If you type IMMEDIATE, you see a picture of Carl's world below a special menu of one-key commands. These commands correspond to Carl's action commands. The immediate mode is a good way of introducing yourself to Carl's world. You can see the result of commands as they are executed one at a time. You can even see the error messages caused by unexecutable commands.

I've used the immediate mode with children as young as first grade. I placed Carl outside a maze, and inside the maze I placed some beepers. Then I challenged the children to move Carl into the maze and pick up the beepers. With a little practice they could do the job easily. ■

Write to David Meredith at Department of Mathematics, San Francisco State University, San Francisco, CA 94132.

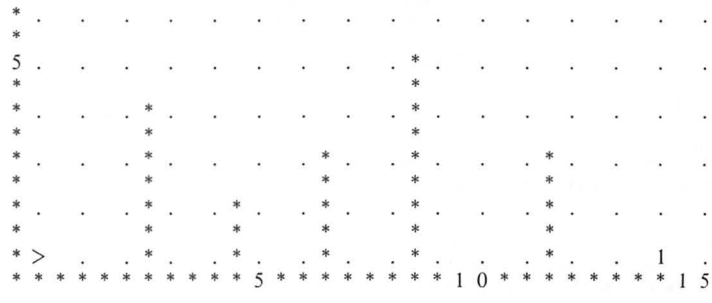


Fig. 8. Hurdles for Carl to Jump

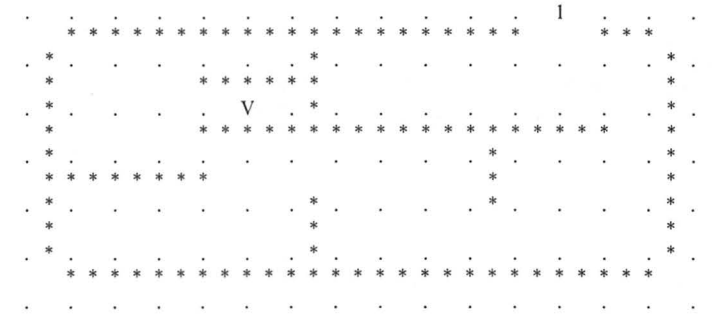


Fig. 9. Can Carl Escape the Maze?

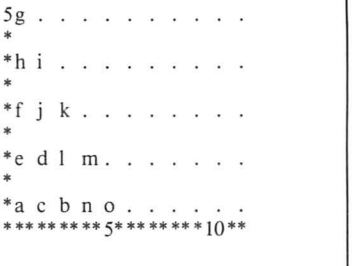


Fig. 10. The Solution to Challenge Problem 3

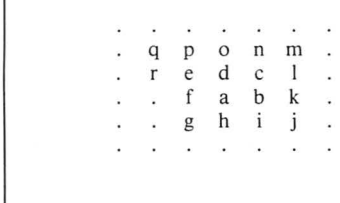


Fig. 11. A Spiral for Carl to Follow

Program Listing 1. ShowCarl, Basic Portion

```
9Ø DATA WALK,WALK,TURNLEFT,TURN
LEFT,TURNRIGHT,TURN RIGHT,NEXT,N
EXT,RETURN,RETURN,PICKBEEPER,PIC
K BEEPER,PUTBEEPER,PUT BEEPER,ST
OP,STOP,DO,DO,GOTO,GOTO,GOSUB,GO
SUB
1ØØ DATA IFFRONTISCLEARTHEN,IF F
RONT IS CLEAR THEN,IFLEFTISCLEAR
THEN,IF LEFT IS CLEAR THEN,IFRIG
HTISCLEARTHEN,IF RIGHT IS CLEAR
THEN,IFFRONTISBLOCKEDTHEN,IF FRO
NT IS BLOCKED THEN,IFLEFTISBLOCK
EDTHEN,IF LEFT IS BLOCKED THEN,I
FRIGHTISBLOCKEDTHEN
11Ø DATA IF RIGHT IS BLOCKED THE
N,IFBEEPERHERETHEN,IF BEEPER HER
E THEN,IFBEEPERNOTHERETHEN,IF BE
EPER NOT HERE THEN,IFBAGEMPTYTHE
N,IF BAG EMPTY THEN,IFBAGNOTEMPT
YTHEN,IF BAG NOT EMPTY THEN,IFFA
CINGNORTHTHEN,IF FACING NORTH TH
EN
12Ø DATA IFFACINGSOUTHTHEN,IF FA
CING SOUTH THEN,IFFACINGEASTTHEN
,IF FACING EAST THEN,IFFACINGWES
TTHEN,IF FACING WEST THEN,IFNOTF
ACINGNORTHTHEN,IF NOT FACING NOR
TH THEN,IFNOTFACINGSOUTHTHEN,IF
NOT FACING SOUTH THEN,IFNOTFACIN
GEASTTHEN
13Ø DATA IF NOT FACING EAST THEN
,IFNOTFACINGWESTTHEN,IF NOT FACI
NG WEST THEN
14Ø DATA FRONT IS BLOCKED AT,NO
BEEPER HERE AT,BAG EMPTY AT, NEX
T WITHOUT DO AT,RETURN WITHOUT G
OSUB AT,DO-NEXT STACK OVERFLOWS
AT,GOSUB STACK OVERFLOWS AT
15Ø DATA UNDEFINED LINE # IN, ST
OP AT, PROGRAM ENDS WITHOUT STOP
 AFTER, TOO MANY BEEPER LOCATION
S AT
16Ø CLEAR5ØØ,&H6FFF:GOTO 223Ø
17Ø REM
18Ø REM SUBROUTINE TO POKE 2-BYT
E UNSIGNED INTEGER N INTO LO,LO+
1
19Ø REM
2ØØ NX=INT(N/256):POKELO,NX:POKE
LO+1,N-256*NX:RETURN
21Ø REM
22Ø REM WAIT SUBROUTINE
23Ø REM
24Ø PRINT"PRESS ANY GREY KEY TO
CONTINUE"
25Ø IFINKEY$<>""THEN RETURNELSE2
5Ø
26Ø REM
27Ø REM INPUT-A-LINE SUBROUTINE
28Ø REM BLANK A LINE ON THE SCRE
EN, INPUT L$, AND ELIMINATE BLAN
K FROM L$.  AT END L=LEN(L$)
29Ø REM
3ØØ PRINT" ":PRINTBS$;:LINEINPUT
L$
31Ø I=1:L=LEN(L$)
32Ø IFI>L THENRETURNELSEJ=INSTR(
I,L$," "):IFJ=ØTHENRETURNELSEL$=
LEFT$(L$,J-1)+RIGHT$(L$,L-J):L=L
-1:I=J:GOTO32Ø
33Ø REM
34Ø REM PRINT CARL AT LOCATION I
N X,Y (CORRESPONDS TO CX,CY IN A
SSEMBLER LISTING).  MOVE WINDOW
INTO WORLD IF CARL OUTSIDE CURRE
NT WINDOW.  BX,BY IS LOWER LEFT
CORNER OF WINDOW.
35Ø REM
36Ø REM
37Ø XX=FNP(X):YY=FNP(Y):IFXX>=BX
 ANDXX<=BX+31ANDYY>=BY ANDYY<=BY
+13THEN 41Ø:REM IF CARL INSIDE C
URRENT WINDOW
38Ø IF XX<32 THENBX=ØELSEBX=XX-1
5:REM RESET LEFT SIDE OF WINDOW
39Ø IF YY<14THENBY=ØELSEBY=YY-7:
REM RESET BOTTOM OF WINDOW
4ØØ GOSUB 67Ø:RETURN
41Ø LO=1Ø24+XX-BX+32*(15-YY+BY):
POKELO,PEEK(CA):RETURN:REM DRAW
CARL IN WORLD
42Ø REM
43Ø REM PRINT N WITH RIGHT END A
T AT SCREEN LOCATION LO
44Ø REM
45Ø N$=STR$(N):L=LEN(N$):PRINT@L
O-L+2,RIGHT$(N$,L-1);:RETURN
46Ø REM
47Ø 'PRINT N BEEPERS AT XX,YY
48Ø REM
49Ø IFXX<BX ORXX>BX+31ORYY<BY OR
YY>BY+13THENRETURN:REM IF LOCATI
ON NOT IN WINDOW
5ØØ LO=XX-BX+32*(15-YY+BY)
51Ø CH=PEEK(LO+1Ø23):IFCH<>15ØTH
ENCH=96
52Ø IFN=ØTHENPOKELO+1Ø24,11Ø:POK
ELO+1Ø23,CH:RETURN
53Ø IFN<=9THENPOKELO+1Ø23,CH
54Ø GOSUB43Ø:RETURN
55Ø PRINT@94,"  ";
56Ø REM
57Ø REM PRINT CONTENTS OF BAG IN
 UPPER RIGHT CORNER OF WORLD
58Ø REM
59Ø N=FNP(BG):LO=95:GOSUB43Ø:RET
URN
6ØØ REM
61Ø REM LIST INSTRUCTION WITH AD
DRESS IN PP.  LIST ON SCREEN OR
PRINTER DEPENDING ON PS
62Ø REM
63Ø P=FNP(PP):Q=FNP(P):IFQ=65535
THENRETURNELSER=PEEK(P+2):IFR<8T
HENPRINT#PS,Q;Cl$(R):RETURNELSES
=FNP(P+3):IFR=8THENPRINT#PS,Q;"D
O";S;"TIMES":RETURNELSEPRINT#PS,
Q;Cl$(R);S:RETURN
64Ø REM
65Ø REM DISPLAY WORLD
66Ø REM
67Ø FORI=64TO384STEP64:PRINT@I,W
1$;:NEXT:PRINT@448,W2$;:POKE1535
,96
68Ø TX=BX+31:TY=BY+13:I=NS:REM B
X,BY LOWER LEFT CORNER OF WINDOW
 INTO WORLD TO BE DISPLAYED.  TX
,TY TOP RIGHT CORNER.  I=ADDRESS
 OF NORTH-SOUTH WALL BUFFER
69Ø XX=FNP(X):YY=FNP(Y):REM CHAN
GE WINDOW IF NECESSARY TO GET CA
RL IN WORLD
7ØØ IFXX>=BX ANDXX<=TX ANDYY>=BY
 ANDYY<=TY THEN 74Ø
71Ø IFXX<32THENBX=ØELSEBX=XX-15
72Ø IFYY<14THENBY=ØELSEBY=YY-7
73Ø TX=BX+31:TY=BY+13
74Ø P=FNP(I):IFP=65535THEN79Ø:RE
M DRAW NS WALLS
75Ø IFBX>P ORP>TX THENI=I+6:GOTO
74Ø
76Ø Q=FNP(I+2):IFQ>TY THENI=I+6:
GOTO74ØELSEIFQ<BY THENQ=BY
77Ø R=FNP(I+4):IFR<BY THENI=I+6:
GOTO74ØELSEIFR>TY THENR=TY
78Ø FORJ=P-BX+32*(15-Q+BY)TOP-BX
+32*(15-R+BY)STEP-32:PRINT@J,CHR
$(15Ø);:NEXT:I=I+6:GOTO74Ø
79Ø I=EW:REM DRAW EW WALLS.  I =
 ADDRESS OF EAST-WEST WALL BUFFE
R
8ØØ P=FNP(I):IFP=65535THEN86Ø
81Ø R=FNP(I+4):IFR<BY ORR>TY THE
NI=I+6:GOTO8ØØ
82Ø IFP>TX THENI=I+6:GOTO8ØØELSE
IFP<BX THENP=BX
83Ø Q=FNP(I+2):IFQ<BX THENI=I+6:
GOTO8ØØELSEIFQ>TX-1 THENQ=TX-1
84Ø FORJ=P-BX+32*(15-R+BY)TOQ-BX
+32*(15-R+BY):PRINT@J,CHR$(15Ø);
:NEXTJ:IFQ=TX-1THENPOKE1Ø55+32*(
15-R+BY),15Ø
85Ø I=I+6:GOTO8ØØ
86Ø I=BE:REM PRINT BEEPERS IN WO
RLD.  I=ADDRESS OF BEEPER LIST
87Ø N=PEEK(I):IFN=255THEN91Ø
88Ø Q=FNP(I+1):IFQ<BX ORQ>TX THE
NI=I+5:GOTO87Ø
89Ø R=FNP(I+3):IFR<BY ORR>TY THE
NI=I+5:GOTO87Ø
9ØØ LO=Q-BX+32*(15-R+BY):GOSUB43
Ø:I=I+5:GOTO87Ø
91Ø B=INT(BX/1Ø):LO=489-BX+1Ø*B:
N=5*(B+1):REM PRINT COORDINATES
ALONG LEFT AND BOTTOM OF WINDOW
92Ø GOSUB43Ø:LO=LO+1Ø:N=N+5:IFLO
<511THEN92Ø
93Ø B=INT(BY/1Ø):L1=192+32*(BY-1
Ø*B):N=5*(B+1)
94Ø LO=L1+LEN(STR$(N))-2:GOSUB43
Ø:L1=L1-32Ø:N=N+5:IFL1>63THEN 94
Ø
95Ø GOSUB55Ø:REM PRINT BAG
96Ø GOSUB 36Ø:REM PRINT CARL
97Ø RETURN: REM FROM DISPLAYING
WORLD
98Ø REM
99Ø REM RUN OR SINGLE STEP A PRO
GRAM
1ØØØ REM
1Ø1Ø LO=PP:N=PR:GOSUB2ØØ:LO=SP:N
=SS-2:GOSUB2ØØ:LO=DP:N=DS-4:GOSU
B2ØØ:REM INITITALIZE PROGRAM POI
NTER, SUBROUTINE POINTER, DO-NEX
T POINTER
1Ø2Ø PRINT@Ø," ":PRINT@32," "
1Ø3Ø GOSUB67Ø:REM DISPLAY WORLD
1Ø4Ø XX=FNP(X):YY=FNP(Y):N=USR1(
Ø):REM GET INFO RE BEEPERS HERE
BEFORE EXECUTING INSTRUCTION
1Ø5Ø IF N<>ØTHENN=PEEK(N):REM N=
# OF BEEPERS
1Ø6Ø IFINKEY$="@"THENPRINT@Ø,"BR
EAK IN ";FNP(FNP(PP)):PRINT"OK":
RETURN ELSEIFSI=ØTHEN1Ø8ØELSEPRI
NTØØ,"";:GOSUB63Ø:REM @ USED AS
BREAK KEY OR SINGLE STEPPING.  S
I<>Ø IF SINGLE STEPPING.
1Ø7Ø I$=INKEY$:IFI$="@"THENPRINT
@Ø,"BREAK IN ";FNP(FNP(PP)):PRIN
T"OK":RETURNELSEIFI$=""THEN1Ø7Ø:
REM WHILE SINGLE STEPPING, BREAK
 WITH @, CONTINUE WITH ANY OTHER
 KEY
1Ø8Ø EXEC RN:REM DO THE NEXT INS
TRUCTION
1Ø9Ø P=PEEK(RV):IFP=ØTHEN1Ø6ØELS
EIFP=1THENGOSUB47Ø:GOSUB36Ø:GOTO
1Ø4ØELSEIFP=2THENGOSUB55Ø:GOTO1Ø
4ØELSEIFP=3THENPOKE1Ø24+FNP(X)-B
X+32*(15-FNP(Y)+BY),PEEK(CA):GOT
O1Ø6ØELSEPRINT@Ø,EM$(P-4);FNP(LI
):RETURN
11ØØ REM P=Ø IS NO CHANGE, P=1 I
S CARL MOVED, P=2 IS BEEPER BAG
CHANGED, P=3 IS CARL TURNED, P=4
 IS ERROR OR STOP
111Ø REM
112Ø REM LIST PROGRAM LINES BETW
EEN N1 AND N2 TO SCREEN OR PRINT
ER.  PS=DEVICE #
113Ø REM
114Ø IF PS=Ø THEN L=L-4 ELSE L=L
-5
115Ø L$=RIGHT$(L$,L):N1=VAL(L$):
IFN1<ØTHENN2=-N1:N1=Ø:GOTO117Ø
116Ø IFL=ØTHENN1=Ø:N2=999:GOTO11
7ØELSEJ=INSTR(L$,"-"):N2=VAL(RIG
HT$(L$,L-J)):IFN2=ØTHENN2=999
117Ø N=N1:LO=LI:GOSUB2ØØ:N=USR2(
Ø):REM FIND FIRST LINE # >= N1
118Ø IF FNP(N)>N2 THEN RETURN EL
SE LO=PP:GOSUB2ØØ:GOSUB63Ø:N=N+5
:GOTO118Ø
119Ø REM
12ØØ REM IMMEDIATE MODE
121Ø REM
122Ø GOSUB67Ø
123Ø PRINT@Ø,"WALK, TURN rIGHT,
TURN lEFT,     PuTBEEPER, PiCKBE
EPER, sTOP"
124Ø I$=INKEY$:IFI$=""THEN124Ø
125Ø N=IC-2:LO=PP:GOSUB 2ØØ:REM
FOOL MACHINE LANGUAGE ROUTINE IN
TO EXECUTING IMMEDIATE COMMAND B
Y MAKING IC THE THIRD BYTE FOLLO
WING THE LOCATION POINTED AT BY
```

Listing 1 continued

```
PP AND PUTTING COMMAND CODE INTO
 IC.
126Ø IF I$="W"THENPOKEIC,ØELSEIF
I$="R"THENPOKEIC,2ELSEIFI$="L"TH
ENPOKEIC,1ELSEIFI$="U"THENPOKEIC
,6ELSEIFI$="I"THENPOKEIC,5ELSEIF
I$="S"THENPOKEIC,7ELSE124Ø
127Ø XX=FNP(X):YY=FNP(Y):N=USR1(
Ø):IFN<>ØTHENN=PEEK(N)
128Ø EXEC RN:REM EXECUTE THE INS
TRUCTION THEN CHECK RV FOR STATU
S
129Ø P=PEEK(RV):IFP=ØTHEN124ØELS
EIFP=1THENGOSUB47Ø:GOSUB36Ø:GOTO
124ØELSEIFP=2THENGOSUB55Ø:GOTO12
4ØELSEIFP=3THENPOKE1Ø24+FNP(X)-B
X+32*(15-FNP(Y)+BY),PEEK(CA):GOT
O124ØELSEIFP=4THENPRINT@Ø,"FRONT
 IS BLOCKED":GOSUB24Ø:GOTO123Ø
13ØØ IFP=5THENPRINT@Ø,"NO BEEPER
S HERE":GOSUB24Ø:GOTO123ØELSEIFP
=6THENPRINT@Ø,"BAG IS EMPTY":GOS
UB24Ø:GOTO123ØELSEIFP=12THENPRIN
T@Ø,"IMMEDIATE MODE STOPPED":PRI
NT@32,"";:RETURN
131Ø REM
132Ø REM BUILD A WORLD FOR CARL
133Ø REM
134Ø CLS:GOSUB67Ø
135Ø PRINT@Ø,"mOVE, tURN, wALL,
eRASEWALL, nEWbEEPERS, BAg, sAVE
, lOAD, rETURN";
136Ø I$=INKEY$:IFI$=""THEN136Ø
137Ø ON INSTR("MTWEBGSLRN",I$)+1
GOTO136Ø,139Ø,145Ø,147Ø,174Ø,178
Ø,189Ø,192Ø,197Ø,2Ø2Ø,2Ø4Ø
138Ø REM MOVE CARL
139Ø PRINT@Ø,"HOW FAR TO MOVE":G
OSUB3ØØ:N=VAL(L$):IFN<1THEN135Ø
14ØØ XX=FNP(X):YY=FNP(Y):NN=USR1
(Ø):IFPEEK(DX+1)=2THENXQ=XX+N*2:
YQ=YY ELSEP=FNP(DY):IFP=2THENYQ=
YY+2*N:XQ=XX ELSEIFP=ØTHENXQ=XX-
2*N:YQ=YY ELSEYQ=YY-2*N:XQ=XX
141Ø IF XQ<Ø OR XQ>3ØØØØ OR YQ<Ø
ORYQ>3ØØØØTHENPRINT@Ø,"YOU TRIED
 TO LEAVE WORLD":GOSUB24Ø:GOTO13
5Ø
142Ø IF NN=Ø THENN=NN ELSE N=PEE
K(NN)
143Ø GOSUB 47Ø:LO=X:N=XQ:GOSUB2Ø
Ø:LO=Y:N=YQ:GOSUB2ØØ:GOSUB36Ø:GO
TO135Ø
144Ø REM TURN CARL
145Ø EXEC TL:GOSUB36Ø:GOTO136Ø
146Ø REM BUILD A WALL
147Ø PRINT@Ø,"HOW LONG A WALL":G
OSUB3ØØ:N=VAL(L$):IFN<1THEN135Ø
148Ø XX=FNP(X):YY=FNP(Y)
149Ø IFPEEK(DX+1)=ØTHEN 162Ø
15ØØ I=NS:IFPEEK(DX)=ØTHENXX=XX+
1ELSEXX=XX-1:REM BUILD A NORTH S
OUTH WALL FROM XX,YY TO XX,Y2.
151Ø Y2=YY+2*N-2:IFY2>3ØØØØTHENY
2=3ØØØØ
152Ø IF PEEK(I)=255 THEN 159Ø:RE
M FIRST SEE IF WALL AMA␣GAMATES
WITH ANOTHER
153Ø IFXX<>FNP(I)ORY2<FNP(I+2)OR
YY>FNP(I+4)THENI=I+6:GOTO152Ø
154Ø REM JOIN THIS WALL TO NEW W
ALL THEN ERASE THIS WALL
155Ø YZ=FNP(I+2):IFYZ<YY THENYY=
YZ
156Ø YZ=FNP(I+4):IFYZ>Y2 THENY2=
YZ
157Ø J=I
158Ø FORK=ØTO5:POKEJ+K,PEEK(J+K+
6):NEXT:IFPEEK(J)=255THEN152ØELS
EJ=J+6:GOTO158Ø
159Ø IF(I-NS)/6>=MW THENPRINT@Ø,
"TOO MANY NORTH-SOUTH WALLS":GOS
UB24Ø:GOTO135Ø
16ØØ LO=I:N=XX:GOSUB2ØØ:LO=I+2:N
=YY:GOSUB2ØØ:LO=I+4:N=Y2:GOSUB2Ø
Ø:LO=I+6:N=&HFFFF:GOSUB2ØØ:GOTO1
32Ø:REM PUT NEW WALL INTO LIST O
F NORTH-SOUTH WALLS
161Ø REM ADD A NEW EAST-WEST WAL
L FROM XX,YY TO X2,YY
162Ø I=EW:IFPEEK(DY)=ØTHENYY=YY+
1ELSEYY=YY-1
163Ø X2=XX+2*N-2:IFX2>29999THENX
2=29999
164Ø IFPEEK(I)=255THEN171Ø:REM F
IRST SEE IF THIS WALL RUNS INTO
ANOTHER
165Ø IFXX>FNP(I+2)ORX2<FNP(I)ORY
Y<>FNP(I+4)THENI=I+6:GOTO164Ø
166Ø REM JOIN THIS WALL TO NEW W
ALL THEN ERASE THIS WALL
167Ø XZ=FNP(I):IF XZ<XX THENXX=X
Z
168Ø XZ=FNP(I+2):IFXZ>X2 THENX2=
XZ
169Ø J=I
17ØØ FORK=ØTO5:POKEJ+K,PEEK(J+K+
6):NEXT:IFPEEK(J)=255THEN164ØELS
EJ=J+6:GOTO17ØØ
171Ø IF(I-EW)/6>=MW THENPRINT@Ø,
"TOO MANY EAST-WEST WALLS":GOSUB
24Ø:GOTO135Ø
172Ø LO=I:N=XX:GOSUB2ØØ:LO=I+2:N
=X2:GOSUB2ØØ:LO=I+4:N=YY:GOSUB2Ø
Ø:LO=I+6:N=&HFFFF:GOSUB2ØØ:GOTO1
32Ø:REM PUT NEW WALL INTO LIST O
F EAST-WEST WALLS
173Ø REM ERASE A WALL
174Ø U=USRØ(Ø):IFU=ØTHENPRINT@Ø,
"NO WALL TO ERASE":GOSUB24Ø:GOTO
135Ø
175Ø IF U=NS OR U=NS+6 OR U=EW O
R U=EW+6 THEN PRINT@Ø,"THAT WALL
 IS PERMANENT":GOSUB24Ø:GOTO135Ø
:REM CANT ERASE BORDER WALL
176Ø IF PEEK(U+6)=255 THEN POKEU
,255:GOTO132ØELSEFORJ=ØTO5:POKEU
+J,PEEK(U+J+6):NEXT:U=U+6:GOTO17
6Ø:REM ERASE WALL RECORD
177Ø REM PUT DOWN BEEPERS WHERE
CARL POINTS
178Ø PRINT@Ø,"HOW MANY BEEPERS G
O HERE?":GOSUB3ØØ:NU=VAL(L$):IFN
U<ØTHEN135ØELSEIFNU>99THENPRINT@
Ø,"TOO MANY BEEPERS!!!":GOSUB24Ø
:GOTO135Ø
179Ø XX=FNP(X):YY=FNP(Y):IFPEEK(
DX+1)=2 THEN N=XX+2:LO=X ELSE IF
 PEEK(DY+1)=2 THEN N=YY+2:LO=Y E
LSE IF PEEK(DX)=ØTHENN=YY-2:LO=Y
 ELSEN=XX-2:LO=X
18ØØ IF LO=X THEN XQ=N:YQ=YY ELS
E XQ=XX:YQ=N:REM XQ,YQ=LOCATION
OF BEEPER
181Ø IFN<Ø OR N>3ØØØØ THEN PRINT
@Ø,"YOU CAN'T PUT BEEPERS THERE"
:GOSUB24Ø:GOTO135Ø
182Ø GOSUB2ØØ:LL=USR1(Ø):LO=X:N=
XX:GOSUB2ØØ:LO=Y:N=YY:GOSUB2ØØ:X
X=XQ:YY=YQ:N=NU:GOSUB47Ø:REM PUT
 LOC OF BEEPER INTO X,Y; GET LL=
ADDR OF RECORD OF BEEPERS THERE(
Ø IF NONE); RESTORE X,Y TO LOC O
F CARL; PRINT NEW # BEEPERS NU A
T XX,YY=LOC OF BEEPERS
183Ø IFLL<>Ø THENIFNU<>Ø THEN PO
KE LL,NU:GOTO 135ØELSE187Ø:REM P
UT NEW # BEEPERS INTO EXISTING B
EEPER RECORD
184Ø IFNU=Ø THEN 135Ø ELSEI=BE:R
EM IF NO BEEPERS WHERE NONE WERE
 DO NOTHING ELSE ERASE OLD BEEPE
R RECORD TO PUT NO BEEPERS WHERE
 SOME WHERE
185Ø IF PEEK(I)<>255 THEN I=I+5:
GOTO185Ø:REM PUTTING A NEW RECOR
D INTO BE
186Ø IF(I-BE)/5>=MB THENPRINT@Ø,
"TOO MANY BEEPER LOCATIONS":GOSU
B24Ø:GOTO135ØELSEPOKEI,N:LO=I+1:
N=XX:GOSUB2ØØ:LO=I+3:N=YY:GOSUB2
ØØ:POKEI+5,255:GOTO135Ø
187Ø IFPEEK(LL+5)=255THENPOKELL,
255:GOTO135ØELSEFORJ=ØTO4:POKELL
+J,PEEK(LL+J+5):NEXTJ:LL=LL+5:GO
TO187Ø:REM IN CASE RECORD IN BE
SET TO Ø, ELIMINATE RECORD
188Ø REM PUT BEEPERS IN BAG
189Ø PRINT@Ø,"HOW MANY IN THE BA
G":GOSUB3ØØ:N=VAL(L$):IFN<ØTHEN1
35ØELSEIFN>999THENPRINT@Ø,"TOO M
ANY FOR THE BAG!!":GOSUB24Ø:GOTO
135Ø
19ØØ LO=BG:GOSUB2ØØ:GOSUB55Ø:GOTO
 135Ø
191Ø REM SAVE A WORLD ON TAPE
192Ø PRINT@Ø,"PREPARE TAPE RECOR
DER AND ENTER NAME OF WORLD--@ T
O RETURN":GOSUB3ØØ:N$=L$:IFLEN(N
$)>8THENN$=LEFT$(N$,8)
193Ø IFL$="@"THEN134Ø
194Ø CSAVEM N$,NS,CA,NS
195Ø PRINT"WORLD ";N$;" IS SAVED
":GOSUB24Ø:GOTO134Ø
196Ø REM LOAD A WORLD FROM TAPE
197Ø PRINT@Ø,"PREPARE TAPE RECOR
DER AND ENTER NAME OF WORLD--@ T
O RETURN":GOSUB3ØØ:N$=L$:IFLEN(N
$)>8THENN$=LEFT$(N$,8)
198Ø IF L$="@" THEN GOTO134Ø
199Ø CLOADM N$
2ØØØ GOTO134Ø
2Ø1Ø REM RETURN FROM WORLD BUILD
ER
2Ø2Ø RETURN
2Ø3Ø REM START A NEW WORLD
2Ø4Ø POKE BE,255:LO=BG:N=Ø:GOSUB
2ØØ:N=&HFFFF:LO=NS+12:GOSUB2ØØ:L
O=EW+12:GOSUB2ØØ:N=7:LO=X:GOSUB2
ØØ:N=5:LO=Y:GOSUB2ØØ:N=2:LO=DX:G
OSUB2ØØ:N=Ø:LO=DY:GOSUB2ØØ:POKEC
A,&H7E:GOTO132Ø:REM NEW WORLD
2Ø5Ø REM END OF SUBROUTINE TO BU
ILD A WORL
2Ø6Ø REM
2Ø7Ø REM 'NEW'--ERASE PROGRAM
2Ø8Ø REM
2Ø9Ø LO=PR:N=&HFFFF:GOSUB2ØØ:RET
URN
21ØØ REM
211Ø REM CLOAD A PROGRAM FROM TA
PE
212Ø REM
213Ø I=INSTR(L$,CHR$(34)):IFI>ØT
HENJ=INSTR(I+1,L$,CHR$(34)):IFJ<
=I+1 THENN$=""ELSEIFJ-I<1ØTHENN$
=MID$(L$,I+1,J-I-1)ELSEN$=MID$(L
$,I+1,8)ELSEN$=""
214Ø CLS:PRINT"LOADING ";N$;" ..
."
215Ø CLOADM N$:PRINTN$;" IS LOAD
ED.":RETURN
216Ø REM
217Ø REM CSAVE A PROGRAM FROM TA
PE
218Ø REM
219Ø I=INSTR(L$,CHR$(34)):IFI>ØT
HENJ=INSTR(I+1,L$,CHR$(34)):IFJ<
=I+1THENN$=""ELSEIFJ-I<1ØTHENN$=
MID$(L$,I+1,J-I-1)ELSEN$=MID$(L$
,I+1,8)ELSEN$=""
22ØØ PRINT"SAVING ";N$;" ..."
221Ø CSAVEM N$,PR,PR+5*ML+2,PR
222Ø RETURN
223Ø REM
224Ø REM INITIALIZE EVERYTHING
225Ø REM
226Ø DEFFNP(X)=256*PEEK(X)+PEEK(
X+1):REM 2-BYTE PEEK
227Ø DEFUSR1=&H78A2:REM RETURNS
ADDRESS OF BEEPER RECORD AT CURR
ENT X,Y--RETURNS Ø IF NO BEEPERS
 THERE
228Ø W1$=" . . . . . .
 . . . . .
          ":W2$=LEFT$(W1$,63)
229Ø DEFUSRØ=&H78CF:REM RETURNS
Ø IF FRONT IS CLEAR ELSE RETURNS
 ADDR OF RECORD IN NS OR EW CONT
AINING BLOCKING WALL
23ØØ DEFUSR2=&H78B4:REM RETURNS
ADDR OF LINE IN PROGRAM WITH LIN
E #>=CONTENTS OF LI.  RV CONTAIN
S Ø IFF LINE FOUND
231Ø DEFUSR3=&H78DF: REM MAKES R
OOM FOR NEW LINE IN PROGRAM BUFF
ER WITH LINE # LI, PUTS LI IN BU
FFER, AND RETURNS ADDRESS OF LIN
E.  RETURNS &HFFFF IF BUFFER FUL
L
232Ø CLS:PRINT@1Ø4,"CARL THE ROB
OT":PRINT:PRINTTAB(7);"BY DAVID
```

Listing 1 continued

Listing 1 continued

```
MEREDITH":PRINT:PRINTTAB(8);"COP
YRIGHT 1983"
233Ø PRINT:PRINT:PRINT"PREPARE T
APE FOR READING MACHINELANGUAGE
COMPONENT OF CARL AND  PRESS ENT
ER."
234Ø IF INKEY$><CHR$(13)THEN234Ø
ELSECLOADM"CARLM"
235Ø DIM CO$(28),C1$(28),EM$(1Ø)
:REM CONTAIN COMPRESSED COMMANDS
, COMMANDS, ERROR MESSAGES
236Ø FOR I=Ø TO 28:READCO$(I),C1
$(I):NEXTI
237Ø FOR I=ØTO1Ø:READEM$(I):NEXT
I
238Ø RU$="RUN":WO$="WORLD":NE$="
NEW":CL$="CLOAD":CS$="CSAVE":LI$
="LIST":ST$="STEP":IM$="IMMEDIAT
E":SK$="SKIPF":LL$="LLIST"
239Ø X=&H74E3:Y=&H74E5:DX=&H74E7
:DY=&H74E9:CA=&H74EB:REM LOCATIO
NS CONTAINING CARL COORDS, DIREC
TION VECTOR, AND CHARACTER REPRE
SENTING CARL
24ØØ PR=&H7ØØØ:PP=&H71F6:RV=&H74
EC:LI=&H74ED:DS=&H74FØ:DP=&H7518
:SS=&H751A:SP=&H7542:REM PROGRAM
 POINTER, PROGRAM POINTER, RETUR
N VALUE, LINE #, DOSTACK AND POI
NTER, SUB STACK AND POINTER
241Ø BE=&H72EC:NS=&H71F8:EW=&H72
72:MB=&H64:MW=&H14:ML=&H64:IC=&H
74EF:REM BEEPERS, NSWALLS,EWWALL
S,MAX NUMBERS OF BEEPERS,WALLS,P
ROGLINES;IMMEDIATE COMMAND BUFFE
R
242Ø RN=&H757E:TL=&H75C2:BG=&H74
E1:REM ADDRESS OF ROUTINE TO EXE
CUTE ONE INSTRUCTION, TURN LEFT,
 AND CARL'S BAG
243Ø BS$=STRING$(32,CHR$(8))
244Ø LO=EW:N=Ø:GOSUB2ØØ:LO=EW+2:
N=29999:GOSUB2ØØ:LO=EW+4:N=Ø:GOS
UB2ØØ:LO=EW+6:N=Ø:GOSUB2ØØ:LO=EW
+8:N=29999:GOSUB2ØØ:LO=EW+1Ø:N=3
ØØØØ:GOSUB2ØØ:LO=EW+12:N=&HFFFF:
GOSUB2ØØ:REM BUILD WALLS AROUND
CARL'S WORLD
245Ø LO=NS:N=Ø:GOSUB2ØØ:LO=NS+2:
N=Ø:GOSUB2ØØ:LO=NS+4:N=29999:GOS
UB2ØØ:LO=NS+6:N=3ØØØØ:GOSUB2ØØ:L
O=NS+8:N=Ø:GOSUB2ØØ:LO=NS+1Ø:N=2
9999:GOSUB2ØØ:LO=NS+12:N=&HFFFF:
GOSUB2ØØ
246Ø POKEBE,255:LO=BG:N=Ø:GOSUB2
ØØ:LO=X:N=7:GOSUB2ØØ:LO=Y:N=5:GO
SUB2ØØ:LO=DX:N=2:GOSUB2ØØ:LO=DY:
N=Ø:GOSUB2ØØ:POKECA,&H7E:LO=PR:N
=&HFFFF:GOSUB2ØØ:REM NO BEEPERS
IN WORLD OR BAG, CARL 2 STEPS EA
ST, 1 STEP NORTH OF SW CORNER PO
INTING EAST, NO PROGRAM.
247Ø GOSUB 67Ø:PRINT@Ø,;
248Ø REM
249Ø REM BEGIN MAIN LOOP WHERE W
E GET LINE FROM USER AND DO IT I
F IT'S A COMMAND OR ADD IT TO PR
OGRAM IF IT'S A PROGRAM LINE
25ØØ REM
251Ø GOSUB3ØØ
252Ø IF INSTR(L$,RU$)=1THENSI=Ø:
GOSUB99Ø:GOTO251Ø:REM RUN
253Ø IF INSTR(L$,WO$)=1THENGOSUB
132Ø:GOTO251Ø:REM WORLD
254Ø IFINSTR(L$,NE$)=1THENGOSUB2
Ø9Ø:GOTO251Ø:REM NEW PROGRAM-ERA
SE OLD ONE
255Ø IFINSTR(L$,CL$)=1THENGOSUB2
13Ø:GOTO251Ø:REM CLOAD
256Ø IFINSTR(L$,CS$)=1THENGOSUB2
19Ø:GOTO251Ø:REM CSAVE
257Ø IFINSTR(L$,LI$)=1THENGOSUB1
14Ø:GOTO251Ø:REM LIST
258Ø IFINSTR(L$,ST$)=1THENSI=1:G
OSUB99Ø:GOTO251Ø:REM SINGLE STEP
259Ø IF INSTR(L$,LL$)=1 THEN PS=
-2:GOSUB114Ø:PS=Ø:GOTO251Ø:REM L
LIST
26ØØ IFINSTR(L$,SK$)=1THENGOSUB2
77Ø:GOTO251Ø
261Ø IFINSTR(L$,IM$)=1THENGOSUB1
2ØØ:GOTO251Ø:REM IMMEDIATE MODE
262Ø LN=VAL(L$):IF LN<1ORLN>999T
HENGOTO272Ø:REM BEGIN DECODING I
NPUT AS LINE STARTING WITH LINE
#
263Ø L$=RIGHT$(L$,LEN(L$)-LEN(ST
R$(LN))+1):CO=Ø
264Ø IF L$<>""THEN268Ø
265Ø N=LN:LO=LI:GOSUB2ØØ:NN=USR2
(Ø):IFPEEK(RV)<>ØTHEN251Ø:REM TR
Y TO ELIMINATE LINE LN FROM PROG
RAM
266Ø N=FNP(NN+5):LO=NN:GOSUB2ØØ
267Ø IF FNP(NN)=&HFFFF THEN251ØE
LSEFORJ=2TO6:POKENN+J,PEEK(NN+J+
5):NEXTJ:NN=NN+5:GOTO267Ø
268Ø IFCO>28THENGOTO272ØELSEIFIN
STR(L$,CO$(CO))<>1THENCO=CO+1:GO
TO268Ø:REM MATCH L$ TO COMMAND
269Ø RI=VAL(RIGHT$(L$,LEN(L$)-LE
N(CO$(CO)))):REM GET LINE # REFE
RENCE OR DO-LOOP COUNT FROM L$
27ØØ LO=LI:N=LN:GOSUB2ØØ:N=USR3(
Ø)
271Ø IFN=-1 THENPRINT"PROGRAM BU
FFER FULL--SORRY":GOTO251Ø ELSE
POKE N+2,CO:LO=N+3:N=RI:GOSUB2ØØ
:GOTO251Ø:REM PUT LINE INTO PROG
RAM
272Ø PRINT"?SN ERROR":PRINT"OK":
GOTO251Ø:REM CAN'T RECOGNIZE LIN
E
273Ø REM END OF MAIN LOOP
274Ø REM
275Ø REM SKIPF
276Ø REM
277Ø REM SKIPF
278Ø I=INSTR(L$,CHR$(34)):IFI>ØT
HENJ=INSTR(I+1,L$,CHR$(34)):IFJ<
=I+1THENN$=""ELSEIFJ-I<1ØTHENN$=
MID$(L$,I+1,J-I-1)ELSEN$=MID$(L$
,I+1,8)ELSEN$=""
279Ø CLS:PRINT"SKIPPING ";N$;" .
.."
28ØØ SKIPFN$
281Ø PRINTN$;" SKIPPED":RETURN
```

END

Program Listing 2. ShowCarl, Assembly Portion

```
00100 *
00110 * CARL THE ROBOT--EXECUTION ROUTINES
00120 * BY DAVID MEREDITH OCT. 8, 1983
00130 *
00140          ORG      $7000
00150 *
00160 * DATA STRUCTURES SHARED WITH BASIC
00170 *
00180 MP       EQU      100       MAX NUMBER PROGRAM STEPS
00190 PR       RMB      5*MP+2    WHERE THE PROGRAM IS STORED
00200 * CARL'S PROGRAM IS STORED IN 5 BYTE RECORDS--2 BYTE
00210 * LINE NUMBER, 1 BYTE COMMAND CODE, AND 2 BYTE
00220 * INTEGER FOR GOTO, GOSUB, IF, OR DO N TIMES.
00230 * THE LAST PROGRAM LINE IS FOLLOWED BY A
00240 * LINE NUMBER = $FFFF
00260 EP       EQU      .-2       ADD OF LAST LINE NUMBER
00270 PP       RMB      2         POINTS TO NEXT INSTRUCTION
00280 * BUFFERS FOR WALLS AND BEEPERS.
00290 * POINTS IN CARL'S WORLD ARE REPRESENTED BY INTEGER
00300 * COORDINATES N,M.  CARL LIVES AT ODD COORDS, AND HIS
00310 * STEPS ARE LENGTH 2. WALLS GO IN BETWEEN CARLS POINTS
00320 MW       EQU      20        MAX NUMBER OF WALLS EACH WAY
00330 * EACH WALL IS GIVEN BY A 6 BYTE RECORD--X,Y1,Y2 FOR NS
00340 * WALLS; X1,X2,Y FOR EW WALLS.  FOR NS WALLS, X IS EVEN
00350 * AND Y1,Y2 ARE ODD.  $FFFF FOLLOWS LAST RECORD
00360 * THE FIRST 2 BYTES FOLLOWING LAST WALL CONTAIN $FFFF
00370 NS       RMB      MW*6+2    NORTH-SOUTH WALLS
00380 EN       EQU      .-2       LAST TWO BYTES OF NS WALLS
00390 EW       RMB      MW*6+2    EW WALLS
00400 EE       EQU      .-2       LAST TWO BYTES OF EW WALLS
00410 MB       EQU      100       MAX # BEEPER LOCATIONS
00420 BE       RMB      MB*5+1    BEEPER BUFFER HOLDING 5 BYTE
00430 * RECORDS N,X,Y; N BEEPERS AT LOCATION X,Y
00440 EB       EQU      .-1       END OF BEEPER BUFFER
00450 BG       RMB      2         BEEPER BAG
00460 CX       RMB      2         CARL'S X-COORDINATE
00470 CY       RMB      2         CARL'S Y-COORDINATE
00480 * (DX,DY) IS CARL'S MOVE VECTOR.
00490 * PERMITTED VALUES ARE (+/-2,0) AND (0,+/-2).
00500 DX       RMB      2
00510 DY       RMB      2
00520 CA       RMB      1         HOLDS POKE # FOR CARL-->,^,<,V
00530 LI       RMB      2         USED TO HOLD LINE NUMBERS
00540 IC       RMB      1         COMMAND CODE FOR IMMEDIATE MODE
00550 RV       RMB      1         RETURN CONDITION CODE TO BASIC
00560 * CONDITION CODES
00570 RVOK     EQU      0         INSTRUCTION EXEC OK; WORLD SAME
00580 RVMOVE   EQU      1         CARL MOVED BY INSTRUCTION
00590 RVBAG    EQU      2         BAG ALTERED BY INSTRUCTION
00600 RVTURN   EQU      3         CARL TURNED BY INSTRUCTION
00610 RVFBLK   EQU      4         FRONT IS BLOCKED
00620 RVNOBE   EQU      5         NO BEEPERS HERE
00630 RVBMTY   EQU      6         BAG EMPTY
00640 RVNWDO   EQU      7         NEXT WITHOUT DO
00650 RVRWGS   EQU      8         RETURN WITHOUT GOSUB
00660 RVDOV    EQU      9         DO STACK OVERFLOW
00670 RVSSV    EQU      10        SUBROUTINE STACK OVERFLOW
00680 RVUNLN   EQU      11        UNDEFINED LINE #
00690 RVSTOP   EQU      12        STOP ENCOUNTERED
00700 RVNSTP   EQU      13        END OF PROGRAM WITHOUT STOP
00710 RVTOOB   EQU      14        TOO MANY BEEPER LOCATIONS
00720 * DO AND SUBROUTINE STACKS
00730 MD       EQU      10        MAX NESTING OF DO'S
00740 DS       RMB      4*MD      DO-NEXT STACK.
00750 * 4-BYTE RECORD IS 2-BYTE COUNT AND 2-BYTE ADDR OF DO
00760 ED       EQU      .-4       END OF DO STACK
00770 DQ       RMB      2         POINTS TO MOST RECENT DO RECORD
00780 MS       EQU      20        MAX NESTING OF SUBROUTINES
00790 SS       RMB      2*MS      SUBROUTINE STACK
00800 * 2-BYTE RECORD IS ADDRESS OF GOSUB
00810 ES       EQU      .-2       END OF SUB STACK
00820 SP       RMB      2         POINTS TO LAST GOSUB RECORD
00830 *
00840 * TABLE OF JUMP ADDRESSES
00850 * COMMAND CODE I MEANS JUMP TO I'TH ADDRESS
00860 *
00870 JUMP     FDB      MOVE
00880          FDB      TURNL
00890          FDB      TURNR
00900          FDB      NEXT
00910          FDB      RETURN
00920          FDB      PICK
00930          FDB      PUT
00940          FDB      STOP
```

Listing 2 continued

Listing 2 continued

```
00950          FDB     DO
00960          FDB     GOTO
00970          FDB     GOSUB
00980          FDB     IFFCLR
00990          FDB     IFLCLR
01000          FDB     IFRCLR
01010          FDB     IFFBLK
01020          FDB     IFLBLK
01030          FDB     IFRBLK
01040          FDB     IFBEEP
01050          FDB     IFNBEE
01060          FDB     IFBMT
01070          FDB     IFBNMT
01080          FDB     IFFN
01090          FDB     IFFS
01100          FDB     IFFE
01110          FDB     IFFW
01120          FDB     IFNFN
01130          FDB     IFNFS
01140          FDB     IFNFE
01150          FDB     IFNFW
01160 *
01170 * THE ALGORITHMS
01180 *
01190 * EXECUTE THE PROGRAM STEP AT [PP]
01200 *
01210 RUN      LDX     PP        ADDRESS OF NEXT PROGRAM STEP
01220          LDY     ,X        GET LINE #
01230          CMPY    #$FFFF
01240          BEQ     RUN1      IF END OF PROGRAM
01250          LDY     #JUMP
01260          LDA     2,X       COMMAND CODE FOR INSTRUCTION
01270          ASLA
01280          JMP     [A,Y]     GO DO THE INSTRUCTION
01290 RUN1     LDD     PP        IF END OF PROGRAM ENCOUNTERED
01300          SUBD    #5
01310          STD     PP        PP POINTS TO LAST INSTRUCTION
01320          LDA     #RVNSTP   INDICATE END WITHOUT STOP ERROR
01330          LBRA    ERROR
01340 *
01350 * EXECUTE WALK INSTRUCTION
01360 *
01370 MOVE     JSR     FCLEAR    CHECK OF FRONT IS CLEAR
01380          BNE     MOV1      FRONT IS BLOCKED
01390          LDD     CX        MOVE CARL
01400          ADDD    DX        CX=CX+DX
01410          STD     CX
01420          LDD     CY
01430          ADDD    DY        CY=CY+DY
01440          STD     CY
01450          LDA     #RVMOVE   TELL BASIC CARL MOVED OK
01460          LBRA    FINISH
01470 MOV1     LDA     #RVFBLK   TELL BASIC FRONT IS BLOCKED
01480          LBRA    ERROR
01490 *
01500 * EXECUTE TURN LEFT INSTRUCTION
01510 *
01520 TURNL    JSR     TLEFT
01530          JSR     SETCRL
01540          LDA     #RVTURN   REPORT CARL TURNED
01550          LBRA    FINISH
01560 *
01570 TLEFT    LDX     DX        TURN CARL LEFT 90 DEGREES
01580          LDD     DY
01590          STX     DY
01600          COMB
01610          COMA
01620          ADDD    #1
01630          STD     DX
01640          RTS
01650 *
01660 * EXECUTE TURN RIGHT INSTRUCTION
01670 *
01680 TURNR    JSR     TRIGHT
01690          JSR     SETCRL
01700          LDA     #RVTURN   TELL BASIC CARL TURNED
01710          LBRA    FINISH
01720 *
01730 TRIGHT   LDX     DY        TURN CARL RIGHT 90 DEGREES
01740          LDD     DX
01750          STX     DX
01760          COMB
01770          COMA
01780          ADDD    #1
01790          STD     DY
01800          RTS
01810 *
01820 SETCRL   LDD     DX        MAKE CHAR CODE IN CA
01830 *                          CORRESPOND TO CARL'S DIRECTION
01840 *                          (DX,DY)
01850          BEQ     SET1      IF CARL POINTING NORTH OR SOUTH
01860          BMI     SET2      IF CARL POINTING WEST
01870          LDA     #$7E      > IF POINTING EAST
01880          STA     CA
01890          RTS
01900 SET2     LDA     #$7C      < IF POINTING WEST
01910          STA     CA
01920          RTS
01930 SET1     LDA     DY
01940          BMI     SET3      IF POINTING SOUTH
01950          LDA     #$5E      ^ IF POINTING NORTH
01960          STA     CA
01970          RTS
01980 SET3     LDA     #$56      V IF POINTING SOUTH
01990          STA     CA
02000          RTS
02010 *
02020 * EXECUTE "NEXT" INSTRUCTION
02030 *
02040 NEXT     LDX     DQ        TOP OF DO-NEXT STACK
02050          CMPX    #DS       BOTTOM OF DO-NEXT STACK
02060          BLO     NX2       IF DO STACK EMPTY
02070          LDD     ,X        COUNT OF LOOPS REMAINING+1
02080          SUBD    #1
02090          BEQ     NX1       DO NOT DO LOOP AGAIN
02100          STD     ,X        PUSH REDUCED COUNT TO STACK
02110          LDD     2,X
02120          STD     PP        ADDR OF DO INSTRUCTION
02130          LDA     #RVOK     TELL BASIC EVERYTHING OK
02140          LBRA    FINISH
02150 NX1      LEAX    -4,X      IF LOOP COUNT REDUCED TO 0
02160          STX     DQ        POP LAST RECORD OFF DO STACK
02170          LDA     #RVOK     TELL BASIC EVERYTHING OK
02180          LBRA    FINISH
02190 NX2      LDA     #RVNWDO TELL BASIC NEXT WITHOUT DO
02200          LBRA    ERROR
02210 *
02220 * EXECUTE RETURN FROM GOSUB
02230 *
02240 RETURN   LDX     SP        TOP OF SUBROUTINE STACK
02250          CMPX    #SS       BOTTOM OF STACK
02260          BLO     RTN1      SUBROUTINE STACK EMPTY
02270          LDD     ,X
02280          STD     PP        ADDR OF GOSUB
02290          LEAX    -2,X
02300          STX     SP        POP RECORD OFF SUBROUTINE STACK
02310          LDA     #RVOK     TELL BASIC EVERYTHING OK
02320          LBRA    FINISH
02330 RTN1     LDA     #RVRWGS IF RETURN WITHOUT GOSUB
02340          LBRA    ERROR
02350 *
02360 * EXECUTE PICK BEEPER
02370 *
02380 PICK     JSR     BEEPHR    CHECK IF CARL AT A BEEPER
02390          BEQ     PICK1     NO BEEPER TO PICK
02400          DEC     ,X        REDUCE BEEPER COUNT HERE
02410          BNE     PICK2
02420 PICK3    LDA     5,X       ELIMINATE RECORD IN BEEPER
02430 *                          BUFFER IF NO MORE BEEPERS HERE
02440          STA     ,X+
02450          CMPA    #-1
02460          BEQ     PICK2     ALL RECORDS MOVED UP
02470          LDD     5,X
02480          STD     ,X++
02490          LDD     5,X
02500          STD     ,X++
02510          BRA     PICK3
02520 PICK2    LDD     BG        ADD A BEEPER TO CARL'S BAG
02530          ADDD    #1
02540          STD     BG
02550          LDA     #RVBAG    TELL BASIC BAG HAS CHANGED
02560          LBRA    FINISH
02570 PICK1    LDA     #RVNOBE TELL BASIC NO BEEPER TO PICK
02580          LBRA    ERROR
02590 *
02600 * EXECUTE PUT BEEPER INSTRUCTION
02610 *
02620 PUT      LDD     BG        # BEEPERS IN BAG
02630          BEQ     PUT1      BAG EMPTY
02640          SUBD    #1
02650          STD     BG        STORE REDUCED COUNT IN BAG
02660          JSR     BEEPHR  X WILL POINT TO BEEPER RECORD
02670          BEQ     PUT2      IF NO BEEPERS ALREADY HERE
02680          INC     ,X        ADD TO BEEPER COUNT HERE
02690          LDA     #RVBAG    TELL BASIC BAG CHANGED
02700          LBRA    FINISH
02710 PUT2     CMPX    #EB       ADD NEW RECORD TO BEEPER BUFFER
02720          BEQ     PUT3      IF NO MORE ROOM IN BEEPER LIST
02730          LDA     #1        BEEPER COUNT FOR NEW RECORD
02740          STA     ,X+
02750          LDD     CX        THEN X-COORDINATE
02760          STD     ,X++
02770          LDD     CY        THEN Y-COORDINATE
02780          STD     ,X++
02790          LDA     #-1       FINALLY END OF DATA MARKER
02800          STA     ,X
02810          LDA     #RVBAG    TELL BASIC BAG CHANGED
02820          LBRA    FINISH
02830 PUT3     LDA     #RVTOOB TELL BASIC BEEPER BUFFER OVLO
02840          LBRA    ERROR
02850 PUT1     LDA     #RVBMTY TELL BASIC BAG EMPTY
02860          LBRA    ERROR
02870 *
02880 * EXECUTE STOP INSTRUCTION
02890 *
02900 STOP     LDA     #RVSTOP TELL BASIC STOP COMMAND EXEC
02910          LBRA    ERROR
02920 *
02930 * EXECUTE DO N TIMES
02940 *
02950 DO       LDY     DQ        CURRENT TOP OF DO-NEXT STACK
02960          CMPY    #ED
02970          BEQ     DO1       DO STACK ABOUT TO OVERFLOW
02980          LEAY    4,Y
02990          STY     DQ        INCREMENT STACK POINTER
03000          LDD     3,X
03010          STD     ,Y++      PUSH LOOP COUNT TO STACK
03020          STX     ,Y        PUSH ADDRESS OF DO TO STACK
03030          LDA     #RVOK     TELL BASIC EVERYTHING OK
03040          LBRA    FINISH
03050 DO1      LDA     #RVDOV    TELL BASIC DO STACK OVERFLOWS
03060          LBRA    ERROR
```

Listing 2 continued

Listing 2 continued

```
03070 *
03080 * EXECUTE GOTO LINE #
03090 *
03100 GOTO    JSR     FINDLI  PP POINTS TO WHERE WE GO
03110         BNE     GO1     NO SUCH LINE
03120         LDA     #RVOK   TELL BASIC EVERYTHING OK
03130         STA     RV      DON'T GOTO FINISH--CHANGES PP
03140         RTS
03150 GO1     LDA     #RVUNLN IF LINE # UNDEFINED
03160         LBRA    ERROR
03170 *
03180 * EXECUTE GOSUB LINE #
03190 *
03200 GOSUB   LDY     SP      POINTER TO TOP OF SUB STACK
03210         CMPY    #ES
03220         BEQ     GOS2    SUB STACK ABOUT TO OVERFLOW
03230         LEAY    2,Y     INCR POINTER TO TOP OF STACK
03240         STY     SP
03250         STX     ,Y      PUSH ADDRESS OF GOSUB TO STACK
03260         JSR     FINDLI  PP POINTS TO WHERE WE GO
03270         BNE     GOS1    UNDEFINED LINE #
03280         LDA     #RVOK   TELL BASIC EVERYTHING OK
03290         STA     RV      DON'T GOTO FINISH--CHANGES PP
03300         RTS
03310 GOS1    LDY     SP      LINE # IN GOSUB UNDEFINED
03320         LEAY    -2,Y    POP FAULTY RECORD OFF SUB STACK
03330         STY     SP
03340         LDA     #RVUNLN TELL BASIC UNDEFINED LINE #
03350         LBRA    ERROR
03360 GOS2    LDA     #RVSSV  TELL BASIC SUB STACK OVERFLOWS.
03370         LBRA    ERROR
03380 *
03390 * LOOK FOR LINE # REFERENCED IN CURRENT LINE.  IF
03400 * FOUND, SET PP=ADDR OF LINE IN PROGRAM BUFFER AND Z=1
03410 * ELSE Z=0.  ALWAYS YREG POINTS TO FIRST LINE WITH
03420 * LINE # >= LINE # SEARCHED FOR.  THIS COULD BE END
03430 * OF PROGRAM MARKER LINE #=$FFFF.
03440 *
03450 FINDLI  LDX     PP
03460         LDX     3,X     LINE # TO SEARCH FOR
03470         LDY     #PR     START OF PROGRAM BUFFER
03480 FIND3   CMPX    ,Y
03490         BLO     FIND1   IF NO SUCH LINE #
03500         BEQ     FIND2   IF TARGET LINE # FOUND
03510         LEAY    5,Y     LOOK AT NEXT LINE
03520         BRA     FIND3
03530 FIND2   STY     PP      PP POINTS TO LINE SEARCHED FOR
03540         ORCC    #$4     Z=1
03550         RTS
03560 FIND1   ANDCC   #$FB    Z=0 IF LINE NUMBER NOT FOUND
03570         RTS
03580 *
03590 * WHAT TO DO IF CONDITIONAL STATEMENT TRUE
03600 *
03610 IFTRUE  JSR     FINDLI  PP=WHERE TO GO
03620         BNE     IFT1    NO LINE TO JUMP TO
03630         LDA     #RVOK   TELL BASIC EVERYTHING OK
03640         STA     RV      DON'T GOTO FINISH--ALTERS PP
03650         RTS
03660 IFT1    LDA     #RVUNLN IF NO LINE TO JUMP TO
03670         LBRA    ERROR
03680 *
03690 * WHAT TO DO IF CONDITIONAL STATEMENT FALSE
03700 *
03710 IFFALS  LDA     #RVOK   TELL BASIC EVERYTHING OK
03720         LBRA    FINISH
03730 *
03740 * EXECUTE CONDITIONAL STATEMENTS
03750 *
03760 IFFCLR  JSR     FCLEAR  IF FRONT IS CLEAR
03770         BEQ     IFTRUE
03780         BRA     IFFALS
03790 IFLCLR  JSR     TLEFT   IF LEFT IS CLEAR
03800         JSR     IFFCLR
03810         JSR     TRIGHT
03820         RTS
03830 IFRCLR  JSR     TRIGHT  IF RIGHT IS CLEAR
03840         JSR     IFFCLR
03850         JSR     TLEFT
03860         RTS
03870 IFFBLK  JSR     FCLEAR  IF FRONT IS BLOCKED
03880         BNE     IFTRUE
03890         BRA     IFFALS
03900 IFLBLK  JSR     TLEFT   IF FRONT IS BLOCKED
03910         JSR     IFFBLK
03920         JSR     TRIGHT
03930         RTS
03940 IFRBLK  JSR     TRIGHT  IF RIGHT IS BLOCKED
03950         JSR     IFFBLK
03960         JSR     TLEFT
03970         RTS
03980 IFBEEP  JSR     BEEPHR  IF BEEPER HERE
03990         BNE     IFTRUE
04000         BRA     IFFALS
04010 IFNBEE  JSR     BEEPHR  IF BEEPER NOT HERE
04020         BEQ     IFTRUE
04030         BRA     IFFALS
04040 IFBMT   LDD     BG      IF BAG EMPTY
04050         BEQ     IFTRUE
04060         BRA     IFFALS
04070 IFBNMT  LDD     BG      IF BAG NOT EMPTY
04080         BNE     IFTRUE
04090         BRA     IFFALS
04100 IFFN    LDD     DY      IF FACING NORTH
04110         LBGT    IFTRUE
04120         LBRA    IFFALS
04130 IFFS    LDD     DY      IF FACING SOUTH
04140         LBLT    IFTRUE
04150         LBRA    IFFALS
04160 IFFE    LDD     DX      IF FACING EAST
04170         LBGT    IFTRUE
04180         LBRA    IFFALS
04190 IFFW    LDD     DX      IF FACING WEST
04200         LBLT    IFTRUE
04210         LBRA    IFFALS
04220 IFNFN   LDD     DY      IF NOT FACING NORTH
04230         LBLE    IFTRUE
04240         LBRA    IFFALS
04250 IFNFS   LDD     DY      IF NOT FACING SOUTH
04260         LBGE    IFTRUE
04270         LBRA    IFFALS
04280 IFNFE   LDD     DX      IF NOT FACING EAST
04290         LBLE    IFTRUE
04300         LBRA    IFFALS
04310 IFNFW   LDD     DX      IF NOT FACING WEST
04320         LBGE    IFTRUE
04330         LBRA    IFFALS
04340 *
04350 * FRONT IS CLEAR RETURNS Z=1 IF FRONT IS CLEAR ELSE Z=0
04360 * ADDRESS OF BLOCKING WALL IN DREG
04370 *
04380 FCLEAR  LDD     DX
04390         BEQ     FCL1    IF FACING NORTH OR SOUTH
04400 * MAKE (DREG,YREG)=POINT IN FRONT OF CARL
04410         ASRA
04420         RORB    DREG = DX/2
04430         ADDD    CX
04440         LDY     CY
04450         LDX     #NS     ADDRESS OF NORTH-SOUTH WALLS
04460 * BEGIN TESTING IF POINT (DREG,YREG) ON WALL [XREG]
04470 FCL3    TST     ,X
04480         BMI     FCL4    IF END OF RECORDS
04490         CMPD    ,X
04500         BNE     FCL2
04510         CMPY    2,X
04520         BLO     FCL2
04530         CMPY    4,X
04540         BHI     FCL2
04550         TFR     X,D     FOUND THE BLOCKING WALL
04560         ANDCC   #$FB    Z=0
04570         RTS
04580 FCL2    LEAX    6,X     LOOK AT NEXT WALL
04590         BRA     FCL3
04600 FCL4    ORCC    #$4     Z=1--FRONT IS CLEAR
04610         RTS
04620 * BEGIN CHECK FOR BLOCKING EW WALL
04630 * MAKE (XREG,DREG)=POINT IN FRONT OF CARL
04640 FCL1    LDD     DY
04650         ASRA
04660         RORB
04670         ADDD    CY
04680         LDX     CX
04690         LDY     #EW     ADDRESS OF EAST-WEST WALLS
04700 * BEGIN TESTING IF POINT (XREG,DREG) ON WALL [YREG]
04710 FCL5    TST     ,Y
04720         BMI     FCL6    IF END OF RECORDS
04730         CMPD    4,Y
04740         BNE     FCL7
04750         CMPX    ,Y
04760         BLO     FCL7
04770         CMPX    2,Y
04780         BHI     FCL7
04790         TFR     Y,D     FOUND THE BLOCKING WALL
04800         ANDCC   #$FB    Z=0
04810         RTS
04820 FCL7    LEAY    6,Y     LOOK AT NEXT WALL
04830         BRA     FCL5
04840 FCL6    ORCC    #4      Z=1--FRONT IS CLEAR
04850         RTS
04860 *
04870 * CHECK FOR BEEPER AT CX,CY. IF ANY HERE THEN
04880 * XREG POINTS TO RECORD IN BEEPER BUFFER AND Z=0
04890 * ELSE Z=1 AND XREG POINTS TO NEXT AVAILABLE RECORD
04900 * IN BEEPER BUFFER
04910 *
04920 BEEPHR  LDU     CX      CARL'S LOCATION
04930         LDY     CY
04940         LDX     #BE     START OF BEEPER BUFFER
04950         LDB     #-1
04960 BP2     CMPB    ,X
04970         BEQ     BP1     IF END OF RECORDS
04980         CMPU    1,X     COMPARE BEEPER RECORD TO
04990         BNE     BP3     CARL'S LOCATION
05000         CMPY    3,X
05010         BNE     BP3
05020         ANDCC   #$FB    Z=0  FOUND BEEPER
05030         RTS
05040 BP3     LEAX    5,X     GET NEXT BEEPER RECORD
05050         BRA     BP2
05060 BP1     ORCC    #4      Z=1  NO BEEPERS AT CX,CY
05070         RTS
05080 *
05090 * NORMAL TERMINATION OF EXECUTION OF INSTRUCTION
05100 *
05110 FINISH  STA     RV      TELL BASIC WHAT HAPPENED
05120         LDX     PP      ADVANCE PROGRAM POINTER
05130         LEAX    5,X     TO NEXT INSTRUCTION
05140         STX     PP
05150         RTS
05160 *
05170 * REPORT AN ERROR (OR STOP INSTRUCTION) BACK TO BASIC
05180 *
```

Listing 2 continued

Listing 2 continued

```
05190 ERROR   STA     RV      TELL BASIC WHAT HAPPENED
05200         LDX     [PP]    LINE NUMBER OF OFFENDING LINE
05210         STX     LI      FOR BASIC TO USE
05220         RTS
05230 *
05240 * USR1 RETURNS TO BASIC THE ADDRESS OF BEEPER RECORD
05250 * CORRESPONDING TO LOCATION CY,CY.  RETURNS 0 IF NONE
05260 *
05270 USR1    JSR     BEEPHR  LOOK FOR BEEPER RECORD
05280         LBEQ    USR11   NO BEEPERS HERE
05290         TFR     X,D     RETURN ADDRESS OF BEEPER RECORD
05300         JSR     $B4F4   TO BASIC VIA USR FUNCTION
05310         RTS
05320 USR11   LDD     #0      RETURN 0 IF NO BEEPERS
05330         JSR     $B4F4
05340         RTS
05350 *
05360 * USR2 TAKES LINE NUMBER IN LI AND MAKES ROOM IN
05370 * IN PROGRAM BUFFER FOR A LINE WITH THAT LINE #.
05380 * IF A LINE WITH THIS LINE # ALREADY PRESENT,
05390 * THEN ITS ADDRESS RETURNED TO BASIC.  OTHERWISE LINES
05400 * ARE MOVED TO MAKE ROOM FOR NEW LINE, NEW LINE # IS
05410 * INSERTED IN PROGRAM BUFFER, AND ADDRESS OF NEW LINE
05420 * IS RETURNED TO BASIC.
05430 * RV=0 IF LINE# ALREADY DEFINED, ELSE RV=-1
05440 *
05450 F       SET     LI-3
05460 USR2    LDX     #F      SET UP PP SO LINE # TO BE
05470         STX     PP      SEARCHED FOR IS IN [PP]+3
05480         JSR     FINDLI
05490         BEQ     USR21   IF LINE NUMBER FOUND
05500         LDA     #-1
05510         STA     RV
05520 USR22   TFR     Y,D     TELL BASIC ADDR OF LINE
05530         JSR     $B4F4   VI USR FUNCTION
05540         RTS
05550 USR21   CLR     RV      IF LINE#=LI FOUND
05560         BRA     USR22
05570 *
05580 * USR0 RETURNS ADDRESS OF RECORD OF WALL IN FRONT OF
05590 * CARL IF ANY, ELSE RETURNS 0.
05600 *
05610 USR0    JSR     FCLEAR  LOOK FOR WALL IN FRONT OF CARL
05620         BEQ     USR01   FRONT IS CLEAR
05630         JSR     $B4F4   RETURN ADDRESS OF WALL TO BASIC
05640         RTS
05650 USR01   LDD     #0      IF FRONT IS CLEAR
05660         JSR     $B4F4   RETURN 0
05670         RTS
05680 *
05690 * USR3 MAKES ROOM FOR A NEW LINE WITH LINE # LI IN
05700 * PROGRAM BUFFER.  IF LINE # NOT ALREADY DEFINED,
05710 * LINES WITH #>LI MOVED UP AND LINE # LI INSERTED
05720 * PROGRAM BUFFER AND ADDRESS WHERE INSERTED RETURNED
05730 * TO BASIC.  ELSE ADDRESS OF CURRENT LINE LI RETURNED
05740 * TO BASIC. IF NO ROOM FOR NEW LINE, RETURNS -1.
05750 *
05760 F       SET     LI-3
05770 USR3    LDX     #F      SET PP SO LINE # TO BE
05780         STX     PP      SEARCHED FOR IS IN [PP]+3
05790         JSR     FINDLI
05800         BEQ     USR31   EQUAL LINE # FOUND
05810         PSHS    Y       ELSE MOVE LINES & INSERT LINE#
05820 USR32   LDA     ,Y      ADVANCE YREG TO END OF PROGRAM
05830         CMPA    #-1
05840         BEQ     USR33   YREG POINTS TO END OF PROGRAM
05850         LEAY    5,Y
05860         BRA     USR32
05870 USR33   CMPY    #EP
05880         BEQ     USR34   IF NO ROOM FOR NEW LINE
05890         LDD     ,Y      MOVE LINES UP
05900         STD     5,Y
05910 USR36   CMPY    ,S      [S] IS LOWEST LINE TO MOVE
05920         BEQ     USR35   ALL LINES MOVED UP
05930         LEAY    -5,Y
05940         LDD     ,Y
05950         STD     5,Y
05960         LDA     2,Y
05970         STA     7,Y
05980         LDD     3,Y
05990         STD     8,Y
06000         BRA     USR36
06010 USR35   LDD     LI      PUT NEW LINE # IN BUFFER
06020         STD     ,Y
06030         LEAS    2,S     CLEAR USE OF S-STACK
06040         TFR     Y,D     REPORT ADDRESS OF NEW LINE
06050         JSR     $B4F4   TO BASIC
06060         RTS
06070 USR34   LDD     #-1     IF NO ROOM FOR NEW LINE
06080         JSR     $B4F4   TELL BASIC
06090         LEAS    2,S
06100         RTS
06110 USR31   LDD     LI      EQUL LINE # FOUND
06120         STD     ,Y      DON'T MOVE LINES
06130         TFR     Y,D
06140         JSR     $B4F4   TELL BASIC ADDRESS OF LINE
06150         RTS
06160         END
```

END

BY COLIN J. STEARMAN

PLAYING THE NUMBERS GAME

If you are not a math whiz, this tutorial will give you a rudimentary grasp of computer logic.

If you are fascinated by computers, understanding numbers makes life much easier. After all, that's all the computer understands.

If you are going to understand numbers relating to computers, you better understand numbers used in everyday life. But rather than dust off those old high-school math books, let's approach the topic from a different angle. I promise not to burden you with heavy math.

Ten Little Indians

Have you ever wondered why we count in tens? Obviously, it's because we have 10 fingers.

A number system that counts from 0 to 9 and then starts over is called a decimal system (from the Latin *decimalis*, meaning ten). Another name for it is a base 10 system.

Let's look at the decimal system more closely. As an example, I'll dissect the decimal number 173. What do those three digits tell us? Instinctively you have a feeling for how big the number is. The "3" says that there are three "ones," the "7" means seven "tens," and the "1" represents one "hundred." Figure 1 shows this, with each digit put into its proper column.

Look at the numbers at the top of each column, which include the decimal number that the column represents and the power of 10 for each. (The power of 10 is the number of times 10 must be multiplied by itself to get the column value. For example, 10^2 is 10 × 10, which is 100. See the sidebar for a better explanation.)

Look at the power numbers for a minute. Notice how they start at 0 and go up one for each column. Each column is worth the number base (10) raised to the power of the column number, with the columns numbered from zero.

Rule 1: In any number system the value of each column is the number system base raised to the power of the column number.

When you add decimal numbers, you automatically carry forward when any column contains more than nine. You don't even have a symbol for "ten" because you express "ten" as 10, meaning one lot of "ten" and zero lots of "one." From now on I want you to think of the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 just as symbols, their value depending on which column they are in and the base of the number.

How Do Computers Fit In?

When all is said and done, computers are just a lot of electronic switches, which can be either on or off. In other words, each switch can represent two digits, a "one" when it's on and a "zero" when it's off. A switch representing one column in a decimal number would have to handle 10 different values (for 0–9), but it can only be on or off. The decimal system, which has done so well for smart humans, is too complex for dumb computers.

Let's look at the requirements of a system designed to accommodate computers. Really, there is only one. It must not need a value higher than one in any column. Then the computer can use one switch to represent each column.

In the base 10 system the highest number in each column is nine. So if you add one to the highest value in the new system (one) you get the new system base (two). I told you the math was going to be simple.

Rule 2: The highest number any column can contain is one less than the base of that number.

Using Rule 1, you can create a picture of the new number system. By taking the base (2) and raising it to the power of the column number, you can find out what each column is worth in the decimal system. The name of this new system is binary, from the Latin *binarius*.

Number Representation

Before looking at the new number system, let's get a few things straight.

Peter Bono

As you will see shortly, numbers in other bases can look very much like those in base 10. To distinguish the other bases, I'll use an abbreviation after the number to indicate its base. For binary it will be (BIN), octal will be (OCT) and hexadecimal (HEX). No abbreviation will follow a decimal number.

Let's create that base 2 picture now. Take a look at Fig. 2. You see columns like those in Fig. 1, but I have changed the column values in accordance with Rule 1. Remember, if the numbers have no suffix, they are base 10. The column values are shown in base 10 to help you relate to them better.

I have included a few more columns that I used in Fig. 1 and have put in a binary number. Take a look at what happens when you dissect it using the same method as in Fig. 1. The binary number I chose comes out to the same as the example decimal number, so you can say that 10101101 (BIN) is the same quantity as 173. Notice how many more digits it took to express that quantity in binary. Each column is worth less in binary, so more are needed.

Rule 3: To convert numbers between bases, multiply each column digit by the value of that column (in the new base) and add up all the results (using arithmetic in the new base throughout).

Rule 3 requires you to do math in other bases if you are not converting to base 10.

Figure 3 shows another conversion technique that lends itself to calculator use. This example converts 94 to its binary equivalent. The number to be converted is divided by the next highest column value (in the new base). The integer portion of the result is the number that appears in that column. The original number is reduced by the product of this integer and the column value. This "conversion procedure" is repeated until the original number is all gone. You can see why a calculator is helpful.

Either Rule 3 or the conversion procedure is awkward to use, so let's consider ways around this problem.

Octal to the Rescue

Normally, you use decimal because you have 10 fingers, and you have to use binary because that's all the computer can understand. Why in the world should we bring in a third number system? There is only one reason—convenience.

It is obvious from Fig. 2, Rule 3, and the conversion procedure that conversion from binary to decimal and back is not easy. You could always get used to binary, but that is also a problem because binary numbers are just too cumbersome.

In Fig. 2 the decimal number 173 is equal to 10101101 (BIN). It took eight digits in binary to describe a value that took only three digits in decimal. Imagine if the decimal number were 14,363,254. That would take 23 digits in binary.

Octal comes from the Latin *octavus*, meaning eight, so an octal number is a base 8 number. According to Rule 2, the highest digit a column can contain is seven. Figure 4 shows 173 expressed as an octal number and dissected as before. Using the same methods used for the decimal and binary numbers, see if you can figure out the dissection.

You should now be convinced that 255 (OCT) is the same quantity as 173. But what's so convenient about that? Quite frankly, nothing when you are going from decimal to octal and back. You still have to use Rule 3 or the conversion procedure. When you want to go from binary to octal, however, things become much simpler. Look at Fig. 5 and see what I mean.

I have rewritten the binary number from Fig. 2 and separated the digits into groups of three. I then treated each group as a separate binary number and calculated its value as shown. It is the same method as in Fig. 2, for each group. The result of each group conversion is written at the bottom of the col-

umn, and the octal number magically appears.

The highest binary number that could exist in a group of three is 111(BIN), which is seven. The highest digit in an octal number is also seven (Rule 2), so they are consistent. This requires remembering the binary only for the digits 0–7.

Rule 4: To convert from binary to octal, separate the number into groups of three digits, starting from the right, and convert each group to its symbol equivalent.

You can easily go from octal to binary by just reversing the procedure.

Rule 5: To convert from octal to binary, convert each digit to its binary equivalent.

Who Needs Hexadecimal?

Many computer people are quite comfortable with octal and never use hexadecimal. This time it's the Greek that gives us hex, meaning six, plus decimal (ten), which adds up to base 16. But Rule 2 says that the highest digit in a number system column is one less than the base, and base 16 has no symbol for the quantity 15. Because 0–9 are just symbols for the quantity they represent, you can just use some other symbols for 10–15. A commonly accepted set of symbols is the letters A through F.

Figure 6 shows a hexadecimal number in a way that should now be familiar. I have also shown the additional symbols that hexadecimal numbers need. As you can see, AD(HEX) is a valid quantity and is the same as 173. Go through the same procedure of dissection and prove it to yourself.

I still haven't answered the question "Who needs it?" In Fig. 5 you had to add a nonsignificant zero to the left of the binary number to give it nine digits so that you could divide it into three groups of three. Unfortunately, computers are normally 8-bit or 16-bit machines. This means they represent numbers in groups (called bytes) of 8 or 16 "switches."

You cannot divide either of these into groups of three without adding extra zeros on the left.

If you look at Fig. 7, I have done the same thing as in Fig. 5, but have divided the binary number into groups of four digits.

Following a process similar to the one you used to convert binary to octal, you can convert binary to hexadecimal. Simply take each group separately and convert it to its symbol equivalent, remembering that the hexadecimal system uses the letters A–F for the values 10–15. The result is a simple conversion, where all you have to remember is the value of 0000(BIN) through 1111(BIN) (i.e., the values 0–15).

Rule 6: To convert from binary to hexadecimal, separate the number into groups of four digits, starting from the right, and convert each group to its symbol equivalent.

Just as before, you can easily go from hexadecimal to binary to reversing the procedure.

Rule 7: To convert from hexadecimal to binary, convert each digit to its binary equivalent.

Hexadecimal is useful because you can divide an 8- or 16-bit (digit) binary number into groups of four without having to add nonsignificant zeros. Also, the conversion to and from binary is easy.

In many microprocessors 8 bits of binary information represent the data. The "address" in memory that stores the data is usually a 16-bit number. For this reason, addresses are often expressed in their hexadecimal form. One side benefit of base 16 numbers is that they use fewer digits to express a quantity. My example took only two (AD).

It's Quiz Time

● As you look through the specifica-

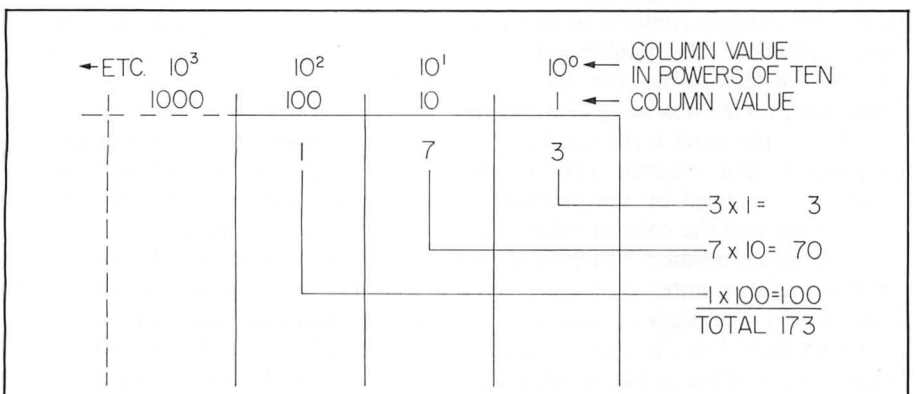


Fig. 1. A Dissected Decimal Number

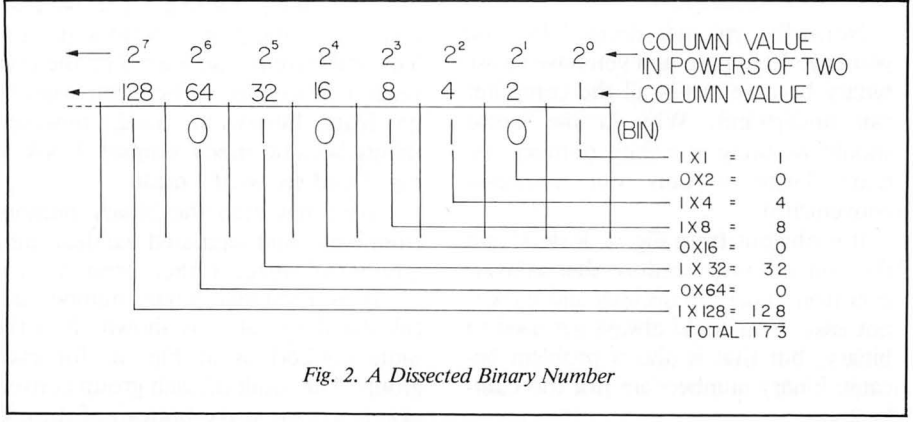
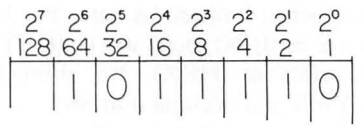


Fig. 2. A Dissected Binary Number



To Convert 94 to Binary

1. Divide 94 by highest column value possible.
   94 ÷ 64 = 1
   remainder = 30
2. Divide remainder by next lower column value.
   30 ÷ 32 = 0
   remainder = 30
3. Repeat Step 2 until remainder is zero.
   30 ÷ 16 = 1
   remainder = 14
   14 ÷ 8 = 1
   remainder = 6
   6 ÷ 4 = 1
   remainder = 2
   2 ÷ 2 = 1
   remainder = 0
4. Put zeros in any columns remaining.
5. Conversion is complete.

94 = 1011110 (BIN)

Fig. 3. A Number-Conversion Procedure

tions on computers, you will see "odd" numbers (in base 10 anyway) that won't look so odd in binary, octal, or hexadecimal. For example, when referring to 16K of memory, the 16 is decimal but the K stands for the quantity 1,024. Therefore, 16K memory means 16,384 bytes. Try converting the 1,024 to binary, octal, and hexadecimal using the conversion procedure, and you will find that the results don't look quite so odd.

- The quantity 1,024 is an exact power of two, so using the definition of powers you should be able to work out which power it is. Use your calculator or computer if you want.
- Other numbers that frequently show up in the computer world are 256 and 512. Try converting these to the other three bases to see why.
- To make sure you have followed all this, try converting the following numbers to the indicated equivalent.

15	to_____(BIN)
110011(BIN)	to_____(OCT)
255(OCT)	to_____(BIN)
FF(HEX)	to_____(BIN)
123(HEX)	to_____(decimal)
C2E(HEX)	to_____(OCT)

On the last one you could either use the conversion procedure or convert the number to binary and then back to octal using Rules 7 and 5.

- If you got this far without tearing your hair out, try the bonus question. To convince yourself you are now a number expert, try converting 173 to the same quantity in base 4. See Table 1 for the answers to the quiz questions.

Summary

I hope that you now have a better understanding of number systems—and that you do not also have a headache. Number systems are not all that bad, but as is the case with anything, practice makes perfect. So by all means try other test examples, and you will eventually acquire an instinctive feeling for questions such as how big FFFF(HEX) is and whether 123 is larger or smaller than 123(OCT). Soon you will be able to look at 11010(BIN) and say without hesitation, "That's 32(OCT)." ■

Address correspondence to Colin J. Stearman, 143 Ash St., Hopkinton, MA 01748.

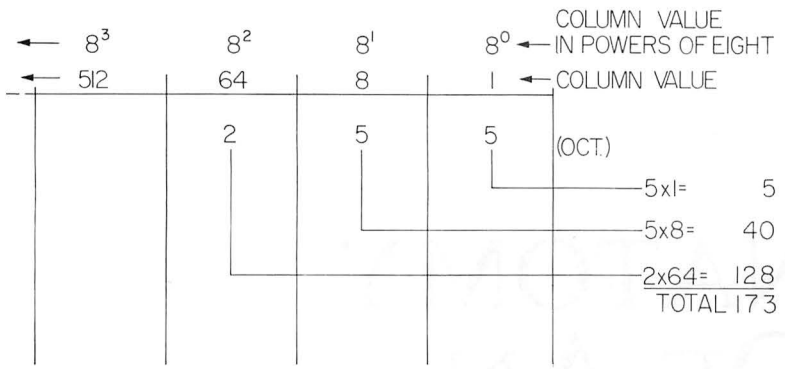


Fig. 4. An Octal Number Dissected

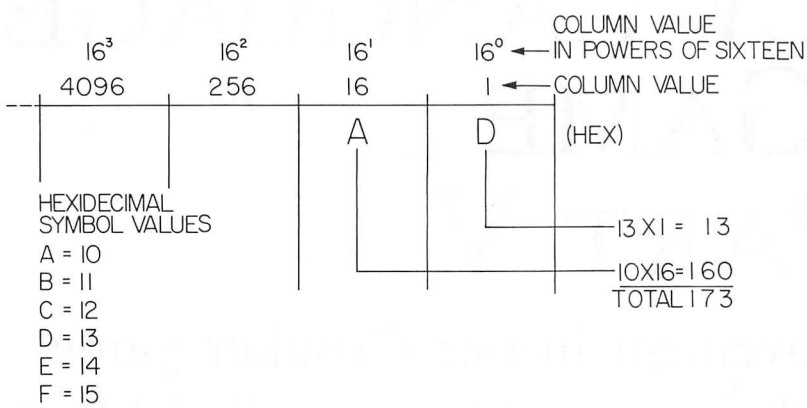


Fig. 6. A Hexadecimal Number Dissected

Raising Numbers to Powers

The explanation in the main article of the meaning of "raising a number to a power" is the commonly given one. But you might have wondered what raising to a power of zero means, and why I said the answer is one. You might also have noticed that any number raised to the power of zero is one. Trying to use the article definition here becomes confusing.

A much better definition is this: A power of a number is the number of times the value one must be multiplied by that number.

In the article example of 10 to the power of two, the new definition works because $1 \times 10 \times 10$ is still 100. But now, when 10 is raised to the power of zero, you do not multiply one by any ten, so the result is still one. Any number raised to the power of zero will therefore be one. ■

- 1024 = 10000000000(BIN) = 2000(OCT) = 400(HEX)
- 1024 = 2 to the power of 10
- 256 = 100000000(BIN) = 400(OCT) = 100(HEX)
- 512 = 1000000000(BIN) = 10000(OCT) = 200(HEX)
- 15 = 1111(BIN)
  110011(BIN) = 63(OCT)
  255(OCT) = 11111111(BIN)
  FF(HEX) = 11111111(BIN)
  123(HEX) = 291(decimal)
  C2E(HEX) = 6456(OCT)
- 173 = 2231(BASE 4)

Table 1. Quiz Answers

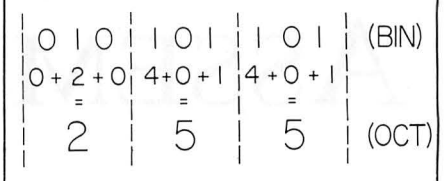


Fig. 5. Converting Binary to Octal

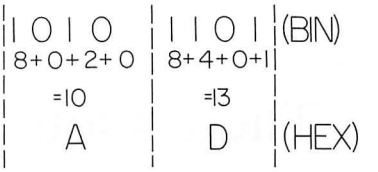
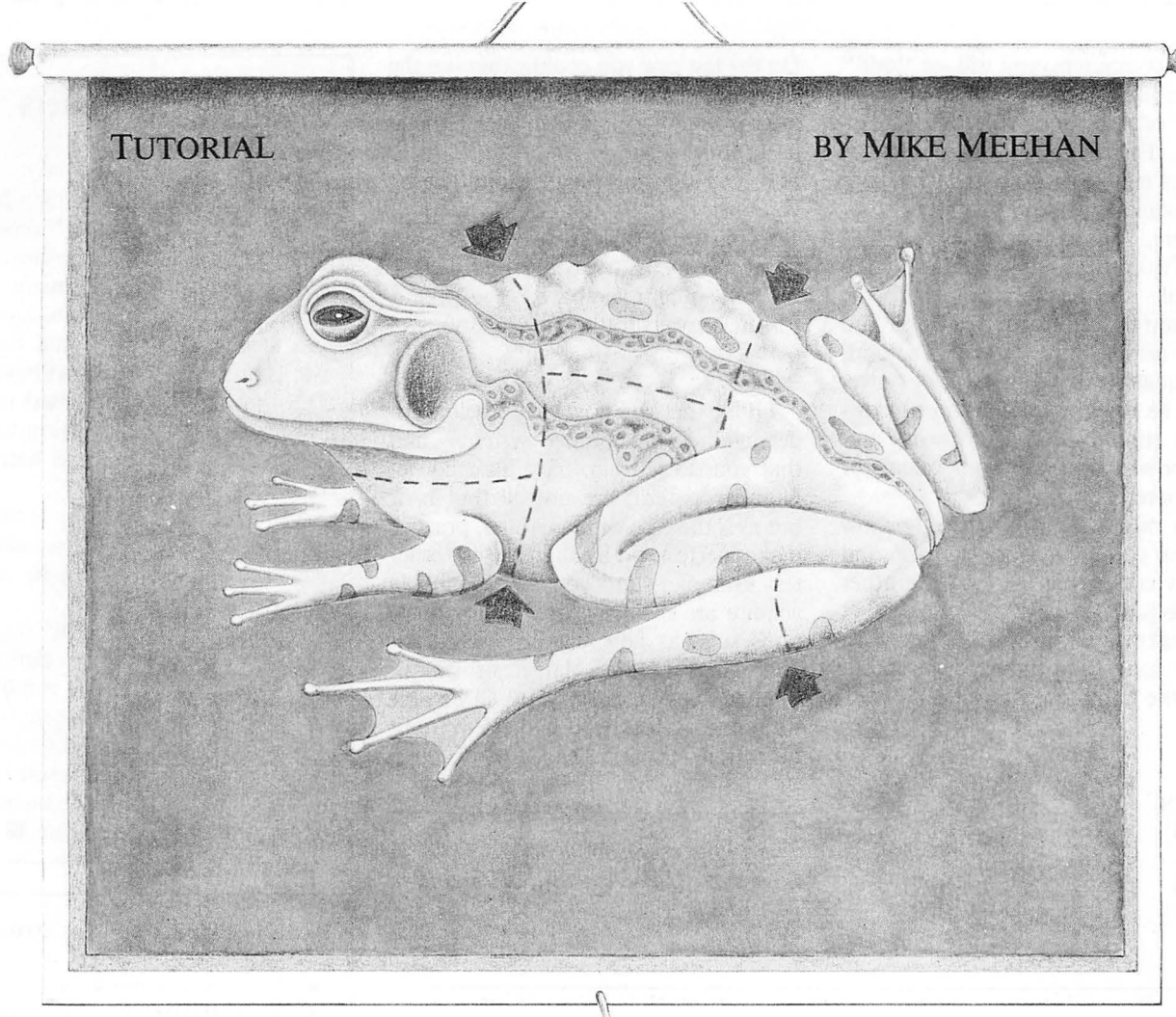


Fig. 7. Hexadecimal-to-Binary Conversion



Annie Gusman

ANATOMY OF AN ASSEMBLY-LANGUAGE GAME PART V

There's lots of movement in our Croaker game, and it's all done with just one routine called Move.

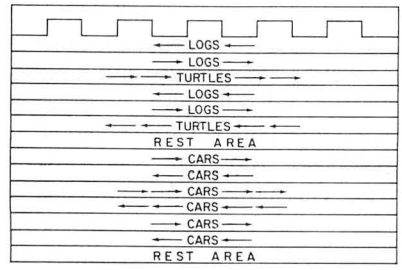


Fig. 1. Directions of Objects on the Screen. Double Arrows Represent Double Speed.

Fig. 2. Example of One Set of Graphics Being Moved to the Right One Place.

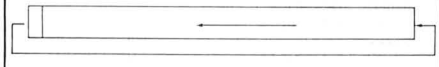


Fig. 3. Example of One Set of Graphics Being Moved to the Left One Place.

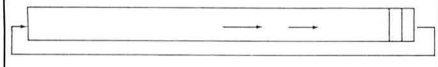


Fig. 4. Example of One Set of Graphics Being Moved to the Right Two Places (Double Speed)

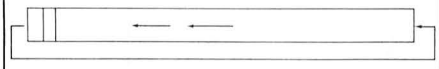


Fig. 5. Example of One Set of Graphics Being Moved to the Left Two Places (Double Speed)

Croaker is full of movement, and all of it, with the exception of the frog itself, is done with one particular subroutine called Move. This month's anatomy lesson takes a close look at this subroutine.

Move is in charge of moving the logs, turtles, and cars. It also keeps track of the timer function and makes the turtles disappear underwater at intervals.

As usual, the Program Listing starts by redefining some of the previously defined labels that are needed by Part V. It then begins the movement of one set of logs. (A set is one horizontal row.) Figure 1 shows that there are four sets of logs, two sets of turtles, and six sets of cars. Each set will move as a block separate from the rest of the screen. You can also see from Fig. 1 that the blocks will move in different directions, and some will move at double speed.

The routine moves one set of graphics by taking the leftmost or rightmost (depending on the direction the set is moving) vertical row of 10 bytes and storing their values in upper memory. It then moves the rest of the set to the right or left and restores the vertical row to the side opposite the one from which it was taken. Figures 2 and 3 show examples of this.

Some of the sets move at double speed and have a slightly different configuration when moved. In these, the two leftmost or two rightmost vertical rows of 10 bytes are stored in upper memory and the rest of the set is moved two places right or left. The two vertical rows are then restored to the side opposite the one from which they were taken. Figures 4 and 5 are examples of this.

Because the frog moves along with the logs and turtles, after each set of logs and turtles is moved, the right/left value and the value of the frog position must be changed appropriately. Also, because the frog remains still when it is sitting on the road, the routine must check to see if the frog is sitting on a set of cars before that set is moved. If it is, the frog must disappear before the cars are moved and then return to the screen after the move is completed.

The routine moves the top set first and proceeds downward. The first two sets of logs are moved and then the routine checks to see if it has to do anything to the first set of turtles. First, it checks the value in the timer function. If the timer contains one of the selected values, it checks location 16294 to see if it contains a 0 or 255. It will contain a 255 if the set of turtles has already been put underwater during this time interval. The routine won't put a set of turtles underwater twice in the same interval.

Location 16294 is reset to zero when the timer is decremented. The routine then checks location 16297 to see if the set of turtles is in the proper position to be put underwater. If it passes all this, a value of 11 is stored in 16294 and the set of turtles is replaced by a set of turtles disappearing underwater.

The frog is redrawn so it won't simply disappear if it is sitting on this set. Each time this routine is run again, location 16294 is decremented. When 16294 reaches eight, a blank area replaces the turtles disappearing underwater. When it reaches three, a set of turtles disappearing underwater (only in this case reappearing) replaces the blank area. Finally, when it reaches one, the regular set of turtles returns to the screen and the frog is redrawn. A value of 255 is then stored in location 16294 so the set won't disappear again in this time interval.

The turtles are moved at double speed and the next two sets of logs are moved. The routine then determines if anything has to be done with the second set of turtles. This functions the same as the first set only location 16293 is used in place of 16294 and location 16295 is used in place of 16297. This set of turtles is then moved at double speed.

The routine next moves the cars on the screen. Remember that if the frog is sitting in the set to be moved, it must disappear and be placed back on the screen at the end of the move. Also, refer to Fig. 1 to tell which sets are moved at double speed.

At the end of the listing, the routine updates the positions of the sets of turtles. It also updates the timer. Location 16299 starts out with a value of 20 and is decremented each time the routine is run. When it reaches zero, 16294 and

System Requirements

32K RAM
Extended Color Basic
Editor/Assembler

16293 are reset to zero and the routine calculates the screen position of the timer and decrements the timer on the screen. It then checks to see if the timer is equal to zero, and if it is, the routine increments location 16227 (which automatically kills the frog as you will see next month) and returns. If not, it checks to see if the timer is equal to three and if it is, it sounds a warning alarm. It then decrements the timer, resets location 16299 to 20, and returns.

Next month I'll finish the series and give the final part of Croaker along with a patch for those of you with disk drives. Questions and comments are still welcome. Please remember to send a self-addressed, stamped envelope for responses. ■

Address correspondence to Mike Meehan, 1300 Fairfield Drive, Clearwater, FL 33546.

Program Listing. Croaker's Move Routine

```
                        00100 *************************
                        00110 ********CROAKER**********
                        00120 *****BY: MIKE MEEHAN*****
                        00130 *****COPYRIGHT 1983******
                        00140 *COLOR HORIZONS SOFTWARE*
                        00150 *************************
                        00160 ********PART FIVE********
                        00170 *************************
2E71                    00180          ORG     $2E71
                0020    00190 NUM      EQU     $20     LABELS DEFINED
                26CA    00200 PWTUR    EQU     $26CA   IN PREVIOUS
                2738    00210 PFROG    EQU     $2738   PROGRAMS THAT
                26EC    00220 PBLK     EQU     $26EC   ARE NEEDED
                2686    00230 PTUR     EQU     $2686   FOR PART 5
                2714    00240 PBLK2    EQU     $2714
                256F    00250 MUSIC    EQU     $256F   ROUTINE TO
                1EF6    00260 SONG2    EQU     $1EF6   MOVE ALL
2E71 108E 3FAF  00270 MOVE     LDY     #16303  OBSTICLES ON
2E75 8E   0A00  00280          LDX     #2560   THE SCREEN
2E78 A6   84    00290 LOOP14   LDA     ,X
2E7A A7   A0    00300          STA     ,Y+
2E7C 1F   10    00310          TFR     X,D
2E7E C3   0020  00320          ADDD    #32
2E81 1F   01    00330          TFR     D,X
2E83 108C 3FB9  00340          CMPY    #16313
2E87 26   EF    00350          BNE     LOOP14
2E89 8E   0A00  00360          LDX     #2560
2E8C C6   00    00370          LDB     #0
2E8E F7   3FC3  00380          STB     16323
2E91 EC   01    00390 LOOP15   LDD     1,X
2E93 ED   80    00400          STD     ,X+
2E95 F6   3FC3  00410          LDB     16323
2E98 CB   01    00420          ADDB    #1
2E9A F7   3FC3  00430          STB     16323
2E9D C1   20    00440          CMPB    #32
2E9F 26   F0    00450          BNE     LOOP15
2EA1 8C   0B3E  00460          CMPX    #2880-2
2EA4 24   09    00470          BHS     EQU1
2EA6 C6   00    00480          LDB     #0
2EA8 F7   3FC3  00490          STB     16323
2EAB A6   80    00500          LDA     ,X+
2EAD 20   E2    00510          BRA     LOOP15
2EAF 8E   0A1F  00520 EQU1     LDX     #2560+31
2EB2 108E 3FAF  00530          LDY     #16303
2EB6 A6   A0    00540 LOOP16   LDA     ,Y+
2EB8 A7   84    00550          STA     ,X
2EBA 1F   10    00560          TFR     X,D
2EBC C3   0020  00570          ADDD    #32
2EBF 1F   01    00580          TFR     D,X
2EC1 108C 3FB9  00590          CMPY    #16313
2EC5 26   EF    00600          BNE     LOOP16
2EC7 BE   3FA3  00610          LDX     16291
2ECA 8C   0A00  00620          CMPX    #2560
2ECD 25   10    00630          BLO     NEQU24
2ECF 8C   0A1F  00640          CMPX    #2560+31
2ED2 22   0B    00650          BHI     NEQU24
2ED4 7A   3FA2  00660          DEC     16290
2ED7 BE   3FA3  00670          LDX     16291
2EDA A6   82    00680          LDA     ,-X
2EDC BF   3FA3  00690          STX     16291
2EDF 108E 3FAF  00700 NEQU24   LDY     #16303
2EE3 8E   0B5F  00710          LDX     #2880+31
2EE6 A6   84    00720 LOOP17   LDA     ,X
2EE8 A7   A0    00730          STA     ,Y+
2EEA 1F   10    00740          TFR     X,D
2EEC C3   0020  00750          ADDD    #32
2EEF 1F   01    00760          TFR     D,X
2EF1 108C 3FB9  00770          CMPY    #16313
2EF5 26   EF    00780          BNE     LOOP17
2EF7 8E   0C7F  00790          LDX     #3200-1
2EFA C6   00    00800          LDB     #0
2EFC F7   3FC3  00810          STB     16323
2EFF EC   1F    00820 LOOP18   LDD     -1,X
2F01 ED   84    00830          STD     ,X
2F03 A6   82    00840          LDA     ,-X
2F05 F6   3FC3  00850          LDB     16323
2F08 CB   01    00860          ADDB    #1
2F0A F7   3FC3  00870          STB     16323
2F0D C1   20    00880          CMPB    #32
2F0F 26   EE    00890          BNE     LOOP18
2F11 8C   0B40  00900          CMPX    #2880
2F14 23   09    00910          BLS     EQU2
2F16 C6   00    00920          LDB     #0
2F18 F7   3FC3  00930          STB     16323
2F1B A6   82    00940          LDA     ,-X
2F1D 20   E0    00950          BRA     LOOP18
2F1F 8E   0B40  00960 EQU2     LDX     #2880
2F22 108E 3FAF  00970          LDY     #16303
2F26 A6   A0    00980 LOOP19   LDA     ,Y+
2F28 A7   84    00990          STA     ,X
2F2A 1F   10    01000          TFR     X,D
2F2C C3   0020  01010          ADDD    #32
2F2F 1F   01    01020          TFR     D,X
2F31 108C 3FB9  01030          CMPY    #16313
2F35 26   EF    01040          BNE     LOOP19
2F37 BE   3FA3  01050          LDX     16291
2F3A 8C   0B40  01060          CMPX    #2880
2F3D 25   10    01070          BLO     NEQU25
2F3F 8C   0B5F  01080          CMPX    #2880+31
2F42 22   0B    01090          BHI     NEQU25
2F44 7C   3FA2  01100          INC     16290
2F47 BE   3FA3  01110          LDX     16291
2F4A A6   80    01120          LDA     ,X+
2F4C BF   3FA3  01130          STX     16291
2F4F B6   3FAC  01140 NEQU25   LDA     16300
2F52 81   12    01150          CMPA    #18
2F54 27   12    01160          BEQ     EQU15
2F56 81   0E    01170          CMPA    #14
2F58 27   0E    01180          BEQ     EQU15
2F5A 81   0C    01190          CMPA    #12
2F5C 27   0A    01200          BEQ     EQU15
2F5E 81   08    01210          CMPA    #8
2F60 27   06    01220          BEQ     EQU15
2F62 81   04    01230          CMPA    #4
2F64 27   02    01240          BEQ     EQU15
2F66 20   24    01250          BRA     NEQU17
2F68 B6   3FA6  01260 EQU15    LDA     16294
2F6B 81   00    01270          CMPA    #0
2F6D 26   1D    01280          BNE     NEQU17
2F6F BE   3FA9  01290          LDX     16297
2F72 8C   0C81  01300          CMPX    #3201
2F75 26   15    01310          BNE     NEQU17
2F77 86   0B    01320          LDA     #11
2F79 B7   3FA6  01330          STA     16294
2F7C BE   3FA9  01340          LDX     16297
2F7F BD   26CA  01350          JSR     PWTUR
2F82 10BE 3FA0  01360          LDY     16288
2F86 BD   2738  01370          JSR     PFROG
2F89 7E   2FD5  01380          JMP     EQU14
2F8C B6   3FA6  01390 NEQU17   LDA     16294
2F8F 81   00    01400          CMPA    #0
2F91 1027 0040  01410          LBEQ    EQU14
2F95 81   FF    01420          CMPA    #255
2F97 1027 003A  01430          LBEQ    EQU14
2F9B 4A         01440          DECA
2F9C B7   3FA6  01450          STA     16294
2F9F 81   08    01460          CMPA    #8
2FA1 26   09    01470          BNE     NEQU18
2FA3 BE   3FA9  01480          LDX     16297
2FA6 BD   26EC  01490          JSR     PBLK
2FA9 7E   2FD5  01500          JMP     EQU14
2FAC B6   3FA6  01510 NEQU18   LDA     16294
2FAF 81   03    01520          CMPA    #3
2FB1 26   09    01530          BNE     NEQU19
2FB3 BE   3FA9  01540          LDX     16297
2FB6 BD   26CA  01550          JSR     PWTUR
2FB9 7E   2FD5  01560          JMP     EQU14
2FBC B6   3FA6  01570 NEQU19   LDA     16294
2FBF 81   01    01580          CMPA    #1
2FC1 26   12    01590          BNE     EQU14
2FC3 BE   3FA9  01600          LDX     16297
2FC6 BD   2686  01610          JSR     PTUR
2FC9 10BE 3FA0  01620          LDY     16288
2FCD BD   2738  01630          JSR     PFROG
2FD0 86   FF    01640          LDA     #255
2FD2 B7   3FA6  01650          STA     16294
2FD5 108E 3FAF  01660 EQU14    LDY     #16303
2FD9 8E   0C9E  01670          LDX     #3200+30
2FDC EC   84    01680 LOOP20   LDD     ,X
2FDE ED   A1    01690          STD     ,Y++
2FE0 1F   10    01700          TFR     X,D
2FE2 C3   0020  01710          ADDD    #32
2FE5 1F   01    01720          TFR     D,X
2FE7 108C 3FC3  01730          CMPY    #16323
2FEB 26   EF    01740          BNE     LOOP20
2FED 8E   0DBE  01750          LDX     #3520-2
2FF0 C6   00    01760          LDB     #0
2FF2 F7   3FC3  01770          STB     16323
2FF5 EC   1E    01780 LOOP21   LDD     -2,X
2FF7 ED   84    01790          STD     ,X
2FF9 A6   83    01800          LDA     ,--X
2FFB F6   3FC3  01810          LDB     16323
2FFE CB   01    01820          ADDB    #1
```

Listing continued

Listing continued

```
3000 F7 3FC3 01830          STB   16323
3003 C1 0F   01840          CMPB  #15
3005 26 EE   01850          BNE   LOOP21
3007 8C 0C82 01860          CMPX  #3200+2
300A 23 09   01870          BLS   EQU3
300C C6 00   01880          LDB   #0
300E F7 3FC3 01890          STB   16323
3011 A6 83   01900          LDA   ,--X
3013 20 E0   01910          BRA   LOOP21
3015 8E 0C80 01920 EQU3     LDX   #3200
3018 108E 3FAF 01930        LDY   #16303
301C BC A1   01940 LOOP22   LDD   ,Y++
301E ED 84   01950          STD   ,X
3020 1F 10   01960          TFR   X,D
3022 C3 0020 01970          ADDD  #32
3025 1F 01   01980          TFR   D,X
3027 108C 3FC3 01990        CMPY  #16323
302B 26 EF   02000          BNE   LOOP22
302D BE 3FA3 02010          LDX   16291
3030 8C 0C80 02020          CMPX  #3200
3033 25 13   02030          BLO   NEQU26
3035 8C 0C9F 02040          CMPX  #3200+31
3038 22 0E   02050          BHI   NEQU26
303A 7C 3FA2 02060          INC   16290
303D 7C 3FA2 02070          INC   16290
3040 BE 3FA3 02080          LDX   16291
3043 A6 81   02090          LDA   ,X++
3045 BF 3FA3 02100          STX   16291
3048 108E 3FAF 02110 NEQU26 LDY   #16303
304C 8E 0DC0 02120          LDX   #3520
304F A6 84   02130 LOOP23   LDA   ,X
3051 A7 A0   02140          STA   ,Y+
3053 1F 10   02150          TFR   X,D
3055 C3 0020 02160          ADDD  #32
3058 1F 01   02170          TFR   D,X
305A 108C 3FB9 02180        CMPY  #16313
305E 26 EF   02190          BNE   LOOP23
3060 8E 0DC0 02200          LDX   #3520
3063 C6 00   02210          LDB   #0
3065 F7 3FC3 02220          STB   16323
3068 EC 01   02230 LOOP24   LDD   1,X
306A ED 80   02240          STD   ,X+
306C F6 3FC3 02250          LDB   16323
306F CB 01   02260          ADDB  #1
3071 F7 3FC3 02270          STB   16323
3074 C1 20   02280          CMPB  #32
3076 26 F0   02290          BNE   LOOP24
3078 8C 0EFE 02300          CMPX  #3840-2
307B 24 09   02310          BHS   EQU4
307D C6 00   02320          LDB   #0
307F F7 3FC3 02330          STB   16323
3082 A6 80   02340          LDA   ,X+
3084 20 E2   02350          BRA   LOOP24
3086 8E 0DDF 02360 EQU4     LDX   #3520+31
3089 108E 3FAF 02370        LDY   #16303
308D A6 A0   02380 LOOP25   LDA   ,Y+
308F A7 84   02390          STA   ,X
3091 1F 10   02400          TFR   X,D
3093 C3 0020 02410          ADDD  #32
3096 1F 01   02420          TFR   D,X
3098 108C 3FB9 02430        CMPY  #16313
309C 26 EF   02440          BNE   LOOP25
309E BE 3FA3 02450          LDX   16291
30A1 8C 0DC0 02460          CMPX  #3520
30A4 25 10   02470          BLO   NEQU27
30A6 8C 0DDF 02480          CMPX  #3520+31
30A9 22 0B   02490          BHI   NEQU27
30AB 7A 3FA2 02500          DEC   16290
30AE BE 3FA3 02510          LDX   16291
30B1 A6 82   02520          LDA   ,-X
30B3 BF 3FA3 02530          STX   16291
30B6 108E 3FAF 02540 NEQU27 LDY   #16303
30BA 8E 0F1F 02550          LDX   #3840+31
30BD A6 84   02560 LOOP26   LDA   ,X
30BF A7 A0   02570          STA   ,Y+
30C1 1F 10   02580          TFR   X,D
30C3 C3 0020 02590          ADDD  #32
30C6 1F 01   02600          TFR   D,X
30C8 108C 3FB9 02610        CMPY  #16313
30CC 26 EF   02620          BNE   LOOP26
30CE 8E 103F 02630          LDX   #4160-1
30D1 C6 00   02640          LDB   #0
30D3 F7 3FC3 02650          STB   16323
30D6 BC 1F   02660 LOOP27   LDD   -1,X
30D8 ED 84   02670          STD   ,X
30DA A6 82   02680          LDA   ,-X
30DC F6 3FC3 02690          LDB   16323
30DF CB 01   02700          ADDB  #1
30E1 F7 3FC3 02710          STB   16323
30E4 C1 20   02720          CMPB  #32
30E6 26 EE   02730          BNE   LOOP27
30E8 8C 0F00 02740          CMPX  #3840
30EB 23 09   02750          BLS   EQU5
30ED C6 00   02760          LDB   #0
30EF F7 3FC3 02770          STB   16323
30F2 A6 82   02780          LDA   ,-X
30F4 20 E0   02790          BRA   LOOP27
30F6 8E 0F00 02800 EQU5     LDX   #3840
30F9 108E 3FAF 02810        LDY   #16303
30FD A6 A0   02820 LOOP28   LDA   ,Y+
30FF A7 84   02830          STA   ,X
3101 1F 10   02840          TFR   X,D
3103 C3 0020 02850          ADDD  #32
3106 1F 01   02860          TFR   D,X
3108 108C 3FB9 02870        CMPY  #16313
310C 26 EF   02880          BNE   LOOP28
310E BE 3FA3 02890          LDX   16291
3111 8C 0F00 02900          CMPX  #3840
3114 25 10   02910          BLO   NEQU28
3116 8C 0F1F 02920          CMPX  #3840+31
3119 22 0B   02930          BHI   NEQU28
311B 7C 3FA2 02940          INC   16290
311E BE 3FA3 02950          LDX   16291
3121 A6 80   02960          LDA   ,X+
3123 BF 3FA3 02970          STX   16291
3126 B6 3FAC 02980 NEQU28   LDA   16300
3129 81 10   02990          CMPA  #16
312B 27 12   03000          BEQ   EQU17
312D 81 0B   03010          CMPA  #11
312F 27 0E   03020          BEQ   EQU17
3131 81 09   03030          CMPA  #9
3133 27 0A   03040          BEQ   EQU17
3135 81 06   03050          CMPA  #6
3137 27 06   03060          BEQ   EQU17
3139 81 03   03070          CMPA  #3
313B 27 02   03080          BEQ   EQU17
313D 20 24   03090          BRA   NEQU21
313F B6 3FA5 03100 EQU17    LDA   16293
3142 81 00   03110          CMPA  #0
3144 26 1D   03120          BNE   NEQU21
3146 BE 3FA7 03130          LDX   16295
3149 8C 1056 03140          CMPX  #4182
314C 26 15   03150          BNE   NEQU21
314E 86 0B   03160          LDA   #11
3150 B7 3FA5 03170          STA   16293
3153 BE 3FA7 03180          LDX   16295
3156 BD 26CA 03190          JSR   PWTUR
3159 10BE 3FA0 03200        LDY   16288
315D BD 2738 03210          JSR   PFROG
3160 7E 31AC 03220          JMP   EQU16
3163 B6 3FA5 03230 NEQU21   LDA   16293
3166 81 FF   03240          CMPA  #255
3168 1027 0040 03250        LBEQ  EQU16
316C 81 00   03260          CMPA  #0
316E 1027 003A 03270        LBEQ  EQU16
3172 4A      03280          DECA
3173 B7 3FA5 03290          STA   16293
3176 81 08   03300          CMPA  #8
3178 26 09   03310          BNE   NEQU22
317A BE 3FA7 03320          LDX   16295
317D BD 26EC 03330          JSR   PBLK
3180 7E 31AC 03340          JMP   EQU16
3183 B6 3FA5 03350 NEQU22   LDA   16293
3186 81 03   03360          CMPA  #3
3188 26 09   03370          BNE   NEQU23
318A BE 3FA7 03380          LDX   16295
318D BD 26CA 03390          JSR   PWTUR
3190 7E 31AC 03400          JMP   EQU16
3193 B6 3FA5 03410 NEQU23   LDA   16293
3196 81 01   03420          CMPA  #1
3198 26 12   03430          BNE   EQU16
319A BE 3FA7 03440          LDX   16295
319D BD 2686 03450          JSR   PTUR
31A0 10BE 3FA0 03460        LDY   16288
31A4 BD 2738 03470          JSR   PFROG
31A7 86 FF   03480          LDA   #255
31A9 B7 3FA5 03490          STA   16293
31AC 108E 3FAF 03500 EQU16  LDY   #16303
31B0 8E 1040 03510          LDX   #4160
31B3 EC 84   03520 LOOP29   LDD   ,X
31B5 ED A1   03530          STD   ,Y++
31B7 1F 10   03540          TFR   X,D
31B9 C3 0020 03550          ADDD  #32
31BC 1F 01   03560          TFR   D,X
31BE 108C 3FC3 03570        CMPY  #16323
31C2 26 EF   03580          BNE   LOOP29
31C4 8E 1040 03590          LDX   #4160
31C7 C6 00   03600          LDB   #0
31C9 F7 3FC3 03610          STB   16323
31CC EC 02   03620 LOOP30   LDD   2,X
31CE ED 81   03630          STD   ,X++
31D0 F6 3FC3 03640          LDB   16323
31D3 CB 01   03650          ADDB  #1
31D5 F7 3FC3 03660          STB   16323
31D8 C1 0F   03670          CMPB  #15
31DA 26 F0   03680          BNE   LOOP30
31DC 8C 117E 03690          CMPX  #4480-2
31DF 24 07   03700          BHS   EQU6
31E1 C6 00   03710          LDB   #0
31E3 F7 3FC3 03720          STB   16323
31E6 20 E4   03730          BRA   LOOP30
31E8 8E 105E 03740 EQU6     LDX   #4160+30
31EB 108E 3FAF 03750        LDY   #16303
31EF EC A1   03760 LOOP31   LDD   ,Y++
31F1 ED 84   03770          STD   ,X
31F3 1F 10   03780          TFR   X,D
31F5 C3 0020 03790          ADDD  #32
31F8 1F 01   03800          TFR   D,X
31FA 108C 3FC3 03810        CMPY  #16323
31FE 26 EF   03820          BNE   LOOP31
3200 BE 3FA3 03830          LDX   16291
3203 8C 1040 03840          CMPX  #4160
3206 25 13   03850          BLO   NEQU29
3208 8C 105F 03860          CMPX  #4160+31
320B 22 0E   03870          BHI   NEQU29
320D 7A 3FA2 03880          DEC   16290
3210 7A 3FA2 03890          DEC   16290
3213 BE 3FA3 03900          LDX   16291
3216 A6 83   03910          LDA   ,--X
3218 BF 3FA3 03920          STX   16291
321B BE 3FA3 03930 NEQU29   LDX   16291
321E 8C 1400 03940          CMPX  #5120
3221 25 08   03950          BLO   NEQU30
3223 8C 141F 03960          CMPX  #5120+31
3226 22 03   03970          BHI   NEQU30
3228 BD 2714 03980          JSR   PBLK2
322B 108E 3FAF 03990 NEQU30 LDY   #16303
322F 8E 141F 04000          LDX   #5120+31
3232 A6 84   04010 LOOP32   LDA   ,X
3234 A7 A0   04020          STA   ,Y+
3236 1F 10   04030          TFR   X,D
3238 C3 0020 04040          ADDD  #32
323B 1F 01   04050          TFR   D,X
323D 108C 3FB9 04060        CMPY  #16313
3241 26 EF   04070          BNE   LOOP32
3243 8E 1521 04080          LDX   #5440-31
3246 C6 00   04090          LDB   #0
3248 F7 3FC3 04100          STB   16323
324B BC 1F   04110 LOOP33   LDD   -1,X
324D ED 84   04120          STD   ,X
324F A6 82   04130          LDA   ,-X
3251 F6 3FC3 04140          LDB   16323
3254 C3 0001 04150          ADDD  #1
3257 F7 3FC3 04160          STB   16323
325A C1 20   04170          CMPB  #32
325C 26 ED   04180          BNE   LOOP33
325E 8C 1400 04190          CMPX  #5120
3261 23 09   04200          BLS   EQU7
3263 C6 00   04210          LDB   #0
3265 F7 3FC3 04220          STB   16323
3268 A6 82   04230          LDA   ,-X
326A 20 DF   04240          BRA   LOOP33
326C 8E 1400 04250 EQU7     LDX   #5120
326F 108E 3FAF 04260        LDY   #16303
3273 A6 A0   04270 LOOP34   LDA   ,Y+
3275 A7 84   04280          STA   ,X
3277 1F 10   04290          TFR   X,D
3279 C3 0020 04300          ADDD  #32
327C 1F 01   04310          TFR   D,X
327E 108C 3FB9 04320        CMPY  #16313
3282 26 EF   04330          BNE   LOOP34
3284 BE 3FA3 04340          LDX   16291
3287 8C 1400 04350          CMPX  #5120
328A 25 0C   04360          BLO   NEQU31
328C 8C 141F 04370          CMPX  #5120+31
328F 22 07   04380          BHI   NEQU31
3291 10BE 3FA0 04390        LDY   16288
3295 BD 2738 04400          JSR   PFROG
3298 BE 3FA3 04410 NEQU31   LDX   16291
329B 8C 1540 04420          CMPX  #5440
329E 25 08   04430          BLO   NEQU32
32A0 8C 155F 04440          CMPX  #5440+31
32A3 22 03   04450          BHI   NEQU32
32A5 BD 2714 04460          JSR   PBLK2
32A8 108E 3FAF 04470 NEQU32 LDY   #16303
32AC 8E 1540 04480          LDX   #5440
32AF A6 84   04490 LOOP35   LDA   ,X
32B1 A7 A0   04500          STA   ,Y+
32B3 1F 10   04510          TFR   X,D
32B5 C3 0020 04520          ADDD  #32
32B8 1F 01   04530          TFR   D,X
32BA 108C 3FB9 04540        CMPY  #16313
32BE 26 EF   04550          BNE   LOOP35
32C0 8E 1540 04560          LDX   #5440
32C3 C6 00   04570          LDB   #0
32C5 F7 3FC3 04580          STB   16323
32C8 EC 01   04590 LOOP36   LDD   1,X
32CA ED 80   04600          STD   ,X+
32CC F6 3FC3 04610          LDB   16323
32CF CB 01   04620          ADDB  #1
32D1 F7 3FC3 04630          STB   16323
32D4 C1 20   04640          CMPB  #32
32D6 26 F0   04650          BNE   LOOP36
32D8 8C 167E 04660          CMPX  #5760-2
32DB 24 09   04670          BHS   EQU8
32DD C6 00   04680          LDB   #0
32DF F7 3FC3 04690          STB   16323
32E2 A6 80   04700          LDA   ,X+
32E4 20 E2   04710          BRA   LOOP36
32E6 8E 155F 04720 EQU8     LDX   #5440+31
32E9 108E 3FAF 04730        LDY   #16303
32ED A6 A0   04740 LOOP37   LDA   ,Y+
32EF A7 84   04750          STA   ,X
32F1 1F 10   04760          TFR   X,D
32F3 C3 0020 04770          ADDD  #32
32F6 1F 01   04780          TFR   D,X
32F8 108C 3FB9 04790        CMPY  #16313
32FC 26 EF   04800          BNE   LOOP37
32FE BE 3FA3 04810          LDX   16291
3301 8C 1540 04820          CMPX  #5440
3304 25 0C   04830          BLO   NEQU33
3306 8C 155F 04840          CMPX  #5440+31
3309 22 07   04850          BHI   NEQU33
330B 10BE 3FA0 04860        LDY   16288
330F BD 2738 04870          JSR   PFROG
3312 BE 3FA3 04880 NEQU33   LDX   16291
3315 8C 1680 04890          CMPX  #5760
3318 25 08   04900          BLO   NEQU34
331A 8C 169F 04910          CMPX  #5760+31
331D 22 03   04920          BHI   NEQU34
331F BD 2714 04930          JSR   PBLK2
3322 108E 3FAF 04940 NEQU34 LDY   #16303
3326 8E 169E 04950          LDX   #5760+30
3329 BC 84   04960 LOOP38   LDD   ,X
332B ED A1   04970          STD   ,Y++
332D 1F 10   04980          TFR   X,D
332F C3 0020 04990          ADDD  #32
3332 1F 01   05000          TFR   D,X
3334 108C 3FC3 05010        CMPY  #16323
```

Listing continued

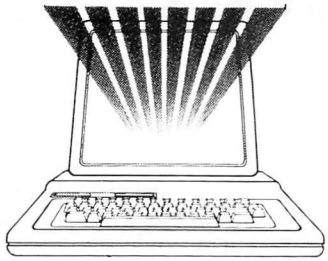
Listing continued

```
3338 26   EF      05020          BNE    LOOP38
333A 8E   17BE    05030          LDX    #6080-2
333D C6   00      05040          LDB    #0
333F F7   3FC3    05050          STB    16323
3342 EC   1E      05060 LOOP39   LDD    -2,X
3344 ED   84      05070          STD    ,X
3346 A6   83      05080          LDA    ,--X
3348 F6   3FC3    05090          LDB    16323
334B CB   01      05100          ADDB   #1
334D F7   3FC3    05110          STB    16323
3350 C1   0F      05120          CMPB   #15
3352 26   EE      05130          BNE    LOOP39
3354 8C   1682    05140          CMPX   #5760+2
3357 23   09      05150          BLS    EQU9
3359 C6   00      05160          LDB    #0
335B F7   3FC3    05170          STB    16323
335E A6   83      05180          LDA    ,--X
3360 20   E0      05190          BRA    LOOP39
3362 8E   1680    05200 EQU9     LDX    #5760
3365 108E 3FAF    05210          LDY    #16303
3369 EC   A1      05220 LOOP40   LDD    ,Y++
336B ED   84      05230          STD    ,X
336D 1F   10      05240          TFR    X,D
336F C3   0020    05250          ADDD   #32
3372 1F   01      05260          TFR    D,X
3374 108C 3FC3    05270          CMPY   #16323
3378 26   EF      05280          BNE    LOOP40
337A BE   3FA3    05290          LDX    16291
337D 8C   1680    05300          CMPX   #5760
3380 25   0C      05310          BLO    NEQU35
3382 8C   169F    05320          CMPX   #5760+31
3385 22   07      05330          BHI    NEQU35
3387 10BE 3FA0    05340          LDY    16288
338B BD   2738    05350          JSR    PFROG
338E BE   3FA3    05360 NEQU35   LDX    16291
3391 8C   17C0    05370          CMPX   #6080
3394 25   08      05380          BLO    NEQU36
3396 8C   17DF    05390          CMPX   #6080+31
3399 22   03      05400          BHI    NEQU36
339B BD   2714    05410          JSR    PBLK2
339E 108E 3FAF    05420 NEQU36   LDY    #16303
33A2 8E   17C0    05430          LDX    #6080
33A5 EC   84      05440 LOOP41   LDD    ,X
33A7 ED   A1      05450          STD    ,Y++
33A9 1F   10      05460          TFR    X,D
33AB C3   0020    05470          ADDD   #32
33AE 1F   01      05480          TFR    D,X
33B0 108C 3FC3    05490          CMPY   #16323
33B4 26   EF      05500          BNE    LOOP41
33B6 8E   17C0    05510          LDX    #6080
33B9 C6   00      05520          LDB    #0
33BB F7   3FC3    05530          STB    16323
33BE EC   02      05540 LOOP42   LDD    2,X
33C0 ED   81      05550          STD    ,X++
33C2 F6   3FC3    05560          LDB    16323
33C5 CB   01      05570          ADDB   #1
33C7 F7   3FC3    05580          STB    16323
33CA C1   0F      05590          CMPB   #15
33CC 26   F0      05600          BNE    LOOP42
33CE 8C   18FE    05610          CMPX   #6400-2
33D1 24   07      05620          BHS    EQU10
33D3 C6   00      05630          LDB    #0
33D5 F7   3FC3    05640          STB    16323
33D8 20   E4      05650          BRA    LOOP42
33DA 8E   17DE    05660 EQU10    LDX    #6080+30
33DD 108E 3FAF    05670          LDY    #16303
33E1 EC   A1      05680 LOOP43   LDD    ,Y++
33E3 ED   84      05690          STD    ,X
33E5 1F   10      05700          TFR    X,D
33E7 C3   0020    05710          ADDD   #32
33EA 1F   01      05720          TFR    D,X
33EC 108C 3FC3    05730          CMPY   #16323
33F0 26   EF      05740          BNE    LOOP43
33F2 BE   3FA3    05750          LDX    16291
33F5 8C   17C0    05760          CMPX   #6080
33F8 25   0C      05770          BLO    NEQU42
33FA 8C   17DF    05780          CMPX   #6080+31
33FD 22   07      05790          BHI    NEQU42
33FF 10BE 3FA0    05800          LDY    16288
3403 BD   2738    05810          JSR    PFROG
3406 BE   3FA3    05820 NEQU42   LDX    16291
3409 8C   1900    05830          CMPX   #6400
340C 25   08      05840          BLO    NEQU37
340E 8C   191F    05850          CMPX   #6400+31
3411 22   03      05860          BHI    NEQU37
3413 BD   2714    05870          JSR    PBLK2
3416 108E 3FAF    05880 NEQU37   LDY    #16303
341A 8E   191F    05890          LDX    #6400+31
341D A6   84      05900 LOOP44   LDA    ,X
341F A7   A0      05910          STA    ,Y+
3421 1F   10      05920          TFR    X,D
3423 C3   0020    05930          ADDD   #32
3426 1F   01      05940          TFR    D,X
3428 108C 3FB9    05950          CMPY   #16313
342C 26   EF      05960          BNE    LOOP44
342E 8E   1A3F    05970          LDX    #6720-1
3431 C6   00      05980          LDB    #0
3433 F7   3FC3    05990          STB    16323
3436 EC   1F      06000 LOOP45   LDD    -1,X
3438 ED   84      06010          STD    ,X
343A A6   82      06020          LDA    ,-X
343C F6   3FC3    06030          LDB    16323
343F CB   01      06040          ADDB   #1
3441 F7   3FC3    06050          STB    16323
3444 C1   20      06060          CMPB   #32
3446 26   EE      06070          BNE    LOOP45
3448 8C   1900    06080          CMPX   #6400
344B 23   09      06090          BLS    EQU11
344D C6   00      06100          LDB    #0
344F F7   3FC3    06110          STB    16323
3452 A6   82      06120          LDA    ,-X
3454 20   E0      06130          BRA    LOOP45
3456 8E   1900    06140 EQU11    LDX    #6400
3459 108E 3FAF    06150          LDY    #16303
345D A6   A0      06160 LOOP46   LDA    ,Y+
345F A7   84      06170          STA    ,X
3461 1F   10      06180          TFR    X,D
3463 C3   0020    06190          ADDD   #32
3466 1F   01      06200          TFR    D,X
3468 108C 3FB9    06210          CMPY   #16313
346C 26   EF      06220          BNE    LOOP46
346E BE   3FA3    06230          LDX    16291
3471 8C   1900    06240          CMPX   #6400
3474 25   0C      06250          BLO    NEQU39
3476 8C   191F    06260          CMPX   #6400+31
3479 22   07      06270          BHI    NEQU39
347B 10BE 3FA0    06280          LDY    16288
347F BD   2738    06290          JSR    PFROG
3482 BE   3FA3    06300 NEQU39   LDX    16291
3485 8C   1A40    06310          CMPX   #6720
3488 25   08      06320          BLO    NEQU40
348A 8C   1A5F    06330          CMPX   #6720+31
348D 22   03      06340          BHI    NEQU40
348F BD   2714    06350          JSR    PBLK2
3492 108E 3FAF    06360 NEQU40   LDY    #16303
3496 8E   1A40    06370          LDX    #6720
3499 A6   84      06380 LOOP47   LDA    ,X
349B A7   A0      06390          STA    ,Y+
349D 1F   10      06400          TFR    X,D
349F C3   0020    06410          ADDD   #32
34A2 1F   01      06420          TFR    D,X
34A4 108C 3FB9    06430          CMPY   #16313
34A8 26   EF      06440          BNE    LOOP47
34AA 8E   1A40    06450          LDX    #6720
34AD C6   00      06460          LDB    #0
34AF F7   3FC3    06470          STB    16323
34B2 EC   01      06480 LOOP48   LDD    1,X
34B4 ED   80      06490          STD    ,X+
34B6 F6   3FC3    06500          LDB    16323
34B9 CB   01      06510          ADDB   #1
34BB F7   3FC3    06520          STB    16323
34BE C1   20      06530          CMPB   #32
34C0 26   F0      06540          BNE    LOOP48
34C2 8C   1B7E    06550          CMPX   #7040-2
34C5 24   09      06560          BHS    EQU12
34C7 C6   00      06570          LDB    #0
34C9 F7   3FC3    06580          STB    16323
34CC A6   80      06590          LDA    ,X+
34CE 20   E2      06600          BRA    LOOP48
34D0 8E   1A5F    06610 EQU12    LDX    #6720+31
34D3 108E 3FAF    06620          LDY    #16303
34D7 A6   A0      06630 LOOP49   LDA    ,Y+
34D9 A7   84      06640          STA    ,X
34DB 1F   10      06650          TFR    X,D
34DD C3   0020    06660          ADDD   #32
34E0 1F   01      06670          TFR    D,X
34E2 108C 3FB9    06680          CMPY   #16313
34E6 26   EF      06690          BNE    LOOP49
34E8 BE   3FA3    06700          LDX    16291
34EB 8C   1A40    06710          CMPX   #6720
34EE 25   0C      06720          BLO    NEQU41
34F0 8C   1A5F    06730          CMPX   #6720+31
34F3 22   07      06740          BHI    NEQU41
34F5 10BE 3FA0    06750          LDY    16288
34F9 BD   2738    06760          JSR    PFROG
34FC FC   3FA9    06770 NEQU41   LDD    16297
34FF C3   0002    06780          ADDD   #2
3502 1083 0CA0    06790          CMPD   #3232
3506 25   03      06800          BLO    NEQU15
3508 83   0020    06810          SUBD   #32
350B FD   3FA9    06820 NEQU15   STD    16297
350E FC   3FA7    06830          LDD    16295
3511 83   0002    06840          SUBD   #2
3514 1083 1040    06850          CMPD   #4160
3518 24   03      06860          BHS    NEQU16
351A C3   0020    06870          ADDD   #32
351D FD   3FA7    06880 NEQU16   STD    16295
3520 B6   3FAB    06890          LDA    16299
3523 4A           06900          DECA
3524 81   00      06910          CMPA   #0
3526 26   4E      06920          BNE    NEQU8
3528 86   00      06930          LDA    #0
352A B7   3FA6    06940          STA    16294
352D B7   3FA5    06950          STA    16293
3530 86   01      06960          LDA    #1
3532 F6   3FAC    06970          LDB    16300
3535 3D           06980          MUL
3536 C3   06A3    06990          ADDD   #1699
3539 1F   01      07000          TFR    D,X
353B 86   55      07010          LDA    #85
353D A7   84      07020          STA    ,X
353F 1F   10      07030          TFR    X,D
3541 C3   0020    07040          ADDD   #32
3544 1F   01      07050          TFR    D,X
3546 86   55      07060          LDA    #85
3548 A7   84      07070          STA    ,X
354A 1F   10      07080          TFR    X,D
354C C3   0020    07090          ADDD   #32
354F 1F   01      07100          TFR    D,X
3551 86   55      07110          LDA    #85
3553 A7   84      07120          STA    ,X
3555 B6   3FAC    07130          LDA    16300
3558 81   00      07140          CMPA   #0
355A 26   04      07150          BNE    NEQU10
355C 7C   3F63    07160          INC    16227
355F 39           07170          RTS
3560 81   03      07180 NEQU10   CMPA   #3
3562 26   0C      07190          BNE    NEQU9
3564 8E   1EF6    07200          LDX    #SONG2
3567 BF   3FE7    07210          STX    $3FE7
356A BD   256F    07220          JSR    MUSIC
356D B6   3FAC    07230          LDA    16300
3570 4A           07240 NEQU9    DECA
3571 B7   3FAC    07250          STA    16300
3574 86   20      07260          LDA    #NUM
3576 B7   3FAB    07270 NEQU8    STA    16299
3579 39           07280          RTS
          0000    07290          END
00000 TOTAL ERRORS

EQU1    2EAF
EQU10   33DA
EQU11   3456
EQU12   34D0
EQU14   2FD5
EQU15   2F68
EQU16   31AC
EQU17   313F
EQU2    2F1F
EQU3    3015
EQU4    3086
EQU5    30F6
EQU6    31E8
EQU7    326C
EQU8    32E6
EQU9    3362
LOOP14  2E78
LOOP15  2E91
LOOP16  2EB6
LOOP17  2EE6
LOOP18  2EFF
LOOP19  2F26
LOOP20  2FDC
LOOP21  2FF5
LOOP22  301C
LOOP23  304F
LOOP24  3068
LOOP25  308D
LOOP26  30BD
LOOP27  30D6
LOOP28  30FD
LOOP29  31B3
LOOP30  31CC
LOOP31  31EF
LOOP32  3232
LOOP33  324B
LOOP34  3273
LOOP35  32AF
LOOP36  32C8
LOOP37  32ED
LOOP38  3329
LOOP39  3342
LOOP40  3369
LOOP41  33A5
LOOP42  33BE
LOOP43  33E1
LOOP44  341D
LOOP45  3436
LOOP46  345D
LOOP47  3499
LOOP48  34B2
LOOP49  34D7
MOVE    2E71
MUSIC   256F
NEQU10  3560
NEQU15  350B
NEQU16  351D
NEQU17  2F8C
NEQU18  2FAC
NEQU19  2FBC
NEQU21  3163
NEQU22  3183
NEQU23  3193
NEQU24  2EDF
NEQU25  2F4F
NEQU26  3048
NEQU27  30B6
NEQU28  3126
NEQU29  321B
NEQU30  322B
NEQU31  3298
NEQU32  32A8
NEQU33  3312
NEQU34  3322
NEQU35  338E
NEQU36  339E
NEQU37  3416
NEQU39  3482
NEQU40  3492
NEQU41  34FC
NEQU42  3406
NEQU8   3576
NEQU9   3570
NUM     0020
PBLK    26EC
PBLK2   2714
PFROG   2738
PTUR    2686
PWTUR   26CA
SONG2   1EF6
```

END

SUPER SCREEN



- A big 51 character by 24 line screen.
- Full upper and lower case characters.
- Easily combine text with hi-res graphics.
- PRINT @ is completely functional on the big screen.
- The powerful ON ERROR GOTO is fully implemented.
- Auto-key repeat for greater keyboard convenience.
- Control codes for additional functions.
- Works with 16K, 32K or 64K computers.
- Available on disc or cassette.
- Works with extended and/or disc BASIC.

51 CHARACTERS BY 24 LINE DISPLAY

Super Screen is a powerful, machine language program that significantly upgrades the performance and usefulness of 16K or greater, Extended and Disc Basic Color Computers. The standard Color Computer display screen is totally inadequate for serious, personal or business applications so Super Screen replaces it with a brand new, 51 character wide by 24 line screen including full upper and lower case characters. Instead of a confusing checkerboard appearance, you now have true lower case letters along with a screen that is capable of displaying 1224 characters. The difference is startling! Your computer takes on new dimensions and can easily handle lines of text that were simply too long and complex to display on the old screen.

COMBINE TEXT WITH HI-RES GRAPHICS

You can now write truly professional looking programs that combine text with hi-res graphics. Super Screen allows you to create graphics displays with the Basic LINE, DRAW and CIRCLE statements and then notate the graphics with descriptive text. You can even use PRINT @ if you wish for greater programming convenience. Super Screen's versatility will amaze you.

PRINT @ IS FULLY IMPLEMENTED

The PRINT @ statement is a valuable asset to the programmer when formatting text on the screen. The standard Color Computer will report an error if you specify a location higher than 511 but Super Screen allows locations all the way to 1223! You get a big screen and a powerful formatting tool as well. Of course, Super Screen also supports the CLS command allowing you to clear the big screen using standard Basic syntax.

ON ERROR GOTO

That's right! Super Screen gives you a full implementation of ON ERROR GOTO including the ERR and ERL functions. Now you can trap errors and take corrective action to prevent crashed programs and lost data using the same standard syntax as other computers. The ON ERROR GOTO capability overcomes a serious deficiency of Color Computer Basic and greatly improves your capability to handle sophisticated tasks. All well written, 'user friendly' programs use error trapping techniques and yours can too! Now that's power!

AUTO KEY REPEAT

No more frustration as you edit a long line in your Basic program; just hold the space bar down and automatically step to the desired position in the line. Need a line of asterisks? Hold the key down and auto repeat will give them to you. Those of you who spend many hours at your keyboard will appreciate this outstanding addition to Super Screen's long list of impressive capabilities.

CONTROL CODES FOR ADDITIONAL FUNCTIONS

Super Screen recognizes several special control code characters that allow selection of block or underline, solid or blinking cursor and other functions. You can 'Home Up' the cursor or you may erase from the cursor to the end of a line or to the end of the screen just like many other computers. These special codes give you an extra dimension of versatility and convenience that put Super Screen in a class by itself.

AND MORE GOOD NEWS...

Super Screen comes with complete, well detailed instructions and is available on cassette or disc. It adjusts automatically to any 16K or greater, Extended or Disc Basic Color Computer or TDP-100 and uses only 2K of memory in addition to the screen memory reserved during power up. Guaranteed to be the most frequently used program in your software library...once you use it, you won't be without it! Super Screen's low price will really please you; only $29.95 on cassette or $32.95 on disc!

64K Memory Expansion Kit

All parts and complete instructions

$64.95

NEW! SUPER BUG

Mark Data Products SUPER BUG is a powerful, relocatable machine code monitor program for your Coco. If you are a beginner, the program and documentation are an indispensable training aid, helping you to gain a better understanding of your Color Computer and machine code programming. If you are an accomplished computerist, SUPER BUG's capabilities, versatility and convenience will prove invaluable during programming and debugging.

SUPER BUG offers so many outstanding features that we are unable to list them all in this limited space. hex and alpha numeric memory display, modify, search and test; full printer support with baud rate and line feed select; up to 220 breakpoints; mini object code disassembler; 64K mode setup; decimal, hex and ascii code conversion routines and extensive documentation. Only $29.95 on cassette or $32.95 on disc.

ORDER ENTRY SYSTEM

The Mark Data Products sales order processing system will give a fast, efficient means to enter orders, print shipping papers and invoices, prepare sales reprots, and monitor receivables. The system automatically enhances the monitor screen to a 51 character by 24 line display. 32K of memory is required along with an 80-column printer, and one or more disc drives.

The MDP order entry system is a family of programs which operate interactively by means of a "menu" selection scheme. Up to 900 products may be defined and a single disc system can hold over 600 transactions. When the operator selects a task to be performed, the computer loads a program designed to handle that task from the system disc. The system disc contains all of the programs required to create, update and maintain data files and prepare the necessary paperwork including shipping and invoice forms, daily sales reports, a monthly (or other period) sales report and a receivables report.

The MDP system:

- Is accurate, user friendly and simple to use.
- Is easy to customize for specific user requirements.
- Produces a traceable invoice.
- Handles receivables as well as closed orders.
- In capable of future expandability.

This accounting software equals or exceeds higher priced packages for other computers and includes a detailed operating manual. For just $99.95.

ACCOUNTING SYSTEM

The Mark Data Products accounting system is ideal for the small businessman needing a fast, efficient means to process income and expenses, prepare detailed reports and maintain most of the information required at tax time. The system is a family of programs which operate by means of a "menu" selection scheme. When the operator selects a task to perform, the computer loads a program designed to handle that task from the system disc. The system disc contains all of the programs required to create, update and maintain data files and prepare the necessary accounting reports including a transaction journal, a P&L or income report, an interim or trial balance and a balance sheet.

Up to 255 separate accounts may be defined and a single disc system can hold over 1,400 transactions. This system automatically enhances the monitor screen to a 51 character by 24 line display. 32K of memory is required along with an 80-column printer and one or more disc drives.

The MDP system:

- Is accurate, user friendly and simple to use.
- Is easy to customize for specific user requirements.
- Immediately updates the chart of accounts.
- Provides an audit trail.
- Includes end of period procedures.
- Is capable of future expandability.

This order entry software equals or exceeds higher priced packages for other computers and includes a detailed operating manual. For just $99.95.

IMPORTANT NEW BOOKS

"Your Color Computer" by Doug Mosher. Over 300 pages of detailed information–A CoCo encyclopedia $16.95.

"Programming the 6809" by Rodnay Zaks and William Labiak. One of the best 6809 machine language texts available—required reference material. $15.95.

WE STOCK SOFTLAW PRODUCTS

The VIP WRITER Text Processor is rated tops by Rainbow, Hot CoCo and Color Computer Magazine. After evaluation we rate it tops too. **Disc $59.95.**

MD

Mark Data Products

24001 ALICIA PKWY., NO. 207 • MISSION VIEJO, CA 92691 • (714) 768-1551

All Orders: Please add $2.00 shipping and handling in the continental U.S. All others, add air shipping and $3.00 handling. California residents add 6% sales tax. Foreign orders please remit U.S. funds. Software authors—Contact us for exciting program marketing details. We accept MasterCard and VISA. Distributed in Canada by Kelly Software

68Ø9 On Line

BULLETIN-BOARD SYSTEMS

by Bobby Ballard

Last month I covered the larger information utilities and the costs involved. This month the cost drops as I discuss bulletin-board systems, or BBSs as they are known among the hackers of America. BBSs are generally free to access although some require passwords or small sign-on fees.

The cost of accessing these systems is much less, but your phone bill could inflate if you live in an area that charges for local calls or if you get the long-distance bug for calling out-of-state bulletin boards. You will have to call long distance if boards do not exist in your local area. In this case you can obtain one of the new long-distance services like MCI or Sprint to save money. If there are no boards in your area, you may wish to start one and become the first SYSOP (system operator) in your city or state.

A bulletin-board service or system is much like a bulletin board you see in the grocery store, school, or in the classified section of a newspaper. You can leave messages for others who access the system, copy (download) programs posted in the board, or just check out local activities of particular interest. Many BBSs center the activities around the host computer, an interest group, or a computer club known as a user's group. If you see a listing for TUG, for example, it indicates the system is operated by a Tandy user's group. A CP/MUG would be a CP/M user's group. This helps guide you to the right boards before you waste a phone call.

A bulletin-board system consists of several parts that, to put it simply, boil down to a computer hooked to a phone. The computer is usually fully dedicated to servicing the subscribers of the board, although some boards are used for business during the day and dedicated to the BBS in the evening.

The host computer, until recently, required disk drives to operate as a BBS. There are programs just coming out that run a simple BBS on 64K with cassette. The tape handles only the simplest of functions, and most systems you call are multiple-drive disk running the latest in BBS software.

Also, most BBSs are run with an auto-answer modem. The software for a BBS is as varied as the systems, and many SYSOPs have developed their own programs or modified existing programs to suit their needs. Although the systems vary, there are some standards that will help you access the information or messages you need.

Knowing something about the software you are calling saves time and cost while on line. Since many boards are listed by and sometimes named for their software you can know what to expect in the way of menus and system protocol. Also, the software indicates which machine the board supports, and therefore you can find programs for that particular machine there.

For example, a TBBS system probably runs on a Tandy machine, and you will find programs for a variety of Radio Shack computers. It is unlikely that you will find much Atari support on a TBBS board, and you won't find Tandy support on an ABBS. This is not a rule, but is generally true and I wouldn't waste a long-distance call on the wrong system. The major software for Radio Shack machines many times ends or begins with 80.

The names of some of the major programs that run on Radio Shack machines include TBBS, Forum-80, Greene Machine, RACS, Trade, MCMS, Bullet-80, and RATS. For the CoCo, you will find Connection-80, Color-80, CoCo Board, and various original software. Some of the software listed for all machines include PMS, Access, DYM, Conference Tree, Networks, and CBBS. By the way, DYM stands for Dial Your Match and is found on BBSs specializing in sexually oriented messages and matching. I've never called a DYM, but I understand you must answer questions that are explicit, your answers are made public, and you must answer in order to gain a password for access.

No matter what machine the BBS operates on, you can probably access that board and enjoy many of its features. CompuServe also operates a National Bulletin Board that you access by typing GO PCS-30 at any prompt and selecting option 3 at the ensuing menu. Remember, you're paying access charges with CompuServe; most other boards are free.

The first time you sign on to a BBS, you are asked to answer a few questions to help the SYSOP keep his board secure. Access to certain areas is often restricted until your registration is complete. Unfortunately, many SYSOPs have had to adopt this policy in order to keep the irresponsible from crashing the system and destroying files.

Within a few days, your application is processed, and by your second call the entire board is open for your use. In the meantime, many features of most BBSs are open on your first call. The system also asks you about the terminal or computer you are using in order to set the proper communication protocol. So check the documentation that came with your communications software, and when calling new boards keep this information handy for quick reference. After some time, you'll have your own protocol settings memorized.

For example, the default protocol settings for my software are 300 baud, even parity, 7 data bits, full duplex, normal delay, and pass line feed. Subsequent calls will not require you to answer these questions again. Some systems ask for your terminal width each time you call, especially systems that are accessed by a wide variety of machines and software systems.

The first time you sign on to a new board, it is a good idea to check the new-user section for tips on using the

system and further information on protocol for downloading and uploading using smart-terminal software. If you have a printer, you can copy the protocol settings for future reference. Also, record your password somewhere for quick retrieval on subsequent calls. SYSOPs do not appreciate finding your name on the new-user list 20 times because you forgot your password.

Many boards limit the time of your call in order to allow others a chance to call. If your favorite BBS has a time limit or limit to the number of times you may call per day, then it is usually with good reason: high demand! Please, observe the rules of the board. One final word: The messages you place are public and most SYSOPs do not tolerate profanity or sexual, racial, religious, and off-color remarks. Keep your language clean or your messages will just get deleted. In a future column, I will give you a form to help keep track of your calls, passwords, and other information to make your telecommunicating easier.

Connection-80 is the system I am most familiar with, and I will use it as an example of some of the menu commands. Even if the system in your area is not a Connection-80 program, you will still find that the overall look is similar to other BBSes.

The menu has many options and includes more than just message bases or programs to download. Many boards sell computer-related merchandise as part of the services provided. You can order with your credit card or COD in many cases, and the items are sent out UPS or through the U.S. postal system. On Connection-80 the menu includes but is not limited to the following:

<B> Boards or Bulletins
<U> User Log
<C> Chat with SYSOP
<M> Merchandise Section
<O> Order Merchandise
<H> Help (this list)
<D> Download
<I> Information (system)
<N> New User Information
<E> Elapsed Time on the System
<T> Terminate Call
<F> Feedback (to SYSOP)
<A> Alter Protocol or Terminal Width
<X> Expert User Mode

Selecting any of the letters in brackets leads you to another menu or information that is completely spelled out as in the above example.

If you select B, another menu appears allowing you to choose between sending or receiving a message:

<R>ead Messages
<L>eave Messages
<S>can Messages
<K>ill Messages

The host computer also indicates how many messages are on the system and the highest-number message accessed on your last call. If the host does not automatically mark your messages for retrieval, you will need to scan for messages left to you. With Connection-80 and many other systems the messages are marked until you've read them once. If you wish to read a message again on a later call, you can select R and the host gives you another choice of read options. The menu may appear as:

<F>orward
<R>everse
<S>elected
<N>ew Messages
<M>arked
<N>umber Message

Pressing S sets up another menu for searching the <T>o, <F>rom, and <S>ubject fields where you can search for specific information or messages. In this case you could search for messages to your name and mark them for retrieval. If you know the number of the message you wish to read, press N and the host will prompt you for the message number, search for it, and print it on your screen.

In the future, I will cover in more detail the advanced features like downloading, uploading, and the expert mode where even more time is saved. If you wish to discover more about telecommunicating, read *The Small Computer Connection*, by Neil L. Shapiro, McGraw-Hill Book Company, 1221 Avenue of the Americas, New York, NY 10020. Table 1 lists some of the CoCo boards that might be of interest to you. The list is not complete by any means.

If you would like more listings of BBSes for the CoCo or other computers, check out the *Bulletin Board Directory of North America,* by Christopher Fisher (P.O. Box 4150, Beach Station, Vero Beach, FL 32964-4150) or *The Computer Phone Book*, by Mike Cane and published by Plume. Published eight times a year, *Plumb* is a small magazine dedicated to personal telecommunicating. You can contact them at Riverside Data Inc., P.O. Box 300, Harrods Creek, KY 40027.

"In the future, I will cover in more detail the advanced features..."

The list in Table 1 is not an endorsement of the boards, and I have not called all of them. They are there for your own edification. Some of the boards may not exist any more, and you can find other numbers of BBSes on other boards and continue from there. Have fun telecommunicating for now and give a few boards a try. ■

Address correspondence to Bobby Ballard, 1207 Eighth Ave. 4-R, Brooklyn, NY 11215.

CA	
LA Color Exchange	213-563-7727
Color-80	408-773-6809
6809 Morning Star BBS	707-257-1485
Color Corner	714-350-2668
CoCo Corner	805-687-9400
Rainbow #5	408-984-7937
FL	
Colorburst	305-525-1192
Color-80	904-264-0335
Dr. D's CoCo Corner	904-456-7195
GA	
CoCo Board II	404-378-4410
MD	
Color-80	301-599-1726
MA	
Color-80	617-646-6809
MO	
TBBS	816-358-6222
NH	
80 Micro's BBS Express	603-924-6985
Magazine-80	603-924-7920
NJ	
CoCoMat	609-468-3844
CoCo Board	201-572-0617
Meadowlands CoCo	201-773-8265
Limericks BBS	201-572-0617
Connection-80	201-667-2504
NY	
Rainbow #1	212-441-3755
Rainbow #2	212-441-3766
Rainbow #3	212-441-5719
Rainbow #4	212-441-5907
CoCo Nest	212-423-4623
Colorama	516-277-1285
Color Channel	516-783-7582
Connection-80	516-588-5836
CoCo Nest OS-9	516-249-3449
OH	
CoCo-Nut Tree	216-788-7910
OK	
TBBS	405-722-6809
TN	
TBBS	615-842-6809
TX	
International Color BBS	214-657-8147
Color Connection	512-285-5028
Commnet-80	817-767-5847
VT	
TBBS	802-862-7023
WV	
21st Century Connection	304-925-3338
Color-80	304-599-0760

Table 1. CoCo-Related BBSes

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

First Class Permit No. 73 Peterborough, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

CW Communications/Peterborough
instant CoCo
Attn: Debbie Bourgault
80 Pine Street
Peterborough, N.H. 03458

HOT PROGRAMS IN AN INSTANT

Finally! Great programs you can load and run in seconds. You'll save hours of keyboarding time and build a terrific library of useful programs for as little as $10.00 a month.

Get the best programs from HOT CoCo magazine... ready to run...for pennies. All the directions or help you need is in your copy of HOT CoCo.

instant CoCo gives you ...*action-packed games—have your own video game entertainment center!

*useful applications—programs for education, business, home and hobby, to delight the whole family.

*utilities—your routine tasks will be taken care of like magic plus you'll be expanding the capabilities of your computer.

Order **instant CoCo** cassettes by the month. Every cassette guarantees you ten to fifteen great programs for only $11.47.*

Save money! A full twelve month subscription is yours for only $99.97*. That's only $8.33 per cassette.

You'll want the "Best of 83."

Over 33 programs from the 1983 issues of HOT CoCo magazine. You'll have a valuable library...for only-$16.47*...less than .45 cents per program.

Order your subscription to **instant CoCo** today!

Mail the coupon below or call TOLL FREE 1-800-258-5473 and use your Visa, Mastercard or American Express. In New Hampshire call 1-924-9471.

instant CoCo lets you start having fun right away...almost like magic!

YES! I want programs in an instant... I want instant CoCo

☐Please send me____copies of the "Best of '83" at $16.47 each.

☐Please send me ____copies of this month's issue at $11.47 each.

☐Please sign me up for a one year subscription beginning with this month's issue at $99.97

*Price includes postage and handling. Foreign Air Mail please add an additional 45¢ per cassette or $25 per subscription. US funds drawn on US banks ONLY.

☐CHECK/MO ☐MC ☐VISA ☐AE

Card# _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4-6 weeks delivery.

instant CoCo • 80 Pine Street • Peterborough, NH 03458

IC8410

The DOSsier

TSC PRECOMPILER, O-PAK, AND MISCHIEF

by Scott Norman

Don't say I told you this, but Apple—of all companies—has produced a good thing for Color Computer owners—those of us who use video monitors, anyway. It's the Apple IIc Monitor, a neat little nine-incher that was introduced along with the IIc computer last April.

It has a green phosphor, which you may not be crazy about, and very high resolution, which should make you feel a whole lot better. Best of all, there is a monitor stand, shaped like a squared-off uppercase U turned on its side, which wraps around the computer to put the screen at the right position for comfortable viewing. The CoCo and the IIc are roughly similar in size and shape, you see.

Although the CoCo in my office is fitted out with a 13-inch Zenith, I find the little Apple display just fine in the closer confines of my den. If you're in the market for a baseband monitor, you should check this one out; it's not cheap, but it's really nice if you spend much time with a word processor or spreadsheet. By the way, the video-driver circuits sold by Computerware and other vendors work perfectly well with it.

This doesn't have much to do with advanced operating systems, except that it would be a shame to lumber along without a good display for those 51-by-24 and 80-by-25 screens of ours.

Meet the TSC Precompiler

In last month's column, I presented a simple budget-management program as a vehicle for discussing TSC (Technical Systems Consultants) Extended Basic, a workhorse among the languages supported by Flex.

The program, which I call SEP, extrapolates my history of spending (on an R&D project, for example) and compares the extrapolation with my budget for the current fiscal year. The intent was to show off some of the features of TSC Extended Basic, including the simple "virtual array" method for managing random disk files.

At the same time, I promised to discuss TSC's Precompiler, which encourages Basic programmers to write more structured, more comprehensible code.

As for last month's program, just remember that it accepts up to 52 inputs, each consisting of the date plus the amount spent since the last report, does a couple of extrapolations to compare the endpoint of the current trend with the year's budget, and reports back to you.

To begin with, there are actually two Precompilers, one each for use with TSC's Basic and their Extended Basic. The Extended versions of the language and Precompiler (XPC) are the ones to have.

Let me be explicit about what XPC is not. As you should have guessed from its name, it is not a true compiler; it does not read high-level code and produce machine language for the computer to use. Instead, it is a program that reads an ASCII-format Extended Basic source file from disk and generates yet another kind of file that runs under Extended Basic.

The new file is generally smaller than the original and can only be run—not listed or edited. In fact, Extended Basic's RUN command is the only way to get XPC's output into the computer at all. LOAD has no effect.

This might be a nice feature for anyone worried about losing his or her programming secrets to prying eyes, but that alone scarcely justifies the purchase of XPC. In truth, TSC Extended Basic already has a COMPILE command that does a splendid conversion job for any conventional program you might want to throw at it.

The real beauty of XPC is that it accepts syntax that has many desirable features, but which is meaningless to TSC Extended Basic. In effect, it expands the vocabulary of the whole language.

A program intended for XPC can take on an appearance very different from more familiar Basics. There is no need to number every statement; the only lines that need labels are those to which the program might have to branch, such as the destinations of GOTOs, GOSUBs, and IF...THEN constructs. You don't need to label lines to which control is returned after the execution of a subroutine.

Notice that the "privileged" lines need labels, which are not necessarily numbers. They can be any series of valid characters, which includes letters, numbers, and the underscore symbol.

Take a look at the Program Listing for the new version of SEP. By one of its conventions, XPC treats any character string that starts in column 1 as a statement label; the statements themselves must begin in column 2 or beyond. Thus there are only a few labeled statements in the program: MENU, PICK, QUIT, NEW_FILE, and so on. You can use the labels in conventional ways within Basic statements; GOTO MENU is perfectly acceptable.

As it happens, I arranged for most of my labeled statements to be REMs. This was just to spread out the listing for legibility, though. I might just as easily have gone ahead with Basic code on the same line. In fact, sometimes I did; note QUIT and QUERY.

Think of XPC source code as nothing but a peculiar form of text file. Any Flex editor capable of producing ASCII material can prepare it. I used TED, the tiny editor included with Frank Hogg Lab's Flex, for the SEP source listing. I have also used Stylograph, and I'm sure that most other editors will work as well.

One that will not work, however, is the "editor" within TSC Extended Basic itself; the poor thing assumes

that any line that doesn't begin with a number is a command-mode statement and tries to execute it!

Using selective labeling in place of the indiscriminate numbering of statements cleans up XPC programs all by itself. The fact that variables can also have names of arbitrary length helps, too. EXPENDITURE and BUDGET are far more convenient than EX and BG, after all. The standard TSC Extended Basic conventions for specifying variable types are still in effect; COUNT% is an integer variable, REPLY$ a string.

The code quickly begins to take on the clean, structured look of Pascal and other modern languages. Somehow it seems natural to think of the statement labels as setting off blocks of code similar to Pascal procedures. It's possible to carry this a little further, using generous indentations to indicate the structure of FOR... NEXT loops and other segments of a program. You won't lose leading blanks when you list the source code.

The Precompiler uses the backslash character \ (ASCII 92) just before a carriage return to indicate that it will continue a logical line (a line of code) on the next physical line of a listing. I used quite a few \'s to make the SEP listing a little more legible. Since XPC treats the entered \ as a blank space on continued lines, there are a couple of restrictions on its use: You can't put it within variable names, and a blank appears if you use it within a string.

Continued lines can be up to 255 characters long, just like conventional Basic statements. XPC's tolerance of extended variable names can lead to wordiness in the code, so apply caution. The idea is to improve Basic by moving in the direction of Pascal and its derivatives—not Cobol.

Using It

After you have prepared some XPC source code, you must run it through XPC and then have TSC Extended Basic call up and run the "compiled" object code.

A major disadvantage of any compiled language is the amount of effort needed to correct the slightest error. XPC, however, has an option that sends it through a compilation without actually generating a binary file; in effect, it does nothing but check for syntax errors. You can also use this to obtain a source-code listing.

Other options let you suppress the source listing during compilation, write the source listing but suppress the logical line numbers normally assigned by XPC, and overwrite any existing binary file that has the same name as the new object file.

Compiled files are generally smaller than the source versions. The original code for SEP occupied most of 12 grans on the disk, while the compiled version came in at 9 grans.

To run a compiled program, you must load TSC Extended Basic and use the RUN "1.SEP.BAC" command. This is one of the few places in which you use quotation marks around file names in Flex. In case you're wondering, .BAC programs don't run any faster than regular TSC Extended Basic routines—another reminder that XPC is not really a Basic compiler. If it's raw speed you're after, you'll have to look elsewhere.

As far as I'm concerned, the Precompiler satisfies other needs. I think the degree to which it encourages more comprehensible code is worth a certain amount of awkwardness in the debugging process. I can see myself using it for programs that I'm likely to use in the same form for a long time, and that I might have to explain to others.

SEP has some more elaborate relatives that I expect to have around for quite a while, and it should be quite a comfort to pull out the source listings a few years hence and be able to dope them out. Some of us need all the help we can get.

Products and Vendors Mentioned in this Month's DOSsier

Apple IIc Monitor ($199)
Monitor Stand ($35.95)
Apple Computer
20525 Mariani Ave.
Cupertino, CA 95014
408-996-1010

TSC Extended Basic ($100)
XBasic Precompiler ($50)
O-Pak ($35.95)
Frank Hogg Laboratory
The Regency Tower, Suite 215
770 James St.
Syracuse, NY 13203
315-474-7856

A First Look at O-Pak

I've had Frank Hogg's O-Pak around for quite a while now, but so far I haven't been able to give it a proper going over. I have had a quick look, though, and should at least share my first impressions.

O-Pak is a three-part screen-enhancement and file-transfer utility that runs under CoCo OS-9. I suspect that its most used component will be HiRes, a program that lets you replace the standard 32-by-16 text display with something more appropriate for serious work.

I also suppose that the 51-by-24 character set supplied as the HiRes default will be the most common replacement, but there are other fonts: a total of 20, in fact. Some simulate the characters generated by common video terminals, but others are pretty exotic. There is even an APL character set; I bet it will be a while before the Color Computer can make sensible use of *that*.

Since HiRes places the computer in high-resolution graphics mode to draw letters, it lets you mix text and graphics on the screen. It also supports a number of advanced terminal functions such as control of the scroll rate, insertion or erasure of a complete line, and windowing. This refers to the software's ability to divide the screen into several independent regions, and is getting a lot of press these days.

Clever applications programmers can do some useful things with separately controllable pieces of video display. O-Pak can support up to eight windows, although I doubt that many people will want to push this limit; the windows are horizontal strips extending the full width of the screen, so you can't have independent rectangular displays side by side.

In case you fail to find something to your liking among the stock HiRes character sets, O-Pak includes a screen editor called CSEdit for generating your own. You don't exactly start from scratch; a clever visual menu lets you pick a character from the set HiRes that is currently running, and use it as a starting point for your development.

You work on an enlarged version of the character, while a second image at the normal size keeps you abreast of how your changes will affect the real thing.

The third component of O-Pak is the following X commands that manipulate disks recorded in Flex, OS-9, and RS DOS formats:

- XCOPY copies data in either direction between any two of the three formats.
- XDIR displays the directory of a Flex or RS DOS disk (OS-9 disks must use the regular DIR command).
- XDUMP produces a combined hex/ASCII dump of a file.
- XLIST lists any ASCII text file.

You must use special identifier prefixes for Flex and RS DOS files, and the format of each file name must be appropriate to the operating system with which you use it. Here, for example, is how to copy an RS-format text file on drive 1 over to the WRITINGS directory of an OS-9 disk on drive 0:

```
XCOPY RS%1:STUFF/TXT
/D0/WRITINGS/newstuff
```

You get used to this after a while, I suppose.

The X commands should become very popular. Who knows? An RS DOS word processor might yet wind up being the most popular source-code editor for Flex or OS-9 programming. ■

Contact Scott Norman at 8 Doris Road, Framingham, MA 01701.

```
      REM   ** "DOSSIER" TEST PROGRAM **
      REM   ** TSC PRECOMPILER VERSION *
      REM   ** SCOTT NORMAN, JUNE 1984 *
      REM
      PRINT CHR$(2): PRINT TAB(11) \
     "SPENDING EXTRAPOLATION PROGRAM"
     PRINT: PRINT TAB(3) \
     "THIS PROGRAM EXTRAPOLATES YOUR"
     PRINT "SPENDING TO YEAR-END BY"
     PRINT "TWO DIFFERENT METHODS."
      REM
MENU   REM
      PRINT TAB(12) "(1) START NEW FILE"
      PRINT
      PRINT TAB(12) "(2) ADD TO OLD FILE"
      PRINT
      PRINT TAB(12) "(3) CHECK SPENDING"
      PRINT
      PRINT TAB(12) "(4) QUIT"
      REM
PICK   REM
      PRINT
      INPUT "YOUR CHOICE"; CHOICE%
      IF CHOICE% <1 OR CHOICE% >4 \
         THEN GOTO PICK
      ON CHOICE% GOTO NEW_FILE, OLD_FILE, \
         OLD_FILE, QUIT
QUIT   END
      REM
NEW_FILE    REM
      PRINT: INPUT \
      "NAME OF NEW FILE"; FILENAME$
      PRINT: PRINT \
      "START OF FISCAL YEAR (MM,DD,YY)";
      GOSUB JULIAN_DATE
      PRINT: INPUT \
      "WHAT IS CURRENT BUDGET"; BUDGET
      OPEN "1."+FILENAME$ AS 1
      DIM #1, ARRAY(52,1)
      ARRAY(0,0)=JULDATE: ARRAY(0,1)=0
      REM (START OF YEAR)
      ARRAY(52,0)=JULDATE+365: \
        ARRAY(52,1)=BUDGET: REM (END OF YR.)
      COUNT%=0
      REM
RECORD_SPENDING     REM
      COUNT%=COUNT%+1
      PRINT:PRINT "EXPENDITURE DATE";
      GOSUB JULIAN_DATE
      INPUT "SPENT THIS PERIOD"; EXPENDITURE
      ARRAY(COUNT%,0)=JULDATE: \
        ARRAY(COUNT%,1)=EXPENDITURE
QUERY     INPUT "MORE DATA (Y/N)"; REPLY$
      IF REPLY$="Y" THEN RECORD_SPENDING
      IF REPLY$<>"N" THEN QUERY
      CLOSE 1
      GOTO MENU
      REM
OLD_FILE   REM
      PRINT: INPUT "NAME OF FILE"; FILENAME$
      OPEN "1."+FILENAME$ AS 1
      DIM #1, ARRAY(52,1)
      COUNT%=1
NEXT_SLOT    REM
      IF ARRAY(COUNT%,0)=0 \
         THEN COUNT%=COUNT%-1 \
         ELSE LOOK_AT_NEXT_SLOT
      ON CHOICE% GOTO DUMMY,RECORD_SPENDING,\
      CHECK_SPENDING, DUMMY
      REM
LOOK_AT_NEXT_SLOT    REM
      COUNT%=COUNT%+1: GOTO NEXT_SLOT
      REM
CHECK_SPENDING    REM
      REM  AVERAGE RATE EXTRAPOLATION
      SPENT=0
      FOR TERM%=1 TO COUNT%
        SPENT=SPENT+ARRAY(TERM%,1)
      NEXT TERM%
      INTERVAL=ARRAY(COUNT%,0)-ARRAY(0,0)
      SLOPE=SPENT/INTERVAL
      EXTRAP_1=SPENT+SLOPE*(ARRAY(52,0)- \
              ARRAY(COUNT%,0))
      REM
      REM MOST-RECENT-RATE EXTRAPOLATION
      LASTSPENT=ARRAY(COUNT%,1)
      LASTINTERVAL=ARRAY(COUNT%,0)- \
                   ARRAY(COUNT%-1,0)
      SLOPE=LASTSPENT/LASTINTERVAL
      EXTRAP_2=SPENT+SLOPE*(ARRAY(52,0)- \
              ARRAY(COUNT%,0))
      REM   PRINT THE RESULTS
      BUDGET=ARRAY(52,1)
      CLOSE 1
      VARIANCE_1=100*(BUDGET-EXTRAP_1)/BUDGET
      IF VARIANCE_1>0 THEN A$="BELOW" \
         ELSE A$="ABOVE"
      VAR1=ABS(VARIANCE_1)
      VARIANCE_2=100*(BUDGET-EXTRAP_2)/BUDGET
      IF VARIANCE_2>0 THEN B$="BELOW" \
         ELSE B$="ABOVE"
      VAR2=ABS(VARIANCE_2)
      PRINT CHR$(2)
      PRINT TAB(19) "** RESULTS **": PRINT
      DIGITS 8,2
      PRINT \
      "AVERAGE-RATE METHOD PREDICTS FINAL"
      PRINT"EXPENDITURES OF $";EXTRAP_1
      PRINT USING "OR ##.#",VAR1:: \
      PRINT "% ";A$;" BUDGET."
      PRINT
      PRINT \
      "MOST-RECENT-RATE METHOD PREDICTS"
      PRINT "EXPENDITURES OF $";EXTRAP_2
      PRINT USING "OR ##.#",VAR2:: \
      PRINT "% ";B$;" BUDGET."
      END
      REM
JULIAN_DATE   REM
      INPUT M%, D%, Y%: Y%=Y%+1900
      JULDATE=3.67E2*Y%- \
        INT(7*(Y%+INT((M%+9)/12))/4)+ \
        INT(275*M%/9)+D%
      RETURN
DUMMY   REM  JUST SATISFIES 'GOTO A,B,...'

TOTAL ERRORS = 0
```

Program Listing. Source code for the TSC Extended Basic Precompiler version of SEP (spending extrapolation program)

↙ See List of Advertisers on page 97

The Educated Guest

Mixing Text And Graphics

by Charles H. Santee

Any serious writer of educational programs eventually needs to mix text and graphics on the same screen. You can do this by using the Basic DRAW statement or by some fancy manipulation of the GET and PUT statements. For the sake of speed, however, you have to rely on a machine-language routine. This month's program is aimed at giving the educational programmer a sufficient routine for mixing text and graphics.

Those of you who are not up to writing an Assembly-language program might want to get Instant CoCo, which contains the merged Basic/machine-code program. For more sophisticated applications there are several very good commercial text-to-graphics-screen programs available, or programs like Graphicom that let you create pictures with text.

In addition, I have included another method for merging an Assembly routine with Basic. In an earlier column, I presented a program that generated Basic program lines. The method used in this month's column yields superior results. The pointers that define the end of a Basic program are moved to capture the assembled machine code. Once completed, you can save and load the machine code and a Basic program as if they were one Basic program.

The Assembly routine is invisible, yet you can add or delete Basic lines without affecting the machine code.

How to Write the Program

First, code and save the Basic listing in the usual fashion. Next code and assemble the Assembly-language listing using an assembler such as the Radio Shack EDTASM+. Make sure to use an absolute address when assembling the final program (&H6000). Finally, follow the next few steps carefully to create a final version. *Please make sure to include the 3 bytes that are set to zero at the end of the assembled program.*

1. Reset the computer with a cold start (turn it off or use POKE 113,0:EXEC 40999).
2. Type in CLEAR 500, &H5FFF to reserve initial space for the assembled code.
3. Load in the assembled code. For example, LOADM "SGRA/BIN".
4. Load in the Basic program. For example, load "NOMIX".
5. Find the location of the end of the Basic program by typing the following:

```
PRINT PEEK(&H1B) * 256 + PEEK(&H1C)
```

6. Execute the first part of the assembled code that will merge the two programs together. To do this type: EXEC &H6000.
7. Find the new location for the end of Basic that has been changed to capture the assembled code:

```
PRINT PEEK(&H1B) * 256 + PEEK(&H1C)
```

8. Subtract the old location for the end of Basic (the value found in step 5) from the new location for the end of Basic (the value found in step 7).
9. Edit line 20 of the Basic program so that the value of ZU = PEEK(&H1B) * 256 + PEEK (&H1C) − (the value found in step 8). The value of ZU is the location where the assembled machine code starts in memory.
10. Save the completed product to tape or disk.

System Requirements
Extended Color Basic
64/32K RAM
Editor/Assembler
Disk (optional)

Assembly-Language Merge Routine

The Assembly-language Program Listing 1 consists of two parts. The first part is a merge routine. It is used only once to create the final Basic/machine-code program. This routine finds the end of a Basic program, moves the second part of the code right after the Basic program, and relocates the end-of-Basic pointer to capture the machine code.

The second part of the Assembly code is the main routine. It takes the location of a string that is found in the Basic program and uses a lookup table to put the appropriate bytes on the graphics screen to generate a letter. Basic also determines the location of the text. Although there is some loss of speed in letting Basic compute the location, this routine allows the person more familiar with Basic greater flexibility in defining screen locations.

The Basic Program Listing 2 demonstrates four different ways that the text-to-graphics screen program might be used. The Assembly routine simply looks for a starting location on the graphics screen. This allows for 32 horizontal positions and 192 vertical positions. You can use Basic to convert from one representation of a screen location to another.

The first option uses the standard horizontal/vertical (0–255 horizontal and 0–191 vertical) coordinate system associated with the high-resolution graphics screen. The second option uses the line number (1–16) and tab position (1–32) of the text screen as a starting reference. The third option demonstrates how animation of text is possible as the word selected is moved around the perimeter of the screen. The final option shows how it is possible to put more text on the graphics screen (25 lines rather than 16) than is possible on the text screen.

Use the Merge Routine With Other Assembly Programs

The first 12 lines of Listing 2 are a general-purpose merge to Basic routine that you can use with many different applications. To use the routine to merge other Assembly-language code follow these rules:

- The Assembly code to be merged with Basic must be relocatable.
- Start the routine with a label that will be identified by the merge routine as a starting point. (I used the label STLEN.)
- End the routine to be merged with Basic with 3 bytes set to zero. For example:

```
FDB  0
FCB  0
```

Basic uses three consecutive zeros to identify the location of the end of the program as well as a pointer in position &H1B and &H1C (decimal 27 and 28).

- Initially use a relatively short Basic program to merge with the Assembly listing. You can always add or change lines after a successful merge.

For You Basic-Only Programmers

Get a copy of Instant CoCo. Not only do you not have to worry about using an assembler, but the version on Instant CoCo contains a more complete character set than is included in this article. You can completely rewrite the Basic program and the Assembly routine will stick with you through most of the changes you might make. See if you can come up with a useful educational application that uses mixed text and graphics. ■

Write to the Educated Guest (Charles Santee) c/o HOT CoCo, *80 Pine St., Peterborough, NH 03458.*

```
6000                    00010          ORG      $6000
6000 9E   1B            00020 FIRST    LDX      $1B *GET THEN END OF BASIC
6002 1F   12            00030          TFR   X,Y
6004 30   8D 0012       00040  LEAX STLEN,PCR
6008 A6   80            00050 TFR LDA ,X+
600A A7   A0            00060  STA ,Y+
600C 8C   6140          00070  CMPX #LAST
600F 26   F7            00080  BNE TFR
6011 109F 1B            00090  STY $1B
6014 86   27            00100  LDA #39 *PROTECT THE LOADER
6016 A7   8C E7         00110  STA FIRST,PCR
6019 39                 00120  RTS
601A      00            00130 STLEN FCB 0 LENGTH OF STRING
601B      03F2          00140 STLOC FDB 1010 *LOCATION OF STRING
601D      03F2          00150 SCLOC FDB 1010 *LOCATION OF SCREEN
601F 10AE 8C F8         00160 START LDY STLOC,PCR
6023 AE   8C F7         00170  LDX SCLOC,PCR
6026 A6   8D FFF0       00180 LOOP1 LDA STLEN,PCR
602A 27   30            00190  BEQ DONE
602C 80   01            00200  SUBA #1  *GET THE NEXT CHARACTER
602E A7   8D FFE8       00210  STA STLEN,PCR
6032 A6   A0            00220  LDA ,Y+
6034 81   3F            00230  CMPA #$3F *CHECK TO SEE IF IN TABLE
6036 25   08            00240  BLO SPA
6038 81   5A            00250  CMPA #$5A
603A 22   04            00260  BHI SPA
603C 80   3F            00270  SUBA #$3F
603E 20   02            00280  BRA MULT
6040 86   01            00290 SPA LDA #$1 *IF NOT IN TABLE USE SPACE
6042 C6   08            00300 MULT LDB #08 FIND CHARACTER IN TABLE
6044 3D                 00310  MUL
6045 33   8D 0014       00320  LEAU TABLE,PCR
6049 33   CB            00330  LEAU D,U
604B C6   80            00340  LDB #$80
604D A6   C0            00350 LOOP2 LDA ,U+ *PUT GRAPHIC ON SCREEN
604F 43                 00360  COMA
6050 A7   85            00370  STA B,X
6052 CB   20            00380  ADDB #$20
6054 C1   80            00390  CMPB #$80
6056 26   F5            00400  BNE LOOP2
6058 30   01            00410  LEAX 1,X
605A 20   CA            00420  BRA LOOP1
605C 39                 00430 DONE RTS
605D      7080          00440 TABLE FDB $7080 PRINT
605F      0810          00450  FDB $0810
6061      0020          00460  FDB $0020
6063      0020          00470  FDB $0020
6065      0000          00480  FDB $0 *SPACE OR NOT IN TABLE
6067      0000          00490  FDB $0
6069      0000          00500  FDB $0
606B      0000          00510  FDB $0
606D      0802          00520  FDB 2050 A
606F      8888          00530  FDB $8888
6071      F888          00540  FDB $F888
6073      8800          00550  FDB $8800
6075      048B          00560  FDB $F048B
6077      4870          00570  FDB $4870
6079      4848          00580  FDB $4848
607B      F000          00590  FDB $F000
607D      088C          00600  FDB $7088C
607F      8080          00610  FDB $8080
6081      8088          00620  FDB $8088
6083      7000          00630  FDB $7000
6085      048D          00640  FDB $F048D
6087      4848          00650  FDB $4848
6089      4848          00660  FDB $4848
608B      F000          00670  FDB $F000
608D      880E          00680  FDB $F880E
608F      80F0          00690  FDB $80F0
6091      8080          00700  FDB $8080
6093      F800          00710  FDB $F800
6095      880F          00720  FDB $F880F
6097      80F0          00730  FDB $80F0
6099      8080          00740  FDB  $8080
609B      8000          00750  FDB $8000
609D      7880          00760  FDB $7880 G
609F      8098          00770  FDB $8098
60A1      8888          00780  FDB $8888
60A3      7800          00790  FDB $7800
60A5      8888          00800  FDB $8888H
60A7      88F8          00810  FDB $88F8
60A9      8888          00820  FDB $8888
60AB      8800          00830  FDB $8800
60AD      7020          00840  FDB $7020 I
60AF      2020          00850  FDB $2020
60B1      2020          00860  FDB $2020
60B3      7000          00870  FDB $7000
60B5      0808          00880  FDB $0808 J
60B7      0808          00890  FDB $0808
60B9      0888          00900  FDB $0888
60BB      7000          00910  FDB $7000
60BD      8890          00920  FDB $8890 K
60BF      A0C0          00930  FDB $A0C0
60C1      A090          00940  FDB $A090
60C3      8800          00950  FDB $8800
60C5      8080          00960  FDB $8080 L
60C7      8080          00970  FDB $8080
60C9      8080          00980  FDB $8080
60CB      F800          00990  FDB $F800
60CD      88D8          01000  FDB $88D8 M
60CF      A8A8          01010  FDB $A8A8
60D1      8888          01020  FDB $8888
60D3      8800          01030  FDB $8800
60D5      88C8          01040  FDB $88C8 N
60D7      A898          01050  FDB $A898
60D9      8888          01060  FDB $8888
60DB      8800          01070  FDB $8800
60DD      F888          01080  FDB $F888 O
60DF      8888          01090  FDB $8888
60E1      8888          01100  FDB $8888
60E3      F800          01110  FDB $F800
60E5      F088          01120  FDB $F088 P
60E7      88F0          01130  FDB $88F0
60E9      8080          01140  FDB $8080
60EB      8000          01150  FDB $8000
60ED      7088          01160  FDB $7088 Q
60EF      8888          01170  FDB $8888
60F1      A890          01180  FDB $A890
60F3      6800          01190  FDB $6800
60F5      F088          01200  FDB $F088 R
60F7      88F0          01210  FDB $88F0
60F9      A090          01220  FDB $A090
60FB      8800          01230  FDB $8800
60FD      7088          01240  FDB $7088 S
60FF      4020          01250  FDB $4020
6101      1088          01260  FDB $1088
6103      7000          01270  FDB $7000
6105      F820          01280  FDB $F820 T
6107      2020          01290  FDB $2020
6109      2020          01300  FDB $2020
610B      2000          01310  FDB $2000
610D      8888          01320  FDB $8888 U
610F      8888          01330  FDB $8888
6111      8888          01340  FDB $8888
6113      7000          01350  FDB $7000
6115      8888          01360  FDB $8888 V
6117      8850          01370  FDB $8850
6119      5020          01380  FDB $5020
611B      2000          01390  FDB $2000
611D      8888          01400  FDB $8888 W
611F      88A8          01410  FDB $88A8
6121      A8D8          01420  FDB $A8D8
6123      8800          01430  FDB $8800
6125      8888          01440  FDB $8888 X
6127      5020          01450  FDB $5020
6129      5088          01460  FDB $5088
612B      8800          01470  FDB $8800
612D      8888          01480  FDB $8888 Y
612F      5020          01490  FDB $5020
6131      2020          01500  FDB $2020
6133      2000          01510  FDB $2000
6135      F808          01520  FDB $F808 Z
6137      1020          01530  FDB $1020
6139      4080          01540  FDB $4080
613B      F800          01550  FDB $F800
613D      0000          01560  FDB 0
613F      00            01570  FCB 0
6140      00            01580 LAST FCB 0
```

Program Listing 1. Merge Routine

```
10 CLEAR 1000
20 ZU=PEEK(27)*256+PEEK(28)-294
30 CLS:PRINT@40,"GRAPHIC/TEXT MI
X":PRINT@111,"by":PRINT@165,"DR.
 CHARLES H. SANTEE":PRINT:PRINTS
TRING$(32,"*"):PRINT@235,"enter
text";
35 W$="":PRINT@293,STRING$(22,17
5);:PRINT@325,CHR$(175);:PRINT@3
46,CHR$(175):PRINT@357,STRING$(2
2,175):PRINT@326,"";
40 X$=INKEY$:IF X$="" THEN 40
60 IF X$=CHR$(13) THEN 100
70 IF X$=CHR$(8) THEN IF LEN(W$)
>1 THEN W$=LEFT$(W$,LEN(W$)-1):P
RINTX$;:GOTO 40 ELSE IF LEN(W$)=
1 THEN W$="":PRINTX$;:GOTO 40 EL
SE GOTO 40
80 W$=W$+X$:IF LEN(W$)>20 THEN P
RINT@448,"text too long - try ag
ain";:PLAY"L2T2BAGFEDC":W$="":PR
INT@448,STRING$(31,32);:PRINT@32
6,STRING$(20,32);:GOTO 35
90 PRINTX$;:GOTO 40
100 CLS:PRINT"SELECT NUMBER":PRI
NT:PRINT" 1. GRAPHIC SCREEN LOCA
TION":PRINT:PRINT" 2. TEXT SCREE
N LOCATION":PRINT:PRINT" 3. ANIM
ATION SAMPLE":PRINT:PRINT" 4. TW
ENTY FIVE LINES":PRINT:PRINT" 5.
 TRY NEW TEXT":PRINT:PRINT" 6. E
ND"
110 X$=INKEY$:IF X$="" THEN 110
ELSE ON VAL(X$) GOTO 200,300,400
,500,30,130
120 GOTO 110
130 CLS:PRINT@163,"NOW TRY YOUR
OWN PROGRAM":PRINT:PRINT:PRINT:E
ND
200 PRINT:PRINT STRING$(32,"*")
210 INPUT "HORIZONTAL LOCATION (
0-255)";H:INPUT"VERTICAL LOCATIO
N (0,183)";V:V=V+4
220 IF V>187 OR V<4 OR H>255 OR
H<0 THEN PRINT"invalid location"
:GOTO 200
230 PMODE4,1:PCLS1:SCREEN 1,0
240 GOSUB 3000
245 T$=W$
250 IF V<95THEN V=183 ELSE V=12
255 PMODE 3,1:COLOR 0,1:LINE(8,V
-8)-(204,V+6),PSET,BF:COLOR 1,1:
LINE(8,V-8)-(204,V+6),PSET,B
260 H=16:W$="PRESS ENTER TO CONT
INUE":GOSUB 3000
265 W$=T$
270 X$=INKEY$:IF X$="" THEN 270
ELSE 100
300 PRINTSTRING$(32,"*")
310 INPUT"ENTER A LINE NUMBER (1
-16)";LN:INPUT"ENTER A tab LOCAT
ION (1-32)";T
320 IF LN<1 OR LN>16 OR T<1 OR T
>32 THEN PRINT"invalid entry":GO
TO 300
330 V=LN*12-6:H=T*8-8
340 GOTO 230
400 W$=" "+W$+" ":LW=255-LEN(W$)
*8
405 V=8:PMODE 4,1:PCLS1:SCREEN 1
,1:PMODE 3,1
410 COLOR 2,1:LINE(20,8)-(235,18
3),PSET,BF
420 FOR H=0 TO LW STEP 8:GOSUB 3
000:NEXT H
430 COLOR 0,1
435 FOR V=8 TO 184:GOSUB 3000:LI
NE(H,V-5)-(255,V-5),PSET:NEXT
440 COLOR 3,1:LINE(60,28)-(195,1
63),PSET,BF
445 COLOR 0,1:LINE(H,V-5)-(255,V
-5),PSET
450 FOR H=LW TO 8 STEP -8:GOSUB
3000:NEXT
455 COLOR 1,1:LINE(100,48)-(145,
143),PSET,BF
457 COLOR 0,1
460 FOR V=184 TO 8 STEP -1:GOSUB
 3000:LINE(0,V+4)-(251-LW,V+4),P
SET:NEXT
470 GOTO 250
500 PMODE 4,1:PCLS1:SCREEN 1,1
510 FOR V=4 TO 192 STEP 8
520 IF 32-V1<LEN(W$) THEN AD=-1
530 IF V1<2 THEN AD=1
540 V1=V1+AD
550 H=V1*8-4
560 GOSUB 3000
570 NEXT
580 GOTO 250
3000 NX=INT(H/8)+V*32:VX=INT(NX/
256):HX=NX-VX*256
3010 X=VARPTR(W$):POKE ZU,PEEK(X
):POKE ZU+1,PEEK(X+2):POKE ZU+2,
PEEK(X+3):POKE ZU+3,PEEK(188)+VX
:POKE ZU+4,HX
3020 DEFUSR=(ZU+5):F=USR(0)
3030 RETURN
3040 '******* MACHINE ROUTINE ST
ARTS AFTER BASIC **************"
```

Program Listing 2. Graphics/Text Demonstration

↙ See List of Advertisers on page 97

Doctor ASCII

by Richard E. Esposito and Ralph E. Ramhoff

Got a problem with your Color Computer? Ask Doctor ASCII to solve it. Write to Doctor ASCII, HOT CoCo, *Pine St., Peterborough, NH 03458. Be sure to include a self-addressed, stamped envelope if you want a reply.*

Q. Where can I get a program that lets you copy a program from disk to disk on a single-drive system? I have a cassette version of Color Pilot. How can I fix it so that it saves programs to disk instead of tape?

Donald M. Dealy
Cumberland, RI

A. You already have it! Although it is not documented in your disk manual, you can type: COPY"file name" and the computer will prompt you when to change disks.

Transporting Pilot to disk is no problem. If it starts below $0E00, you can fix it with "Tapefix," *HOT CoCo*, September 1983, p. 135. Getting it to save and load programs to disk is another matter. If you are willing to tackle the machine-language patching process, Roger Schrag's article "It's Superpatch," *Rainbow*, September 1983, p. 66, shows how it is done for EDTASM+. The technique for Pilot would be similar.

Q. I just got a CoCo X-Pad. Using the menu program included with it, I created a nice PMODE3 graphics screen. There is, however, no explanation as to how to save the display to tape. Would you please explain?

Gail Allinson
Brookfield, IL

A. Extended Basic when first booted up reserves four 1,536-byte graphics pages for a total of 6,144 bytes of graphics memory. This is the same result that you get when reserving graphics memory with a PCLEAR4. In a tape system, this graphics memory starts at address 1536 and continues through address 7679. If you had a disk system, the addresses would start and end 2,048 bytes higher in memory. These four pages of screen memory can be saved by typing:

```
CSAVEM"PICTURE",1536,7679,41175
```

for a tape system or:

```
SAVEM"PICTURE",3584,9727,41175
```

for a disk system.

I use an EXEC address such as 41175 because it refers to a known ROM subroutine so that if you accidentally try to execute your picture as a machine-language program, your computer will not go to never-never land. Once you have saved a picture, you can relocate it at any time by running this short Basic program:

```
10 PCLEAR4
20 PMODE3,1
30 SCREEN1,0
40 CLOADM"PICTURE"
50 GOTO 50
```

Q. Where can I get a program to check the speed of my drives?

Douglas Savadage
Dover, NJ

A. If you purchase Radio Shack's OS-9, one comes on the OS-9 Boot disk. Also, J & M Systems, Ltd. (137 Utah NE, Albuquerque, NM 87108) sells the Disk Drive Analyzer for $79.

Q. Why do you bother to print a letter in Doctor ASCII if you are not going to answer it? You all too often refer to an answer in a previous issue or another magazine.

Peter Engelhaut
Watertown, WI

A. There are two reasons: limitations due to space (many of the references go on for pages) and copyright (other magazines own the articles). Besides, regular subscribers would be short-changed if I kept repeating myself or engaged in rehashing in this column what was done by other authors. If I can make a contribution by improving upon a program, I will. Besides, back issues and reprints of *HOT CoCo* and most other magazines are available, and they are much cheaper than most commercial software.

Q. The RANDOM instruction is not available for the CoCo as it is for the TRS-80 Model III. How can I write a program so that it does not use the same sequence of "random" numbers every time I power up?

Barry Hornstein
E. Rockaway, NY

A. The RND function is reseeded whenever it is used with a negative argument. Radio Shack inadvertently left this fact out of their documentation for the CoCo. If you have Extended Basic, the statement X = RND(−TIMER) will seed the random-number generator with the time since power-on in microseconds, and unless you can type RUN consistently at the very same microsecond from power-up, it will give you the equivalent of the Model III's RANDOM instruction.

Q. What does "bubble memory" mean? What are "multicolor rainbow disks"? When you use an EPROM programmer, can you use the EPROM as a ROM pack?

Kent Jakway
Garrett, IN

A. Intel Corp. (Intel Literature Dept., 3065 Bowers Ave., Santa Clara, CA 95051) offers a booklet, free of charge, entitled *Primer on Magnetic Bubble Memory*, which describes bubble memories in this fashion:

> Magnetic bubble memories are serial high-density storage devices like electromechanical disk memories. However, in a disk, the stored bits are stationary on a moving medium, whereas in the magnetic bubble memory, the medium is stationary and the bits move.

When bubble memories were first introduced, floppy disk drives were considerably more expensive than they are today. Bubble-memory manufacturers aimed at the then current price of disk drives with the intention of competing in the floppy market. After a disk-drive price war, most semiconductor companies either put bubble memories on the back burner or withdrew from the market totally.

The rainbow multicolor disks are ordinary disks that are packaged in color jackets. Currently, they are priced somewhat higher than standard disks, but you may like the colored disks better than the plain old black ones. You could use these disks to distinguish your various disk formats (Disk Basic, Flex, OS-9, etc.).

You can buy a ROM-pack board to support 2716 or 2732 EPROMs. These boards are available from Green Mountain Micro, Bathory Road, Roxbury, VT 05669 ($29.95 assembled, $19.95 kit).

Q. I want to make programs with passwords. Do you know of a way to make the password invisible so that no one else can see it?

Stephen Slack
Bear, DE

A. There are a number of ways to change passwords so that they are "invisible." One of the simplest involves using an exclusive-or operation on the original password to produce an encrypted version. Program Listing 1 prompts you for your new password and your encryption code. All hex values that fit in a byte are legal codes; however, a value of 00 results in no encryption being done.

When Listing 1 executes, it prints three DATA statements that correspond to lines 350, 360, and 370 in Program Listing 2. Listing 2 illustrates how a program can be constructed to accept a password from the user and check it against the encrypted password.

You can use this method to encrypt the data in a file in the same manner as was used in the first program, and you decrypt the data in the same manner as in the second program.

Q. In your May column, you discussed version 1.1 vs. 1.2 Color Basic ROM and 1.0 vs. 1.1 Disk Basic ROM. Software written specifically for one ROM will not necessarily work on another. Radio Shack provides a look-up table at the beginning of each ROM to prevent this problem, if programmers would just use it. In Color Basic 1.2, POLCAT was moved, and in Disk Basic 1.1, DSKCON was moved. These two subroutines are the cause of a lot of software incompatibility.

Also, the disk controller in the CoCo 2 does not use 12 volts, it uses 5 volts for the 7416 driver.

Glenn Little

A. I realize that the incompatibility between the ROMs is due to changes in the routines that you cited. The problem in the question you refer to was that the old version Telewriter (1.0) has sluggish keyboard response. This problem can be solved in many ways. First, you can disassemble Telewriter and fix it. Second, you can buy an upgraded version of Telewriter or third, you can use the old ROM. If you have a machine around with the 1.1 ROM, the simplest and least expensive solution is the one that was given. We have had successful results in loading the 1.1

```
10 INPUT "NEW PASSWORD";P$
20 INPUT"ENCRYPTION CODE (HEX)";
R$
30 BIT = VAL("&H"+R$)
40 GOSUB 60
50 STOP
60 '************ PRINT THE
70 '         ENCRYPTED VALUES
80 '
90 PRINT"350 DATA &H";HEX$(BIT)
100 PRINT "360 DATA ";LEN(P$)
110 PRINT "370 DATA ";
120 FOR X = 1 TO LEN(P$)
130 Y=ASC(MID$(P$,X,1))
140 Y1=(Y OR BIT) AND NOT(Y AND
BIT)
150 IF X=1 THEN PRINT Y1; ELSE P
RINT ",";Y1;
160 NEXT X
170 RETURN
```

Program Listing 1

```
10 READ BIT
20 GOSUB 240
30 INPUT "PASSWORD: ";R$
40 GOSUB 70
50 IF Z=1 THEN PRINT"ILLEGAL PAS
SWORD" ELSE PRINT"PASSWORD OK"
60 STOP
70 '************ CHECK
80 '            PASSWORD
90 '
100 Z = 0
110 'BIT IS AN ENCRYPTION CODE
120 'IT IS XOR'ED WITH THE USER
130 'PASSWORD TO PRODUCE THE
140 'SECRET CHARACTER STRING
150 IF LEN(P$) <> LEN(R$) THEN Z
=1:RETURN
160 FOR X = 1 TO LEN(P$)
170 Y = ASC(MID$(P$,X,1))
180 Y1 = (Y OR BIT) AND NOT (Y A
ND BIT)
190 'LINE 100-300 XOR'S Y AND BIT
200 Y2 = ASC(MID$(R$,X,1))
210 IF Y1 <> Y2 THEN Z=1
220 NEXT X
230 RETURN
240 '********** CONVERT DATA
250 '            TO THE STRING
260 '
270 P$=""
280 READ Z
290 FOR X = 1 TO Z
300 READ Y
310 P$ = P$ + CHR$(Y)
320 NEXT X
330 RETURN
340 '****** PASSWORD DATA
350 DATA &HAA
360 DATA 8
370 DATA 238,248,138,235,249,233
,227,227
```

Program Listing 2

ROM from both tape and disk replacing the 1.2 code, and with loading the 1.2 ROM from tape and disk replacing the 1.1 code.

Thank you for the comment on the disk power. It is never possible to get all questions right. We try to base our answers on our personal expertise; sometimes we have to rely on other sources. The disk answer was received from a usually reliable source. Since we know of people who have modified their older controllers using the 12 volts from the disk power supply, we were inclined to accept this as fact.

Q. In your November column, you stated that "the full power of a 64K machine can only be unleashed with . . . Flex and OS-9." I'd like to include Star-DOS on that list. You also said "Only Flex can use the double-sided feature of your disks." Not really. Both Star-DOS and JDOS (by J & M Systems) can use two-sided drives. In addition, Star-DOS can use up to 80 tracks and also single and double density. On another occasion, you mentioned some programs with hi-res displays. Star-DOS offers a choice of 40-by-24, 51-by-24, or 64-by-24 displays, selectable either directly from the keyboard or through system calls.

Peter A. Stark, President
Star-Kits
Mt. Kisco, NY

A. I guess it all boils down to what I mean by "the full power of a 64K machine." DOSes are purchased for various reasons. One must decide what need a particular DOS fills.

I consider programming in Assembly language a necessary evil. Others I know love it. My definition of "unleashing the full power of a 64K machine" is gaining access to languages other than Basic such as C, Cobol, Fortran, and Pascal. In this category, neither Star-DOS nor JDOS fills the bill. If one considers "unleashing the power" to be a development system with which you can write disk-oriented programs that use the standard Radio Shack format, but does not want to depend on using routines that are resident in Radio Shack's ROMs, then Star-DOS may be what he is looking for.

If he is a purveyor of Flex-oriented software and wants to broaden his market to include the standard CoCo Disk Basic user, Star-DOS could help cut down on his conversion effort. I know of one commercial product for the CoCo that is based on Star-DOS, and it saved the authors a lot of development time.

I did not mean to slight the Star-DOS operating system. It is a matter of opinion. It is just not what I look for in a DOS. My favorite language is APL, which is about as far-removed from Assembly language as you can get. (By the way, if anyone has transported University of Waterloo's APL from the 6809-based Commodore Super Pet, please let me know.)

Overall, Star-DOS is a nice package for those who are in the market for this type of specialized system. ■

Reader's Forum

Who's Minding The Subroutines?

If you have trouble keeping track of your subroutines, you can list the beginning line numbers at the end of your program by using the GOTO and REM statements.

For example:

```
1000 GOTO 100 : 'SET-UP GRAPHICS
1010 GOTO 205 : 'MOVE WIDGET
1020 GOTO 357 : 'ADD SCORE
1030 GOTO 419 : 'END GAME
```

Always use the GOTO statement first or your CoCo will treat the whole line as a REM statement.

Using this routine will let you renumber the program without losing track of subroutines.

Edward J. Niklas
Nokomis, FL

Sound Discoveries

While searching through the computer, I found that address 223 stores the volume of the PLAY statement in this manner:

```
PEEK (223) = 128 + (volume*4)
```

I have also found that you can make a different tone by POKEing 223 to a number lower than 128. The resulting buzzing tone is good for honky-tonk music such as that used in Donkey King.

You can alter the volume of this by using this equation:

```
POKE 223, (31 – volume)*4
```

Note that a statement like V+ or V– gives you an illegal-function-call error while a statement such as V15 returns you to a normal tone. When you give the command RUN, the volume is automatically reset to V15.

Gordy Dow
Bellevue, WA

CLOAD Both

These steps enable you to CLOAD two different Basic programs and use either one when you need it.

- CLOAD the first program as usual.
- PRINT PEEK(25), PEEK(26), PEEK(27), PEEK(28).
- Write out the value numbers of these PEEKs: (A,B,C,D; I got A = 30, B = 1, C = 44, D = 109 with my CoCo).
- POKE 25, PEEK(27) :POKE 26, PEEK(28) then press enter changing the Basic pointers.
- CLOAD the second program.

Now, when you want the first program just type POKE25,A:POKE26, B, press enter (POKE25,30:POKE 26,1 in my case).

For the second program type POKE25, C:POKE26, D, press enter (POKE25,44:POKE26,109 in my case).

If the programs you CLOAD are too long and exceed your CoCo's total memory, you will get an OM error message.

Tho Phuc Luong
Flushing, NY

Listing Change

You can change the screen listing produced by the LIST command using the following techniques.

Add these lines to the beginning of your program:

```
1 D = PEEK(25)*256
2 FOR Y = D TO D + (the number of bytes taken by your program)
3 IF PEEK(Y) = 42 THEN POKE Y,8
4 NEXT Y
5 DEL – 5
```

Then, format the rest of your program as follows:

```
10 (whatever your line says):'******(anything goes here).
```

(****** is the number of characters in your line, plus six).

Format all your lines like this, then run. The five lines will replace each asterisk with a back-space, then delete themselves. When you type LIST, only the things you put behind the asterisks will appear.

If you dump the listing to a printer the real listing will appear. Here is an example:

```
1 D = PEEK(25)*256
2 FOR Y = D TO D + 200
3 IF PEEK(Y) = 42 THEN POKEY,8
4 NEXT Y
5 DEL – 5
10 FORX = 0TO8:'***************HELP
20 CLS(X): '************SOMEONE
30 FORI = 1TO100:NEXTI: '************************STOLE
40 NEXTX:'***********MY
50 END:'*********LISTING
```

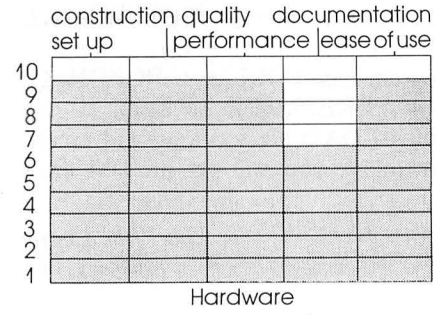
When run and listed the screen would look like this:

```
HELP
SOMEONE
STOLE
MY
LISTING
```

You can also edit lines when they're like this. Experiment.

John Sciarabba
Rochester, NY

REVIEWS



CoCo Port
Green Mountain Micro
Bathory Road
Roxbury, VT 05669
802-485-6112
$49.95 assembled and tested
$39.95 complete kit
$20 bare printed circuit board
$19.95 optional parallel printer cable
$5 printer-interface software on cassette

by Robert P. Bussell

The CoCo Port is a parallel interface with a feed-through connection that lets you use it with a cartridge and a disk at the same time.

It's very easy to use. Simply connect the optional printer cable and plug the unit into the cartridge slot. If you have a disk drive, plug it into the CoCo Port, turn on the power, and load the accompanying program.

The software that comes with the interface lets the operating system run Basic programs without modification. If you don't know how to make software patches to machine-language programs, you should make sure the interface is compatible with your favorite software before you spend your money.

I bought the CoCo Port kit. Although there are very few parts to install, it is by no means simple. If you decide to go this way and do it yourself, make sure you heed the warning on the instructions and read them very carefully before you break the seal on the parts.

My instructions came with a sheet of changes that made assembling the kit pretty confusing. Change #1 told me to solder the 74S139 chip directly to the board. However, change #2 says to insert a 0.1 μF flat capacitor into pins 8 and 16 before you insert the 74S139 chip. The original instructions tell you to connect the capacitor between pins 1 and 16.

When I noticed these discrepancies, I looked for a schematic but found none. However, the kit does include reprints of two of Dennis Kitsz's articles on the CoCo Port that give very good explanations of the parallel port. These should be invaluable to the hardware buff.

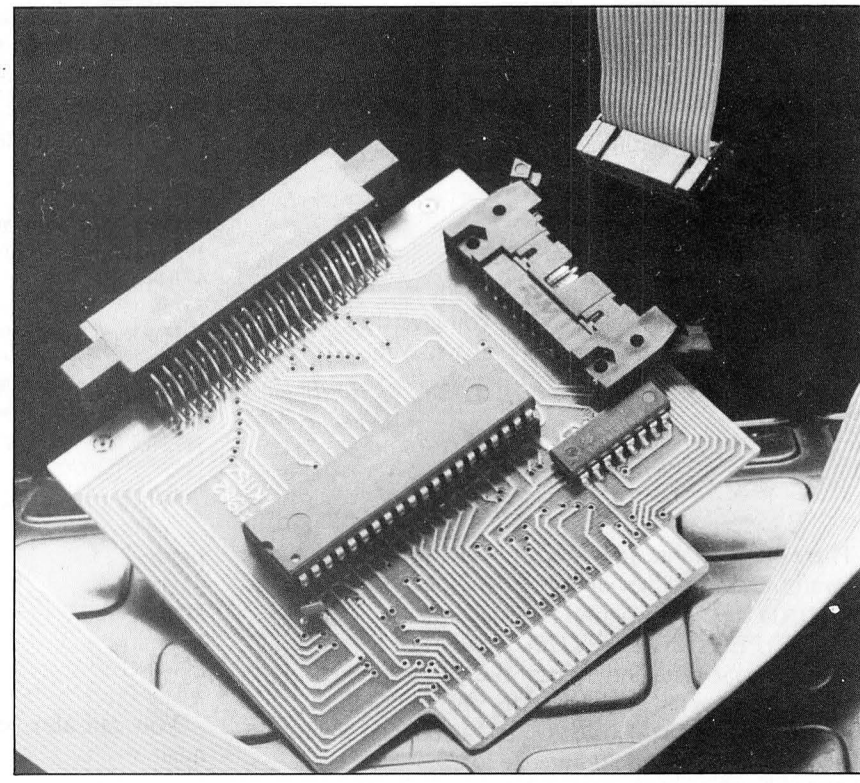
Those of us without advanced hardware-construction skills, though, would like to see a single, clear instruction sheet, a picture of the board with the components and jumper installed, and a schematic of the parallel port.

The quality of the kit is outstanding. All connectors are gold plated, and the PIA chip is socketed. The optional printer cable is 36 inches long, and you get a cable-wiring diagram with the instructions in case you want to make your own.

I'm very impressed with the CoCo Port, but I can't recommend that you try to build your own unless you are an

CONTENTS

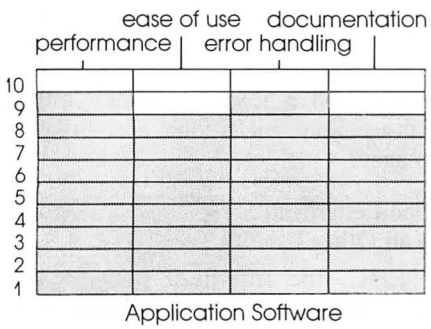
edited by Mark E. Reynolds



Green Mountain Micro's CoCo Port

experienced hardware hacker. It's not an easy project.

For those who like to experiment, the CoCo Port is ideal for data acquisition and control applications such as measurement, switching, and signaling. The port lets you use all 18 control lines. With an extra power supply, you can have up to four CoCo Ports for a total of 64 control lines. The documentation includes a control-process idea sheet to get you going. ■



Master Design
Derringer Software
P.O. Box 5300
Florence, SC 29502
803-665-5676
32K, Extended Color Basic
$34.95 disk

by Scott L. Norman

Here is a versatile program that uses the Color Computer's graphics capabilities to produce both pictures and text in a variety of formats. You use the keyboard to edit images on the TV screen, and then save the material to tape or disk, or print it out.

The package includes subroutines that you can adapt to drive most dot-matrix printers, as well as material that lets you link Master Design to your own Basic programs. There are also detailed instructions for interfacing with Telewriter-64, so you can design letterheads and other material and have the word processor call it up whenever you want.

Master Design is fun to use, too. Although there are plenty of commands to keep track of, this is still the most useful graphic-design package I've yet worked with. The ability to mix text and graphics freely has a lot to do with it. I have quite a few uses for images accompanied by text, and now I can create text in large, varied formats to go along with my graphic material.

There are a number of Basic files on the program disk, and the most important are MD (Master Design itself), LHU (the letterhead utility, which stores any portion of a high-resolution display in a binary file that you can later read and send to a printer), and SUB (a subroutine package that you can merge with your program to print out Master Design images).

Setting Up

Before getting to work, you must copy the entire disk and customize those portions of the Basic programs that call the printer-driver routines. The manual contains a brief discussion of 7- and 8-bit printers, as well as instructions for editing a half-dozen lines of code to accommodate Epson, C. Itoh, and Okidata printers (Master Design will drive most Radio Shack printers without modification).

By the way, my copy of the manual included an error in the C. Itoh customization listing. In MD, the Basic line containing the variables needed to put the printer into graphics mode should read:

```
90 GM$ = CHR$(27) + CHR$(83) + "0256"
```

You should change the corresponding lines in the other routines (LHU and SUB) as well. I have also tested the system successfully with an Epson FX-80, using the Basic code recommended in the documentation.

If you like to send data to your printer at more than the 600-baud default rate, you should also add the appropriate POKE 150,n command to any convenient early section of the code. If your printer gives you a choice between bidirectional and unidirectional printing, choose the latter; it will give much straighter vertical lines.

The Master Design File

Master Design has two modes of operation: Edit and Graphic. Edit enters text: You issue commands to set up a font and type away. It also includes the commands for saving and loading graphics pages. Edit supports both tape and disk image storage, and you can specify the starting page and the number of pages you want to save. You can also get page-to-page transfers, which gives a lot of flexibility in copying material.

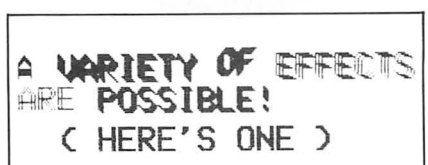


Fig. 1. Different Fonts

In Graphic mode, you can create images on the video screen using an arrow-key controlled cursor. Extended Color Basic's LINE, BOX, CIRCLE, and PAINT commands are available, although Master Design uses its own syntax for them.

Much of the program's power comes from its ability to combine material created with the two modes. In effect, it treats the video screen as a single piece of RAM and ignores any distinction between text and graphics. This is certainly logical—I'm sure the program actually draws the alphabetic characters on the display using graphics commands—but it is different from conventional CoCo operation.

One benefit is that you can use certain commands and features in both modes. Two examples are block erasure, which amounts to filling a designated rectangle with the background color, and reflection, which creates the impression that the material written or drawn on the TV screen is obliquely illuminated so that it casts a shadow on a horizontal surface.

You can select the PMODE in which you will work, and you can write or draw in one PMODE and then switch to another for display. You can create different parts of an image in different modes, too. The result is that a tremendous number of visual effects are possible—too many to easily catalog in a review, in fact.

The two-key combination, shift/clear, performs the control function for both Edit and Graphic modes, providing access to a common set of 16 commands. The Graphic mode has 17 additional commands of its own, and most of them require only a single keystroke, although there are commands in both sets that require additional information.

Like most graphics programs, Master Design is command driven; there is no menu on the screen to remind you of your possible choices at each step. The manual manages to get a complete command summary onto a single page, although explanations of all the options take another seven.

Some of the subcommands consist

of a series of numbers or letters separated from each other by the enter key, and if you forget the syntax, you might spend quite a while wondering where the program went. There are no prompts, and the cursor usually disappears once you start to enter a command.

Fortunately, you can survive without losing a partially completed creation: Press break, type RUN, and you will find yourself back where you were before things went awry. Good marks for error handling.

Master Design always wakes up in Edit mode and draws its own title display. To create your own text, you must clear the screen with the shift/clear combination, choose a type size and font, and put the cursor into position. Unfortunately, you can't produce lowercase letters.

You can set the size of your letters and also choose a font, which is where you first get an idea of Master Design's flexibility (see Fig. 1). You can draw or shadow characters with a single stroke and choose the direction of shadowing: horizontal, slanting up to the right, or slanting down to the right. Shadowing can be solid or open, so that the letters might look as though constructed from a set of wire frames.

This gives you plenty of opportunity to create good-looking block lettering, but there are still more possibilities. If none of the standard fonts please you, you have a certain amount of leeway to create your own.

Normally, you use the arrow keys to move the cursor by one character position in Edit mode; if you use shifted arrows, however, the cursor only moves by one graphics dot. Therefore, you can go over a piece of text several times, with a slight offset between each pass, thereby thickening the lines in any direction.

You can also create unique effects by entering text in one PMODE and then switching to another. The differences in resolution can convert solid letters to zebra-striped ones, or give contrast reversal: light letters on a dark background. (A contrast-reversal option is also available when you print out a display.) The choices of line and background colors interact with the PMODE selection, as well.

The Master Design package provides some design help in a sample file and a sample printout, but experimentation is far and away the best teacher. This is especially true when it comes to predicting the effects of switching PMODEs or color sets.

In graphic mode, you get a marker and a cursor that perform several useful jobs. The cursor moves around under arrow-key control and can draw or erase lines. The marker stays put until you jump it all the way to the cursor's position or exchange it with the cursor.

The marker and cursor can represent two points that you connect with a line, or they can define opposite corners of a rectangular box. Master Design has commands for outlining such a box or filling it in with dots, lines, or a solid color—very useful in creating graphics.

To create the word, "reflection" (Fig. 2), I merely had to place the marker and cursor just outside the word and type the letter R followed by a number representing the cast factor—the angle at which I wanted the shadow thrown.

"It's always a pleasure to see a program that capitalizes on the strengths of the Color Computer."

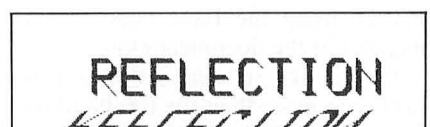


Fig. 2. Reflection

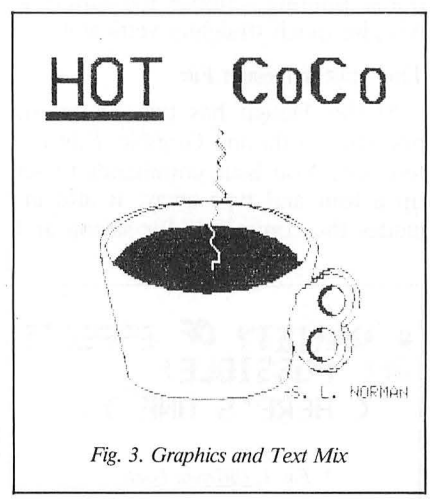


Fig. 3. Graphics and Text Mix

My *HOT CoCo* logo (Fig. 3) is no artistic triumph, but it does illustrate the way Master Design can mix text and graphics. I drew the whole thing on one PMODE 4 display using a few tricks that I had to discover by experimentation, and Master Design makes experimenting pretty easy.

Because the program uses canned functions for lines, circles, and rectangles, it was surprisingly easy to draw with the arrow keys, although I expected to find them clumsy when compared with a joystick. Pixel-by-pixel editing can result in some nice-looking images.

The Letterhead And Other Utilities

LHU, the letterhead utility, converts any high-resolution display to a special file that you can reload and send directly to the printer. Normally, you must load an image into the Color Computer's graphics area in low RAM, and your printout will then reflect the contents of the entire area, but LHU permits you to read and print smaller images—exactly the kind of capability needed for letterheads. Anyone who has ever received mail from Derringer Software and noticed the small Derringer pistol logo has had a preview of this routine's usefulness.

You can check the appearance of your selected graphic by getting a printout from LHU itself. You can change PMODE before printing, just as in Master Design, and you can send any special control codes to your printer. You can't store the codes along with the letterhead, however.

You can only store LHU graphics on disk, not tape. You can build up complex letterheads by appending images from different graphics pages, though, and you can store imagery in reverse video. This can be handy because the programs will generally use the graphics without a contrast-reversal option.

The Master Design disk includes a small file that you merge with the subroutine package to present you with a list of options for printing your letterhead with Telewriter-64.

Summing Up

I am enthused about Master De-

READER SERVICE

This card valid through November 30, 1984

A. On the average, how many of each HOT CoCo issue's program listings do you actually type in?
1. 1
2. 2
3. 3
4. 4
5. 5 or more
6. None

B. How much do you plan to spend on software during the next 12 months?
1. Nothing
2. Less than $100
3. $100-$250
4. $250-$500
5. $500-$1,000
6. Over $1,000

C. How much do you plan to spend on hardware during the next 12 months?
1. Nothing
2. Less than $250
3. $250-$500
4. $500-$750
5. $750-$1,000
6. $1,000-$1,500
7. Over $1,500

D. Which of the following items do you plan to purchase during the next 12 months?
1. Printer
2. Modem
3. Keyboard
4. Expansion Device
5. Joystick
6. Voice Synthesizer
7. Disk Drive(s)
8. Monitor
9. Other

E. Which of the following models do you own? Check all that apply.
1. 4K
2. 16K
3. 32K
4. 64K
5. MC-10
6. TDP-100
7. Dragon 64K
8. Extended Basic
9. Standard Basic

F. Which of the following publications do you read monthly?
1. HOT CoCo
2. 80 Micro
3. Rainbow
4. Color Computer Magazine
5. Advanced Computing
6. Computer User
7. Micro
8. Color Micro Journal

G. What types of articles in these magazines interest you most? Check 3 only.
1. Editorials
2. Tutorials
3. Industry News
4. Reviews
5. New Products
6. Hardware Construction
7. Software Modification
8. Programming Techniques
9. Application Programs

H. If offered a choice, how would you prefer to purchase Instant CoCo?
1. 6 month subscription
2. 12 month subscription
3. Longer subscription
4. Buying one issue at a time

I. What would you like to see improved in HOT CoCo? Check up to 3.
1. Legibility of program listings
2. Clarity of instructions
3. Explanation of figures and tables
4. Explanation of terminology

J. On a scale of 1 (no interest) to 5 (great interest) rate your interest in the following HOT CoCo columns:
___ 1. The Basic Beat
___ 2. 6809 On Line
___ 3. Digressions
___ 4. Reviews
___ 5. Reader's Forum
___ 6. Doctor ASCII
___ 7. Product News
___ 8. DOSsier
___ 9. The Educated Guest

K. If you are not a subscriber please circle 500

Reader Service: Return this card to receive full information on the products advertised in this issue. Refer to the ad. You will find numbers near the logo of each advertiser. Each represents the advertiser's individual Reader Service number. Circle the corresponding numbers on the card on this page. Include your name, address & zip, and drop in a mailbox. In 4-6 weeks you'll hear from the advertiser directly.

1 6 11 16 21	151 156 161 166 171	301 306 311 316 321	451 456 461 466 471
2 7 12 17 22	152 157 162 167 172	302 307 312 317 322	452 457 462 467 472
3 8 13 18 23	153 158 163 168 173	303 308 313 318 323	453 458 463 468 473
4 9 14 19 24	154 159 164 169 174	304 309 314 319 324	454 459 464 469 474
5 10 15 20 25	155 160 165 170 175	305 310 315 320 325	455 460 465 470 475
26 31 36 41 46	176 181 186 191 196	326 331 336 341 346	476 481 486 491 496
27 32 37 42 47	177 182 187 192 197	327 332 337 342 347	477 482 487 492 497
28 33 38 43 48	178 183 188 193 198	328 333 338 343 348	478 483 488 493 498
29 34 39 44 49	179 184 189 194 199	329 334 339 344 349	479 484 489 494 499
30 35 40 45 50	180 185 190 195 200	330 335 340 345 350	480 485 490 495 500
51 56 61 66 71	201 206 211 216 221	351 356 361 366 371	501 506 511 516 521
52 57 62 67 72	202 207 212 217 222	352 357 362 367 372	502 507 512 517 522
53 58 63 68 73	203 208 213 218 223	353 358 363 368 373	503 508 513 518 523
54 59 64 69 74	204 209 214 219 224	354 359 364 369 374	504 509 514 519 524
55 60 65 70 75	205 210 215 220 225	355 360 365 370 375	505 510 515 520 525
76 81 86 91 96	226 231 236 241 246	376 381 386 391 396	526 531 536 541 546
77 82 87 92 97	227 232 237 242 247	377 382 387 392 397	527 532 537 542 547
78 83 88 93 98	228 233 238 243 248	378 383 388 393 398	528 533 538 543 548
79 84 89 94 99	229 234 239 244 249	379 384 389 394 399	529 534 539 544 549
80 85 90 95 100	230 235 240 245 250	380 385 390 395 400	530 535 540 545 550
101 106 111 116 121	251 256 261 266 271	401 406 411 416 421	551 556 561 566 571
102 107 112 117 122	252 257 262 267 272	402 407 412 417 422	552 557 562 567 572
103 108 113 118 123	253 258 263 268 273	403 408 413 418 423	553 558 563 568 573
104 109 114 119 124	254 259 264 269 274	404 409 414 419 424	554 559 564 569 574
105 110 115 120 125	255 260 265 270 275	405 410 415 420 425	555 560 565 570 575
126 131 136 141 146	276 281 286 291 296	426 431 436 441 446	576 581 586 591 596
127 132 137 142 147	277 282 287 292 297	427 432 437 442 447	577 582 587 592 597
128 133 138 143 148	278 283 288 293 298	428 433 438 443 448	578 583 588 593 598
129 134 139 144 149	279 284 289 294 299	429 434 439 444 449	579 584 589 594 599
130 135 140 145 150	280 285 290 295 300	430 435 440 445 450	580 585 590 595 600

Name_____

Address_____

City_____ State_____ Zip_____

PLACE
STAMP
HERE

HOT CoCo
POB 347
Dalton, Mass 01227

HOT CoCo INDEX TO ADVERTISERS

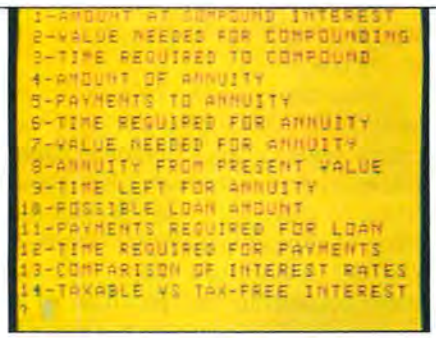
Director: Stephen Twombly, Sales Manager: Raino Wirein, Ad Coordinator: Beverly Poirier
Advertising Offices: (603) 924-7138 or (800) 441-4403

*This advertiser prefers to be contacted directly. For further information from our advertisers, please user the Reader Service card.

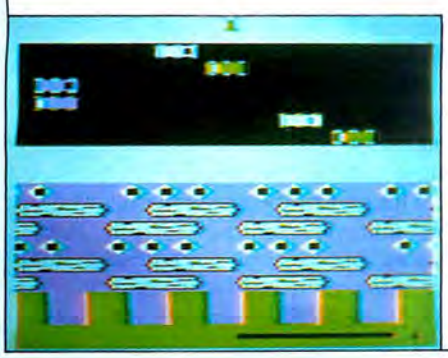
COMING NEXT MONTH

You bought your Color Computer to have a little fun and do a few household clerical chores. Right? The November *HOT CoCo* certainly fits this mold. Next month we'll offer both practical and entertaining software that's yours for the typing in.

"Interesting Interest," by Anna Reeves, is the most complete loan analysis program we've seen. It offers 14 menu selections that will let you do "what if" projections before you go to the bank.



"QType," by Robert Cutter, stands for "quick type." It is a very short Basic text processor, but it is so convenient to use that you'll shelve your fancy $50 word processor in favor of QType for common letter-writing tasks.

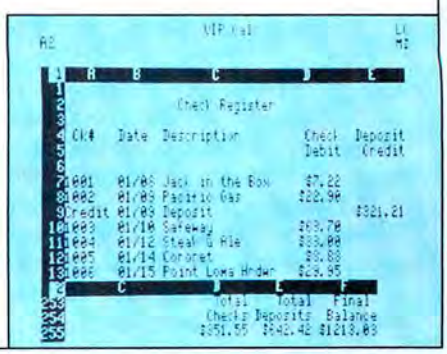


"Night Racer" is for all you Parnelli Jones types. It is a 16-screen, scrolling road race game by James Wood. How long can you stay on course?

The long-awaited VIP Calc gets the once-over from Scott Norman. Does it live up to its claims? Find out next month. We'll also review Radio Shack's Disk EDTASM+ for the Assembly-language aficionados among you.

Many of you are anxiously awaiting November for the final installment of Croaker in "Anatomy of an Assembly-Language Game," by Mike Meehan. By then you'll not only have a great commercial-quality game, but a little more knowledge of Assembly-language programming.

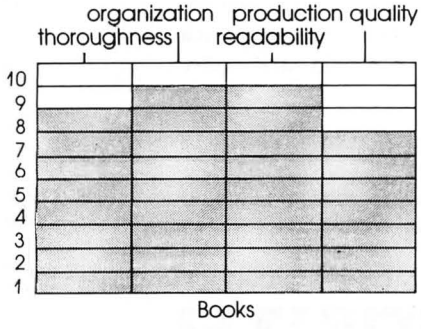
November will bring much more—too much to list here. Pick up a copy and enjoy. ■



sign, and for a couple of reasons. It is always a pleasure to see a program that capitalizes on the strengths of the Color Computer, and decent graphics capability at a low price is certainly one such strength. It is also nice to receive a complete package; it would not have been outlandish if author Dennis Derringer had omitted a printer driver at this price, for example.

As far as specific improvements are concerned, I can think of only two. First, there are enough commands in Master Design to warrant a duplicate copy of the reference sheet, perhaps on something like manila stock for durability. I'd also like to see a lowercase alphabet.

Even without lowercase, though, I find a lot to like in Master Design. ■



Things to Do
With Your Color Computer
by Jerry Willis, Merl Miller,
and D. Lamont Johnson
prepared for New American Library
1633 Broadway
New York, NY 10019
Dilithium Press
8285 S.W. Nimbus
Suite 151
Beaverton, OR 97005
$3.95, 216 pp.

by Gary A. Ludwick

Where was this book when I needed it?

Things to Do with Your Color Computer is probably the best introduction to the Radio Shack Color Computer I have seen. It doesn't try to teach you Basic. It's not a compilation of type-'em-yourself programs. It is a consumer's buying guide to the wonderful world of CoCo.

Things to Do is the kind of book that's most valuable before you buy a computer, or during the first few months after your purchase.

Beginning with a good layman's overview of the Color Computer, *Things to Do* anticipates most questions a potential buyer might have about the machine, and it offers the information in a refreshingly opinionated way. For instance, the authors term the Color Computer 2's standard Basic as "good, but not great," and go on to urge you to spring for Extended Color Basic if possible.

The rest of this 216-page book offers resource information that every new CoCo owner should have.

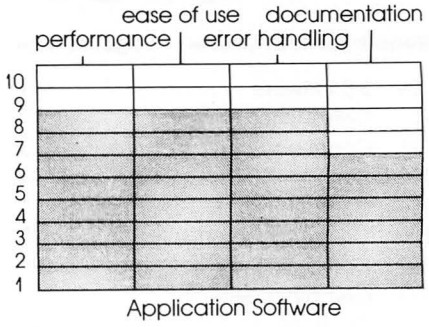
Topics covered include available publications, games, educational programs, telecommunications, word processing, programming languages, peripherals, and more.

In each chapter, the authors attempt to review what they feel are the best examples of currently available software. Along with this are some good tips about what program features to look for, how the programs compare, and where-to-buy information. The reviews are succinct, informative, and candid, just the way one user would talk to another.

Of course, with the ever-changing market, the book might have a problem with timeliness. My copy was issued in December 1983 and was still current at the time I wrote this review. However, semiannual revisions would almost seem a necessity. We'll have to wait and see if Signet/Dilithium is up to the task.

Given that caveat, *Things to Do with Your Color Computer* is an excellent guide to Color Computer capabilities and programming. For those trying to make their way through the maze of competing software claims, it provides a fine road map. And for those considering the purchase of a computer, it's a great way to receive a fast education. It's the closest thing yet to a *Consumer's Report* buying guide for the Color Computer. ■

> "Things to Do With Your Color Computer *is an excellent guide to Color Computer capabilities and programming.*"



Teachers' Database
Tom Mix Software
3424 College N.E.
Grand Rapids, MI 49505
616-364-4791
32K, Extended Color Basic
$39.95 cassette
$42.95 disk

by Thomas and Anne Taulli

The Teachers' Database helps teachers keep a computerized file of students' grades. You can alphabetize records and sort, rank, and weight them by various criteria. You can then analyze the data and print it out.

When you load the data-entry program, a menu shows your options. You can enter the class roster, load a previously saved file, change an item, and delete or add categories. The program will accept up to 100 names and up to 20 items per student, but you can change these parameters to suit your needs.

If you use a point system in which your students know in advance how much the tests, quizzes, and projects are worth, you can choose the Combine option to display the categories that contain numeric data. You can then select those you wish to combine. You can also add extra weight to any category by entering it twice.

The second half of the program sorts the data you entered and statistically analyzes the student's records in relationship to his other grades, or to the class as a whole. Either one gives you mean, mode, and median values. You can also get variance and standard deviation, as well as minimum and maximum values in the file.

This part of the program only works on numeric records, not on letter grades, but you can easily resolve this by using numbers for grades (A = 4, B = 3, etc.). The program will, how-

ever, sort, rank, and print letter grades.

You can print out your files. However, if you are used to a gradebook that has room for 25 grades per page, you will be disappointed. Teachers' Database will only give you a name and five columns of grades per page, if your printer is set to an 80-character line. A 132-character line will give you an extra three categories per name. If you need to print out many grades, you might find this limiting.

The program will automatically append a $ to any category name to alert you that there is nonnumeric data in the file. This also prevents the program from attempting to perform statistical analysis incorrectly. If you try to delete some or all of a file, the program will ask whether this command is correct, because the deletion is irrevocable.

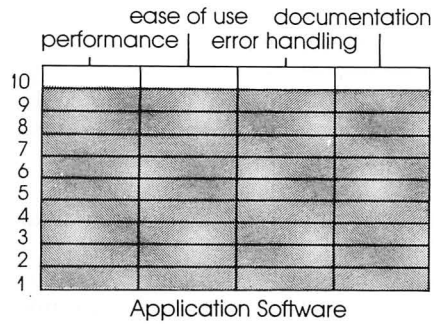
Cassette users get some good tips on saving data and protecting your information against bad saves or errors.

Documentation is the weak link in this otherwise fine program. The 4-by-4-inch booklet presents information in the order that it appears in the menu and not in the sequence that you would normally use. A demo program would also have been a help, especially to teachers who aren't comfortable yet with computers.

The print in the documentation is also quite small. It's a little rough to read when you try to scan for specific information.

All in all, though, Teachers' Database is a worthwhile program. It can save you time as well as give you a clearer statistical understanding of your class's performance. ■

"Teacher's Database... can give you a clearer statistical understanding of your class's performance."



Key-264K
Key Color Software
P.O. Box 360
Harvard, MA 01451
617-263-1727
32K, Extended Color Basic
$39.95 cassette
$44.95 disk

by Martin Klaver

Would you like to upgrade your 32K Color Computer to 64K? Have you held back because the upgrade wouldn't really double your useful

✓ See List of Advertisers on page 97

memory for Basic? Have you ever wanted a second CoCo to run two programs at the same time? How would it be if the two programs could communicate with each other and work together?

Well, Key-264K might just be a software solution that is better than hardware.

To use Key-264K, your 32K computer should not be a piggy-back upgrade, and your chips should not be "half-good" ones that might have been installed in early 32K versions. Also, if you have a particular machine-language program you want to run with Key-264K, you should first make sure that both are compatible. Key-264K should work with E, F, or modified D boards.

For me, the program worked perfectly with an early 32K machine, and I had no compatibility problems. There should be few difficulties with Basic programs.

I found Key-264K to be a fine program with outstanding documentation, even if its name is a little confusing (264?). Its advertising does not do it justice.

Key Color Software has simply unlocked the "backside" of the 32K chips (that half that Radio Shack didn't access for the 32K CoCos). In 64K computers, it opens up the backside for Basic and other uses. The extra memory becomes truly useful.

In order to get maximum use of the second 32K, Key Color Software divided the 64K into two separate 32K memory banks that operate independently, but can also communicate with each other. The program occupies the upper 3,225 bytes of each bank of memory. In order to make the two banks communicate, the software adds a whole new set of simple commands to Basic.

The first bank comes up with the usual green screen. The second comes up in orange. That makes it easy to distinguish one from the other when you start to work with the program. Later you can change the background colors to be the same for both banks, which makes switching banks invisible.

The manual moves you easily into the program. That is important, because operating two computers at the same time is a little complex. I spent a half hour reading the 75-page manual before I fired up the program and another half hour working through the manual on the computer. That is all I needed to get thoroughly familiar with Key-264K. To use its many capabilities will take much more thought and imagination, however. This is a big extension to the CoCo.

"Key-264K gives my CoCo capabilities I don't even have with my IBM PC compatible."

The manual deserves special praise. It makes everything easy. Although it's printed in dot matrix and lacks charts, the text is obviously inspired by Radio Shack's fine manuals and is free of jargon and full of examples. It consists of three parts: a keyboard-learning guide, a Basic-learning guide, and a Basic reference section.

The keyboard commands are simple. Using the space bar with the down arrow switches to the B side (orange screen) and the space bar with the right arrow switches back (green screen).

You can load and run a program into either side while you do something else on the other. This is one of the few programs that makes the most of the CoCo's multitasking ability, which uses interrupts to shift control during operation from one function to another. Key-264K lets you use simple commands to move back and forth between two programs.

Of course, you don't have to use the multitasking feature. The program offers several other options. For example, you can simply use the B side as an extension of the A side, which gives you an enormous increase in usable memory.

You can transfer variables between banks with the PUSH and PULL commands, or move Basic or blocks of memory from one side to the other with the DUP, LCOPY, or CMCOPY commands. You can move memory within a bank.

You can PEEK and POKE the other bank with CPEEK and CPOKE, and you can view the text or graphics screen from either bank with a variety of VIEW commands. You can access the graphics screen in both banks in an instant, and thus enhance your CoCo's graphics capabilities.

You also get commands for break, reset, and cold starts that address one side or the other, but not both. You can run Disk Basic on one side and Extended Color Basic on the other, which means that you can run Extended Color Basic programs that are otherwise incompatible with Disk Basic.

A command that lets you switch from one program on one side to a subroutine or second program on the other side allows you to run very large Basic programs. Whether you want long Basic programs or just more graphics pages, or LLISTing from one side while editing on the other, Key-264K will do the job.

The VIEW commands actually consist of 13 separate commands to control graphic and text screens. These let you build many more screens into your programs and easily access the newfound memory to increase your graphics capabilities.

The VIEW commands let you view the graphics and text screens from the same and from opposite sides of memory. You can also set up and view special displays and graphics modes not normally available from Basic, such as the semi-graphics modes.

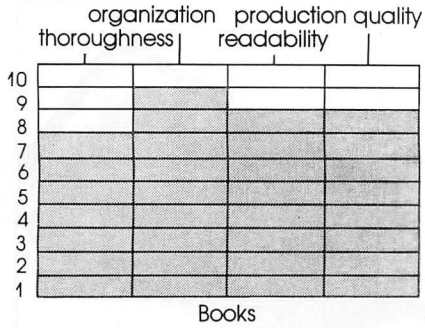
Additional commands refine those mentioned. For example, there are multitasking options and a multitasking PAUSE command.

Even after I read the documentation, I was not expecting the simple smooth performance I discovered when I put Key-264K to work. It is impressive enough to observe software addressing 64K on a 32K computer. It is even more impressive to watch the computer perform as two computers working together. But best of all is the ability to integrate the second 32K back into your programs.

Key-264K gives my CoCo capabilities I don't even have with my IBM PC compatible. ■

Eds. note—Key Color Software recently informed us that registered Key-264K owners can receive a back-up disk for $10, plus $2 shipping.

Those with the cassette version can switch to disk by sending in the tape and $5, plus $2 shipping.



Color Basic Unravelled
Extended Basic Unravelled
Disk Basic Unravelled
Spectral Associates
3416 South 90th
Tacoma, WA 98409
206-581-6938
$19.95 (all three for $49.95)

by Mark S. Rothstein

These three books are the first publications to collect full disassembly information on the current editions of the Color Basic and Disk Basic ROMs. They offer thorough, commented disassembly listings of Standard Color Basic 1.2, Extended Basic 1.1 (although 1.2 has recently been released), and Disk 1.1. Obviously, the books are for those who are familiar with 6809 Assembly language.

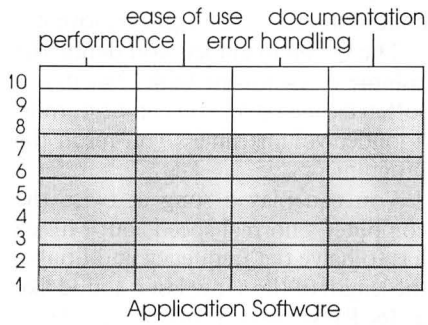
Each is about 125 pages long and covers a separate ROM. Each offers a very detailed and helpful introduction, a brief how-to-use section, a description of that particular Basic, and several appendices.

One of these appendices is the commented disassembly; others give the Basic entry points and the differences between the ROM versions.

Spectral Associates has listed and included comments for every instruction in all the ROMs. For the most part, these comments are sufficient to explain what the code is doing. However, they have omitted the instruction field, and, in *Color Basic Unravelled*, they've omitted the operand field as well. The disassembly listing does not include a full cross reference.

Although the books provide thorough listings of the three Color Computer ROMs, the listings are missing the instruction operand codes, the symbols usually appear as addresses rather than as real functional names, and there is no cross-reference table of symbol names. However, the comments are complete.

The books are paper-bound and should have a reasonably long life. However, since those who need them will most likely use them often, spiral binding would have been more convenient. They'll be a good set of references for the 6809 Assembly-language buff who wants to know more about the different ROM functions. ■



Musica
Speech Systems
38W255 Deerpath Road
Batavia, IL 60510
312-879-6880 (voice)
$34.95 16K cassette
$39.95 32K disk

by Eric Grammer

Speech Systems' Musica is a software-based music composer that lets you create four-part harmony and play it through your monitor's speaker.

After you load this Basic program, you'll see a bass and treble clef displayed, with the cursor blinking at middle C. The screen shows the amount of free memory in the lower left corner and the voice with which you're currently working (1–4) in the lower right.

The up- and down-arrow keys move the cursor up or down the clef, one note at a time. A shifted up or down arrow moves the cursor a whole octave. You are limited to notes between high C above the treble clef, and low C below the bass clef.

Pressing the enter key adds or erases single notes. The number keys set the length of a note. The numeral 1 designates a whole note, 2 a half, 3 a triplet, and so on. This doesn't let you do things like create a dotted quarter-note, so, to get the effect, enter the same note twice, once as a quarter-note and once as an eighth.

✓ See List of Advertisers on page 97

Musica won't let you enter a time signature, so if you want to divide music into measures, press the M key to create a barline in the music.

You can't use key signatures either, so you must put a sharp (S) or flat (F) with each note, if necessary. Once you've sharped or flatted a note, that sharp or flat doesn't carry over to similar notes that follow. For example, if you have to use two consecutive B flats in the treble clef, you must press F after each note to make both flat.

To select the voice with which you want to work, press V and the number (1–4) that corresponds to the desired voice. You enter all music as block chords. If a measure in part one is all eighth notes, and the same measure for part two is a whole note, you must enter that whole note as eight eighth notes. This can be tricky when you're working with four parts, but it works well.

Single-key commands let you delete entire chords, or insert a chord between two others. Although you enter parts one at a time, they are in block form, which displays all four voices at the same time.

You can also highlight one voice over the others if the display becomes messy. Adding a rest to notes in block form applies it to all four voices. B sends you to the beginning of the song, E sends you to the end, and P plays your composition.

Each voice has its own tone table, a description of the amplitude of each of the first eight harmonics in a tone. This can get pretty complicated and requires some knowledge of music theory, but the 23-page manual gives a good explanation of these tables.

By changing tone tables, you can create tones that sound like clarinets, violins, organs, and so on. Therefore, you can make each of your four voices sound like a different instrument, a capability not found in many hardware-based music synthesizers. You can also swap timbres from voice to voice and fully control the tempo of a piece.

The Y command lets you copy from one voice to another (which only makes sense in three-part harmony). Y also lets you create a vibrato in a voice, though the effect isn't very realistic. True vibrato consists of a pitch wavering slightly above and below the normal; Musica creates vibrato by lowering the pitch of a note and creating a pulsating beat against the other notes of a chord.

"Musica is a Gem."

If the vibrato voice is solo, there is no pulsating beat, and the voice merely sounds flat. Furthermore, once the vibrato is in effect, you can't remove it.

The Q command lets you define the volume of each tone table, but if you soften a tone table, you can't return to a louder volume unless you rerun the program.

You can play a song at twice the computer's normal speed, but Musica has to halve the frequency and double the length of the song, lest it sound like a 16 RPM record playing at 33⅓. When the double-speed playing is over, Musica returns all the halved frequencies and doubled lengths to their normal values, but it often fails to match the original frequencies of the notes. This error is usually unnoticeable, but it does become part of the song data.

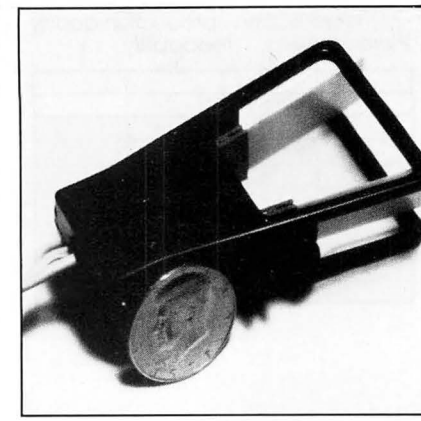
If a note is even a bit off its normal frequency, Musica will display an "unrecognized pitch value" message. This same thing happens when you give a voice vibrato, since the frequency of each individual note is slightly lowered. The program doesn't display the notes of unrecognized pitches. Double-speed playing is desirable, since it is often clearer.

Finally, the disk version of Musica also lets you lower the fourth voice by a whole octave or eliminate the notes of an entire voice. The latter is especially welcome if you wish to play a solo line by yourself, with the Color Computer playing the accompaniment.

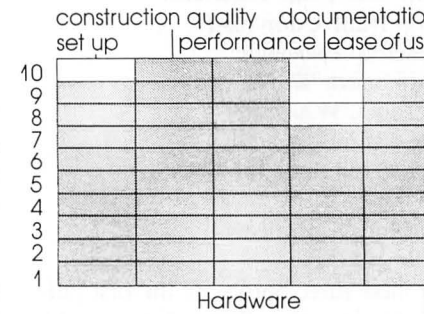
Speech Systems also sells the Stereo Composer, a plug-in board that lets you play Musica songs in stereo.

Musica comes with six preprogrammed songs and a special program that lets you play Musica songs from your own Basic programs. Examples of this are outlined in the manual.

Musica is a gem. I have never seen software that lets you control four voices with individual, programmable timbres. Once you get used to it, it's likely that you'll spend hours composing your own four-part harmonies. ■



The Dragonfly Fan



The Dragonfly Fan
Dragonfly Software
729 Westview St.
Philadelphia, PA 19119
Manufactured by
Piezo Electric Products Inc.
$18 + $1 shipping

by Bobby Ballard

If you're always short on space for CoCo peripherals, but you would like to keep your CoCo running cool, without overheating your wallet, the Dragonfly Fan might be your answer.

Not only is it great for the CoCo, but I also plan to mount one in my printer, my monitor, and my home TV. At $19, you can buy three of these fans for a few dollars more than you would spend on one of the conventional fans that mount outside the case and take up valuable desktop space.

The fan itself is very small (approximately 3-by-4-by-1½ inches) and arrives with an information sheet from Piezo, the manufacturer, and an installation instruction page from Dragonfly Software, the distributor. Although the drawings are somewhat primitive, the instructions are clear. The fan has its own power cord, and you're supposed to cut a notch for it in the computer's case. However, I tapped my CoCo's ac power supply, thus eliminating the need for another

✓ See List of Advertisers on page 97

plug at my power strip.

The Dragonfly Fan uses only 0.11 watts at 115 VAC 60 Hz, so you can tie it in with just about any peripheral without affecting the power requirements. This arrangement also lets me leave the fan running after shutting down the CoCo, much like the cooling cycle in a slide projector.

Besides that, installation is a snap. The instructions tell you to mount the fan on the right side, under the case top, so that it blows towards the cartridge-slot side vents. The fan comes with its own adhesive pads, so mounting isn't much of a problem.

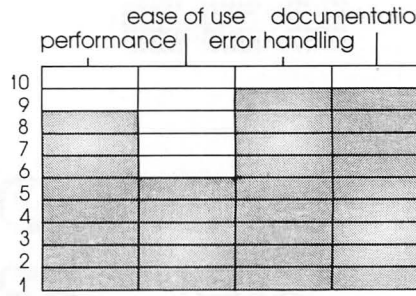
I feel, however, that it's best to mount the fan on the right, blowing left, so that the hot air formed on the left side of the computer passes out the vents, instead of being pulled across the already hot SAM and 6809E chips. This arrangement draws cool air in through the cartridge slot and pushes it across the PC board and ICs and then out past the voltage transformer on the left side.

The heat you usually feel on the left side of the CoCo dissipates once you install the Dragonfly Fan. That keeps your chips cool and, consequently, lengthens their life. The fan moves 5 cubic feet of air per minute, more than enough to keep everything in the case cool.

This fan, explains Piezo, is powered electrostatically instead of magnetically, like rotary models. Therefore, it has no starting surge, no wearing parts, 1/15th the power consumption of rotary fans, and less noise. No wearing parts means a long service life. In fact, the life-expectancy data states that there have been no failures after 30,600 hours of continuous duty. Therefore, the life expectancy is not yet determined.

This solid-state device has mylar blades that bend from the influence of two Piezoceramic elements. This means no electromagnetic interference (EMI) or radio-frequency interference (RFI).

I'm impressed with the Dragonfly Fan's efficiency. It's durable, easy to install, quiet, and inexpensive. ■



Paintpot
Tim Skene
6073 Durocher Ave.
Montreal, Quebec H2V 3Y7
514-288-4233
16K–32K, Extended Color Basic
$20 cassette (includes 16K and 32K versions)
$25 32K disk

by Beth Norman

Paintpot lets you use the arrow keys, a joystick, or the Color Mouse to draw pictures on your computer screen. You can then store your creations on tape or disk, incorporate

The Joystick that sets you free!

The one-hand operation of this fantastic new joystick will truly set you free and increase the pleasure of playing your favorite video games. The smoothness and responsiveness of this unique joystick that operates completely without a base is something to be experienced. Available direct from us or from your independent computer retail store. (See below)

$49.95 suggested retail

STOP changing Printer and Modem Cables! Our **Parallel Printer Interface** provides **Switch Selectable Printer** or **Modem** operations for both CoCo and MC10. It features switchable baud rates from 300 to 9600. It comes complete with power supply, modem cable and **"Centronics"** type printer cable. For Basic 1.1 and later revisions.

Available direct from us or from your independent computer retail store. (See below)

Only $89.95 suggested retail

pbh Computer Products, Inc.
P. O. Drawer 55868
Houston, Texas 77055
713/956-0207

↙20

When ordering direct from **PBH** please enclose $3.00 per item for shipping.

Stocking Distributors

Compukit
Houston, TX. 77059

Spectrum Projects
Woodhaven, N.Y. 11421

Authorized Dealers

Endicott Computer
Software & Accessories
Huntsville, AL. 35801

The Computer Store
Jasper, IN. 47456

The Software Connection
Ft. Lauderdale, FL. 33319

Colorware, Inc.
Woodhaven, N.Y. 11421

TRS TECH
Computer Services
Houston, TX. 77033

Computers & More
Huntsville, TX. 77340

The Photo Shop Radio Shack
Wilmar, MN. 56201

Patterson Electronics
Mountain View, AR. 72560

Computers, Etc.
Austin, TX. 78745

Cinsoft
Cincinnati, OH. 45237

EDC Industries
Los Angeles, CA. 90042

Sound Center Radio Shack
Whiterock, N.M. 87644
& Los Alamos, N.M. 87544

Chips, Incorporated
Atlanta, GA. 30340

Computer Associates
West Fargo, N.Dak. 58078

Computer Plus, Inc.
Littleton, MA. 01460

Turtle Micro Ware
East Lime, CT. 06333

↙ See List of Advertisers on page 97

them into other Basic programs, and even show them one after another in rapid succession as short "animated" sequences. The program is very easy to use and involves a minimum number of control keys.

The program draws in PMODE 1, giving you a 128-by-86 screen to work with. The screen really contains 128-by-96 points, but Paintpot uses the bottom 10 rows for a "color palette." Although the resolution is coarse, you will have to be patient as you draw—there's not much room for error.

Paintpot presents you with four blank screens in the 32K version or three in the 16K version, four color choices (red, blue, green, and the "white" of the background), and a help screen. You "dip" the cursor into a color on the palette and move it to a point at which you want to begin.

I prefer to use the joystick to move the cursor, as it is faster than any of the three available keyboard speeds. However, as the manual warns, the joystick does not let you put the cursor on every point of the screen—it can only address 64-by-64 points.

Holding down the joystick button or the enter key leaves a permanent trail of color behind the cursor. Pressing E erases the entire screen, and you can also fill an entire region with color by putting the cursor inside the border of the area to be painted and pressing P.

The P command is a little less flexible than Extended Color Basic's PAINT command. In Paintpot, you can only paint with the current color, and the border of the painted region must also be in the same color.

When you've finished a drawing, press the space bar to bring up the next blank screen. Pressing C copies your drawing onto the next screen.

Paintpot has very few commands to memorize, but you must draw freehand; there is no built-in shape library. If you've ever tried to use the joystick to draw a freehand circle, you know how much of a problem this can be. (It's impossible with the arrow keys!)

Actually, the program does give you a way out. You can break out of the system and enter the LINE, CIRCLE, or DRAW commands you need to construct a complicated figure. The results will show up on the current screen when you press the enter key.

You can even use a FOR...NEXT loop to construct a series of similar shapes, as long as you can enter everything on one command line of less than 255 characters. The manual describes how to use some of the options of the LINE command to get rid of the palette image.

It is possible to get good results using these methods, but it is time consuming, and you must be familiar with the Extended Color Basic graphics commands. Paintpot really should include a listing of these commands with examples. As it is, you may have to juggle two pieces of documentation to find the information you need to create a drawing.

I found it difficult to erase mistakes without wiping out a complete drawing. You have to position the cursor precisely on part of the drawing and retrace it to erase. You can only remove lines of the current color.

It is almost impossible to erase a straight line with the joystick, and very difficult with the keyboard. Your erasing has to be as precise as your original drawing, and that can be a problem. Maybe these things are easier with the Color Mouse.

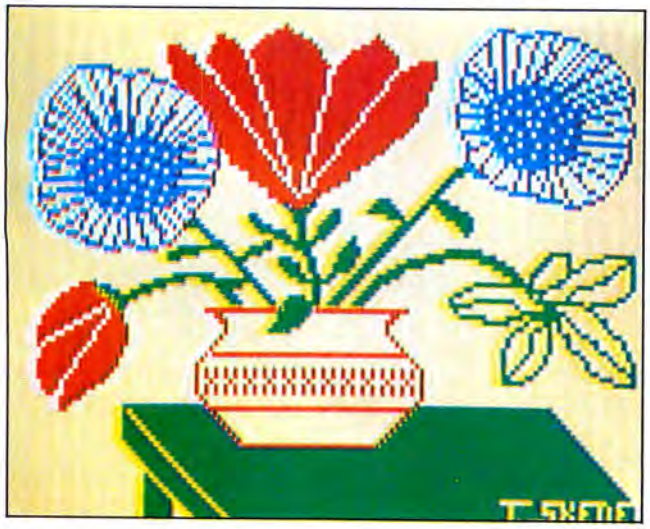
Paintpot's most interesting feature is its animation option. You can copy whatever you have drawn on the first screen onto the second, make a few changes, copy the new version to the third screen, and so on. When you've finished the four screens, you can press the A key and watch the computer flashes as a short animated loop. The space bar stops the action.

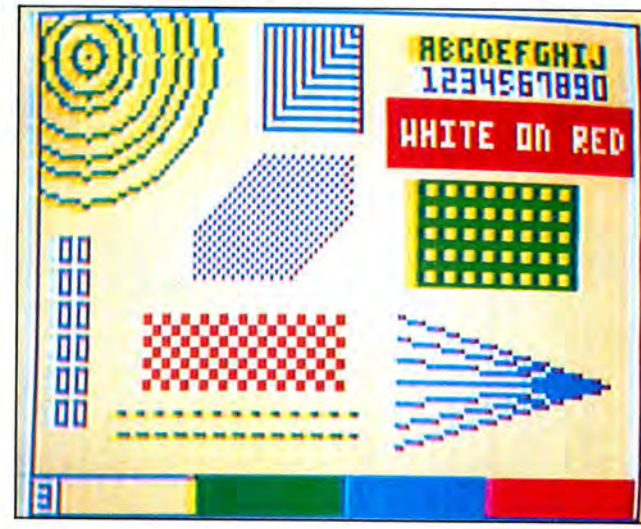
This is a very nice feature; I just wish it were easier to control erasure so it would be easier to draw successive frames.

Since you can save your creations on tape or disk, you can reload a whole sequence of animated images, or just a single frame, as you wish. The documentation also tells you how to include a Paintpot drawing in one of your own Basic programs. That's a very big plus; it lets you use the program to create your own title screens or video game boards. ■

Eds. note—Tim Skene has recently informed us of three changes he has made in the latest version of Paintpot. He has removed the feature that caused the cursor to erase in the joystick mode. The cursor now erases in all modes when you press the clear key.

You can also change the current drawing color by pressing a key or by dipping the cursor into the palette. Both the tape and disk versions of Paintpot now include a sample drawing.

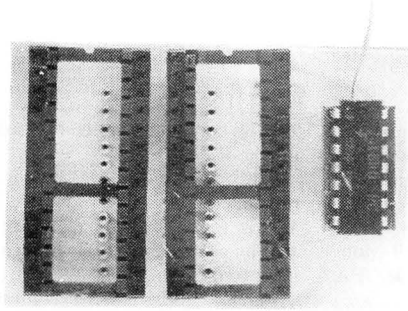




Photos. Sample Paintpot Graphics

16 K DOS CARD

- ☐ PLUGS INTO YOUR J-M DISK CONTROLLER AND ALLOWS YOU TO MAP ON AN EXTRA 8 K E-PROM ABOVE DOS.
- ☐ USE YOUR OWN 24 PIN, 8 K DOS AND ONE 2764 E-PROM OR TWO 2764 E-PROMS.
- ☐ GREAT FOR UTILITIES OR A MACHINE LANGUAGE MONITOR.
- ☐ ON BOARD DE-CODING, ONLY ONE WIRE TO SOLDER. COMPLETE WITH INSTRUCTIONS.



$19.95

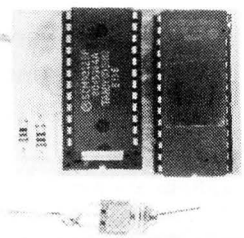
RGS DUAL DOS CARD

WITH SWITCH SELECTOR

DESIGNED TO ACCOMODATE TWO DIFFERENT DOS CHIPS INSIDE YOUR J-M DISK CONTROLLER.

- ☐ PIN TO PIN COMPATIBLE WITH RS-DOS AND J-DOS CHIPS.
- ☐ THE SWITCH ALLOWS YOU TO HARD SELECT ANY ONE OF THE TWO DOS SYSTEMS OF YOUR CHOICE.
- ☐ IN CENTER POSITION, THE SWITCH DISCONNECTS FROM THE DOS AND BRINGS YOU BACK TO BASIC.
- ☐ DESIGNED FOR ONE 24 PIN ROM AND A 28 PIN E-PROM OR TWO 28 PIN E-PROM CONFIGURATION.
- ☐ EASILY MODIFIED BY CUTTING TWO TRACES ON THE BACK OF THE BOARD.

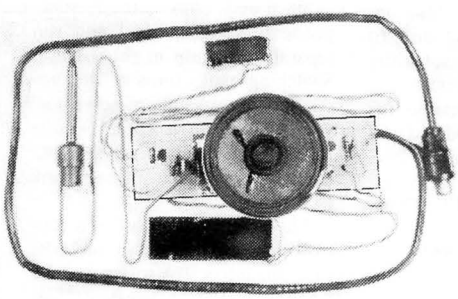
$19.95
(Board with switch only)



VIDEO PAL

- ☐ AUDIO-VIDEO INTERFACE
- ☐ MONOCHROME COMPOSITE OUTPUT
- ☐ EASY TO INSTALL, FITS UNDER YOUR KEYBOARD
- ☐ NO SOLDERING!
- ☐ BUILT-IN SPEAKER
- ☐ DOES NOT DISABLE YOUR REGULAR T.V. OUTPUT
- ☐ FULLY TESTED AND ASSEMBLED
- ☐ COMPLETE WITH INSTRUCTIONS.

ALSO AVAILABLE FOR COLOR MONITORS



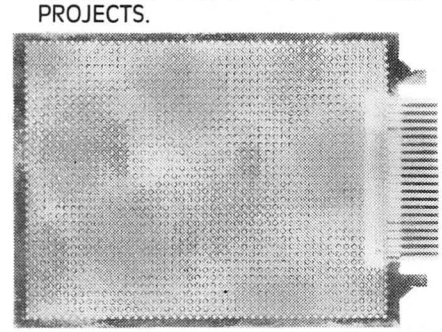
$29.95

PROJECT BOARD

A MUST FOR EXPERIMENTS

- ☐ UNLIMITED CHIP POSITIONS
- ☐ GOLD PLATED EDGE-CARD CONNECTOR
- ☐ FITS INTO ANY RS DISK PACK
- ☐ HOLES PLATED THROUGH BOTH SIDES
- ☐ EASY TO WIRE - WRAP

GREAT TO BUILD YOUR "TURN OF THE SCREW" PROJECTS.



$19.95
(TWO FOR $34.95)

USA
RGS MICRO INC.
MAIN STREET
DERBY LINE, VERMONT
ZIP 05830
TEL: 802-873-3386
ORDER LINE 800-361-4970

RGS MICRO INC.
185

CANADA
RGS MICRO INC.
759, VICTORIA SQUARE 405
MONTRÉAL H2Y 2J3
TEL.:(514) 287-1563
ORDER LINE ONLY ★★★
QUÉBEC - ONTARIO - MARITIMES
800-361-5338
WESTERN CANADA 800-361-5155

TERMS: VISA - MASTER CARD - AMERICAN EXPRESS

HOURS: MONDAY - SATURDAY 10:00 AM - 6:00 PM

See List of Advertisers on page 97

PRODUCT NEWS

edited by Celeste Wrenn

Computerware's Newest Games

Computerware has come out with two new adventure games for the Color Computer.

● **Middle Kingdom**—In this real-time graphics adventure your goal is to become ruler of the Middle Kingdom, which can be achieved only by returning the three magic rings to the sanctuary. You must search the rooms of the catacombs, temple, and pyramid. Be warned! Many monsters lurk in these rooms.

Middle Kingdom requires 32K and sells for $24.95 on cassette and $27.95 on disk (plus $2 shipping).

● **Star Trader**—In this graphics simulation you are a merchantship captain in the far future. You travel in real time between solar systems, trading cargo, encountering pirate ships, stopping at starports for fuel or repairs, and making money. Your goal is to collect 1,000 credits to retire in luxury.

With different skill levels and many variable factors, this simulation offers a new adventure every time you play.

Star trader requires 32K cassette or 64K disk, with one joystick or mouse and Extended Basic. It sells for $24.95 on cassette and $27.95 on disk (plus $2 shipping).

Both games are available from Computerware dealers or directly from Computerware at Box 668, 4403 Manchester Ave., Suite 103, Encinitas, CA 92024, 619-436-3515.

Reader Service ✓ 550

Information used in the Product News section is supplied through manufacturers' press releases. *HOT CoCo* has not tested or reviewed these products and cannot guarantee any manufacturer's claim.

Sprightly Tapes

The HHCI Tape System can read and write data via unmodified CTR-41, CTR-80, or CCR-81 cassette recorder at approximately 7,000 baud. Programs can be loaded four times faster. A switch selects high-speed or standard tape operation.

The HHCI Tape System interfaces with Color Basic ROMs and Extended Color Basic ROMs. It sells for $54.95 with installation instructions and user's manual. The user's manual is available separately for $9.95, but that amount is credited to the purchase of a full system. Contact HHCI Tape Systems, 725 Idlewild Road, Bel Air, MD 21014, 301-838-7692.

Reader Service ✓ 552

Home Energy Manager

The Micro-Man Home Energy Manager by Digital Systems saves up to 30 percent on heating and cooling costs when used with the Color Computer, TDP-100, or MC-10.

Connected to the RS-232 port and the home's low-voltage thermostat wiring, the Micro-Man interrupts the furnace burner or the ac compressor on an automatically varying optimal energy-saving cycle.

The Micro-Man interface controller connects to the Color Computer via the standard four-pin DIN RS-232 port and to the thermostat wiring with four color-coded wires and wire nuts. The interface controller can be mounted at the furnace or at any other point in the thermostat wiring and can be disconnected at the computer at any time without affecting normal furnace and air-conditioner operation.

Micro-Man is distributed on cassette and includes a separate Home Energy Audit program for evaluating the home's energy efficiency. It sells for $39.95 from Digital Systems Corp., 319 Monroeville Mall, Monroeville, PA 15146, 412-373-3802.

Reader Service ✓ 555

Let Your Fingers Do the Walking

There is something new being offered in the world of communications. It's the BBS Directory. This listing of bulletin boards in North America offers over 700 listings by state, area code, and computer type. With each telephone number there is a comment line with information about free program downloading; the type of computer supported; if the bulletin board is for shopping, education, games, adult, etc.; and if a password is required. Also included is an overview of how to use a BBS, the major software, and the computers it supports.

The directory is available for $5.95 postpaid and is updated and published quarterly. Contact BBS Directory, P.O. Box 4150, Beach Station, Vero Beach, FL 32964, 305-466-5515.

Reader Service ✓ 553

Parents, Kids, And Computers

Sybex Computer Books has published a book for parents who want their children to know about computers but weren't sure where to begin learning themselves.

Parents, Kids, and Computers will help parents teach their children learning skills and show them how to have fun. Parents find out:

- How computers are being used in schools.
- The best way to work with children at computers.
- How to select the best software.
- Why girls need extra encouragement to use computers.

Parents, Kids, and Computers is available at book and computer stores, or contact Sybex Press, 2344 Sixth St., Berkeley, CA 94710, 415-848-8233.

Reader Service ✓ 556

128K Memory Expanders

Dynamic Electronics Inc. has introduced a complete line of 128K memory expanders that mount inside the computer and are compatible with all existing software. The memories consist of two 64K memory banks that you can select via a miniature three-position switch (included) or via software.

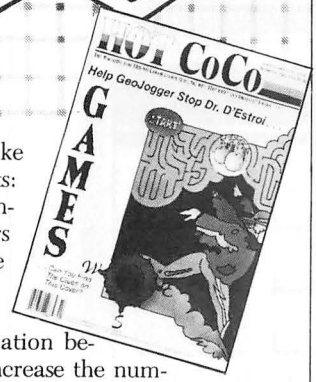
Since each bank is totally independent, you can load and run separate programs in either bank. Switching banks turns off the unselected bank but preserves all variables and vectors. Simple memory POKEs and PEEKs transfer variables from one bank to the other.

The expanders consist of a control circuit mounted in modules that plug into a PIA socket and the SAM socket, two banks of 64K RAM, and a three-position toggle switch for either hardware or software selection of the banks.

The 128K memory expanders simply plug in, requiring no wire cutting or changes, and they come with a one-year warranty.

There are three models available. The ME-128D, for upgrading D and E types to 128K, sells for $269. The ME-128F, for upgrading 285 types to 128K, sells for $259. The ME-12-64, for upgrad-

Dealers Dealers Dealers SELL!



Selling **HOT CoCo** will make money for you. Consider the facts:
Fact 1: Selling **HOT CoCo** increases store traffic—our dealers tell us that **HOT CoCo** is one of the hottest-selling computer magazines on the newsstands.
Fact 2: There is a direct correlation between store traffic and sales—increase the number of people coming through your door and you'll increase sales.
Fact 3: Fact 1 + Fact 2 = INCREASED SALES, which means more money for you. And that's a fact.

For information on selling **HOT CoCo**, call 800-343-0728 (In N.H. call 924-9471) and speak with Ginnie Boudrieau, our bulk sales manager. Or write to her at **HOT CoCo**, 80 Pine Street, Peterborough, NH 03458.

HOT CoCo

80 Pine Street
Peterborough, NH 03458
800-343-0728

I HAVE SOMETHING TO SAY!

For as little as 46 cents a program you can enjoy great games like **FROGJUMP, FRENZY, CATERPILLAR CAVE, SHOOTIN GALLERY,** and great adventures. Use powerful home management programs such as **BUDGET, STOCK MARKET, BARTENDER, TINY CALC,** and **COST OF LIVING.** Also, enjoy the use of powerful programming utilities such as **COCO MONITOR, GRAPHIC SCREEN PRINT, RAM TEST, HIGH RESOLUTION TEXT,** and **SINGLE DISK COPY.** Increase your knowledge with educational programs such as **BASIC SPEED UP TUTORIAL, FLASH CARD, TRIGONOMETRY TUTOR,** and our **8 PART SERIES ON MACHINE LANGUAGE.**

⁂ Many programs use machine language! ⁂

Every month you will receive a cassette tape with 10 ready to run documented programs.

16K Extended Required

ORDERS SENT SAME DAY!
FIRST CLASS SHIPPING!
Back Issues Available from July, 1982.
Call or write for our free catalog.
Dealer Inquiries Invited.

PRICES

1 YR (12 ISSUES)	$55.00
6 MO (6 ISSUES)	$30.00
SINGLE COPIES	$ 6.00

MICHIGAN RESIDENTS ADD 4% TO ORDER
OVERSEAS ADD $10.00 TO SUBSCRIPTION AND $1.00 TO SINGLE COPIES
PERSONAL CHECKS WELCOMED!

★★★★ **FREE PROGRAMS** ★★★★
FOR EACH ORDER OF 20 OR MORE TAPES YOU WILL RECEIVE TWO COLOR COMPUTER PROGRAMS OF THE ABOVE PRODUCT!

✓236

T&D

COMPUTER CASSETTES

★ 5 SCREW SHELL
★ FREE LABELS
★ ERROR FREE TAPE
★ IMMEDIATE SHIPPING

MasterCard VISA

T&D Computer Products
P.O. Box 256-C
Holland, MI 49423
(616)396-7577

ITEM	PRICE	QUANT.	TOTAL
C-10	.59		
C-20	.69		
C-60	.85		
C-90	.99		
Outside USA Add $3.00		Subtotal	
		Shipping	3.00
USA funds only		Mich. Res.	4%
		TOTAL	

Name _____
Address _____
City _____ St. _____ Zip _____
Credit Card _____ Exp. _____

The choice is yours — and you can get your first issue free if you subscribe now

If you want to make the most of your new Dragon computer, then you need **Dragon User**. This independent, international magazine for all Dragon owners is packed with software and hardware advice.
Regular features:
- Pages of program listings
- Chance to win $300 prizes
- Advice on which software to buy
- In-depth hardware evaluations
- Technical advisory service
- All the latest news

If you've ever been killed by the evil goblin, flamed by a dragon or turned to stone by a wizard, then you need **Micro Adventurer** — the new magazine devoted to all microcomputer adventures, war games and simulations.
Each issue features:
- Helpline and Contact columns
- Reviews of the latest adventures
- Competitions with exciting prizes
- Adventures to type in and play
- Advice on how to write your own adventures
- Profiles of famous adventurers

Subscription form

Fill in this form and send it to the appropriate magazine's subscription department, **c/o Business Press International, 205 East 42nd Street, New York, NY 10017.**

Please send me 13 issues of
☐ **Micro Adventurer** at US$33.95
☐ **Dragon User** at US$29.95

This is the usual rate for a year's subscription (12 issues).

Name _____
Address _____
_____
_____
Signature _____
Date _____

Please start my subscription from the following issue
_____

This form should be accompanied by a check made payable to the magazine to which you are subscribing.

✓27

✓ See List of Advertisers on page 97

ing all 64K computers to 128K, sells for $199.

For further information contact Dynamic Electronics Inc., P.O. Box 896, Hartselle, AL 35640, 205-773-2758.

Reader Service ✓ 557

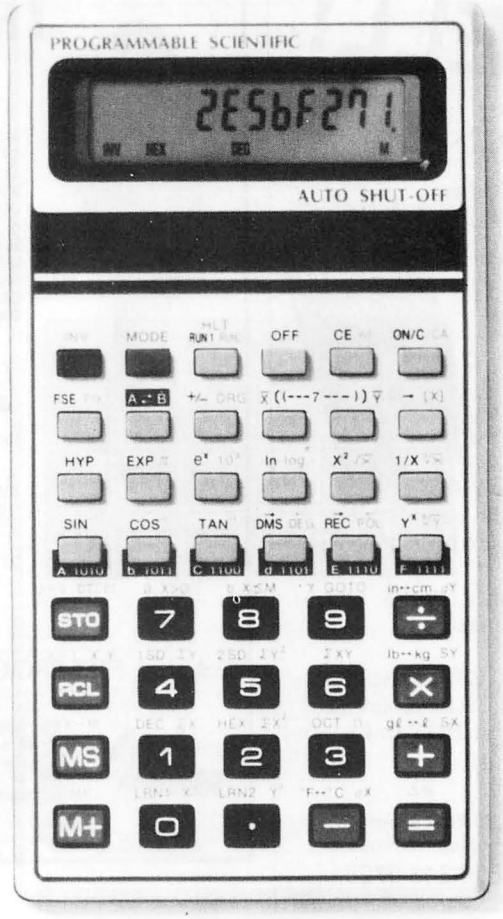
The Spreadsheet Zapper

Now you can integrate a spreadsheet program with high-resolution graphs and charts. The Spreadsheet Zapper bridges the gap between the great computational power of Radio Shack's Spectaculator and the high-quality graphical display capability of Southern Software Systems' Graph, Bar, and Pie Zapper.

The Spreadsheet Zapper converts Spectaculator files (ROM pack or disk version) into data files that can be used directly by the Graph, Bar, and Pie Zappers to produce the graphs and charts that make your spreadsheet calculations crystal clear. Now virtually anything you can compute with Spectaculator can be graphically displayed.

The Spreadsheet Zapper is priced at $17.95 for the 32K-64K tape version and $25.95 for the 32K-64K disk version, plus $1 for shipping. Both require at least 32K of memory and Extended Color Basic. It is available with a 14-day, money-back guarantee from Southern Software Systems, 485 South Tropical Trail, Suite 109, Merritt Island, FL 32952, 305-452-2217.

Reader Service ✓ 558



Low-Priced Programmable Scientific calculator

K-Basic Compiler

Lloyd I/O has introduced a new K-Basic Compiler. It is a Basic-language compiler that converts Basic programs to machine code.

K-Basic has three general data types: real, string, and integer. There are four integer sizes: 8 bit, 16 bit, 32 bit, and 64 bit. Real numbers are 15-digit precision with an exponent of +/−99. Each type can be dimensioned and defined to absolute addresses. There are many directives, statements, and functions not found in Basic interpreters. Line numbers or labels are not required on every line. Variable names are significant to 12 characters. Labels are significant to 16 characters. K-Basic includes debugging features and error processing.

K-Basic sells for $199. The manual may be purchased separately for $15 which is applied to the purchase of the complete package. For more information, contact Lloyd I/O, 19535 NE Glisan, Portland, OR 97230, 503-666-1097.

Reader Service ✓ 562

Programmable Scientific Calculator

The new CI-560 Programmable Scientific is an advanced scientific and statistical calculator designed to meet highly technical needs.

The calculator operates in seven modes. It has 45-step operational programming, 70 preprogrammed scientific, statistical, and mathematical functions, seven-level algebraic-logic parentheses, seven permanent-storage memories (including one independent-user memory), 10-digit LCD display, color-coded keyboard, and automatic shut-off.

The CI-560 Programmable Scientific includes a money-back guarantee, one-year limited warranty, instruction manual, carrying case, and long-life batteries. It sells for $39.95 plus $3 shipping.

For further information contact Calculated Industries, 2010 North Tustin Ave., Suite B, Orange, CA 92665, 1-800-854-8075 (in California: 1-714-921-1800).

Reader Service ✓ 561

Free Software

CoCo Freeware has introduced two new programs for the Color Computer.

● **Assembler Language Programming Toolkit** (ALPT) (v1.0) is a two-pass editor/assembler. The ALPT provides you with an easy way to introduce yourself to the world of machine language. It comes with a separate program to produce the needed seven pages of documentation. It requires 32K Extended Basic (printer initially required to print the documentation). It works with either tape or disk systems. Request programs 301A, B, and C. Disk users should also request program 301D.

● **Spooler** (v1.0) is an easy-to-use spooler program to speed up your printer use. It creates a buffer of nearly 8,000 characters. Just load and execute this machine-language program and recover the time that you used to waste waiting for your printer to finish its assigned task. Spooler comes with a separate program to produce the needed three pages of documentation. It requires 64K Extended Basic (printer initially required to print the documentation) and is disk compatible. Request programs 351A and B.

CoCo users can obtain a copy of these programs and their associated documentation programs by forwarding a blank, computer-grade cassette (or formatted disk) along with a postage-paid return mailer to the address below. If your return mailer is large enough a copy of the latest Users Info-Pak will also be included. There is no purchase price involved, but contributions in any amount are accepted on the behalf of the author once users have received and have had a chance to use the program. Send no money now.

For more information contact the CoCo Freeware Clearinghouse, P.O. Box 1084, Morgantown, WV 26507, 304-599-4493.

Reader Service ✓ 563

Peeper

The Peeper is an interrupt-based program tracer that lets you monitor operation of a machine-language program while it is running. You instantly switch between watching your program's regular output and watching Peeper's changing trace output of registers and stacks on your screen or printer.

Execution speed can be varied while the program is running from moderate slow-down to several thousand times slower than normal to a complete freeze.

Peeper supports single-stepping, breakpoints, memory examine/change, and movable window to view memory in any graphics mode. It can be used with arcade games to watch fine details of animation effects in slow motion, inspect hidden screens, and play in unusual graphics modes.

The documentation includes "A Guided Tour Through CoCo's Memory," which uses Peeper's capabilities as the basis for a tutorial on the functioning of the Basic interpreter. Peeper is available on cassette for $21.95 or with assembler listing for $24.95 plus $2 shipping. Contact Spectrosystems, 11111 North Kendall Drive, Suite A108, Miami, FL 33176, 305-274-3899.

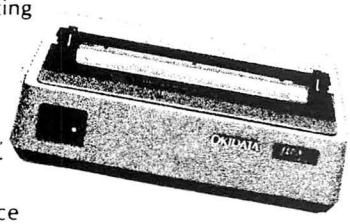
Reader Service ✓ 565

PRINTER SALE

$189

ML 80 Features

- 80 cps, unidirectional printing
- 80 columns standard, 132 condensed
- TRS-80 character set
- Graphics, 64 block shapes
- Long-life, 7- pin print head, warranted for one full year
- Friction & pin paper feeds
- Parallel Centronics interface



OKIDATA

SUNLOCK SYSTEMS
4217 Carolina Ave.
Richmond, Va. 23222

✓456

ADDITIONAL PRINTER SPECIALS

Epson		Okidata		Brother 15	$479
RX80	$299	ML82	$349	Brother 25	739
80FT	369	ML83	569	Gemini 10X	279
RX100	529	ML84	975	Man/Tal 80	319
FX80	479	ML92	429	Qume 1140	1349
FX100	649	ML93	699	SCM TP-2+	449

CItoh 8510	$359
CItoh F10	1099
Sv.Reed 500	429
Diablo 620	949
NEC 2000	849
NEC 3510	1499

TO ORDER CALL TOLL FREE 800-368-9191
In Virginia call 804-321-9191
We accept MasterCard, Visa and CODs

GIVE SOMEONE THE GIFT OF LEARNING

Experience the joy of giving and Learning. There is no better gift. Three great programs for you and your kids/ students.

MATH JOY: Addition & Subtraction

MOTIVATES children to learn math in a FUN style. Superb graphics, animation & sound. Teaching & drills in Addition and Subtraction. 3 skill levels from 1-to-2 digit numbers. Unconventional 3" bold NOs. Correct answers suitably rewarded & wrong answers crossed off & displayed correctly. No jystk req. 16K ECB Tape $11.95, Disk $16.95

SPELL 'N WIN

A GREAT SPELLINGS program for Beginners, Intermediate, Advanced & Highly Advanced levels. Helps you and children achieve high scores in School or competitive exams. 800 assorted well-balanced words appearing in a random sequence & not in the same order. 16K ECB. No jystk. Disk $16.75

COLOR PAD

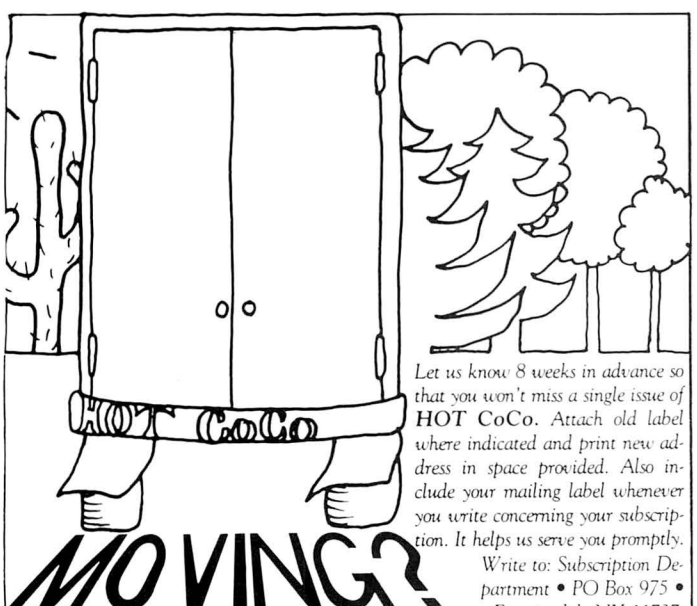
A "FUN PAD" for children & adults. Draw anything from Planes to Landscapes. Create dazzling patterns. Edit, paint, erase and save your creations to tape or disk. BONUS: Airplanes Color Sketch Book Program.
No jystk req. Tape $16.95 ..... Disk $19.95

COLOR SKETCH BOOK PROGRAMS: Solar System, ships, airplanes, bldgs. & Landscapes. The Color Sketch Book programs Require Color Pad. Tape $6.00(ea.) Disk $9.00(ea.)

TO ORDER: VISA, MC, CHECK, MO, COD($2.50). Please add $2.00 shipping and handling. NYS residents, please add Sales Tax. Immediate Shipment.

MICROCOM SOFTWARE
P.O. BOX 214, FAIRPORT, N.Y. 14450
((716) 223-1477
Dealer Inquiries invited

✓40



MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of HOT CoCo. *Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly. Write to: Subscription Department • PO Box 975 • Farmingdale NY 11737.*

☐ *Extend my subscription one additional year for only $24.97.*
☐ *Payment Enclosed* ☐ *Bill Me*

Canada & Mexico $27.97/1 yr. only US funds drawn on US bank. Foreign surface $44.97/1 yr. only US funds drawn on US bank.

affix old label *new address*

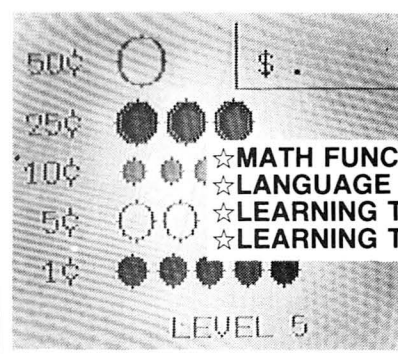
*name*_____ *name*_____
*address*_____ *address*_____
*city*_____ *state*_____ *zip*_____ *city*_____ *state*_____ *zip*_____

HOT CoCo • PO Box 975 • Farmingdale NY 11737

Educational Programs for the TRS-80 Color Computer

All TRS-80 programs require Extended Basic
Available for both tape and disk

Used successfully in classrooms across the country on a daily basis, B-5 programs make learning fun! Each program can be geared to the individual needs of each student.



NEW!
Recess Games
Grades 3-8

☆MATH FUNCTIONS
☆LANGUAGE ARTS
☆LEARNING TO COUNT MONEY
☆LEARNING TO TELL TIME

and more!

Priced from $9.95 to $26.95

Write today for a free catalog, or ask for a dealer demonstration.

B-5 Software Co.
1024 Bainbridge Place
Columbus, Ohio 43228
Phone (614) 276-2752

Teachers: Have you written the "ultimate" program? We'd like to take a look . . .

✓138

✓ See List of Advertisers on page 97

ON SALE NOW!

Please sign me up for a one-year subscription (twelve issues) to *Digital Audio* magazine at $19.97, a 43% savings over the newsstand price.

☐ Check enclosed
☐ MC ☐ VISA ☐ AE
☐ Bill me

Card no._____
Exp. date_____
Name_____
Address_____
City_____
State_____ Zip_____

74A405

Send Coupon or a Photocopy to:

Digital Audio Magazine
P.O. Box 976
Farmingdale, NY 11737

or call toll free **1-800-227-1053**

DIGITAL AUDIO MAGAZINE

A WGE PUBLICATION

452

You've invested a lot of time and money into your computer . . . . .
It's time that investment paid off!

THE COLOR ACCOUNTANT

Introducing The Color Accountant (from The Programmer's Institute), the only complete personal financial package specifically designed for the TRS-80 Color Computer. This unique package includes:

1. **Complete Checkbook Maintenance**
2. **Chart of Accounts Maintenance**
3. **Income/Expense Accounts**
4. **Net Worth Statement**
5. **Payments/Appointments Calendar**
6. **Color Graph Design Package (graphs any files)**
7. **Check Search**
8. **Home Budget Analysis**
9. **Decision Maker**
10. **Mailing List**

After the initial setup, **THE COLOR ACCOUNTANT** requires less than an hour of data input each month.

The checkbook maintenance program is the key to the entire package. Once your checkbook is balanced, the checkbook summary file will automatically update the home budget analysis, net worth, and income/expense statements. You can then graph any file, record bills and appointments, make decisions, print a mailing list, analyze various accounts or stocks, and even calculate taxes.

All programs are menu-driven and allow add/change/delete. Each file and statement can be listed to screen or printer, and saved to cassette or diskette. **THE COLOR ACCOUNTANT** also comes with 40 pages of documentation that leads you step-by-step through the entire package. The TRS-80 COLOR Ext. Basic requires 16K for this package.

$74.95 Cassette **$79.95** Diskette

Send $2 For Our New 64-Page Catalog (#11) Which Contains More Than 500 Software Products.
(Catalog is provided FREE with any order)

To Order:
Write or call Toll-Free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

THE COMPUTRONICS INC. -9

50 NORTH PASCACK ROAD
SPRING VALLEY, N.Y. 10977

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

AMERICAN EXPRESS MasterCard VISA

30-DAY MONEY BACK GUARANTEE

***** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE *****
DELIVERY SUBJECT TO AVAILABILITY

BUSINESS PAC 100

Now Available for the TRS-80 Color Computer!

Includes Inventory Control, Payroll, Accounts Receivable, Accounts Payable, Checkbook Maintenance, and more. Comes with 128-page user's manual.

Interest apportionment by Rule of 78's • Annuity computation program • Time between dates • Day of year a particular date falls on • Interest rate on lease • Breakeven analysis • Straightline depreciation • Sum of the digits depreciation • Declining balance depreciation • Double declining balance depreciation • Cash flow vs. depreciation tables • Print checks with daily register • Checkbook maintenance program • Mortgage amortization table • Compute time needed for money to double, triple, etc. • Determine salvage value of an investment • Rate of return on investment with variable inflows • Rate of return on investment with constant inflows • Effective interest on a loan • Future value of an investment (compound interest) • Present value of a future amount • Amount of payment on a loan • Equal withdrawals from investment to leave zero over • Simple discount analysis • Equivalent & nonequivalent dated values for obligations • Present value of deferred annuities • Percent markup analysis for items • Sinking fund amortization program • Value of a bond • Depletion analysis • Black-Scholes options analysis • Expected return on stock via discounts dividends • Value of a warrant • Estimate of future earnings per share for company • Compute alpha and beta variables for stock • Portfolio selection model • Option writing computations • Value of a right • Expected value analysis • Bayesian decisions • Value of perfect information • Value of additional information • Derive utility function • Linear programming solution by simplex method • Transportation method for linear programming • Economic order quantity inventory model • Single server queueing model • Cost-volume-profit analysis • Conditional profit tables • Opportunity loss tables • Fixed quantity economic order quantity model • Cost-benefit waiting line analysis • Net cash-flow analysis for simple investment • Profitability index of a project • Weighted average cost of capital • True rate on loan with compensating balance required • True rate on discounted loan • Merger analysis computations • Financial ratios for a firm • Net present value of project • Laspeyres price index • Paasche price index • Construct seasonal quantity indices for company • Time series analysis linear trend • Time series analysis moving average trend • Future price estimation with inflation • Mailing list system • Letter writing system (links with mailing list system) • Sort lists of names • Shipping label maker • Name label maker • DOME business bookkeeping system • Compute week's total hours from timeclock info. • In-memory accounts payable system • Generates invoices on screen and print on printer • In-memory inventory control system • Computerized telephone directory • Time use analysis • Use of assignment algorithm for optimal job assignment • In-memory accounts receivable system • Compare 3 methods of repayment of loans • Compute gross pay required for given net • Compute selling price for given after-tax amount • Arbitrage computations • Sinking fund depreciation • Computerized UPS zone table • Type envelope with return address • Automobile expense analysis • Insurance policy file • In-memory payroll system • Dilution analysis • Loan amount a borrower can afford • Purchase price for rental property • Sale-leaseback analysis • Investor's rate of return on convertable bond

100 Ready-To-Run Business Programs

only $99.95

(Available on diskette only)
Also available for TRS-80 Models I, II and III, Apple, Atari, Osborne, Kaypro, all Commodore Computers and most CP/M systems.

The HJL-57 Keyboard

Now available for all models, including CoCo 2.

Compare it with the rest. Then, buy the best.

If you've been thinking about spending good money on a new keyboard for your Color Computer, why not get a good keyboard for your money?

Designed from scratch, the HJL-57 Professional Keyboard is built to unlock ALL the potential performance of your Color Computer. Now, you can do real word processing and sail through lengthy listings...with maximum speed; minimum errors.

At $79.95, the HJL-57 is reasonably priced, but you can find other CoCo keyboards for a few dollars less. So, before you buy, we suggest that you compare.

Compare Design.
The ergonomically-superior HJL-57 has sculptured, low profile keycaps; and the three-color layout is identical to the original CoCo keyboard.

Compare Construction.
The HJL-57 has a rigidized aluminum baseplate for solid, no-flex mounting. Switch contacts are rated for 100 million cycles minimum, and covered by a spill-proof membrane.

Compare Performance.
Offering more than full-travel, bounce-proof keyswitches, the HJL-57 has RFI/EMI shielding that eliminates irritating noise on displays; and four user-definable function keys (one latchable), specially-positioned to avoid inadvertent actuation.

Free Function Key Program
Your HJL-57 kit includes usage instructions and decimal codes produced by the function keys, plus a free sample program that defines the function keys as follows: F1 = Screen dump to printer. F2 = Repeat key (latching). F3 = Lower case upper case flip (if you have lower case capability). F4 = Control key; subtracts 64 from the ASCII value of any key pressed. Runs on disc or tape; extended or standard Basic.

Compare Installation.
Carefully engineered for easy installation, the HJL-57 requires no soldering, drilling or gluing. Simply plug it in and drop it right on the original CoCo mounting posts. Kit includes a new bezel for a totally finished conversion.

Compare Warranties.
The HJL-57 is built so well, it carries a full, one-year warranty. And, it is sold with an exclusive 15-day money-back guarantee.

Compare Value.
You know that a bargain is a bargain only so long as it lasts. If you shop carefully, we think you will agree...The HJL-57 is the last keyboard your CoCo will ever need. And that's **real** value.

Order Today.
Only $79.95, the HJL-57 is available for immediate shipment for either the original Color Computer (sold prior to October, 1982) or the F-version and TDP-100 (introduced in October, 1982), and the new 64K CoCo. **Now also available for CoCo 2.**

Call Toll Free
1-800-828-6968
In New York 1-800-462-4891

Ordering Information: Specify model (Original, F-version, or CoCo 2). Payment by C.O.D., check, MasterCard or Visa. Credit card customers include complete card number and expiration date. Add $2.00 for shipping ($3.50 for Canada). New York state residents add 7% sales tax. Dealer inquiries invited.

HJL PRODUCTS
Div. of Touchstone Technology Inc.
955 Buffalo Road • P.O. Box 24954
Rochester, New York 14624
Telephone: (716) 235-8358

440